



ViPNet Coordinator HW 4

Настройка с помощью командного интерпретатора

Версия продукта: 4.5.8

ViPNet Coordinator HW10

ViPNet Coordinator HW50

ViPNet Coordinator HW100

ViPNet Coordinator HW1000

ViPNet Coordinator HW2000

ViPNet Coordinator HW5000

ViPNet Coordinator VA

© АО «ИнфоТеКС», 2025

ФРКЕ.00130-03 90 04

Версия продукта 4.5.8, документ обновлен 08.07.2025.

Этот документ входит в комплект поставки продукта ViPNet, и на него распространяются все условия лицензионного соглашения.

Ни одна из частей этого документа не может быть воспроизведена, опубликована, сохранена в электронной базе данных или передана в любой форме или любыми средствами, такими как электронные, механические, записывающие или иначе, для любой цели без предварительного письменного разрешения АО «ИнфоТеКС».

ViPNet[®] является зарегистрированным товарным знаком АО «ИнфоТеКС».

Все названия компаний и продуктов, которые являются товарными знаками или зарегистрированными товарными знаками, принадлежат соответствующим владельцам.

АО «ИнфоТеКС»

127083, Москва, улица Мишина, д. 56, стр. 2, этаж 2, помещение IX, комната 29

Телефон: +7 (495) 737-6192, 8 (800) 250-0260 — бесплатный звонок из России (кроме Москвы)

Сайт: infotecs.ru

Служба поддержки: hotline@infotecs.ru

Содержание

Введение.....	12
О документе	13
Для кого предназначен документ	13
Соглашения документа	13
Общие сведения о ViPNet Coordinator HW	14
Обратная связь	15
 Глава 1. Работа с командным интерпретатором.....	16
Основные сведения по работе с командным интерпретатором	17
Интерфейс командного интерпретатора	18
Сокращённый ввод команд	18
Автодополнение команд	18
Контекстная справка	19
Сочетания клавиш	20
Локальное подключение к ViPNet Coordinator HW с помощью консоли	21
Выбор консоли при локальном подключении к ViPNet Coordinator HW	21
Удаленное подключение к ViPNet Coordinator HW с помощью протокола SSH	23
Перевыпуск ключей локального SSH-сервера	24
Мониторинг активности удаленных пользователей.....	24
Аутентификация пользователя на ViPNet Coordinator HW	25
Аутентификация администратора на ViPNet Coordinator HW	27
 Глава 2. Настройка подключения к сети.....	28
Настройка сетевых интерфейсов Ethernet	29
Назначение дополнительных IP-адресов	31
Подключение к сети Wi-Fi	32
Подключение по PPPoE	34
Дополнительные настройки PPPoE-интерфейса	35
Подключение к мобильной сети	36
Настройка подключения	36
Предварительная настройка	36
Автоматическое определение оператора	37
Настройка модема вручную	37
Настройка учета мобильного трафика	39
Использование динамических интерфейсов	41

Изменение значения MTU для сетевых интерфейсов	42
Поддержка технологии VLAN	43
Настройка обработки трафика из нескольких VLAN	44
Использование агрегированных сетевых интерфейсов	46
Режимы работы агрегированного интерфейса	46
Создание агрегированного сетевого интерфейса	49
Причины сбоя подчиненного интерфейса	50
Настройка агрегированного канала между ViPNet Coordinator HW и коммутатором	51
Настройка агрегированного канала между двумя ViPNet Coordinator HW	53
Настройка получения параметров от DHCP-сервера	55
Глава 3. Настройка VPN.....	56
Общие принципы настройки VPN	57
Настройка параметров сетевого подключения координатора	58
Режимы сетевого подключения координатора	58
Настройка режима «Без использования межсетевого экрана»	59
Настройка режима «Координатор»	60
Настройка режима «Со статической трансляцией адресов»	61
Настройка режима «С динамической трансляцией адресов»	63
Настройка виртуальных IP-адресов	67
Общие сведения о виртуальных IP-адресах	67
Настройка параметров виртуальных IP-адресов	68
Настройка видимости узлов	69
Настройка туннелирования на сетевом уровне	73
Общие сведения о туннелировании	73
Ручная настройка туннелирования	74
Примеры настройки туннелирования	76
Организация защищенного взаимодействия открытых узлов (туннелирование)	76
Ограничение доступа туннелируемых узлов	78
Организация туннелей между защищенными и открытыми узлами	79
Настройка IP-адресов доступа к узлу и их приоритета	82
Настройка TCP-туннеля	84
Настройка сервера TCP-туннелей	85
Настройка TCP-клиента	86
Сценарии использования VPN	88
Организация защищенного взаимодействия между двумя удаленными офисами	88

Использование ViPNet Coordinator HW в качестве защищенного интернет-шлюза	89
Настройка туннелирования на канальном уровне по технологии L2OverIP	92
Общее описание технологии L2OverIP	92
Особенности реализации L2OverIP	94
Организация защиты соединения между удаленными сегментами сети на канальном уровне модели OSI	95
Примеры настройки функции L2OverIP	96
Настройка функции L2OverIP при отсутствии VLAN	96
Настройка функции L2OverIP в случае использования VLAN	98
Настройка функции L2OverIP для ViPNet Coordinator VA	100
Глава 4. Настройка сетевых фильтров	102
Основные принципы фильтрации трафика	103
Виды трафика	103
Принципы фильтрации	104
Общие сведения о сетевых фильтрах	108
Группы объектов	110
Системные группы объектов	111
Пользовательские группы объектов по умолчанию	112
Создание пользовательской группы объектов	113
Просмотр групп объектов	114
Удаление пользовательских групп объектов	115
Создание сетевого фильтра	116
Адрес отправителя	118
Адрес получателя	120
Протокол	120
Расписание	121
Действие	122
Просмотр сетевых фильтров	123
Изменение сетевого фильтра	125
Удаление сетевого фильтра	126
Блокирование сайтов по доменным именам с помощью сетевых фильтров	127
Глава 5. Настройка правил трансляции адресов	128
Трансляция сетевых адресов (NAT)	129
Трансляция адреса назначения	130
Трансляция адреса источника	131
Создание правила трансляции адресов	132

Просмотр, изменение, удаление правил трансляции сетевых адресов	134
Примеры использования правил трансляции адресов	136
Организация доступа пользователей к интернету	136
Размещение общедоступного сервера в демилитаризованной зоне	137
Настройка доступа к узлам локальной сети с помощью дополнительных IP-адресов	139
Обеспечение доступа удаленных клиентов к серверу по технологии PPTP VPN	139
Глава 6. Тонкая настройка межсетевого экрана	141
Обработка фрагментированных пакетов	142
Настройка антиспуфинга	144
Настройка дополнительных параметров межсетевого экрана	146
Глава 7. Настройка обработки прикладных протоколов	148
О прикладных протоколах	149
Поддерживаемые прикладные протоколы	150
Настройка параметров обработки прикладных протоколов	151
Особенности обработки прикладных протоколов DNS, FTP, SIP и H.323	152
Глава 8. Настройка прокси-сервера	157
Общие сведения о работе прокси-сервера	158
Настройка основных параметров прокси-сервера	160
Настройка антивирусной проверки трафика	163
Настройка правил контент-фильтрации трафика	164
Адрес отправителя	166
Адрес получателя	166
HTTP-метод	166
MIME-тип	167
Действие	168
Глава 9. Настройка сетевых служб	169
Настройка DHCP-сервера	170
Настройка DHCP-relay	172
Примеры настройки DHCP-сервера и DHCP-relay	174
Организация работы клиентов с локальным DHCP-сервером	174
Организация работы клиентов с удаленным DHCP-сервером	175
Использование DHCP-сервера и DHCP-relay в разных сетях	177
Использование сторонних DHCP-серверов	179
Одновременное использование DHCP-сервера и DHCP-relay	181

Использование запасного DHCP-сервера	182
Настройка DNS-сервера	184
Настройка NTP-сервера	186
Организация работы клиентов удаленного офиса с корпоративными DNS- и NTP-серверами.....	189
Настройка точки доступа сети Wi-Fi	191
Глава 10. Настройка маршрутизации.....	193
Общие сведения о маршрутизации.....	194
Принципы маршрутизации	195
Просмотр таблиц маршрутизации	199
Настройка статической маршрутизации	201
Создание статических маршрутов	201
Удаление статических маршрутов	202
Настройка балансировки IP-трафика	203
Просмотр статических маршрутов	204
Настройка параметров маршрутов от DHCP/PPP-протокола	205
Настройка административной дистанции для маршрутов DHCP-протокола	205
Настройка метрики для маршрутов DHCP-протокола	206
Настройка метрики для маршрутов PPP-протокола	207
Изменение метрики по умолчанию для маршрутов протоколов DHCP и PPP	208
Просмотр настроек DHCP в режиме клиента.....	208
Просмотр маршрутов DHCP/PPP-протокола.....	209
Настройка динамической маршрутизации	210
Общие сведения о протоколе OSPF	210
Настройка параметров динамической маршрутизации по протоколу OSPF	212
Настройка протокола OSPF	213
Настройка перераспределения маршрутов	215
Просмотр настроек протокола OSPF	216
Настройка протокола OSPF в сетях со множественным доступом	217
Просмотр состояния каналов связи между маршрутизаторами	218
Просмотр информации о соседних маршрутизаторах	219
Просмотр маршрутов, сформированных по протоколу OSPF	220
Настройка функции L2OverIP для обеспечения работоспособности протоколов динамической маршрутизации	220
Настройка параметров L2OverIP	221
Настройка протокола OSPF	222
Настройка функции MultiWAN	225
Резервирование каналов доступа в интернет	226

Обеспечение связи между прокси-сервером и его клиентами, находящимися в разных сетях	228
Глава 11. Настройка параметров безопасности	231
Изменение пароля пользователя	232
Изменение способа аутентификации пользователя	233
Подключение к веб-интерфейсу по HTTPS	235
Просмотр информации об установленных ключах	239
Обновление справочников и ключей	240
Добавление резервного набора персональных ключей (РНПК)	243
Когда требуется добавлять РНПК.....	243
Порядок добавления РНПК	243
Обновление ключей при истечении срока их действия	245
Контроль целостности файлов ViPNet Coordinator HW	247
Глава 12. Настройка системных параметров	249
Настройка даты и времени.....	250
Установка серийного номера	251
Настройка расписания перезагрузки	252
Настройка параметров использования файла подкачки	253
Установка режима повышенной производительности	254
Управление копиями конфигурации VPN	255
Виды копий конфигурации VPN	255
Создание копии конфигурации VPN	255
Просмотр списка копий конфигурации VPN	256
Восстановление настроек из копии конфигурации VPN	257
Удаление копии конфигурации VPN	257
Резервное копирование и восстановление настроек	259
Назначение экспорта и импорта справочников, ключей и настроек.....	259
Экспорт справочников, ключей и настроек.....	260
Импорт справочников, ключей и настроек	262
Использование резервного раздела в исполнениях ViPNet Coordinator HW с двумя накопителями.....	263
Сброс настроек ViPNet Coordinator HW к заводским	264
Глава 13. Обеспечение отказоустойчивости	267
Настройка системы защиты от сбоев.....	268
Назначение системы защиты от сбоев.....	268
Работа системы защиты от сбоев в одиночном режиме	268

Работа системы защиты от сбоев в режиме кластера горячего резервирования	269
Развертывание кластера горячего резервирования	272
Настройка параметра checkonlyidle	278
Типовая схема кластера	280
Развертывание кластера совместно с DHCP-сервером и DHCP-relay.....	283
Запуск и завершение работы системы защиты от сбоев	284
Переключение кластера горячего резервирования в одиночный режим работы	284
Замена сбойного узла кластера на новый без прерывания трафика	285
Просмотр информации о работе системы защиты от сбоев	286
Текущее состояние системы защиты от сбоев	286
Журнал переключений	288
Работа кластера с коммутационным оборудованием	289
Организация электропитания от ИБП	290
Глава 14. Настройка транспортного сервера MFTP	293
Назначение транспортного сервера MFTP	294
Настройка протоколирования событий транспортного сервера MFTP	295
Настройка прямой маршрутизации между сетями ViPNet.....	296
Настройка прямой маршрутизации без обработки транспортных конвертов на шлюзовых координаторах.....	296
Настройка маршрутизации без обработки транспортных пакетов на шлюзовом координаторе своей сети.....	298
Глава 15. Обновление программного обеспечения	300
Порядок обновления ПО	301
Особенности обновления ПО	302
Проверка ЭП файла обновления	304
Предварительные действия	304
Установка корневого сертификата и списка отозванных сертификатов	304
Проверка электронной подписи.....	306
Проведение обновления ПО	308
Удаленное обновление ПО	308
Локальное обновление ПО	309
Обновление ПО на кластере горячего резервирования	310
Проверка корректности обновления	312
Глава 16. Мониторинг состояния ViPNet Coordinator HW и просмотр журналов	313
Мониторинг по SNMP.....	314
Настройка SNMP-агента	316

Мониторинг кластера	319
Пример настройки системы мониторинга	321
Просмотр статистики передачи данных	324
Работа с журналом регистрации IP-пакетов	325
Просмотр журнала регистрации IP-пакетов	325
Экспорт журнала регистрации IP-пакетов на USB-носитель	332
Экспорт журнала регистрации IP-пакетов по сети в формате CEF	333
Работа с журналом MFTP	336
Работа с системным журналом	338
Просмотр системного журнала	339
Настройка параметров ведения системного журнала	340
Экспорт диагностической информации на компьютер	342
Экспорт диагностической информации на USB-носитель	344
Работа с журналом DNS-запросов	345
 Приложение А. Сетевые фильтры по умолчанию	 347
 Приложение В. Пользовательские группы протоколов по умолчанию	 351
 Приложение С. Типы событий в журнале регистрации IP-пакетов	 353
 Приложение D. Структура записи журнала регистрации IP-пакетов для экспорта по сети в формате CEF	 358
 Приложение Е. Записи системного журнала, связанные с аутентификацией и настройкой оборудования	 361
 Приложение F. Базы управляющей информации (MIB) для протокола SNMP	 371
MIB-II System	371
IF-MIB	375
IP-MIB	378
IP-FORWARD-MIB	384
Ethernet-Like MIB	388
TCP-MIB	389
UDP-MIB	392
HOST-RESOURCES-MIB	393
LM-SENSORS-MIB	399
NOTIFICATION-LOG-MIB	404
UCD-SNMP-MIB	405
UCD-DISKIO-MIB	409

NET-SNMP-MIB.....	410
SNMP-MPD-MIB	415
SNMP-TARGET-MIB	415
SNMP-USER-BASED-SM-MIB	415
SNMP-VIEW-BASED-ACM-MIB	417
SNMP-FRAMEWORK-MIB	419
VIPNET-MIB.....	420
MonitoringAgent-MIB.....	422
System-MonitoringAgent-MIB.....	423
 Приложение G. Список служб в составе ПО ViPNet Coordinator HW	 424
 Приложение H. Поддерживаемые MIME-типы	 427
 Приложение I. Использование ViPNet Coordinator HW в качестве МЭ типа А четвертого класса защиты	 431
 Приложение J. Возможные неполадки	 432
Блокирование медиа-сессии SIP-соединения при завершении сигнальной сессии по тайм-ауту	432
Для новых вариантов ключей отображается старая дата обновления	432
Истек срок действия пароля администратора	433
При переносе клиента за другой координатор обновления из ЦУСа не применяются	433
Проблема с обновлением ПО в кластере с ограничением по IP-адресам.....	434
Проблемы при работе системы защиты от сбоев	434
Циклическая перезагрузка	435
Циклическая перезагрузка ViPNet Coordinator HW после смены пользователя в ViPNet ЦУС или Prime.....	436
Некорректные настройки параметров работы системы защиты от сбоев.....	436
Нарушена целостность конфигурационных файлов	437
Циклическая перезагрузка узла кластера после смены пароля пользователя.....	437
Блокируется защищенный транзитный трафик	438
Блокируется исходящий трафик от ViPNet Coordinator HW в режиме «Со статической трансляцией адресов»	439
Получено сообщение от сервера мониторинга о недостатке дискового пространства	440
 Приложение K. Термины и сокращения.....	 441



Введение

О документе	13
Общие сведения о ViPNet Coordinator HW	14
Обратная связь	15

О документе

В документе описаны основные сценарии настройки ViPNet Coordinator HW с помощью командного интерпретатора, а также работа с журналами ViPNet Coordinator HW.

Для кого предназначен документ

Документ предназначен для администраторов сетей ViPNet.

Соглашения документа

Обозначение	Описание
Название	Название элемента интерфейса: окна, вкладки, поля, кнопки, ссылки
Клавиша+Клавиша	Сочетание клавиш: нажмите первую клавишу и, не отпуская её, нажмите вторую
Меню > Команда	Последовательность элементов или действий
Код	Имя файла, путь, фрагмент кода или команда в командной строке



Примечание. В документе могут присутствовать снимки интерфейса из предыдущих версий продукта. Поэтому некоторые элементы интерфейса, которые не влияют на понимание текста, могут выглядеть не так, как в продукте.

Обозначения при описании команд в документе:

- Команды, которые участвуют в сценарии администратора, обозначены символом #
`hostname# admin config list`
- Команды, которые участвуют в сценарии пользователя, обозначены символом >
`hostname> firewall local show`
Все команды, которые доступны пользователю, доступны и администратору.
- Параметры заключены в угловые скобки:
`inet bonding delete <номер>`
- Необязательные параметры или ключевые слова заключены в квадратные скобки:
`firewall <тип> add name @<имя> <состав> [exclude <исключения>]`
- Допустимые варианты заключены в фигурные скобки и разделены вертикальной чертой:
`inet ntp mode {on | off}`

Общие сведения о ViPNet Coordinator HW

ViPNet Coordinator HW — программно-аппаратный комплекс, который выполняет функции VPN-шлюза в сети ViPNet.

ViPNet Coordinator HW обеспечивает безопасную передачу данных между защищенными сегментами сети ViPNet, а также фильтрует IP-трафик. Он выполняет роль [координатора в сети ViPNet](#). Роль назначается администратором сети.

Кроме аппаратных исполнений ViPNet Coordinator HW поставляется в виртуализированном исполнении ViPNet Coordinator VA, которое разворачивается на платформе виртуализации.

Описание всех функций и исполнений ViPNet Coordinator HW см. в документе «Подготовка к работе».

Обратная связь

Контактная информация

- Единый многоканальный телефон:
+7 (495) 737-6192,
8 (800) 250-0-260 — бесплатный звонок из России (кроме Москвы).
- Служба поддержки: hotline@infotecs.ru.
Форма для обращения в службу поддержки через сайт.
Телеграм-канал поддержки: t.me/vhd21
Телефон для клиентов с расширенной поддержкой: +7 (495) 737-6196.
- Отдел продаж: soft@infotecs.ru.

Дополнительная информация на сайте ИнфоТеКС

- [О продуктах ViPNet](#).
- [О решениях ViPNet](#).
- [Часто задаваемые вопросы](#).
- [Форум пользователей продуктов ViPNet](#).

Если вы обнаружили уязвимости в продуктах компании, сообщите о них по адресу security-notifications@infotecs.ru. Распространение информации об уязвимостях продуктов ИнфоТеКС регулируется [политикой ответственного разглашения](#).

1

Работа с командным интерпретатором

Основные сведения по работе с командным интерпретатором	17
Интерфейс командного интерпретатора	18
Локальное подключение к ViPNet Coordinator HW с помощью консоли	21
Удаленное подключение к ViPNet Coordinator HW с помощью протокола SSH	23
Аутентификация пользователя на ViPNet Coordinator HW	25
Аутентификация администратора на ViPNet Coordinator HW	27

Основные сведения по работе с командным интерпретатором

Работа в командном интерпретаторе возможна при следующих способах подключения к ViPNet Coordinator HW:

- [Локальное подключение к ViPNet Coordinator HW с помощью консоли.](#)
- [Удаленное подключение к ViPNet Coordinator HW с помощью протокола SSH.](#)

Командный интерпретатор независимо от способа подключения к ViPNet Coordinator HW запускается автоматически после успешной аутентификации пользователя (см. [Аутентификация пользователя на ViPNet Coordinator HW](#)). После запуска командный интерпретатор находится в режиме пользователя (обозначается символом «>» в приглашении интерпретатора). В режиме пользователя доступны не все команды.

Чтобы перейти в режим администратора, пройдите аутентификацию администратора (см. [Аутентификация администратора на ViPNet Coordinator HW](#)). В режиме администратора доступны все возможные настройки.

В случае одновременной работы нескольких командных интерпретаторов (независимо от типа подключения — локального или удаленного) только один из них может находиться в режиме администратора. При попытке перейти в некотором командном интерпретаторе в режим администратора проверяется режим работы в остальных запущенных сеансах интерпретаторов. Если остальные интерпретаторы находятся в режиме пользователя, то данный интерпретатор переходит в режим администратора. Если один из интерпретаторов находится в режиме администратора, то выводится информация об узле, на котором он запущен, и предложение принудительно завершить работу этого интерпретатора. В случае согласия, работа интерпретатора, находящегося в режиме администратора, завершается, после чего данный интерпретатор переходит в режим администратора.

- Чтобы просмотреть информацию обо всех запущенных сессиях командного интерпретатора, выполните команду `who`.
- Чтобы принудительно завершить любую сессию командного интерпретатора, выполните команду `admin kick {tty<N> | ttyS<N> | pts/<N>}`. Выполнить команду можно только в режиме администратора.
- Чтобы вернуться в режим пользователя, выполните команду `exit`.
- Чтобы выйти из командного интерпретатора, нажмите **Ctrl+D** (в режиме пользователя или администратора) или выполните команду `exit` (в режиме пользователя). В случае выхода из командного интерпретатора в консоли снова появится приглашение для аутентификации пользователя.
- Чтобы прервать длительное выполнение команды, нажмите **Ctrl+C**.

Интерфейс командного интерпретатора

Командный интерпретатор принимает текстовые команды, состоящие из слов, разделённых пробелами, латинских и специальных символов. Для облегчения ввода команд в интерпретаторе предусмотрены:

- [сокращённый ввод команд](#);
- [автодополнение команд](#);
- [контекстная справка](#);
- [сочетания клавиш](#).

Сокращённый ввод команд

Слова команды распознаются по минимальному числу символов, отличающих вводимую команду от других. Например, если есть только две команды, первое слово которых начинается с символа «i» (`inet` и `iplir`), то для указания первой команды достаточно набрать `in`, для указания второй — `ip`. Если после слова `iplir` в качестве второго слова возможны только `stop` и `start`, то для их указания достаточно набрать `sto` и `sta` соответственно. Таким образом, вместо полной команды `iplir start` достаточно ввести `ip sta`.

Если среди введённых символов есть ошибочные, то вся команда считается ошибочной, даже если её можно однозначно определить по первым правильно введённым символам (то есть в приведённом выше примере команда `iplor sta` будет ошибочной).

Автодополнение команд

Если введённые символы однозначно определяют команду или параметр команды, то вы можете автоматически дополнить недостающие символы, нажав **Tab**:

```
hostname> en<Tab>
```

```
hostname> enable _
```

Автодополнение работает только для слова, на котором находится курсор. Автодополнение не работает для параметров, значение которых вы задаёте (например, имена создаваемых объектов).

Контекстная справка

Контекстная справка позволяет вам просмотреть информацию о командах и параметрах, ввод которых возможен в текущей ситуации. Контекстная справка вызывается с помощью символа «?».

Чтобы просмотреть список всех доступных вам групп команд, в приглашении командного интерпретатора введите символ «?»:

```
hostname# ?

inet          a group of commands intended for working with routing,
               interfaces and network tools

failover      manage command for Failover daemon

iplir         manage command for IpLir daemon

webui         manage WebUI service

admin         perform management tasks

mftp          manage MFTP daemon

enable        switch to the administrator mode

exit          exit the command line interface

version       view the versions of the appliance

who           view currently running sessions of the command line interface

vpn           manage VPN services

machine       a group of system commands

debug         manage displaying syslog messages on the current console

ups

service       manage additional services

firewall      firewall's objects commands

alg           manage application-level gateway

hostname# _
```

Левая колонка списка содержит первое слово группы команд, правая — краткое описание ее назначения.

В случае ввода символа «?» в процессе набора команд, интерпретатор предложит вам варианты завершения текущего или следующего слова команды, в зависимости от положения курсора:

```
hostname> machi?
machine a group of system commands
hostname> machi_
hostname> machine ?

halt          turn off the software and hardware appliance

reboot        reboot the software and hardware appliance

reboot-schedule set reboot-schedule option

show          a group of commands for viewing system settings

hostname> machine _
```

После информации о вариантах завершения команды отображается приглашение интерпретатора с ранее введенной командой для редактирования. Изменять команды можно как обычно: стирать символы клавишей **Backspace** или **Delete**, перемещаться по тексту с помощью клавиш со стрелками влево и вправо.

Сочетания клавиш

Сочетание клавиш	Действие
Ctrl+U	стереть всю команду
Ctrl+K	стереть всё от курсора до конца строки
Ctrl+A	перейти в начало строки
Ctrl+E	перейти в конец строки
Ctrl+B	перейти на один символ назад
Ctrl+F	перейти на один символ вперед
Ctrl+H	стереть символ перед курсором
Ctrl+W	стереть слово перед курсором

Локальное подключение к ViPNet Coordinator HW с помощью КОНСОЛИ



Примечание. Локальное подключение к ViPNet Coordinator HW с помощью консоли доступно только для аппаратных платформ.

Чтобы локально подключиться к ViPNet Coordinator HW:

- 1 Выберите и настройте консоль для подключения к ViPNet Coordinator HW (см. [Выбор консоли при локальном подключении к ViPNet Coordinator HW](#)).
- 2 Выполните аутентификацию пользователя (см. [Аутентификация пользователя на ViPNet Coordinator HW](#)).
- 3 Чтобы настраивать параметры ViPNet Coordinator HW, выполните аутентификацию администратора (см. [Аутентификация администратора на ViPNet Coordinator HW](#)).



Примечание. Если на ViPNet Coordinator HW истекают или уже истекли сроки действия [персонального ключа пользователя](#) или [ключа защиты ключей обмена](#), то на экране появится соответствующее сообщение. Обратитесь к администратору сети ViPNet для обновления ключей.

Выбор консоли при локальном подключении к ViPNet Coordinator HW

При локальном подключении к ViPNet Coordinator HW в качестве консоли вы можете использовать:

- монитор и клавиатуру;
- компьютер, подключенный к консольному порту ViPNet Coordinator HW (кроме HW10 F1).



Примечание. Разъем консольного порта может отличаться на разных аппаратных платформах ViPNet Coordinator HW. Подробнее об аппаратных платформах см. документ «Подготовка к работе».

При подключении к ViPNet Coordinator HW (например, с помощью PuTTY) установите значения параметров консольного порта.

Таблица 1. Параметры консольного порта для подключения консоли

Параметр	Значение
Speed (скорость обмена данными)	38400
Data (размер данных)	8
Parity (четность)	None
Stopbits (стоповые биты)	1
Тип терминала	VT100+
Flow Control	None



Примечание. В ViPNet Coordinator HW поддерживается только кодировка KOI8-R. Поэтому при подключении к ViPNet Coordinator HW с использованием другой кодировки возможны проблемы при вводе с клавиатуры и отображении на экране символов нелатинского алфавита.

К ViPNet Coordinator HW локально можно подключиться с помощью нескольких консолей. Одновременная работа во всех запущенных консолях возможна только в режиме пользователя. В режиме администратора можно работать только в одной консоли. Поскольку возможна одновременная работа сразу в нескольких консолях, при каждом включении ViPNet Coordinator HW выберите консоль для дальнейшей работы следующим образом:

- 1 На все консоли, подключенные к ViPNet Coordinator HW, выводится приглашение нажать любую клавишу. Если в течение 10 секунд вы не нажмете клавишу ни на одной из консолей, то считается, что клавиша нажата на обычной консоли.
- 2 На консоль, на которой нажата клавиша, выводится список консолей и приглашение выбрать нужную. Если в течение 5 секунд вы не выберите консоль, то автоматически выбирается обычная консоль.

Все последующие сообщения о загрузке ОС и приглашение для входа в систему будут выводиться на выбранной консоли.

Удаленное подключение к ViPNet Coordinator HW с помощью протокола SSH

Удаленное подключение к командному интерпретатору следует осуществлять только с других [защищенных узлов ViPNet](#), связанных с ViPNet Coordinator HW. Связи между узлами [сети ViPNet](#) задаются в ViPNet ЦУС или Prime.



Внимание! Предоставлять доступ к ViPNet Coordinator HW с незащищенных узлов запрещено. Для рабочего места администратора на ViPNet Coordinator HW настройте фильтры, которые разрешают только удаленное управление и передачу данных по служебным протоколам ViPNet.

Для удаленного подключения вы можете использовать любой SSH-клиент с парольным типом аутентификации.



Примечание. В ViPNet Coordinator HW поддерживается только кодировка KOI8-R. Поэтому при подключении к ViPNet Coordinator HW с использованием другой кодировки возможны проблемы при вводе с клавиатуры и отображении на экране символов латинского алфавита.

Чтобы удаленно подключиться к ViPNet Coordinator HW:

- 1 С помощью SSH-клиента подключитесь к ViPNet Coordinator HW, указав необходимые параметры подключения и пароль.
- 2 Если в момент вашего подключения к ViPNet Coordinator HW установлены удаленные сессии других пользователей, и количество сессий максимальное, то соединение закрывается.
- 3 Если во время удаленной сессии с ViPNet Coordinator HW вы не будете выполнять никаких действий в течение 30 минут, то сессия будет прервана. Подробнее см. Мониторинг неактивных удаленных сессий (см. [Мониторинг активности удаленных пользователей](#)).



Примечание. Если на ViPNet Coordinator HW истекают или уже истекли сроки действия [персонального ключа пользователя](#) или [ключа защиты ключей обмена](#), то на экране появится соответствующее сообщение. Обратитесь к администратору сети ViPNet для обновления ключей.

Перевыпуск ключей локального SSH-сервера

Удаленное подключение к ViPNet Coordinator HW обеспечивает локальный SSH-сервер. Если вам необходимо заменить ключи SSH-сервера, например, в вашей компании изменилась политика безопасности, выполните:

```
hostname# admin ssh reset-key local
```

Старый ключ будет удален и сгенерирован новый.

Для просмотра информации о ключах локального SSH-сервера, выполните:

```
hostname# admin ssh show-key local
```

Мониторинг активности удаленных пользователей

При одновременном удаленном подключении по протоколу SSH большого количества пользователей нагрузка на ViPNet Coordinator HW значительно увеличивается, что может привести к сбою в работе. Чтобы этого избежать, ViPNet Coordinator HW выполняет мониторинг активности пользователей в удаленных сеансах.

Активность пользователя в каждом удаленном сеансе проверяется каждые 5 минут. Если пользователь не выполняет никаких действий в течение 30 минут, его сеанс принудительно завершается.

Вы можете изменить максимально допустимое время неактивности сессии пользователя:

- 1 Посмотрите текущее значение:

```
hostname> machine show session-timeout
```

- 2 Установите новое значение:

```
hostname# machine set session-timeout <время>
```

где <время> – время неактивности в минутах, кратное 10. При значении равном 0 сессия не будет завершена при неактивности.

При работе ViPNet Coordinator HW в режиме [кластера горячего резервирования](#) указанные настройки передаются с активного узла кластера на пассивный.

Аутентификация пользователя на ViPNet Coordinator HW

ViPNet Coordinator HW предусмотрено два типа аутентификации пользователя:

- «Пароль». При аутентификации требуется ввести имя учетной записи и [пароль пользователя](#). Каждый раз при вводе пароля вычисляется парольный ключ, который используется для доступа к вашему [персональному ключу](#).
- «Устройство». При аутентификации требуется ввести имя учетной записи, подключить устройство, на котором сохранен персональный ключ, и ввести ПИН доступа к устройству. Этот способ аутентификации применим только при локальном подключении к ViPNet Coordinator HW с помощью консоли.

По умолчанию способ аутентификации задается администратором сети ViPNet при создании [дистрибутива ключей](#). Администратор ViPNet Coordinator HW может изменить способ аутентификации. Подробнее см. раздел [Изменение способа аутентификации пользователя](#).

Особенности аутентификации

- При использовании ViPNet Prime в качестве управляющего ПО, ViPNet Coordinator HW поддерживает только способ аутентификации «Пароль».
- При удаленном подключении по SSH или через веб-интерфейс применяется способ аутентификации «Пароль».
- В текущей версии ViPNet Coordinator HW для аутентификации могут использоваться только внешние устройства Rutoken Lite производства компании «Актив».
- Для исполнения ViPNet Coordinator VA: платформы виртуализации Microsoft Hyper-V и Oracle VM Server не поддерживают подключение USB-устройств к виртуальной машине. Поэтому аутентификация с помощью устройства на этих платформах невозможна.

Аутентификация пользователя

- 1 Введите имя учетной записи пользователя `user`.
- 2 Для аутентификации с помощью пароля — введите пароль пользователя. Пароль совпадает с паролем дистрибутива ключей, если он не изменялся в процессе работы с ViPNet Coordinator HW. При вводе пароля на экране ничего не отображается. Если введен неверный пароль, появится соответствующее сообщение и приглашение для повторной аутентификации.



Примечание. Для предотвращения подбора пароля методом перебора в ViPNet Coordinator HW реализована прогрессивная задержка ввода пароля. После трех неудачных попыток аутентификации возможность ввода пароля станет доступна через 3 секунды, а каждая последующая ошибка увеличит эту задержку. Счетчик ошибок сбрасывается при успешном вводе пароля, а также после десятой ошибки. Перезагрузка ViPNet Coordinator HW не сбрасывает счетчик ошибок.

Все неудачные попытки ввода пароля автоматически записываются в системный журнал.

3 Для аутентификации с помощью устройства:

3.1 Подключите к ViPNet Coordinator HW USB-носитель с [персональным ключом](#).

3.2 Введите ПИН доступа к подключенному устройству в ответ на сообщение `Enter PIN code`.

Если введен неверный ПИН, аутентификация не будет выполнена.

В случае успешной аутентификации командный интерпретатор будет запущен в режиме пользователя. Вы можете приступить к работе с ViPNet Coordinator HW. О том, как перейти в режим администратора, см. раздел [Аутентификация администратора на ViPNet Coordinator HW](#).

Аутентификация администратора на ViPNet Coordinator HW

Вы можете перейти в режим администратора, если выполнили аутентификацию в качестве пользователя на ViPNet Coordinator HW:

- 1 Выполните команду `enable`.
- 2 Введите пароль администратора ViPNet Coordinator HW. Если у вас нет пароля администратора, обратитесь к [администратору вашей сети ViPNet](#).

При вводе пароля на экране ничего не отображается. Если введен неверный пароль, появится соответствующее сообщение и приглашение для повторной аутентификации.



Примечание. Для предотвращения подбора пароля методом перебора в ViPNet Coordinator HW реализована прогрессивная задержка ввода пароля. После трех неудачных попыток аутентификации возможность ввода пароля станет доступна через 3 секунды, а каждая последующая ошибка увеличит эту задержку. Счетчик ошибок сбрасывается при успешном вводе пароля, а также после десятой ошибки. Перезагрузка ViPNet Coordinator HW не сбрасывает счетчик ошибок.

Все неудачные попытки ввода пароля автоматически записываются в системный журнал.

В случае ввода верного пароля командный интерпретатор перейдет в режим администратора — появится символ «#» в приглашении интерпретатора. В этом режиме вы можете выполнить те настройки, которые были недоступны в режиме пользователя.

На ViPNet Coordinator HW могут авторизоваться три типа администраторов:

- администратор сетевого узла ViPNet Coordinator HW;
- администратор группы сетевых узлов ViPNet;
- администратор сети ViPNet.

2

Настройка подключения к сети

Настройка сетевых интерфейсов Ethernet	29
Назначение дополнительных IP-адресов	31
Подключение к сети Wi-Fi	32
Подключение по PPPoE	34
Подключение к мобильной сети	36
Использование динамических интерфейсов	41
Изменение значения MTU для сетевых интерфейсов	42
Поддержка технологии VLAN	43
Использование агрегированных сетевых интерфейсов	46
Настройка получения параметров от DHCP-сервера	55

Настройка сетевых интерфейсов Ethernet



Внимание! Не изменяйте параметры сетевых интерфейсов, задействованных в работе кластера (см. [Развертывание кластера горячего резервирования](#)). Это может вызвать сбой в его работе.

Сетевым интерфейсам Ethernet ViPNet Coordinator HW присваиваются имена `eth0`, `eth1` и так далее по количеству интерфейсов. Чтобы настроить подключение к сети, задайте параметры сетевых интерфейсов Ethernet:

1 Задайте класс интерфейса:

```
hostname# inet ifconfig <имя интерфейса> class {access | trunk | slave}
```

- По умолчанию назначен класс `access`.
- Если требуется, чтобы данный интерфейс обрабатывал трафик из нескольких VLAN, назначьте ему класс `trunk`.
- Если вы хотите объединить данный интерфейс с другими в составе агрегированного интерфейса, назначьте ему класс `slave`.



Примечание. Исполнение HW10 не поддерживает агрегирование сетевых интерфейсов.

2 Присвойте интерфейсу IP-адрес одним из способов:

- Чтобы включить на интерфейсе режим автоматического получения параметров от DHCP-сервера:

```
hostname# inet ifconfig <имя интерфейса> dhcp
```

Дополнительно см. [Настройка получения параметров от DHCP-сервера](#).



Внимание! IP-адрес, получаемый от DHCP-сервера, не должен принадлежать подсетям, в которые входят IP-адреса других интерфейсов ViPNet Coordinator HW.

- Чтобы присвоить интерфейсу статический IP-адрес:

```
hostname# inet ifconfig <имя интерфейса> address <IP-адрес> netmask <маска сети>
```

Например, чтобы присвоить интерфейсу `eth1` IP-адрес `192.168.10.1` и маску подсети `255.255.255.0`, выполните:

```
hostname# inet ifconfig eth1 address 192.168.10.1 netmask 255.255.255.0
```

Если до выполнения этой команды интерфейс работал в режиме DHCP, данные о DNS- и NTP-серверах, полученные по протоколу DHCP, будут потеряны.

Внимание! При задании IP-адреса действуют ограничения:



- нельзя задать IP-адрес 0.*.*.*;
 - нельзя задать маски подсети 0.0.0.0, 255.255.255.254 и 255.255.255.255;
 - для разных сетевых интерфейсов нельзя задать IP-адреса, относящиеся к одной подсети или широковещательному каналу (broadcast).
-

3 Включите сетевой интерфейс:

```
hostname# inet ifconfig <имя интерфейса> up
```

4 Если хотя бы на одном используемом интерфейсе Ethernet задано автоматическое получение маршрутов от DHCP-сервера, проверьте, что хотя бы один маршрут по умолчанию получен:

```
hostname# inet show routing
```

Если маршрут по умолчанию не был получен, [задайте статический маршрут по умолчанию](#). Маршрут по умолчанию также необходимо задать при ручной настройке параметров интерфейса Ethernet.



Внимание! Убедитесь, что заданный в маршруте шлюз по умолчанию доступен для сетевого узла ViPNet Coordinator HW. Если шлюз по умолчанию окажется недоступен, многие функции ViPNet Coordinator HW не будут работать (например, виртуальные IP-адреса, обработка прикладных протоколов, сетевые службы и другие).

5 Выполните дополнительные настройки:

- Задайте дополнительные IP-адреса для сетевого интерфейса.
- [Настройте маршрутизацию](#).
- [Измените значение MTU для сетевого интерфейса](#).

6 При необходимости вы можете сбросить параметры сетевого интерфейса:

```
hostname# inet ifconfig <имя интерфейса> reset
```

Назначение дополнительных IP-адресов

С помощью дополнительных IP-адресов вы можете разделить сеть на логические подсети. Например, чтобы разрешить доступ в интернет только некоторым узлам, когда в остальной сети доступ закрыт.



Примечание. Максимальное количество сетевых интерфейсов в ViPNet Coordinator HW — 128 (как физических, так и виртуальных, включая loopback).

Чтобы добавить дополнительный IP-адрес:

- 1 Проверьте, что на интерфейсе задан класс `access`:

```
hostname# inet show interface <имя интерфейса>
```

В выводе команды в строке `Class` значение `access`.

- 2 Добавьте дополнительный IP-адрес:

```
hostname# inet ifconfig <имя интерфейса> address add <IP-адрес> netmask <маска подсети>
```

После выполнения команды создается виртуальный сетевой интерфейс с именем `<интерфейс>:номер`, где `номер` — очередной свободный номер (нумерация дополнительных IP-адресов начинается с 0). На созданном виртуальном интерфейсе задаются указанные IP-адрес и маска подсети.

Например, чтобы добавить на интерфейс `eth1` дополнительный IP-адрес 192.168.10.2 с маской подсети 255.255.255.0:

```
hostname# inet ifconfig eth1 address add 192.168.10.2 netmask 255.255.255.0
```

Будет создан виртуальный интерфейс `eth1:0` с IP-адресом 192.168.10.2 и маской подсети 255.255.255.0.

Подключение к сети Wi-Fi

Внимание!



- Подключение к сети Wi-Fi возможно только в исполнениях ViPNet Coordinator HW50 A, B и ViPNet Coordinator HW100 C.
- Адаптер Wi-Fi отображается как сетевой интерфейс wlan0.
- Не поддерживается подключение к сети Wi-Fi в кластере.
- Не поддерживается одновременное использование ViPNet Coordinator HW как клиента и точки доступа Wi-Fi.
- Все настройки подключения в сети Wi-Fi должны выполняться при выключенном сетевом интерфейсе Wi-Fi (wlan0). Если интерфейс включен, выключите его:

```
hostname# inet wifi mode off
```

Чтобы подключить ViPNet Coordinator HW к сети Wi-Fi:

- 1 Проверьте параметры работы и текущие настройки адаптера Wi-Fi:

```
hostname# inet show wifi
```

- 2 Переключите интерфейс Wi-Fi в режим клиента, если он работает в режиме точки доступа:

```
hostname# inet wifi role client
```

- 3 Укажите параметры доступа к сети Wi-Fi, к которой вы хотите подключиться:



Примечание. Список доступных сетей Wi-Fi (кроме скрытых) можно посмотреть командой:

```
hostname# inet wifi scan
```

- Если аутентификация в сети не требуется:

```
hostname# inet wifi client authentication open
```

Затем введите имя сети.

- Если для аутентификации используется режим WPA-PSK или WPA2-PSK:

```
hostname# inet wifi client authentication {wpa-psk | wpa2-psk}
```

Затем введите имя сети и пароль.

- 4 Сетевой интерфейс wlan0 работает в режиме автоматического получения параметров от DHCP-сервера, но вы можете настроить дополнительные параметры:

- выключить автоматическое получение адресов DNS-серверов:

```
hostname# inet ifconfig <имя интерфейса> dhcp dns off
```

- выключить автоматическое получение NTP-серверов:

```
hostname# inet ifconfig <имя интерфейса> dhcp ntp off
```

- выключить автоматическое получение маршрутов от DHCP-сервера:

```
hostname# inet ifconfig <имя интерфейса> dhcp route off
```

- задать специфичное значение метрики для маршрутов, полученных от DHCP-сервера:

```
hostname# inet ifconfig <имя интерфейса> dhcp route-metric <1-255>
```

Вы можете задать ряд параметров, которые будут присваиваться маршрутам DHCP-сервера (см. [Настройка параметров маршрутов от DHCP/PPP-протокола](#)). Также вы можете просмотреть параметры [режима DHCP на сетевых интерфейсах](#).

5 Включите сетевой интерфейс wlan0:

```
hostname# inet wifi mode on
```

6 Проверьте параметры подключения к сети Wi-Fi:

```
hostname# inet show wifi
```

Подключение по PPPoE

Внимание!



- При настройке используйте доверенный PPPoE-сервер.
 - Убедитесь, что у PPPoE-сервера, который вы используете, включена опция CHAP Rechallenge — периодическая аутентификация установленного соединения. Это защитит PPPoE-сервер от подмены координатора в сети.
-

Вы можете настроить PPPoE-соединение для подключения узлов к сети провайдера:

1 Создайте виртуальный PPPoE-интерфейс:

```
hostname# inet pppoe add <номер> [comment <комментарий>],
```

где <номер> — номер PPPoE-интерфейса. Возможные значения: 0–100.

<комментарий> — комментарий к PPPoE-интерфейсу, например, имя провайдера.

2 Свяжите виртуальный PPPoE-интерфейс с физическим интерфейсом:

```
hostname# inet ifconfig pppoe <номер> if-set <Ethernet-интерфейс>
```

Вы можете добавить несколько PPPoE-интерфейсов на один Ethernet-интерфейс. Включенным может быть только один интерфейс, остальные интерфейсы — резервные. При потере соединения вы сможете переключиться на резервный PPPoE-интерфейс.

3 Задайте способ аутентификации:

```
hostname# inet ifconfig pppoe <номер> auth {chap | pap}
```

4 Задайте имя и пароль пользователя для PPPoE-подключения:

```
hostname# inet ifconfig pppoe <номер> username <имя пользователя>
```

```
hostname# inet ifconfig pppoe <номер> password <пароль пользователя>
```

Допустимые символы для имени и пароля пользователя: A–Z, a–z, 0–9, ! # \$ % () * + , . / : ; < = > @ [] _ { | } ~. Максимальная длина — 255 символов. Лишние символы обрезаются.

5 Выполните [дополнительные настройки](#).

6 Включите PPPoE-интерфейс:

```
hostname# inet ifconfig pppoe <номер> enable
```

Примечание. PPPoE-интерфейс можно включить, только если:



- Включен связанный физический интерфейс.
 - Указаны основные настройки: физический интерфейс, имя и пароль пользователя, тип аутентификации.
-

7 Чтобы посмотреть информацию об установленном соединении:

```
hostname> inet show pppoe session
```

После включения PPPoE-интерфейса установится соединение с PPPoE-сервером.

При потере соединения ViPNet Coordinator HW будет автоматически пытаться подключиться к PPPoE-серверу, пока соединение не будет восстановлено. Это может занять длительное время. Для подключения к другому PPPoE-серверу добавьте на Ethernet-интерфейс резервный PPPoE-интерфейс и включите резервный интерфейс:

```
hostname# inet ifconfig pppoe <номер> enable
```

Резервный интерфейс станет активным, подключение к PPPoE-серверу будет выполняться через него.

Особенности при работе в кластере

- PPPoE-соединение в кластере не резервируется: при потере соединения активный узел пытается подключиться к PPPoE-серверу до восстановления соединения. Узлы кластера не переключаются, даже если PPPoE-соединение не будет восстановлено.
- На пассивном узле кластера доступен только просмотр PPPoE-интерфейсов и их настроек.
- При переключении узлов кластера PPPoE-соединение восстанавливается автоматически.

Дополнительные настройки PPPoE-интерфейса



Примечание. Выполнять настройки можно только на выключенном PPPoE-интерфейсе. Чтобы выключить PPPoE-интерфейс, выполните команду:

```
hostname# inet ifconfig pppoe <номер> disable
```

- Получать маршрут по умолчанию.

Чтобы при первом подключении к сети провайдера на ViPNet Coordinator HW получать маршрут по умолчанию, в котором в качестве шлюза указан адрес сети вашего провайдера:

```
hostname# inet ifconfig pppoe <номер> defaultroute on
```

- Изменить метрику маршрута по умолчанию:

```
hostname# inet ifconfig pppoe <номер> route-metric <метрика>,
```

где <метрика> — число из интервала 1–255 (по умолчанию — 70).

- Отказаться от получения адреса DNS-сервера от PPPoE-сервера:

```
hostname# inet ifconfig pppoe <номер> usepeerdns off
```

- Вручную задать IP-адрес PPPoE-интерфейса.

По умолчанию PPPoE-интерфейс получает динамический IP-адрес от PPPoE-сервера. Если провайдер передал вам IP-адрес, назначьте его статически:

```
hostname# inet ifconfig pppoe <номер> address <IP-адрес> netmask <маска подсети>
```

- Изменить значение MTU:

```
inet ifconfig pppoe <номер> mtu <значение MTU>,
```

где возможные значения MTU — 1280–9000 (по умолчанию — 1492 байта).

Подключение к мобильной сети



Внимание! Подключение к мобильной сети возможно только в исполнениях со встроенным модемом (см. исполнения в документе «Описание аппаратных платформ»).

Подключение к мобильной сети выполняется по протоколу PPP. Для подключения:

- 1 Приобретите SIM-карту. Вы можете пользоваться услугами любого оператора мобильной связи.
- 2 Подключите необходимые услуги мобильного оператора.
- 3 Убедитесь, что SIM-карта не имеет ограничений на использование в модемах. Подробную информацию об условиях подключения к интернету можно получить у оператора мобильной связи.
- 4 Установите SIM-карту в соответствующий слот ViPNet Coordinator HW (см. документ «Описание аппаратных платформ», раздел «Установка SIM-карты»).
- 5 Перед подключением к мобильной сети вы можете выполнить предварительную настройку ViPNet Coordinator HW (см. [Предварительная настройка](#)).

Настройка подключения

Настройки для подключения к мобильной сети зависят от оператора связи. При включении модема ViPNet Coordinator HW запрашивает коды MCC и MNC, комбинация которых используется для идентификации оператора в списке мобильных операторов. Если нужная комбинация найдена, устанавливаются настройки подключения к мобильной сети соответствующего оператора.

Предварительная настройка

Перед организацией подключения к мобильной сети вы можете:

- Отказаться от получения маршрута по умолчанию

При первом подключении модема к мобильной сети на ViPNet Coordinator HW добавляется новый маршрут по умолчанию, в котором в качестве шлюза указан адрес узла сети вашего мобильного оператора. Вы можете отказаться от маршрута по умолчанию:

```
hostname# inet usb-modem route off
```

- Изменить метрику маршрута по умолчанию

Первоначально для нового маршрута задана метрика по умолчанию. Вы можете задать другое значение:

```
hostname# inet usb-modem route-metric <метрика>,
```


где <метрика> число из интервала 1–255.

- Отказаться от добавления адреса DNS-сервера

При первом подключении модема оператор добавляет адрес DNS-сервера. Вы можете отказаться от добавления адреса DNS-сервера:

```
hostname# inet usb-modem dns off
```

Автоматическое определение оператора

- 1 Если на вашей SIM-карте установлен ПИН, задайте его:

```
hostname# inet usb-modem pin <ПИН>
```

- 2 Включите модем:

```
hostname# inet usb-modem mode on
```

- 3 Проверьте параметры подключения к мобильной сети:

```
hostname# inet show usb-modem
```

- Если в выводе присутствуют сообщения о неверном ПИН или отсутствующей (поврежденной) SIM-карте, например:

```
hostname# inet show usb-modem
```

```
3G modem is found
```

```
Cellular provider: MTS
```

```
SIM card: error
```

```
PIN code: 1234
```

```
PIN verification: failed
```

```
...
```

исправьте их (вставьте или замените SIM-карту, задайте правильный ПИН) и заново установите соединение с мобильной сетью:

```
hostname# inet usb-modem mode off
```

```
hostname# inet usb-modem mode on
```

- Если параметры подключения не определились, настройте модем вручную.

Настройка модема вручную

Выбор оператора

- 1 Отключите модем:

```
hostname# inet usb-modem mode off
```

- 2 Проверьте параметры подключения к мобильной сети вашего оператора:

```
hostname# inet show usb-modem provider <код страны> name <оператор>
```

- 3 Если параметры подключения верные, укажите, что для подключения к сети следует использовать заданного оператора:

```
hostname# inet usb-modem select provider <код страны> name <оператор>
```

- 4 Укажите ПИН, если ваша SIM-карта защищена:

```
hostname# inet usb-modem pin <ПИН>
```

- 5 Включите модем:

```
hostname# inet usb-modem mode on
```

Если вашего оператора с необходимыми параметрами подключения нет в базе, добавьте нового.

Добавление нового оператора

- 1 Отключите модем:

```
hostname# inet usb-modem mode off
```

- 2 Запросите настройки параметров у оператора услуг.

- 3 Добавьте нового оператора:

```
hostname# inet usb-modem add provider <код страны> name <имя оператора>
```

- 4 Задайте настройки подключения для нового оператора:

- Укажите коды MCC и MNC:

```
hostname# inet usb-modem provider <код страны> name <оператор> mc <mcc mnc>
```

- Укажите APN:

```
hostname# inet usb-modem provider <код страны> name <оператор> type <apn>
```

- Задайте номер доступа для соединения:

```
hostname# inet usb-modem provider <код страны> name <оператор> phone <телефон>
```

- Укажите имя пользователя и пароль подключения:

```
hostname# inet usb-modem provider <код страны> name <оператор> username <имя>
```

```
hostname# inet usb-modem provider <код страны> name <оператор> password <пароль>
```

- Укажите, что для подключения к сети следует использовать параметры вновь заданного оператора:

```
hostname# inet usb-modem select provider <код страны> name <оператор> [type <apn>]
```

- Укажите ПИН, если ваша SIM-карта защищена:

```
hostname# inet usb-modem pin <ПИН>
```

- 5 Включите модем:

```
hostname# inet usb-modem mode on
```

- 6 Проверьте текущие параметры подключения к мобильной сети:

```
hostname# inet show usb-modem
```

Настройка учета мобильного трафика

ViPNet Coordinator HW позволяет отслеживать количество используемого мобильного трафика и отправлять уведомления о превышении установленного ограничения. Эта функция может быть полезна, когда ViPNet Coordinator HW выходит в интернет через мобильную сеть, а оператор мобильной связи ограничивает объем потребляемого трафика. Превышение допустимого объема трафика может привести к ненужным расходам или потере связи между ViPNet Coordinator HW и управляющим ПО.

Чтобы этого не произошло, вы можете настроить учет используемого трафика и уведомления о превышении разрешенного объема.

Настройка учета используемого трафика

- 1 Укажите день месяца, с которого начинается учет трафика. Например, включите учет трафика с 12 числа каждого месяца:

```
hostname# inet usb-modem traffic-limit billing-day 12
```

- 2 Установите объем мобильного трафика, при котором устройство отправляет уведомление. Например, если ежемесячный пакет мобильного трафика, предоставляемый мобильным провайдером, составляет 1 Гбайт (1024 Мбайта):

```
hostname# inet usb-modem traffic-limit set 1024
```

- 3 Включите счетчик мобильного трафика:

```
hostname# inet usb-modem traffic-limit counter on
```

- 4 Проверьте текущие настройки измерения трафика:

```
hostname# inet usb-modem traffic-limit show
```

Настройка оповещения

- 1 Отключите SMTP-оповещения:

```
hostname# inet smtp mode off
```

- 2 Укажите параметры подключения к SMTP-серверу:

```
hostname# inet smtp server <адрес сервера> login <имя пользователя> password <пароль>
```



Примечание. Вы можете поменять параметры подключения к SMTP-серверу позже:

- Поменять имя пользователя: `hostname# inet smtp login <имя пользователя>`
 - Поменять пароль: `hostname# inet smtp password <пароль>`
-

- 3 При необходимости укажите протокол для защиты подключения к SMTP-серверу:

```
hostname# inet smtp secure {tls | starttls}
```



Примечание. Если вы не хотите использовать защищенное соединение выполните:

```
hostname# inet smtp secure off.
```

При выборе протокола защиты автоматически устанавливается порт подключения:

- o `tls` — 465;
 - o `starttls` — 587;
 - o без использования защиты — 25.
-



Примечание. При смене протокола защиты ViPNet Coordinator HW предложит поменять порт подключения на порт по умолчанию другого протокола.

- 4 Если вы хотите, чтобы ViPNet Coordinator HW подключался к SMTP-серверу через порт, отличный от порта по умолчанию, измените его:

```
hostname# inet smtp port <порт>
```

- 5 Укажите адрес электронной почты, с которого будут приходить оповещения:

```
hostname# inet smtp sender <email>
```

- 6 Укажите адреса электронной почты, на которые будут приходить оповещения:

```
hostname# inet smtp recipients add <email>
```

Чтобы добавить нескольких получателей выполните команду несколько раз с разными адресами электронной почты.

- 7 Проверьте настройки:

```
hostname# inet smtp show
```

- 8 Включите уведомления:

```
hostname# inet smtp mode on
```

- 9 Отправьте тестовое письмо, чтобы проверить работу SMTP-сервера:

```
hostname# inet smtp test
```

Если появится сообщение об ошибке проверьте:

- o настройки SMTP-сервера;
 - o настройки электронной почты;
 - o интерфейс модема (должен быть включен).
-



Примечание. После включения SMTP-оповещений автоматически создается разрешающий сетевой фильтр.

Оповещения отправляются, как только потребляемый трафик достигает 75%, 85% и 95% от указанного ограничения трафика.

Использование динамических интерфейсов

При подключении к мобильной сети (см. [Подключение к мобильной сети](#)) или сети Wi-Fi (см. [Подключение к сети Wi-Fi](#)) на ViPNet Coordinator HW автоматически добавляются и удаляются [динамические интерфейсы](#). По умолчанию разрешена работа динамических интерфейсов, которые входят в одну из следующих групп:

- `ppp` — для подключения к мобильной сети через встроенный модем;
- `pppoe` — для подключения к сети провайдера по PPPoE;
- `wifi` — для подключения к сети Wi-Fi.

Если созданный динамический интерфейс входит в одну из перечисленных групп, он автоматически становится активным и начинает пропускать IP-трафик. Дополнительные настройки динамических интерфейсов, аналогичные настройкам статических интерфейсов через файл `iplir.conf`, не требуются. Для каждой группы динамических интерфейсов на ViPNet Coordinator HW создан отдельный журнал IP-пакетов (файл `iplir.conf`-<интерфейс или группа интерфейсов>), в котором регистрируются записи обо всех IP-пакетах, разрешенных и заблокированных на интерфейсах этой группы.

Вы можете просмотреть список активных динамических сетевых интерфейсов:

```
hostname# iplir show adapters groups
```

Активные динамические интерфейсы также отображаются в списке интерфейсов:

```
hostname# iplir show adapters
```

Также вы можете изменить настройки журнала IP-пакетов для группы интерфейсов:

```
hostname# iplir config <группа интерфейсов>
```

Подробнее о настройке журнала см. «Справочник команд и конфигурационных файлов», раздел «Файл `iplir.conf`-<интерфейс или группа интерфейсов>».

Изменение значения MTU для сетевых интерфейсов

По умолчанию для передачи данных через сетевой интерфейс используется размер **MTU** 1500 байт. Чтобы оптимизировать передачу данных через интерфейс, измените размер MTU:

```
hostname# inet ifconfig <имя интерфейса> mtu <значение MTU>
```

Значение MTU задается в байтах, допустимый диапазон значений — 1280–9000. Чтобы вернуть значение MTU по умолчанию, в качестве значения MTU укажите `auto`.

При изменении значения MTU учитывайте:

- Значение MTU, заданное для интерфейса класса `trunk`, автоматически применяется ко всем виртуальным интерфейсам класса `vlan`, созданным на этом физическом интерфейсе. Изменять значения MTU для отдельных интерфейсов класса `vlan` нельзя.
- Значение MTU, заданное для агрегированного интерфейса, будет использоваться на всех подчиненных физических интерфейсах класса `slave`. Изменять значения MTU для отдельных интерфейсов класса `slave` нельзя. По умолчанию при создании агрегированного интерфейса для всех подчиненных физических интерфейсов автоматически задается значение MTU 1500 байт. Если вы исключаете подчиненный физический интерфейс из агрегированного, для исключенного интерфейса устанавливается значение MTU 1500 байт.
- В зависимости от платформы виртуализации, на которой развернуто исполнение ViPNet Coordinator VA, накладываются следующие ограничения:
 - VMware vSphere — для изменения MTU необходимо предварительно настроить платформу виртуализации, разрешив использование Jumbo-кадров.
 - Oracle VM VirtualBox — изменение MTU не поддерживается при использовании сетевых устройств AMD.

Особенности обработки транзитных IP-пакетов

Если MTU на входящем интерфейсе транзитного IP-пакета меньше, чем MTU на исходящем интерфейсе, то IP-пакет будет отправлен без изменений. Иначе IP-пакет будет фрагментирован в соответствии с MTU исходящего интерфейса.

Поддержка технологии VLAN

Вы можете использовать ViPNet Coordinator HW для обработки трафика в физической локальной сети, разделённой на несколько независимых виртуальных локальных сетей — [VLAN](#). Это возможно благодаря поддержке виртуальных сетевых интерфейсов и тегированию (маркировке) трафика в соответствии со стандартом IEEE 802.1Q. Каждой VLAN присваивается идентификатор из диапазона от 1 до 4094 (значения 0 и 4095 зарезервированы).

Настройте обработку трафика из нескольких VLAN:

- 1 Подключите один из сетевых интерфейсов ViPNet Coordinator HW к коммутатору, объединяющему виртуальные сети.

- 2 Назначьте физическому интерфейсу класс `trunk`:

```
hostname# inet ifconfig <имя физического интерфейса> class trunk
```

- 3 Задайте номера виртуальных интерфейсов, которые будут соответствовать виртуальным сетям за коммутатором:

```
hostname# inet ifconfig <имя физического интерфейса> vlan add <номер виртуального интерфейса>
```

В результате будет создан виртуальный интерфейс вида `<имя физического интерфейса>.номер виртуального интерфейса>`. Например, `eth1.10`.



Примечание. Максимальное количество сетевых интерфейсов в ViPNet Coordinator HW — 128 (как физических, так и виртуальных, включая `loopback`).

- 4 Задайте параметры созданных виртуальных интерфейсов:

- Включите на сетевом интерфейсе режим автоматического получения параметров от DHCP-сервера:

```
hostname# inet ifconfig <имя интерфейса> dhcp
```

При необходимости задайте дополнительные параметры для режима DHCP (см. [Настройка получения параметров от DHCP-сервера](#)).

- Присвойте сетевому интерфейсу статический IP-адрес:

```
hostname# inet ifconfig <имя интерфейса> address <IP-адрес> netmask <маска подсети>
```

Если до выполнения этой команды интерфейс работал в режиме DHCP, данные о DNS- и NTP-серверах, полученные по протоколу DHCP, будут потеряны.

- 5 Добавьте новый виртуальный интерфейс в конфигурацию ViPNet Coordinator HW:

```
hostname# iplir adapter add <имя виртуального интерфейса> traffic on,
```

где `<имя виртуального интерфейса>` — `<имя физического интерфейса>.номер виртуального интерфейса>`. Например, `eth1.10`.

При выполнении команды вы можете не использовать параметр `traffic {on | off}`, в этом случае прохождение трафика через создаваемый виртуальный интерфейс будет включено.

Чтобы включить прохождение трафика через интерфейс, выполните:

```
hostname# iplir adapter traffic <имя виртуального интерфейса> on
```

6 Чтобы оптимизировать передачу данных через интерфейс, **измените значение MTU**.

7 Включите физический интерфейс, к которому подключен коммутатор:

```
hostname# inet ifconfig <физический интерфейс> up
```

При этом автоматически будут включены созданные виртуальные интерфейсы.

Настройка обработки трафика из нескольких VLAN

Две подсети (VLAN10, VLAN20) расположены за координатором, обеспечивающим выход в интернет. Для их подключения на координаторе используется один физический интерфейс `eth1`.

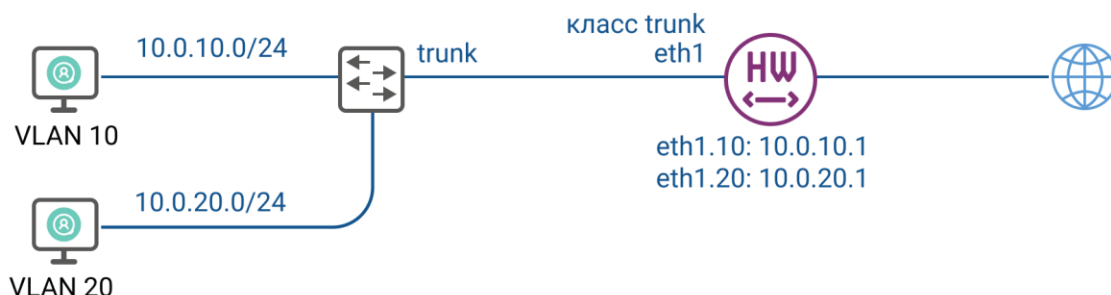


Рисунок 1. Обработка трафика из нескольких VLAN

Чтобы настроить обработку трафика из нескольких VLAN:

1 Подключите сетевой интерфейс `eth1` к коммутатору, объединяющему VLAN.

2 Установите для порта коммутатора режим `trunk`.

3 Назначьте интерфейсу `eth1` класс `trunk`:

```
hostname# inet ifconfig eth1 class trunk
```

4 Создайте виртуальные интерфейсы `eth1.10`, `eth1.20` соответствующие VLAN 10 и 20:

```
hostname# inet ifconfig eth1 vlan add 10
```

```
hostname# inet ifconfig eth1 vlan add 20
```

5 Задайте параметры виртуальных интерфейсов:

```
hostname# inet ifconfig eth1.10 address 10.0.10.1 netmask 255.255.255.0
```

```
hostname# inet ifconfig eth1.20 address 10.0.20.1 netmask 255.255.255.0
```

6 Добавьте виртуальные интерфейсы в конфигурационный файл `iplir.conf` и включите прохождение трафика через них:

```
hostname# iplir adapter add eth1.10 traffic on
```

```
hostname# iplir adapter add eth1.20 traffic on
```


Если вы не указали параметр `traffic on`, прохождение трафика через создаваемый виртуальный интерфейс будет выключено. Вы можете включить его позже:

```
hostname# iplir adapter traffic <имя виртуального интерфейса> on
```

7 Включите интерфейс `eth1`:

```
hostname# inet ifconfig eth1 up
```

При этом автоматически будут включены виртуальные интерфейсы `eth1.10`, `eth1.20`.

Использование агрегированных сетевых интерфейсов



Примечание. Исполнение HW10 не поддерживает агрегирование сетевых интерфейсов.

Агрегирование сетевых интерфейсов — объединение нескольких физических сетевых интерфейсов в один логический. При этом соответствующие каналы связи объединяются на канальном уровне сетевой модели OSI.

Агрегированные сетевые интерфейсы можно использовать:

- Чтобы увеличить пропускную способность сетевых интерфейсов ViPNet Coordinator HW. Например, если вам нужен канал связи с пропускной способностью выше 1 Гбит/с, а на вашем исполнении ViPNet Coordinator HW есть только гигабитные сетевые интерфейсы, вы можете объединить два или три из них в один агрегированный.
- Чтобы повысить надежность канала связи. Если один из объединенных интерфейсов выйдет из строя, агрегированный канал связи продолжит работать.
- Для резервирования канала связи, без увеличения его пропускной способности. Весь трафик будет передаваться через один из подчиненных физических интерфейсов, остальные же начинают работать только при его сбое.

Агрегированные каналы связи целесообразно использовать в отказоустойчивых сетях, по каналам которых передаются большие объемы данных, например, в сетях центров обработки данных.

Чтобы задать распределение нагрузки по подчиненным физическим интерфейсам, выберите один из [режимов работы агрегированного интерфейса](#).



Внимание! Работоспособность агрегированных интерфейсов в ViPNet Coordinator VA не гарантируется. Если вам необходимо использовать агрегированный канал для ViPNet Coordinator VA, настройте его на платформе виртуализации.

Режимы работы агрегированного интерфейса

При создании агрегированного интерфейса указывайте режим его работы, наиболее подходящий для решения ваших задач. В ViPNet Coordinator HW предусмотрено несколько режимов, позволяющих по-разному распределять нагрузку между подчиненными физическими интерфейсами.

balance-rr

Подходит как для балансировки нагрузки на подчиненных интерфейсах, так и для защиты от сбоев. Может применяться в сетях с простой топологией.

Исходящие пакеты, попадающие на агрегированный интерфейс, отправляются через подчиненные поочередно: первый пакет отправляется через один подчиненный интерфейс, второй — через следующий и так далее.

balance-xor

Предназначен для защиты от сбоев и распределения нагрузки таким образом, чтобы пакеты от одного и того же отправителя к одному и тому же получателю всегда отправлялись через один и тот же подчиненный интерфейс.

Для определения подчиненного физического интерфейса, через который отправляется пакет, используется специальная хэш-функция, алгоритм вычисления которой вы можете задать с помощью списка **Алгоритм хэширования** после создания агрегированного интерфейса:

- **layer2** — используются MAC-адреса отправителя и получателя пакета. Одинаковыми считаются сетевые узлы с одинаковыми MAC-адресами;
- **layer2+3** — используются MAC-адреса отправителя и получателя, а также IP-адреса отправителя и получателя (для протокола IPv4). Одинаковыми считаются сетевые узлы с одинаковыми MAC-адресами и IP-адресами;
- **layer3+4** — используются IP-адреса отправителя и получателя, а также номера портов TCP и UDP (при наличии). Одинаковыми считаются сетевые узлы с одинаковыми IP-адресами, а также портами TCP или UDP.

balance-tlb

Подходит для балансировки нагрузки на подчиненных интерфейсах. Используется при передаче большого числа пакетов разного размера, когда более простые режимы не распределяют нагрузку равномерно.

Учитывается размер исходящих пакетов, переданных через каждый из подчиненных физических интерфейсов, и на основе этого выполняется выбор интерфейса, через который будет передан пакет, попавший на агрегированный интерфейс.



Примечание. Для работы в режиме balance-tlb необходимо, чтобы все подчиненные интерфейсы были подключены к сети через коммутатор.

802.3ad

Режим динамического агрегирования с использованием протокола LACP для комплексной балансировки нагрузки. В этом режиме агрегированный интерфейс работает следующим образом:

- Среди подчиненных интерфейсов формируются группы — «агрегаторы», с одинаковой скоростью передачи данных (например, группа гигабитных интерфейсов). Параметры `mtu`, `duplex`, `autoneg` для физических интерфейсов, объединяемых в группу, должны совпадать.
- Один из агрегаторов выбирается активным в соответствии с режимом выбора, задаваемым с помощью списка **Режим агрегатора** после создания агрегированного интерфейса. Вы можете задать один из режимов:
 - **stable** — первоначально выбирается агрегатор с наибольшей суммарной пропускной способностью подчиненных интерфейсов, а в дальнейшем выбор нового агрегатора выполняется только в случае сбоя всех подчиненных интерфейсов текущего агрегатора.
 - **bandwidth** — первоначально выбирается агрегатор с наибольшей пропускной способностью подчиненных интерфейсов, а в дальнейшем, при добавлении, удалении или сбое подчиненных интерфейсов, в агрегаторах производится перегруппировка подчиненных интерфейсов и выполняется выбор нового агрегатора.
 - **count** — первоначально выбирается агрегатор с наибольшим количеством подчиненных интерфейсов, а в дальнейшем, при добавлении, удалении или сбое подчиненных интерфейсов, в агрегаторах производится перегруппировка подчиненных интерфейсов и выполняется выбор нового агрегатора.
- Внутри агрегатора подчиненный интерфейс, через который отправляются исходящие пакеты, выбирается аналогично режиму **balance-xor**.
- С другим сетевым оборудованием происходит обмен пакетами LACP с периодичностью, задаваемой с помощью списка **Частота обмена пакетами** после создания агрегированного интерфейса. В случае выбора параметра **slow** обмен пакетами по протоколу LACP выполняется каждые 30 секунд, в случае выбора параметра **fast** — каждую секунду.

Обмен пакетами позволяет определить сбой подчиненного интерфейса даже в том случае, если этот интерфейс подключен к другому сетевому узлу не напрямую.



Примечание. Механизм MC-LAG (MLAG) не поддерживается. Поэтому физические порты разных ViPNet Coordinator HW нельзя использовать в одном агрегированном интерфейсе.

active-backup

Предназначен для защиты от сбоев, но не для балансировки нагрузки на подчиненных интерфейсах.

Один из подчиненных интерфейсов назначается основным, и все исходящие пакеты отправляются через него. При этом в случае сбоя на основном подчиненном интерфейсе пакеты будут отправляться через другие подчиненные интерфейсы.

broadcast

Предоставляет наибольшую защиту от сбоев. Пакеты, попадающие на агрегированный интерфейс, отправляются через все подчиненные интерфейсы одновременно.



Примечание. Режим broadcast требует специальной настройки сетевого оборудования, предотвращающей дальнейшую передачу по сети нескольких копий пакетов данных.

Если агрегированный канал связи работает в режиме **broadcast**, организовать на этом канале защиту соединения по технологии **L2OverIP** невозможно.

Создание агрегированного сетевого интерфейса



Примечание. Максимальное количество сетевых интерфейсов в ViPNet Coordinator HW — 128 (как физических, так и виртуальных, включая loopback).

- 1 Выберите физические интерфейсы ViPNet Coordinator HW, которые вы хотите объединить, и установите для них **класс slave**:

```
hostname# inet ifconfig <имя интерфейса> class slave
```

- 2 Выберите **режим работы создаваемого агрегированного сетевого интерфейса**, затем создайте агрегированный канал связи:

```
hostname# inet bonding add <номер> mode <режим> slaves <интерфейс 1> [<интерфейс N>],
```

где:

- о <номер> — номер агрегированного сетевого интерфейса. Вы можете создать до восьми агрегированных интерфейсов с номерами от 0 до 7.

В результате выполнения команды создается агрегированный сетевой интерфейс с именем `bond<номер>`.

- о <режим> — режим работы агрегированного интерфейса.
- о <интерфейс 1>, <интерфейс N> — имена физических интерфейсов, подчиненных создаваемому агрегированному сетевому интерфейсу. Количество подчиненных интерфейсов ограничено только общим количеством интерфейсов на ViPNet Coordinator HW. При создании агрегированного интерфейса необходимо задать не менее одного подчиненного интерфейса.

Примечание. Вы можете добавлять или удалять подчиненные интерфейсы в любой момент времени:



- Добавить подчиненный интерфейс: `hostname# inet ifconfig <агрегированный интерфейс> bonding add <имя подчиненного интерфейса>`
- Удалить подчиненный интерфейс: `hostname# inet ifconfig <агрегированный интерфейс> bonding delete <имя подчиненного интерфейса>`

- 3 Измените значение **MTU** созданного агрегированного интерфейса, если хотите оптимизировать передачу данных через него.

- 4 Задайте частоту проверки соединения на подчиненных интерфейсах. По умолчанию соединение на подчиненных физических интерфейсах проверяется каждые 0,1 секунды. Вы можете изменить это значение и задать частоту от 1 до 1000 миллисекунд:

```
hostname# inet ifconfig <имя агрегированного интерфейса> bonding mimon <частота в миллисекундах> ,
```

где <имя агрегированного интерфейса> — bond{0-7}.

- 5 Настройте дополнительные параметры для режимов работы агрегированного интерфейса. Подробнее в разделе [Режимы работы агрегированного интерфейса](#).

- 6 Назначьте агрегированному интерфейсу динамический или статический IP-адрес:

- Чтобы включить на агрегированном интерфейсе режим автоматического получения параметров от DHCP-сервера:

```
hostname# inet ifconfig <имя агрегированного интерфейса> dhcp
```

Дополнительно см. [Настройка получения параметров от DHCP-сервера](#).

- Чтобы присвоить агрегированному интерфейсу статический IP-адрес:

```
hostname# inet ifconfig <имя агрегированного интерфейса> address <IP-адрес> netmask <маска подсети>
```

Если до выполнения этой команды агрегированный интерфейс работал в режиме DHCP, данные о DNS- и NTP-серверах, полученные по протоколу DHCP, будут потеряны.

- 7 Добавьте агрегированный интерфейс в конфигурационный файл `iplir.conf`:

```
hostname# iplir adapter add <имя агрегированного интерфейса> traffic on,
```

Если пока не нужно разрешать трафик через агрегированный интерфейс, выполните команду без параметра `traffic on`. Позже вы сможете разрешить прохождение трафика:

```
hostname# iplir adapter traffic <имя агрегированного интерфейса> on
```

Чтобы агрегированный интерфейс [обрабатывал трафик из нескольких VLAN](#), назначьте ему класс `trunk`. По умолчанию созданному агрегированному сетевому интерфейсу назначается класс `access`.

- 8 Последовательно включите все подчиненные физические интерфейсы:

```
hostname# inet ifconfig <имя физического интерфейса> up
```

- 9 Включите агрегированный интерфейс:

```
hostname# inet ifconfig <имя агрегированного интерфейса> up
```

В результате будет создан агрегированный сетевой интерфейс с именем `bond<номер>`. В дальнейшем вы можете работать с агрегированным интерфейсом так же, как с обычным физическим.

Причины сбоя подчиненного интерфейса

Причинами сбоев в работе подчиненного интерфейса могут быть:

- Неисправность сетевого оборудования (сетевой карты).

- Несогласованность дуплексных скоростных режимов работы сетевого интерфейса.
- Неисправность физического тракта (медного, оптического соединения).

Чтобы выяснить причины сбоя в работе подчиненного интерфейса:

1 Проверьте работоспособность интерфейса:

1.1 Посмотрите информацию об интерфейсе:

```
hostname# inet show interface <интерфейс>
```

1.2 Проверьте вывод в строке Link detected:

Если в строке отображается no, интерфейс не работоспособный.

2 Проверьте режим работы смежного устройства, с которым устанавливается соединение по агрегированному интерфейсу. Режимы работы должны совпадать на обоих устройствах.

3 Посмотрите подробную информацию:

- Об агрегированном интерфейсе:

```
hostname# inet show bonding <интерфейс>
```

- Обо всех агрегированных интерфейсах:

```
hostname# inet show bonding
```

Настройка агрегированного канала между ViPNet Coordinator HW и коммутатором

Настроим агрегированный канал между ViPNet Coordinator HW и коммутатором. Его можно использовать, чтобы обеспечить отказоустойчивость и избежать эффекта "бутылочного горлышка" при передаче данных между координатором и коммутатором.

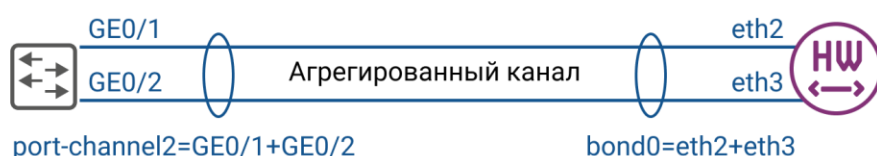


Рисунок 2. Агрегированный канал между ViPNet Coordinator HW и коммутатором

При передаче пакетов от ViPNet-клиентов через коммутатор MAC-адрес, IP-адрес и порт сетевого узла отправителя не подменяется, поэтому в схеме можно использовать режимы, связанные с распределением нагрузки по MAC-адресам, IP-адресам и портам: `balance-xor`, `balance-tlb` и `802.3ad`.

Используем режим `802.3ad`. В этом режиме для определения подчиненного физического интерфейса, через который отправляется пакет, используется хэш-функция, поэтому пакеты от одного и того же отправителя к одному и тому же получателю всегда будут отправляться через один и тот же подчиненный физический интерфейс. Хэш-функция вычисляется по алгоритму `layer3+4`, поэтому учитываются IP-адреса и порты TCP или UDP отправителей и получателей.

Чтобы настроить агрегированный канал:

- 1 Установите для физических интерфейсов eth2 и eth3 класс slave:

```
hostname# inet ifconfig eth2 class slave
hostname# inet ifconfig eth3 class slave
```

- 2 Создайте агрегированный интерфейс с режимом работы 802.3ad:

```
hostname# inet bonding add 0 mode 802.3ad slaves eth2 eth3
```

- 3 Задайте алгоритм вычисления хэш-функции:

```
hostname# inet ifconfig bond0 bonding xmit-hash-policy layer3+4
```

- 4 Добавьте агрегированный интерфейс в конфигурационный файл iplir.conf и включите прохождение трафика:

```
hostname# iplir adapter add bond0 traffic on
```

Если вы не указали параметр traffic on, прохождение трафика через создаваемый виртуальный интерфейс будет выключено. Вы можете включить его позже:

```
hostname# iplir adapter traffic bond0 on
```

- 5 Задайте IP-адрес агрегированного интерфейса:

```
hostname# inet ifconfig bond0 address 10.10.10.1 netmask 255.255.255.0
```

- 6 Соедините интерфейсы eth2 и eth3 с интерфейсами коммутатора.

- 7 Последовательно включите все подчиненные физические интерфейсы:

```
hostname# inet ifconfig eth2 up
hostname# inet ifconfig eth3 up
```

- 8 Включите агрегированный сетевой интерфейс:

```
hostname# inet ifconfig bond0 up
```

Чтобы настроить подключение к агрегированному каналу на коммутаторе, например Cisco Catalyst 2960-S:

- 1 Предварительно объедините интерфейсы коммутатора, к которым подключены ViPNet-клиенты, а также интерфейсы, которые будут подключены к агрегированному каналу, в виртуальную сеть, например VLAN 116.
- 2 Создайте логический интерфейс port-channel, например с номером 2:



Примечание. Для настройки коммутатора:

- перейдите в режим администратора: switch> enable
 - перейдите в режим конфигурирования: switch# configure terminal
-

```
interface Port-channel2
    switchport access vlan 116
    switchport mode access
exit
```


- 3 Укажите физические интерфейсы, например `GigabitEthernet0/1` и `GigabitEthernet0/2`, которые будут входить в состав логического интерфейса `port-channel2`. Они будут использоваться для подключения к агрегированному каналу.

```
interface GigabitEthernet0/1
    switchport access vlan 116
    switchport mode access
    channel-group 2 mode active
exit
interface GigabitEthernet0/2
    switchport access vlan 116
    switchport mode access
    channel-group 2 mode active
exit
```

- 4 Настройте интерфейсы, к которым подключены ViPNet-клиенты:

```
interface GigabitEthernet<номер интерфейса>
    switchport access vlan <номер VLAN>
    switchport mode access
end
```

Теперь трафик между коммутатором и координатором будет идти по агрегированному каналу, состоящему из двух подчиненных физических каналов. В случае отказа одного из подчиненных каналов оставшийся продолжает работу и сессии с отказавшего канала перенаправляются на оставшийся.

Настройка агрегированного канала между двумя ViPNet Coordinator HW

Чтобы обеспечить обмен информацией между сетевыми узлами, стоящими за координаторами, между HW-1 и HW-2 настроим агрегированный канал: объединим на каждом координаторе интерфейсы `eth2` и `eth3` в агрегированный интерфейс `bond0`.

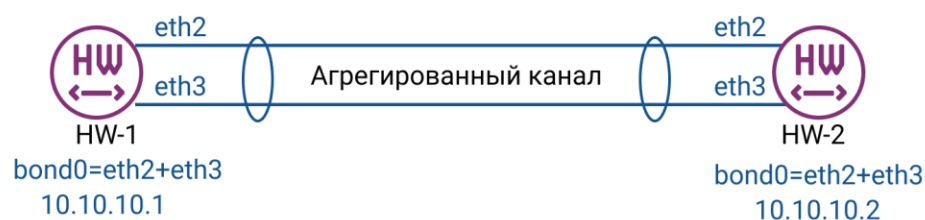


Рисунок 3. Агрегированный канал между двумя ViPNet Coordinator HW

Так как при передаче IP-пакетов через ViPNet Coordinator HW MAC-адрес, порт и IP-адрес сетевого узла отправителя подменяются, в схеме можно использовать режимы `balance-rr`, `active-backup` и `broadcast`.

Используем режим `balance-rr`. В этом режиме исходящие пакеты, попадающие на агрегированный интерфейс, отправляются через подчиненные физические интерфейсы поочередно: первый пакет отправляется через один подчиненный интерфейс, второй пакет — через следующий подчиненный интерфейс и так далее. По агрегированному каналу будет отправляться как защищенный, так и открытый трафик.



Примечание. При использовании режима `balance-rr` порядок следования IP-пакетов может измениться, что недопустимо при передаче некоторых типов данных, например, при передаче видео или звука.

Чтобы настроить агрегированный канал:

1 На обоих HW:

1.1 Установите для физических интерфейсов `eth2` и `eth3` класс `slave`:

```
hostname# inet ifconfig eth2 class slave
hostname# inet ifconfig eth3 class slave
```

1.2 Создайте агрегированный интерфейс с режимом работы `balance-rr`:

```
hostname# inet bonding add 0 mode balance-rr slaves eth2 eth3
```

1.3 Добавьте агрегированный интерфейс в конфигурационный файл `iplir.conf` и включите прохождение трафика:

```
hostname# iplir adapter add bond0 traffic on
```

2 Задайте IP-адреса агрегированных интерфейсов:

○ На HW-1:

```
hostname# inet ifconfig bond0 address 10.10.10.1 netmask 255.255.255.0
```

○ На HW-2:

```
hostname# inet ifconfig bond0 address 10.10.10.2 netmask 255.255.255.0
```

3 Соедините интерфейс `eth2` HW-1 с интерфейсом `eth2` HW-2.

4 Соедините интерфейс `eth3` HW-1 с интерфейсом `eth3` HW-2.

5 На обоих HW:

5.1 Последовательно включите подчиненные физические интерфейсы:

```
hostname# inet ifconfig eth2 up
hostname# inet ifconfig eth3 up
```

5.2 Включите агрегированный интерфейс:

```
hostname# inet ifconfig bond0 up
```

Теперь трафик между координаторами будет идти по агрегированному каналу, состоящему из двух подчиненных физических каналов. В случае отказа одного из подчиненных каналов оставшийся продолжает работу. При возобновлении работы подчиненного канала работа агрегированного канала продолжится в режиме `balance-rr`.

Настройка получения параметров от DHCP-сервера

При использовании режима автоматического получения параметров от DHCP-сервера сетевой интерфейс получает:

- собственный IP-адрес и маску подсети;
- IP-адреса DNS-серверов;
- IP-адреса NTP-серверов;
- маршруты.



Примечание. Отправка указанных параметров должна быть настроена на DHCP-сервере.

Вы можете отключить получение параметров от DHCP-сервера:

- Чтобы отключить получение IP-адресов DNS-серверов:

```
hostname# inet ifconfig <имя интерфейса> dhcp dns off
```

- Чтобы отключить получение IP-адресов NTP-серверов:

```
hostname# inet ifconfig <имя интерфейса> dhcp ntp off
```

- Чтобы отключить получение маршрутов:

```
hostname# inet ifconfig <имя интерфейса> dhcp route off
```

Чтобы задать специфичное значение метрики для маршрутов, полученных от DHCP-сервера:

```
hostname# inet ifconfig <имя интерфейса> dhcp route-metric <1-255>
```

Подробнее см. в документе «Справочник команд и конфигурационных файлов».

Вы можете задать ряд параметров, которые будут присваиваться маршрутам DHCP-сервера (см. [Настройка параметров маршрутов от DHCP/PPP-протокола](#)). Чтобы просмотреть параметры, которые заданы для режима DHCP на всех сетевых интерфейсах, см. [Просмотр настроек DHCP в режиме клиента](#).

3

Настройка VPN

Общие принципы настройки VPN	57
Настройка параметров сетевого подключения координатора	58
Настройка виртуальных IP-адресов	67
Настройка туннелирования на сетевом уровне	73
Настройка IP-адресов доступа к узлу и их приоритета	82
Настройка TCP-туннеля	84
Сценарии использования VPN	88
Настройка туннелирования на канальном уровне по технологии L2OverIP	92

Общие принципы настройки VPN

Настройка VPN выполняется в конфигурационном файле `iplir.conf`. Большинство параметров в этом файле задается автоматически при установке справочников и ключей на узле — параметрам присваиваются значения, заданные администратором при настройке сети ViPNet и отдельных сетевых узлов в ViPNet ЦУС или Prime.

С помощью параметров файла `iplir.conf` вы можете выполнить следующие виды настроек VPN:

- [Настройка подключения к внешней сети.](#)
- [Настройка видимости узлов.](#)
- [Настройка параметров виртуальных IP-адресов.](#)
- [Настройка туннелирования на сетевом уровне.](#)

Настройка параметров сетевого подключения координатора

Для подключения к ViPNet Coordinator HW других узлов, находящихся во внешней сети или стоящих за другими межсетевыми экранами, настройте режим подключения [координатора](#) к внешней сети.

Настройка режимов выполняется двумя способами:

- С помощью ViPNet ЦУС или Prime — рекомендуемый способ.
Режим устанавливается на ViPNet Coordinator HW при установке дистрибутива ключей или при удаленном обновлении справочников и ключей.
- Путем редактирования файла конфигурации `iplir.conf` на ViPNet Coordinator HW.

Режимы сетевого подключения координатора



Внимание! Не рекомендуем использовать режимы подключения «Без использования межсетевого экрана» и «Координатор», так как они являются устаревшими и будут удалены в следующих версиях продукта. При выборе этих режимов возможна потеря связи с узлами.

Используйте:

- режим «Со статической трансляцией адресов» вместо «Без межсетевого экрана»;
- режим «С динамической трансляцией адресов» вместо «Координатор».

Режим [«Со статической трансляцией адресов»](#) выбирается в случае:

- если для выхода во внешнюю сеть установлен межсетевой экран или иное устройство NAT, на котором можно настроить статические правила трансляции адресов, обеспечивающие взаимодействие с определенным внутренним адресом сети по протоколу UDP с заданным портом;
- если в локальной сети имеется несколько координаторов, связанных между собой как по внутренней, так и по внешней сети, и к ним необходим доступ из внешней сети со стороны удаленных узлов, работающих в режиме динамической трансляции адресов;
- если ViPNet Coordinator HW имеет непосредственное подключение к внешней сети, то есть может быть доступен напрямую со стороны других сетевых узлов (например, имеет публичный адрес).

Режим [«С динамической трансляцией адресов»](#) выбирается в случае:

- если для выхода во внешнюю сеть уже используется другой координатор, выполняющий функции межсетевого экрана. В этом случае установленный координатор служит межсетевым экраном для ViPNet Coordinator HW;
- если для выхода во внешнюю сеть установлен межсетевой экран или иное устройство NAT, на котором затруднительно настроить статические правила трансляции адресов. Данный режим наиболее универсален, и ViPNet Coordinator HW в этом режиме будет работоспособен и при других способах подключения к сети.



Примечание. Если узлы находятся в одной локальной сети и могут обмениваться широковещательными пакетами, то независимо от выбранного типа межсетевого экрана взаимодействие между ними всегда происходит напрямую по IP-адресу узла.

Настройка режима «Без использования межсетевого экрана»



Внимание! Режим «Без использования межсетевого экрана» является устаревшим и будет удален в следующих версиях продукта. При выборе этого режима возможна потеря связи с узлами.

Вместо него рекомендуем использовать режим «Со статической трансляцией адресов».

Подключение без использования межсетевого экрана следует настраивать на координаторе, если ни один из его сетевых интерфейсов не находится за устройством NAT, то есть когда координатор доступен из маршрутизируемой сети. Если координатор должен быть доступен для других узлов, находящихся во внешних сетях, то один из его интерфейсов должен иметь публичный IP-адрес.

Для настройки работы ViPNet Coordinator HW без использования межсетевого экрана:

- 1 Завершите работу управляющей службы:

```
hostname# iplir stop
```

- 2 Откройте конфигурационный файл `iplir.conf` для редактирования:

```
hostname# iplir config
```

- 3 В собственной секции `[id]` (первая секция) установите параметр `usefirewall` в значение `off`.
- 4 В секции `[dynamic]` установите параметр `dynamic_proxy` в значение `off`.
- 5 Для всех используемых сетевых интерфейсов в секциях `[adapter]` установите параметр `type` в значение `internal`.
- 6 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).

При появлении сообщения `Change ProxyId to SelfId for own 'Coordinator' station because 'usefirewall' is off` укажите идентификатор координатора в параметре `proxid` (файл `iplir.conf` > секция `[id]`, подробнее см. «Справочник команд и конфигурационных файлов»).

7 Запустите управляющую службу:

```
hostname# iplir start
```

Настройка режима «Координатор»



Внимание! Режим «Координатор» является устаревшим и будет удален в следующих версиях продукта. При выборе этого режима возможна потеря связи с узлами. Вместо него рекомендуем использовать режим «С динамической трансляцией адресов».

Если необходимо защитить трафик отдельного сегмента внутри локальной сети, на границе которой уже установлен координатор, выполняющий функции межсетевого экрана для клиентов этой локальной сети, то на границе такого сегмента может быть установлен второй координатор.

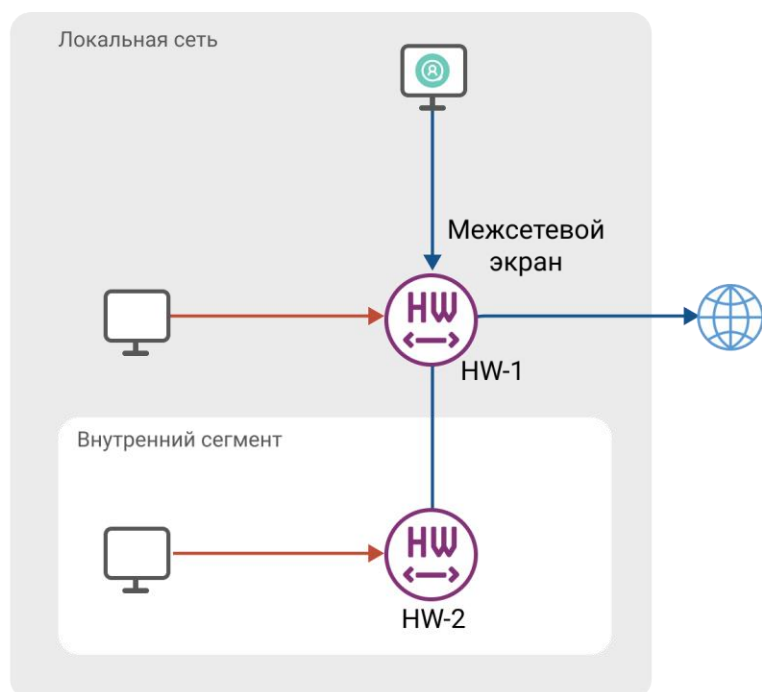


Рисунок 4. Подключение координатора через другой координатор

При этом координатор 1 (см. рисунок выше) должен быть выбран в качестве межсетевого экрана для координатора 2. Между двумя координаторами не должно быть никаких устройств, осуществляющих трансляцию адресов (NAT).

Такое включение координаторов называется каскадным включением. В результате для координаторов будет реализована автоматическая маршрутизация зашифрованного трафика из внутреннего сегмента сети как в локальную, так и в глобальную сеть.

Для настройки подключения к сети ViPNet Coordinator HW через координатор:

1 Завершите работу управляющей службы:

```
hostname# iplir stop
```


2 Откройте конфигурационный файл `iplir.conf` для редактирования:

```
hostname# iplir config
```

3 В секции `[id]` координатора укажите:

- o `ip` — реальный IP-адрес координатора, доступный ViPNet Coordinator HW (если он не указан);
- o `port` — номер порта назначения, на который будут посылаться пакеты для координатора.

4 В собственной секции `[id]`:

- o установите параметр `usefirewall` в значение `on`;
- o для параметра `proxyid` укажите идентификатор выбранного координатора.

5 В секции `[dynamic]` установите параметр `dynamic_proxy` в значение `off`.

6 В секции `[adapter]` сетевого интерфейса, со стороны которого находится выбранный координатор, установите параметр `type` в значение `external`.

7 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).

8 Запустите управляющую службу:

```
hostname# iplir start
```

После соединения с координатором и при его правильной настройке в секции `[id]` этого координатора будут установлены значения параметров `firewallip`, `port` и `proxyid`. Кроме того, в соответствии со значениями этих параметров координатора могут измениться значения параметров `firewallip` и `port` собственной секции `[id]`.

Настройка режима «Со статической трансляцией адресов»

Если ViPNet Coordinator HW установлен за устройством со статической трансляцией сетевых адресов (NAT), настройте на координаторе режим «Со статической трансляции адресов».



Рисунок 5. Подключение координатора через устройство со статической трансляцией адресов

Перед включением режима «Со статической трансляцией адресов» настройте маршрутизацию на устройство со статическим NAT (шлюз по умолчанию или маршруты для удаленных подсетей). Также рекомендуем использовать ViPNet Coordinator HW в качестве сервера соединений для клиентов локальной сети.

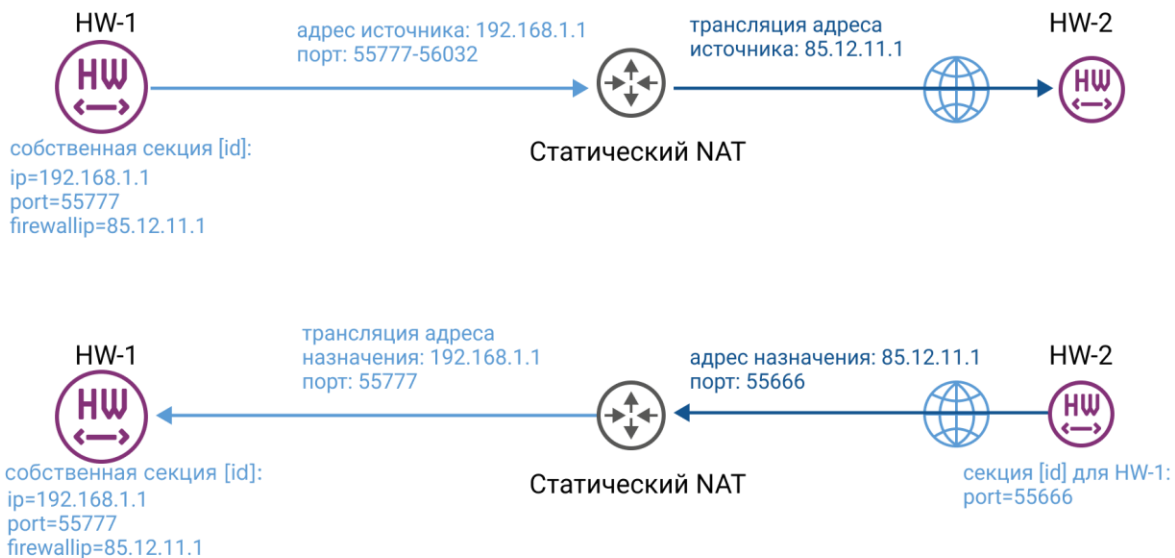


Рисунок 6. Пример работы координатора в режиме «Со статической трансляцией адресов»

На устройстве со статическим NAT настройте следующие фильтры и правила трансляции:

- пропускать UDP-пакеты с трансляцией адреса источника на внешний IP-адрес устройства NAT:
 - адрес источника: IP-адрес доступа HW-1;
 - порт источника: от port до port+255 (port — параметр в собственной секции [id] HW-1);
 - адрес назначения: любой;
 - порт назначения: любой;
- пропускать UDP-пакеты с трансляцией адреса назначения на IP-адрес и порт HW-1:
 - адрес источника: любой;
 - порт источника: любой;
 - адрес назначения: внешний IP-адрес устройства NAT;
 - порт назначения (параметр port в iplir.conf HW-2 в секции [id] для HW-1).



Примечание. Внешний IP-адрес устройства NAT отображается в параметре firewallip собственной секции [id] HW-1.

На HW-1 настройте режим «Со статической трансляцией адресов»:

- 1 Завершите работу управляющей службы:


```
hostname# iplir stop
```
- 2 Откройте конфигурационный файл iplir.conf для редактирования:


```
hostname# iplir config
```
- 3 В собственной секции [id] установите следующие параметры:
 - usefirewall в значение on;

- o `proxyid` в значение `0x00000000`;
- o `fixfirewall` в значение:
 - `on` — если на устройстве с NAT есть возможность настроить статические правила для входящих IP-пакетов, т.е. перенаправлять пакеты, у которых задан IP-адрес ViPNet Coordinator HW и порт назначения. В этом случае внешний IP-адрес межсетевого экрана с NAT и порт доступа к нему могут быть жестко заданы на межсетевом экране, и их необходимо указать в параметрах `firewallip` и `port` собственной секции `[id]`.



Внимание! При задании фиксированного адреса возможна потеря связи с узлами, находящимися за межсетевым экраном с динамической трансляцией адресов. Режим со статической трансляцией и фиксированным IP-адресом внешнего межсетевого экрана является устаревшим и будет удален в следующих версиях продукта.

- `off` — если внешний IP-адрес устройства с NAT в процессе работы может измениться, то на других сетевых узлах IP-адрес доступа к ViPNet Coordinator HW будет регистрироваться по внешним параметрам пакета. Параметр `firewallip` в данном режиме определяется автоматически по информации, полученной от узлов.
- 4 Если вы установили параметр `fixfirewall` в значение `on`, то в секции `[adapter]` сетевого интерфейса, со стороны которого находится внешний межсетевой экран, установите параметр `type` в значение `external`.
 - 5 В секции `[dynamic]` установите параметр `dynamic_proxy` в значение `off`.
 - 6 Если в локальной сети через один межсетевой экран (или устройство NAT) работает несколько узлов с программным обеспечением ViPNet, задайте для таких узлов разные номера портов. Для этого в собственной секции `[id]` узла измените значение параметра `port`, определяющего номер порта, через который преобразованные в UDP-формат пакеты уходят с узла ViPNet Coordinator HW (порт источника) и приходят на данный узел (порт назначения).
 - 7 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).
 - 8 Запустите управляющую службу:


```
hostname# iplir start
```

Информация об IP-адресе межсетевого экрана и порте доступа автоматически передается всем остальным узлам, с которыми связан ViPNet Coordinator HW.

Настройка режима «С динамической трансляцией адресов»

Если ViPNet Coordinator HW подключен к внешней сети через устройство NAT, на котором затруднительно настроить статические правила трансляции, и есть необходимость во взаимодействии узлов локальной сети с другими узлами, находящимися во внешней, относительно

устройства NAT, сети, то на узлах локальной сети нужно настроить подключение к сети через межсетевой экран с динамической трансляцией адресов.

Режим подключения с использованием такого типа межсетевого экрана наиболее универсален и может использоваться практически во всех случаях. Однако основное его назначение — обеспечить надежное двустороннее соединение с узлами, подключенными к сети через устройства NAT, на которых настройка статических правил трансляции сетевых адресов затруднена или невозможна. Такая ситуация типична, например, для простых устройств NAT с минимумом настроек: DSL-модемов, беспроводных устройств и других. Затруднительно также произвести настройки на устройствах NAT, установленных у провайдера (в домашних сетях, мобильных и других сетях, где провайдер предоставляет частный IP-адрес).

Для обеспечения возможности двусторонней работы координатора, подключенного к внешней сети через устройство NAT, и всех его клиентов на координаторе устанавливается режим работы через межсетевой экран с динамической трансляцией адресов. Одновременно во внешней сети должен присутствовать постоянно доступный координатор, который будет являться [сервером соединений](#).

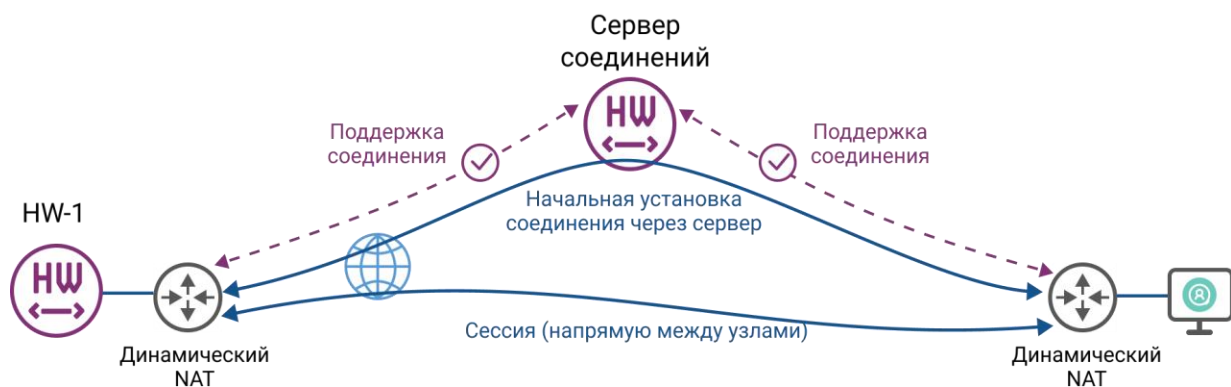


Рисунок 7. Подключение координатора через межсетевой экран с динамической трансляцией адресов

Организация соединений между узлами происходит в этом случае следующим образом:

- Узел в режиме работы через межсетевой экран с динамическим NAT после подключения к сети периодически отправляет IP-пакеты на свой сервер соединений. Период отправки таких IP-пакетов по умолчанию составляет 25 секунд. Этого, как правило, достаточно для работы через большинство устройств NAT. При необходимости вы можете изменить этот период (тайм-аут).
- После того как связь между узлом и его сервером соединений будет установлена, узел начинает устанавливать соединение с другим узлом. Он начинает передавать тестовые IP-пакеты удаленному узлу через свой сервер соединений. Одновременно с этим узел передает тестовые IP-пакеты на сервер соединений удаленного узла и напрямую на удаленный узел.
- Если тестовые IP-пакеты дошли до удаленного узла, то удаленный узел регистрирует соединение и начинает передавать ответный IP-трафик напрямую. Узел при получении

ответного IP-трафика от удаленного узла свой последующий IP-трафик ему также начинает передавать напрямую.

Если тестовые IP-пакеты дошли только до сервера соединений удаленного узла, то сервер соединений регистрирует это соединение и отправляет напрямую узлу ответные IP-пакеты удаленного узла.

То есть с удаленным узлом устанавливается прямое соединение или соединение через его сервер соединений. Если ответный IP-трафик так и не поступил от удаленного узла или его сервера соединений, то узел по-прежнему устанавливает соединение с удаленным узлом через свой сервер соединений.

- Возможность прямого соединения с удаленным узлом, находящимся за устройством с динамической трансляцией сетевых адресов, сохраняется по умолчанию в течение 75 секунд (трех интервалов отправки IP-пакетов) с момента окончания предыдущего соединения.

Для настройки работы ViPNet Coordinator HW в режиме «С динамической трансляцией адресов»:

1 Завершите работу управляющей службы:

```
hostname# iplir stop
```

2 Откройте конфигурационный файл `iplir.conf` для редактирования:

```
hostname# iplir config
```

3 В собственной секции `[id]` (первая секция) параметр `usefirewall` установите в значение `on`.

4 В секции `[dynamic]`:

- параметр `dynamic_proxy` установите в значение `on`;
- в параметре `forward_id` укажите идентификатор внешнего координатора (сервера соединений), с помощью которого ViPNet Coordinator HW будет устанавливать соединение с другими узлами.



Внимание! Выбранный сервер соединений должен быть доступен напрямую или через устройство со статической трансляцией адресов.

При подключении через устройство с динамическим NAT параметры `firewallip` и `port` секции `[id]` определяются автоматически по информации, полученной от внешних узлов, редактировать их вручную не следует.

5 При необходимости в секции `[dynamic]`:

- измените значение параметра `timeout` (период отправки IP-пакетов серверу соединений);
- параметр `always_use_server` установите в значение `on`, если требуется направлять весь IP-трафик через сервер соединений. При этом учтите, что передача всего IP-трафика через сервер соединений может привести к снижению скорости обмена данными между узлами.

6 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).

7 Запустите управляющую службу:

```
hostname# iplir start
```



Примечание. На ViPNet Coordinator HW, который является сервером соединений для узла, работающего в режиме «С динамической трансляцией адресов», в секции [id] этого узла параметру `proxyid` будет присвоено значение `0xFFFFFFFF`.

На всех остальных координаторах в секции [id] этого узла в параметре `proxyid` будет указан идентификатор узла, выполняющего функции его сервера соединений.

Настройка виртуальных IP-адресов

Общие сведения о виртуальных IP-адресах

При подключении удаленных пользователей к ресурсам локальной сети, а также при организации взаимодействия локальных сетей с использованием технологии VPN может возникать проблема пересечения IP-адресов. Например, два сетевых узла, находящиеся в разных сетях, которые взаимодействуют с помощью VPN, могут иметь одинаковые частные IP-адреса, и в случае обращения по такому адресу возникнет неоднозначность. Технология [виртуальных IP-адресов](#) позволяет решить эту проблему без изменения текущей адресной структуры сетей.

Каждый узел сети ViPNet при получении справочников автоматически формирует виртуальные IP-адреса для доступа ко всем [защищенным узлам](#), с которыми у него имеется связь. Если среди них есть координаторы, виртуальные IP-адреса формируются также для всех узлов, которые эти координаторы [туннелируют](#). Каждому реальному IP-адресу узла ставится в соответствие виртуальный IP-адрес. В результате каждый защищенный узел имеет собственный список виртуальных IP-адресов для доступа к узлам сети ViPNet.

Приложения на защищенном узле могут использовать виртуальные IP-адреса для соединения с соответствующими узлами. При отправке IP-пакета VPN-драйвер защищенного узла подменяет виртуальный IP-адрес получателя на реальный и отправляет адресату. В момент приема ответного IP-пакета из сети ViPNet-драйвер передает его приложению после подмены реального IP-адреса отправителя на виртуальный. Это происходит только в случае успешной расшифровки IP-пакета с помощью ключей отправителя, что обеспечивает защиту от подмены адреса отправителя.

Виртуальные IP-адреса для защищенных узлов закрепляются за уникальными идентификаторами узлов ViPNet. Виртуальные IP-адреса для [туннелируемых узлов](#) закрепляются за реальными IP-адресами.

При добавлении новых связей с защищенными узлами, а также новых реальных IP-адресов для узлов, с которыми уже существует связь, защищенный узел присваивает им свободные виртуальные IP-адреса из используемого диапазона виртуальных адресов. При удалении связей между узлами или удалении реальных IP-адресов, виртуальные IP-адреса освобождаются и могут использоваться повторно. Виртуальные IP-адреса, выделенные для туннелируемых узлов, могут измениться при добавлении новых IP-адресов туннелируемых узлов.

Если защищенный узел имеет несколько сетевых интерфейсов или несколько IP-адресов на интерфейсе, то виртуальные адреса для них формируются по правилу: для каждого следующего IP-адреса второй октет виртуального адреса увеличивается на единицу, начиная с нуля, например 11.0.0.1, 11.1.0.1, 11.2.0.1 и далее.

Настройка параметров виртуальных IP-адресов

ViPNet Coordinator HW формирует виртуальные IP-адреса для защищенных узлов и узлов, туннелируемых другими координаторами, с которыми у ViPNet Coordinator HW имеются связи.

По умолчанию начальный IP-адрес для генератора виртуальных IP-адресов — 11.0.0.1 (маска подсети: 255.0.0.0). Автоматическое формирование виртуальных IP-адресов для защищенных узлов и одиночных туннелируемых узлов начинается с этого IP-адреса.



Примечание. Одиночный туннелируемый узел — это узел, IP-адрес которого явно (а не в составе диапазона IP-адресов) указан в настройках туннелирования координатора.

Для диапазонов IP-адресов туннелируемых узлов начальным виртуальным IP-адресом по умолчанию является 12.0.0.1 (маска подсети 255.0.0.0).

Вы можете изменить используемые диапазоны IP-адресов для формирования виртуальных IP-адресов:



Внимание! При изменении начального IP-адреса для генератора виртуальных IP-адресов все виртуальные IP-адреса на ViPNet Coordinator HW будут сформированы заново.

1 Завершите работу управляющей службы:

```
hostname# iplir stop
```

2 Откройте файл `iplir.conf` для редактирования:

```
hostname# iplir config
```

3 В секции `[virtualip]` задайте диапазон IP-адресов для формирования виртуальных IP-адресов защищенных узлов и одиночных туннелируемых узлов одним из следующих способов:

- Задайте начальный и конечный IP-адреса диапазона виртуальных IP-адресов в параметрах `startvirtualip` и `maxvirtualip`. Последний октет IP-адресов не должен быть равен 0. При этом `maxvirtualip` должен быть больше `startvirtualip`.
- Задайте начальный IP-адрес диапазона виртуальных IP-адресов в параметре `startvirtualip`. Значение должно быть меньше 11.0.255.254 (так как начальный IP-адрес 12.0.0.1 задан по умолчанию для формирования виртуальных IP-адресов туннелей). При этом значение параметра `maxvirtualip` останется равным 11.0.255.254. Если значение `startvirtualip` задать в виде `x.0.0.1`, то параметру `maxvirtualip` будет автоматически присвоено значение `x.255.255.254`.
- Задайте начальный IP-адрес диапазона виртуальных IP-адресов с маской подсети в параметре `startvirtualip` (допустимые маски подсети — /8 или /16, например: `startvirtualip = 11.0.0.1/8`). При этом не присваивайте значение параметру `maxvirtualip`, и сам параметр не будет использоваться в конфигурационном файле, а диапазон виртуальных IP-адресов будет ограничен IP-адресом, указанным в параметре `startvirtualip` с маской подсети /16 (пример: `11.0.0.1/16`).

Внимание! Чтобы избежать ошибок при назначении начальных IP-адресов для генератора виртуальных IP-адресов, следует учитывать:



- Значение первого октета должно быть в диапазоне 1–239.
 - Значение второго и третьего октетов должно быть в диапазоне 0–255.
 - Значение четвертого октета должно быть в диапазоне 1–254.
-

4 Начальный IP-адрес для формирования диапазонов виртуальных IP-адресов туннелируемых узлов задается автоматически: `starttunnelvirtualip = <x>.0.0.1`, где `x` больше первого октета `startvirtualip` на 1. Если адресное пространство `<x>` исчерпано, то начинается назначение из следующего (`<x+1>`) и так далее. Вы можете задать его вручную. Для этого введите значение в параметр `starttunnelvirtualip`. При этом первый октет должен быть больше первого октета `startvirtualip` и не должен быть равен 0, 127, 224–239, 255.



Примечание. Если диапазон виртуальных IP-адресов пересекаются с реальными адресами туннелируемых узлов измените диапазон `starttunnelvirtualip`.

5 Сохраните изменения (**Ctrl+O**) и закройте редактор (**Ctrl+X**).

6 Запустите управляющую службу:

```
hostname# iplir start
```

Следует учитывать, что любой диапазон виртуальных IP-адресов (как диапазон по умолчанию, так и заданный вручную) будет пересекаться с одним из диапазонов реальных IP-адресов интернета. Поэтому при взаимодействии узла с открытыми ресурсами интернета может возникнуть конфликт, если у открытого ресурса IP-адрес совпадет с виртуальным IP-адресом одного из узлов сети ViPNet. Соединение с таким открытым ресурсом будет невозможно. В этом случае доступ к этому ресурсу можно настроить одним из способов:

- сменить диапазон виртуальных IP-адресов;
- работать с ресурсом через прокси-сервер.

При этом работать с защищенными узлами через прокси-сервер нельзя. Поэтому при использовании прокси-сервера требуется указать IP-адреса защищенных узлов в качестве исключений.

Настройка видимости узлов

В настройках ViPNet Coordinator HW вы можете задать видимость защищенных узлов и узлов, туннелируемых другим координатором, по реальным или виртуальным IP-адресам. При этом узлы, туннелируемые ViPNet Coordinator HW, будут подключаться к защищенным и туннелируемым узлам других координаторов по тем же IP-адресам, что и ViPNet Coordinator HW. Например, если в настройках ViPNet Coordinator HW задано использование виртуальных IP-адресов для

подключения к определенным защищенным узлам, то узлы, туннелируемые этим ViPNet Coordinator HW, также будут подключаться к ним по виртуальным IP-адресам.

По умолчанию на ViPNet Coordinator HW используются следующие настройки видимости:

- защищенные узлы, находящиеся в одном и том же широковещательном сегменте, доступны по реальным IP-адресам;
- туннелируемые узлы (узлы, туннелируемые другими координаторами) доступны по реальным IP-адресам;
- остальные узлы сети ViPNet доступны по виртуальным IP-адресам.

Видимость каждого защищенного узла сети ViPNet определяется с помощью параметров файла `iplir.conf` следующим образом:

- 1 Если в секции `[id]` узла присутствует параметр `visibility`, то видимость узла определяется значением этого параметра.
- 2 Если в секции `[id]` узла нет параметра `visibility`, но при этом сеть, которой принадлежит узел, указана в параметре `subnet_real` или `subnet_virtual` секции `[visibility]`, то видимость узла определяется значением этого параметра.
- 3 В иных случаях видимость узла определяется значением параметра `default` секции `[visibility]`.

То есть индивидуальные настройки видимости узлов приоритетнее настроек видимости сетей, а настройки видимости сетей приоритетнее настроек видимости по умолчанию.

Видимость туннелируемых узлов определяется с помощью параметров файла `iplir.conf` следующим образом:

- 1 Если в секции `[id]`, которая относится к туннелирующему координатору, присутствует параметр `tunnelvisibility`, то видимость туннелируемых узлов определяется значением этого параметра.
- 2 Если в секции `[id]`, которая относится к туннелирующему координатору, нет параметра `tunnelvisibility`, то видимость туннелируемых узлов определяется значением параметра `tunneldefault` секции `[visibility]`.

Настройка видимости защищенных узлов

- 1 Завершите работу управляющей службы:

```
hostname# iplir stop
```
- 2 Откройте файл `iplir.conf` для редактирования:

```
hostname# iplir config
```
- 3 Чтобы настроить параметры видимости защищенных узлов по умолчанию и параметры видимости сетей, в секции `[visibility]`:
 - установите параметр `default` в одно из значений:
 - `auto` (по умолчанию) — видимость узлов определяется автоматически;

- `real` — для доступа к узлам используются их реальные IP-адреса;
 - `virtual` — для доступа к узлам используются их виртуальные IP-адреса.
- при необходимости укажите идентификаторы вашей сети ViPNet и [доверенных сетей](#) в одном из параметров:
- `subnet_auto` — идентификаторы сетей ViPNet, для которых необходимо настроить автоматическую видимость узлов;
 - `subnet_real` — идентификаторы сетей ViPNet, для которых необходимо настроить видимость узлов по реальным IP-адресам;
 - `subnet_virtual` — идентификаторы сетей ViPNet, для которых необходимо настроить видимость узлов по виртуальным IP-адресам.

Вы можете указать один либо несколько идентификаторов через запятую.

4 Чтобы настроить параметры видимости отдельных защищенных узлов, в соответствующих секциях `[id]` параметр `visibility` установите в одно из значений:

- `auto` — определять видимость узла автоматически, в зависимости от его текущего адреса видимости.
- `real` — всегда обращаться к узлу по его реальному адресу.
- `virtual` — всегда обращаться к узлу по его виртуальному адресу.



Внимание! Настраивать параметры видимости отдельных узлов нужно с осторожностью. В случае если необходимо избежать конфликта реальных IP-адресов, установите параметр `visibility` в значение `virtual`.

5 Сохраните изменения (**Ctrl+O**) и закройте редактор (**Ctrl+X**).

6 Запустите управляющую службу:

```
hostname# iplir start
```

Настройка видимости туннелируемых узлов

1 Завершите работу управляющей службы:

```
hostname# iplir stop
```

2 Откройте файл `iplir.conf` для редактирования:

```
hostname# iplir config
```

3 Чтобы настроить параметры видимости туннелируемых узлов по умолчанию, в секции `[visibility]` установите параметр `tunneldefault` в одно из значений:

- `real` (по умолчанию) — для доступа к туннелируемым узлам используются их реальные IP-адреса.
- `virtual` — для доступа к туннелируемым узлам используются их виртуальные IP-адреса.

4 Чтобы настроить параметры видимости туннелируемых узлов, в секциях `[id]`, которые относятся к туннелирующим координаторам, в параметре `tunnelvisibility` установите одно из значений:

- o `real` — всегда обращаться к туннелируемым узлам по их реальным адресам.
- o `virtual` — всегда обращаться к туннелируемым узлам по их виртуальным адресам.

5 Сохраните изменения (**Ctrl+O**) и закройте редактор (**Ctrl+X**).

6 Запустите управляющую службу:

```
hostname# iplir start
```

Настройка туннелирования на сетевом уровне

Общие сведения о туннелировании

Технология туннелирования применяется:

- для защиты данных при их передаче на потенциально опасном участке сети;
- для организации доступа к открытому узлу, на котором установка ПО ViPNet невозможна. Как правило, это специализированные устройства (например, IP-АТС или аппаратные IP-телефоны) или серверы (SQL, 1С, DHCP), установка дополнительного ПО на которые нежелательна.

Технология туннелирования предполагает направление трафика узла не напрямую на другой узел, а через координатор, где трафик фильтруется и защищается криптографическими методами. Также настройки туннелирования открытых узлов, которые задает администратор сети ViPNet, хранятся на координаторах сети ViPNet, выполняющих туннелирование этих узлов.

Обычно параметры туннелирования рассылаются на узлы в составе справочников. Если туннелируемые IP-адреса координатора заданы в ViPNet ЦУС или Prime, то другие узлы получают информацию об этом автоматически.



Примечание. Если защищенный узел имеет связь с определенным координатором, то этот узел будет иметь доступ ко всем туннелируемым узлам этого координатора. Чтобы ограничить доступ к определенным туннелируемым узлам, настройте сетевые фильтры (см. [Настройка сетевых фильтров](#)).

Если туннелируемые IP-адреса заданы на координаторе вручную (см. [Ручная настройка туннелирования](#)), эти IP-адреса также необходимо указать вручную на каждом узле, который будет подключаться к туннелируемым узлам посредством сети ViPNet.

Если настройки туннелирования на узлах были заданы вручную, а затем поступили настройки из ViPNet ЦУС или Prime, то выполняется следующий алгоритм:

- если настройки туннелирования для ViPNet Coordinator HW не менялись в ViPNet ЦУС или Prime, то локально заданные настройки остаются без изменений;
- если с настройками туннелирования из ViPNet ЦУС или Prime получены новые IP-адреса туннелируемых узлов, то они добавляются в конфигурационный файл `iplir.conf`;
- если заданные локально IP-адреса туннелируемых узлов были удалены из настроек туннелирования, полученных из ViPNet ЦУС или Prime, то эти IP-адреса будут удалены.

После изменения настроек туннелирования может возникнуть конфликт между реальными IP-адресами узлов из широковещательного сегмента вашего ViPNet Coordinator HW и реальными IP-адресами узлов, туннелируемых удаленным координатором, или между реальными IP-адресами

туннелируемых узлов. Чтобы устранить возможные конфликты, на координаторе, туннелирующем узлы с конфликтующими IP-адресами, задайте видимость туннелируемых узлов по виртуальным IP-адресам (см. [Настройка видимости узлов](#)).

Ручная настройка туннелирования



Примечание. Настраивайте туннелирование на координаторе только когда централизованная настройка туннелирования в ViPNet ЦУС или Prime не используется, иначе настройки на координаторе могут быть заменены настройками из ViPNet ЦУС или Prime.

Чтобы настроить туннелирование для открытых узлов, подключенных к ViPNet Coordinator HW, или настроить доступ к туннелируемым узлам удаленного координатора:

- 1 Завершите работу управляющей службы:

```
hostname# iplir stop
```

- 2 Откройте конфигурационный файл `iplir.conf` для редактирования:

```
hostname# iplir config
```

- 3 Выберите режим задания виртуальных IP-адресов для туннелируемых узлов. Для этого в секции `[misc]` установите параметр `tunnel_virt_assignment` в одно из значений:

- o `auto` (по умолчанию) — при задании реальных IP-адресов туннелируемых узлов виртуальные IP-адреса для них назначаются автоматически.



Внимание! Если все возможные адреса из диапазонов виртуальных IP-адресов для туннелируемых узлов заняты, то следующим узлам в качестве виртуальных копируются реальные адреса. При этом режим назначения принудительно переключается в ручной (`manual`).

- o `manual` — виртуальные IP-адреса для туннелируемых узлов задаются вручную. При этом реальные IP-адреса туннелируемых узлов могут быть заданы как вручную, так и получены из ViPNet ЦУС или Prime.
- 4 Для туннелирования открытых узлов, подключенных к ViPNet Coordinator HW, в собственной секции `[id]` (первая секция) измените значение параметра `tunnel`:

- o при автоматическом режиме задания виртуальных IP-адресов (см. предыдущий шаг):

```
tunnel = <ip1>-<ip2>
```

где `<ip1>-<ip2>` — диапазон реальных IP-адресов туннелируемых узлов (начальный и конечный IP-адреса диапазона).

Виртуальные IP-адреса для заданного диапазона будут сформированы автоматически.

Например, чтобы ViPNet Coordinator HW туннелировал IP-адреса с 192.168.201.5 по 192.168.201.10, а виртуальные IP-адреса для этого диапазона были назначены автоматически, задайте:

```
tunnel= 192.168.201.5-192.168.201.10
```

- о при ручном режиме задания виртуальных IP-адресов (см. предыдущий шаг):

```
tunnel = <ip1>-<ip2> to <ip3>-<ip4>
```

где <ip1>-<ip2> — диапазон реальных IP-адресов туннелируемых узлов (начальный и конечный IP-адреса диапазона),

<ip3>-<ip4> — диапазон виртуальных IP-адресов (начальный и конечный IP-адреса диапазона), которые будут использоваться вместо реальных IP-адресов туннелируемых узлов, если на узле, который к ним обращается, настроена видимость по виртуальным IP-адресам.

Параметр <ip4> можно не указывать, он формируется путем прибавления к <ip3> разницы между <ip2> и <ip1>.



Внимание! При задании виртуальных IP-адресов в ручном режиме автоматическая проверка повторного использования одних и тех же IP-адресов не производится. Поэтому убедитесь, что задаваемые диапазоны виртуальных IP-адресов не пересекаются.

Например, чтобы координатор туннелировал IP-адреса с 192.168.201.5 по 192.168.201.10, а виртуальными IP-адресами для этого диапазона были IP-адреса с 192.168.111.5 по 192.168.111.10, задайте:

```
tunnel = 192.168.201.5-192.168.201.10 to 192.168.111.5-192.168.111.10
```



Внимание! На всех туннелируемых узлах сети настройте статические маршруты для защищаемого трафика через ViPNet Coordinator HW или укажите ViPNet Coordinator HW в качестве шлюза по умолчанию.

- 5 Для настройки доступа к узлам, туннелируемым удаленным координатором, в секции [id] соответствующего координатора измените значение параметра tunnel:

```
tunnel = <ip1>-<ip2> to <ip3>-<ip4>
```

где <ip1>-<ip2> — диапазон реальных IP-адресов узлов, туннелируемых данным координатором (начальный и конечный IP-адреса диапазона),

<ip3>-<ip4> — диапазон виртуальных IP-адресов (начальный и конечный IP-адреса диапазона), которые используются вместо реальных IP-адресов туннелируемых узлов данного координатора, если на узле, который к ним обращается, настроена видимость по виртуальным IP-адресам.

- 6 Задайте использование IP-адресов <ip3>-<ip4> на каждом защищенном узле, с которого необходим доступ к туннелируемым узлам.
- 7 Сохраните изменения (Ctrl+O) и закройте редактор (Ctrl+X).
- 8 Запустите управляющую службу:

```
hostname# iplir start
```

Примеры настройки туннелирования

Организация защищенного взаимодействия открытых узлов (туннелирование)



Примечание. В примере ниже указаны реальные IP-адреса туннелируемых узлов и настроена видимость узлов по реальным адресам.

Рассмотрим пример:

- В одном из офисов находится сервер данных, к которому обращаются пользователи из другого офиса.
- IP-адрес сервера 192.168.1.2. IP-адреса пользователей 192.168.2.2-192.168.2.4.
- Установка ПО ViPNet на сервер данных и компьютеры пользователей невозможна.

Необходимо, чтобы весь обмен данными через интернет происходил с шифрованием трафика.

Чтобы решить эту задачу, в каждом из офисов устанавливается ViPNet Coordinator HW, который выступает в роли криптошлюза для компьютеров локальной сети:

- HW-1 имеет частный IP-адрес 192.168.1.1 и идентификатор 0x00010101 в сети ViPNet;
- HW-2 имеет частный IP-адрес 192.168.2.1 и идентификатор 0x00020201 в сети ViPNet.

В результате компьютеры 1, 2, 3 смогут обращаться к серверу по адресу 192.168.1.2. При этом на участке между HW 1 и 2 пакеты будут передаваться в зашифрованном виде.

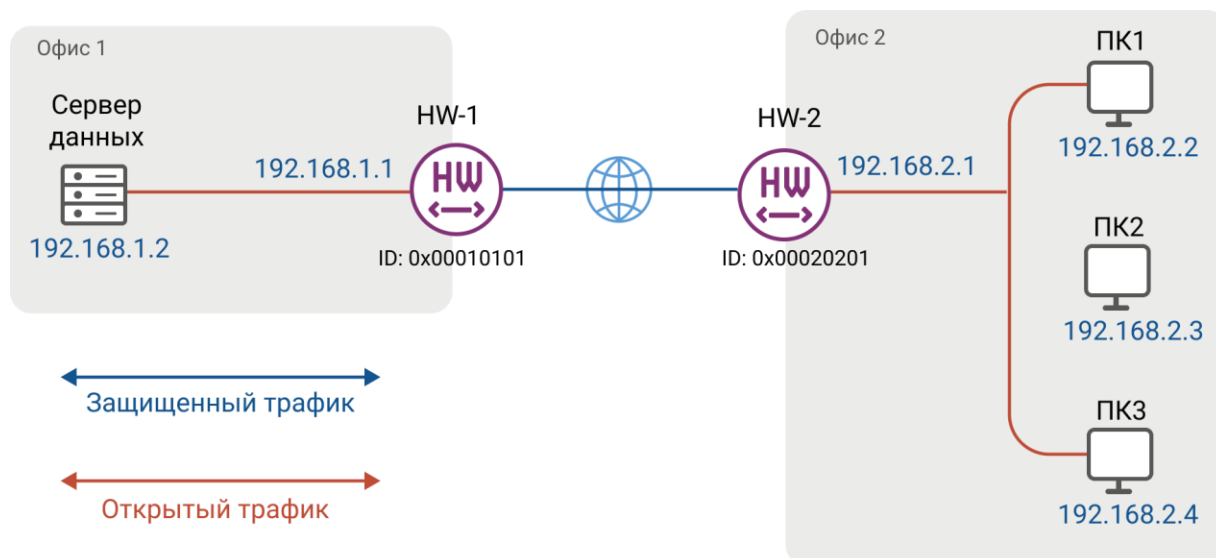


Рисунок 8. Организация туннелей с использованием ViPNet Coordinator HW

Чтобы настроить работу туннелей:

- 1 На компьютерах 1, 2, 3 в качестве шлюза по умолчанию задайте частный IP-адрес HW-2, на сервере — частный IP-адрес HW-1.
- 2 На каждом ViPNet Coordinator HW задайте IP-адреса компьютеров, которые туннелирует он сам и противоположный ViPNet Coordinator HW. Адреса туннелируемых узлов можно задать двумя способами:
 - Централизованно в ViPNet ЦУС или Prime. В этом случае для каждого ViPNet Coordinator HW задаются только адреса узлов, которые туннелирует он сам. После рассылки обновлений справочников и ключей заданные адреса появятся в настройках обоих ViPNet Coordinator HW. Этот способ является предпочтительным.
 - Локально на ViPNet Coordinator HW путем редактирования конфигурационного файла `iplir.conf` (подробнее см. в документе «Справочник команд и конфигурационных файлов», в разделе «Файл `iplir.conf`»).

Для локальной настройки туннелей:

2.1 На HW-1:

- Завершите работу управляющей службы:
`hostname# iplir stop`
- Откройте конфигурационный файл `iplir.conf` для редактирования:
`hostname# iplir config`
- В собственную секцию `[id]` (содержащую параметр `id= 0x00010101`) добавьте параметр:
`tunnel= 192.168.1.2-192.168.1.2 to 192.168.1.2-192.168.1.2`



Примечание. Если на вашем ViPNet Coordinator HW виртуальные адреса для туннелируемых узлов задаются автоматически (по умолчанию), то вместо этой строки вы можете записать только первую ее часть:

`tunnel= 192.168.1.2-192.168.1.2`

В этом случае строка будет автоматически дополнена дублирующимся IP-адресом после запуска управляющей службы. Это справедливо для всех строк, начинающихся с `tunnel=`.

- В секцию `[id]`, описывающую HW-2 (содержащую параметр `id= 0x00020201`), добавьте параметр:
`tunnel= 192.168.2.2-192.168.2.4 to 192.168.2.2-192.168.2.4`
- Сохраните изменения (**Ctrl+O**) и закройте редактор (**Ctrl+X**).
- Запустите управляющую службу:
`hostname# iplir start`

2.2 На HW-2:

- Завершите работу управляющей службы:
`hostname# iplir stop`
- Откройте конфигурационный файл `iplir.conf` для редактирования:

```
hostname# iplir config
```

- В собственную секцию [id] (содержащую параметр id= 0x00020201) добавьте параметр:

```
tunnel= 192.168.2.2-192.168.2.4 to 192.168.2.2-192.168.2.4
```

- В секцию [id], описывающую HW-1 (содержащую параметр id= 0x00010101), добавьте параметр:

```
tunnel= 192.168.1.2-192.168.1.2 to 192.168.1.2-192.168.1.2
```

- Сохраните изменения (Ctrl+O) и закройте редактор (Ctrl+X).
- Запустите управляющую службу:

```
hostname# iplir start
```

- 3 Убедитесь, что в настройках каждого ViPNet Coordinator HW разрешен трафик для заданных туннелируемых узлов. По умолчанию на ViPNet Coordinator HW разрешен трафик для всех туннелируемых узлов, но эта настройка могла быть изменена. Подробнее см. в разделе [Ограничение доступа туннелируемых узлов](#).

Ограничение доступа туннелируемых узлов

В приведенном выше примере доступ к серверу разрешен всем открытым компьютерам, туннелируемым HW-2, так как по умолчанию на ViPNet Coordinator HW разрешен трафик для всех туннелируемых узлов.

При необходимости можно запретить доступ к серверу каким-либо компьютерам, не исключая их из списка туннелируемых узлов. Для этого на HW-2 добавьте сетевые фильтры туннелируемых узлов, запрещающие доступ для этих компьютеров:

Пусть требуется запретить доступ к серверу компьютеру 1 и разрешить компьютерам 2 и 3. Для этого на HW-2 добавьте фильтры:

```
hostname# firewall tunnel add 1 src 192.168.2.2 dst 192.168.1.2 drop
```

```
hostname# firewall tunnel add 2 src 192.168.1.2 dst 192.168.2.2 drop
```

Таким образом, будет разрешен трафик в обоих направлениях только между туннелируемыми компьютерами 2, 3 и HW-1, туннелирующим сервер. Для компьютера 1 защищенное взаимодействие с сервером будет недоступно.

Взаимодействие компьютеров с сервером разрешено по всем протоколам и портам. Чтобы ограничить это взаимодействие конкретными протоколами и портами, создайте соответствующие сетевые фильтры:

Пусть на сервере установлен веб-сервис, к которому разрешено обращаться только компьютерам 2 и 3. В этом случае на обоих ViPNet Coordinator HW необходимо создать фильтры туннелируемых узлов, ограничив их протоколом TCP и номером порта 80:

- На HW-1 удалите фильтры по умолчанию и создайте фильтры, разрешающие TCP-трафик в обоих направлениях между туннелируемым веб-сервером и HW-2:

```
hostname# firewall tunnel delete src @any dst @tunneledip pass
```

```
hostname# firewall tunnel delete src @tunneledip dst @any pass
hostname# firewall tunnel add src 192.168.1.2 dst 192.168.2.1 tcp sport 80 pass
hostname# firewall tunnel add src 192.168.2.1 dst 192.168.1.2 tcp dport 80 pass
```

- На HW-2 вместо приведенных выше фильтров туннелируемых узлов создайте следующие:

```
hostname# firewall tunnel add src 192.168.2.3-192.168.2.4 dst 192.168.1.1 tcp dport 80 pass
hostname# firewall tunnel add src 192.168.1.1 dst 192.168.2.3-192.168.2.4 tcp sport 80 pass
```

В результате компьютерам 2 и 3 будет доступно защищенное взаимодействие с веб-сервисом по протоколу TCP через порт 80.

Организация туннелей между защищенными и открытыми узлами



Примечание. Описанный в этом разделе сценарий вы можете реализовать только в ручном режиме назначения виртуальных адресов для туннелируемых узлов (параметру `tunnel_virt_assignment` присвоено значение `manual` в секции `[misc]` файла `iplir.conf`). Подробнее см. раздел [Ручная настройка туннелирования](#).

Рассмотрим модифицированную схему использования туннелей, когда на компьютеры 1, 2, 3 установлена программа ViPNet Client for Windows, а сервер остается незащищенным.

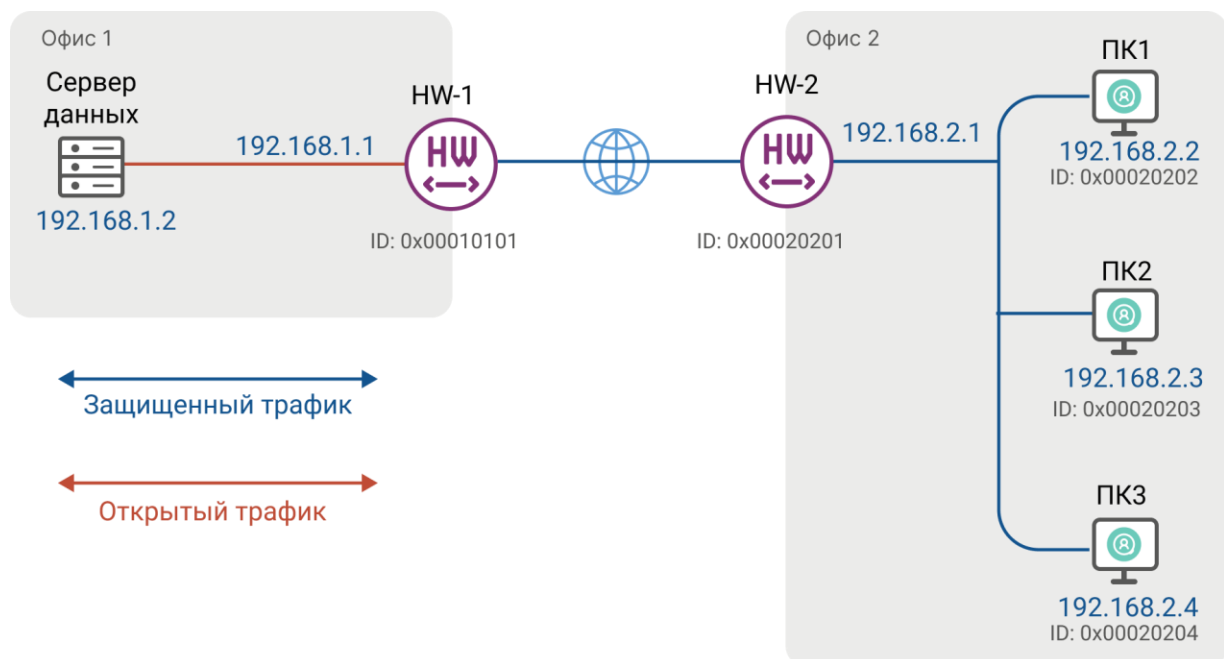


Рисунок 9. Схема организации туннелей между защищенными и открытыми узлами

Пусть компьютерам 1, 2, 3 присвоены в сети ViPNet идентификаторы 0x00020202, 0x00020203, 0x00020204 соответственно.

В этом случае, когда туннелируется только сервер, организовать туннель можно двумя способами:

- Централизованно с помощью ViPNet ЦУС или Prime. В этом случае задайте для HW-1 в качестве IP-адреса туннелируемого соединения IP-адрес сервера и разошлите справочники на HW-1 и все связанные с ним узлы вашей сети ViPNet.

Мы рекомендуем использовать этот способ.

- Локально редактированием конфигурационного файла `iplir.conf` HW-1 и настройки программы ViPNet Client for Windows на компьютерах 1, 2, 3.

Чтобы организовать туннель локально:

1 На HW-1 задайте адрес туннелируемого сервера:

1.1 Завершите работу управляющей службы:

```
hostname# iplir stop
```

1.2 Откройте конфигурационный файл `iplir.conf` для редактирования:

```
hostname# iplir config
```

1.3 В собственную секцию `[id]` добавьте параметр:

```
tunnel= 192.168.1.2-192.168.1.2 to 192.168.1.2-192.168.1.2
```

1.4 Сохраните изменения (**Ctrl+O**) и закройте редактор (**Ctrl+X**).

1.5 Запустите управляющую службу:

```
hostname# iplir start
```

2 Если в настройках HW-1 для туннелируемых узлов задан фильтр по умолчанию, то никакие дополнительные настройки не нужны. Иначе на HW-1 задайте фильтры туннелируемых узлов, разрешающие трафик между сервером и компьютерами 1, 2, 3:

2.1 Просмотрите фильтры туннелируемых узлов, заданных на HW-1:

```
hostname# firewall tunnel show
```

2.2 Удалите фильтры туннелируемых узлов:

```
hostname# firewall tunnel delete <номер фильтра>
```

2.3 Добавьте фильтры туннелируемых узлов:

```
hostname# firewall tunnel add src 192.168.1.2 dst 0x00020202,0x00020203,0x00020204 pass
```

```
hostname# firewall tunnel add src 0x00020202,0x00020203,0x00020204 dst 192.168.1.2 pass
```

Эти фильтры разрешают трафик в обоих направлениях между туннелируемым сервером и защищенными компьютерами 1, 2, 3.

3 На компьютерах 1, 2, 3 в программе ViPNet Client for Windows задайте для HW-1 в качестве IP-адреса для туннелирования IP-адрес 192.168.1.2 (подробнее см. документ «ViPNet Client for Windows. Руководство пользователя», раздел «Настройка доступа к туннелируемым узлам»).

После произведенных настроек компьютеры 1, 2, 3 смогут обращаться к серверу по адресу 192.168.1.2. При этом на участке между ViPNet Coordinator HW 1 и 2 пакеты будут передаваться в зашифрованном виде. Если требуется ограничить доступ к серверу со стороны некоторых узлов,

например запретить доступ к серверу компьютеру 1, то на HW-1 вместо приведенных выше фильтров туннелируемых узлов добавьте следующие:

```
hostname# firewall tunnel add src 192.168.1.2 dst 0x00020203,0x00020204 pass
```

```
hostname# firewall tunnel add src 0x00020203,0x00020204 dst 192.168.1.2 pass
```

В результате изменения настроек защищенное взаимодействие с сервером будет доступно только компьютерам 2 и 3.

Для случая, когда на сервере установлен веб-сервис, к которому разрешено обращаться компьютерам 2 и 3, но запрещено компьютеру 1, приведенные выше фильтры надо изменить следующим образом:

```
hostname# firewall tunnel add src 192.168.1.2 dst 0x00020203,0x00020204 tcp sport 80 pass
```

```
hostname# firewall tunnel add src 0x00020203,0x00020204 dst 192.168.1.2 tcp dport 80 pass
```

После изменения настроек компьютерам 2 и 3 будет доступно защищенное взаимодействие только с веб-сервисом.

Настройка IP-адресов доступа к узлу и их приоритета

Каждый узел ViPNet взаимодействует с другим узлом, с которым у него есть связь, по определенному адресу доступа. Однако нередко некоторые узлы подключены к нескольким независимым каналам связи и имеют несколько адресов доступа, каждый из которых может использоваться для связи с другими узлами.

Приоритеты каналов связи зависят от метрик, настроенных на сетевом узле, которому необходимо получить доступ к защищенному сегменту сети.

Исходя из значений метрик каналов, ViPNet Coordinator HW будет выбирать более приоритетный канал связи. Для этого при включении ViPNet Coordinator HW отправляет служебные сообщения на все известные адреса доступа узлов с определенной задержкой, равной метрике, заданной для этих адресов в миллисекундах. Выбор канала для связи производится с учетом следующего:

- Адрес с наименьшей метрикой считается наиболее приоритетным. Соединение с узлом устанавливается по адресу с наименьшей метрикой всегда, когда этот адрес доступен.
- Если адрес с наименьшей метрикой недоступен или если все метрики равны, то соединение с узлом устанавливается по тому адресу, доступность которого определяется быстрее в результате опроса.
- Если помимо адресов с заданными метриками, есть адреса, для которых метрика установлена в значение `auto` (определяется автоматически), значение `auto` всегда будет на 100 миллисекунд больше максимального значения заданной метрики. То есть канал с автоматически определяемой метрикой всегда будет иметь самый низкий приоритет по сравнению с каналами, для которых метрика задана вручную.
- После установления соединения с узлом определение доступности других его адресов выполняется только при разрыве текущего соединения. Задержка в выборе альтернативного канала связи может составлять от 7 до 70 секунд.

Адреса доступа для узлов, связанных с ViPNet Coordinator HW, настраиваются в соответствующих секциях `[id]` файла `iplir.conf`. Каждый адрес доступа узла указывается в параметре `accessiplist` этой секции, количество данных параметров в секции не ограничено.



Примечание. IP-адреса доступа узла могут быть настроены централизованно в ViPNet ЦУС или Prime. В этом случае они будут указаны в соответствующих секциях `[id]` файла `iplir.conf`. Для таких адресов вы также можете настроить приоритет, изменив соответствующие значения метрик (по умолчанию они определяются автоматически).

Чтобы указать IP-адреса доступа к какому-либо узлу, связанному с ViPNet Coordinator HW, и определить приоритеты этих адресов:

- 1 Завершите работу управляющей службы:

```
hostname# iplir stop
```

2 Откройте файл `iplir.conf` для редактирования:

```
hostname# iplir config
```

3 В параметрах `accessiplist` секции `[id]` соответствующего узла выполните одно из действий:

- Укажите IP-адреса доступа к узлу и задайте метрики маршрутов для каждого из этих адресов в следующем виде:

```
accessiplist= <IP-адрес узла>, <метрика>
```

При задании значения метрики необходимо учитывать следующее:

- рекомендуется указывать значения метрик больше максимального времени задержки при прохождении служебного сообщения и ответа на него по соответствующему каналу связи, но не превышающее 9999 миллисекунд;
- рекомендуемая минимальная разница между метриками — 100 миллисекунд.
- Если вы уже указали IP-адреса доступа ранее или они были получены из ViPNet ЦУС или Prime автоматически, в параметре `accessiplist` задайте метрики маршрутов для этих адресов в соответствии с рекомендациями, описанными выше.

4 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).

5 Запустите управляющую службу:

```
hostname# iplir start
```

В результате в параметры `accessiplist` файла `iplir.conf` автоматически будут добавлены реальный IP-адрес сетевого интерфейса узла, через который будут передаваться IP-пакеты для выбранного IP-адреса доступа, условный номер этого интерфейса и тип регистрации данного IP-адреса доступа узла в виде:

```
accessiplist= <IP-адрес узла>, <метрика>, <IP-адрес интерфейса>, <номер интерфейса>, <тип регистрации>
```

Например, запись об автоматическом добавлении IP-адреса 89.135.15.2 с метрикой 1 будет выглядеть следующим образом:

```
accessiplist= 89.135.15.2, 1, 0.0.0.0, 0, auto
```

Подробнее о параметре `accessiplist` см. в документе «Справочник команд и конфигурационных файлов», описание секции `[id]` файла `iplir.conf`.

Настройка TCP-туннеля

При удаленном подключении клиентов к сетям ViPNet может возникать проблема с передачей IP-пакетов по протоколу UDP из-за того, что данный протокол блокируется некоторыми интернет-провайдерами. Но в этих случаях обычно разрешен доступ по протоколу TCP через порты 80 (HTTP) и 443 (HTTPS).

Для решения проблемы доступа можно организовать взаимодействие клиентов с другими узлами сети ViPNet через **TCP-туннель**, настроенный на **сервере соединений** этих клиентов.

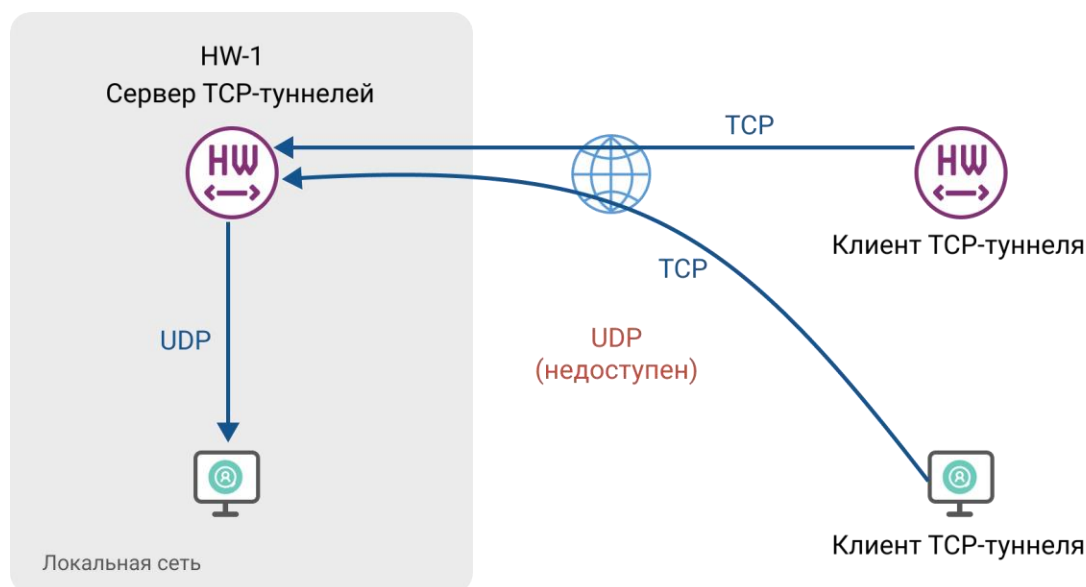


Рисунок 10. Настройка TCP-туннеля

В данном сценарии:

- Клиент TCP-туннеля — удаленный узел, подключенный через TCP-туннель. Клиентом может быть ViPNet-клиент или координатор.
- Сервер TCP-туннелей — координатор, выполняющий функцию сервера соединений, через который происходит подключение TCP-клиентов.

Если клиент TCP-туннеля не может связаться со своим сервером соединений (сервером TCP-туннелей) по протоколу UDP, он автоматически начинает устанавливать с ним соединение через TCP-туннель. Сервер соединений (сервер TCP-туннелей) передает полученные от клиента IP-пакеты дальше на узлы назначения по протоколу UDP.

ViPNet Coordinator HW может работать в режиме сервера TCP-туннелей или клиента TCP-туннеля, подключаясь к своему серверу соединений. Исполнения HW10 и HW50 не поддерживает режим сервера TCP-туннелей.

Настройка сервера TCP-туннелей



Примечание. ViPNet Coordinator HW может поддерживать не более 100 соединений через TCP-туннель.

При соединении через TCP-туннель клиентам не будут приходить уведомления в случае рассинхронизации времени.

Для работы координатора в роли сервера TCP-туннелей:

- На ViPNet Coordinator HW в файле `iplir.conf` в списке адресов доступа (параметр `accessiplist` в собственной секции `[id]`) задайте хотя бы один публичный адрес, поскольку для установления TCP-соединения клиент использует именно публичный IP-адрес доступа сервера соединений.
- На ViPNet Coordinator HW включите функцию сервера TCP-туннелей и настройте порт для входящих TCP-соединений (см. описание ниже).
- На клиентах TCP-туннеля проверьте заданный порт. Номер порта должен совпадать с тем, который используется на ViPNet Coordinator HW для TCP-соединений.

Номер порта должен поступать клиенту от сервера соединений в составе служебной информации. Если служебная информация не поступила, то порт на клиенте можно задать вручную.

Настроить сервер TCP-туннелей можно с помощью команд или путем редактирования конфигурационного файла `iplir.conf`.

Настройка с помощью команд



Внимание! Служба `iplircfg` должна быть запущена.

- 1 Назначьте ViPNet Coordinator HW роль Сервер TCP-туннелей:

```
hostname# iplir tcptunnel role server
```

- 2 Укажите порт для TCP-соединений:

```
hostname# iplir tcptunnel server port <номер порта>
```

После изменения номера порта на все сетевые узлы, для которых данный координатор является сервером соединений, будет отправлена служебная информация с новым номером порта для TCP-соединений.

- 3 Включите сервер TCP-туннелей:

```
hostname# iplir tcptunnel server on
```

Настройка с помощью редактирования конфигурационного файла

- 1 Завершите работу управляющей службы:

```
hostname# iplir stop
```

2 Откройте файл `iplir.conf` для редактирования:

```
hostname# iplir config
```

3 В секции `[misc]`:

- о параметру `tcptunnel_role` присвойте значение `server`.
- о параметру `tcptunnel_establish` присвойте значение `on`.

4 Если порт для TCP-соединений, заданный в настройках ViPNet Coordinator HW, используется какой-либо другой службой, во избежание конфликта в собственной секции `[id]` в параметре `tcptunnelport` укажите номер порта, на который будут приходить TCP-пакеты от защищенных узлов.

5 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).

6 Запустите управляющую службу:

```
hostname# iplir start
```

После изменения номера порта на все сетевые узлы, для которых данный координатор является сервером соединений, будет отправлена служебная информация с новым номером порта для TCP-соединений.

Примечание. В случае если протокол UDP перестал блокироваться, соединение по данному протоколу восстановится при следующих условиях:



- по истечении тайм-аута отправки служебных IP-пакетов с клиента на сервер соединений (по умолчанию тайм-аут равен 5 минутам);
 - после проверки соединения этого узла с другим;
 - после смены IP-адреса узла (например, при изменении точки доступа к интернету).
-

Настройка TCP-клиента

Чтобы ViPNet Coordinator HW мог работать как клиент TCP-туннеля, он должен быть подключен к сети в режиме «С динамической трансляцией адресов» (см. [Настройка параметров сетевого подключения координатора](#)).

Включить координатор в режим клиента TCP-туннеля можно с помощью команд или путем редактирования конфигурационного файла `iplir.conf`.

Настройка с помощью команд



Внимание! Служба `iplircfg` должна быть запущена.

1 Назначьте ViPNet Coordinator HW роль Клиент:

```
hostname# iplir tcptunnel role client
```

2 Выберите режим подключения:

```
hostname# iplir tcptunnel client mode {auto | forced}, где
```

- o auto — автоматическое включение TCP-туннеля при пропадании соединения по UDP и выключение при его появлении;
- o forced — TCP-туннель устанавливается всегда, когда возможно, даже при доступе к серверу соединения по UDP.

3 Укажите порт для TCP-соединений:

```
hostname# iplir tcptunnel client port <номер порта>
```

4 Включите режим клиента TCP-туннелей:

```
hostname# iplir tcptunnel client on
```

Настройка с помощью редактирования конфигурационного файла

1 Завершите работу управляющей службы:

```
hostname# iplir stop
```

2 Откройте файл `iplir.conf` для редактирования:

```
hostname# iplir config
```

3 В секции `[misc]`:

- o параметру `tcptunnel_role` присвойте значение `client`;
- o параметру `tcptunnel_client` присвойте значение `on`;
- o параметру `tcptunnel_client_mode` присвойте одно из значений:
 - auto — автоматическое включение TCP-туннеля при пропадании соединения по UDP и выключение при его появлении;
 - forced — TCP-туннель устанавливается всегда, когда возможно, даже при доступе к серверу соединения по UDP.
- o Если порт для TCP-соединений, заданный в настройках ViPNet Coordinator HW, используется какой-либо другой службой, во избежание конфликта в параметре `tcptunnel_client_port` укажите номер порта, на который будут приходить TCP-пакеты от защищенных узлов. TCP-порт должен быть таким же, как и на сервере TCP-туннелей.

4 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).

5 Запустите управляющую службу:

```
hostname# iplir start
```

Сценарии использования VPN

Организация защищенного взаимодействия между двумя удаленными офисами

ViPNet Coordinator HW позволяет организовать защиту трафика, передаваемого между удаленными локальными сетями. Такая задача обычно возникает при наличии у организации нескольких офисов, в каждом из которых развернута своя локальная сеть. С использованием программного обеспечения ViPNet можно объединить локальные сети офисов в виртуальную сеть с защитой трафика как внутри каждой локальной сети, так и при взаимодействии офисов через сеть общего пользования.

Рассмотрим пример взаимодействия двух офисов через интернет (в общем случае офисов может быть больше). В одном из офисов компьютеры объединены в локальную сеть с частными IP-адресами 192.168.1.0/24, в другом — в локальную сеть с частными IP-адресами 192.168.2.0/24. На границе каждой сети установлен шлюз с публичным IP-адресом.

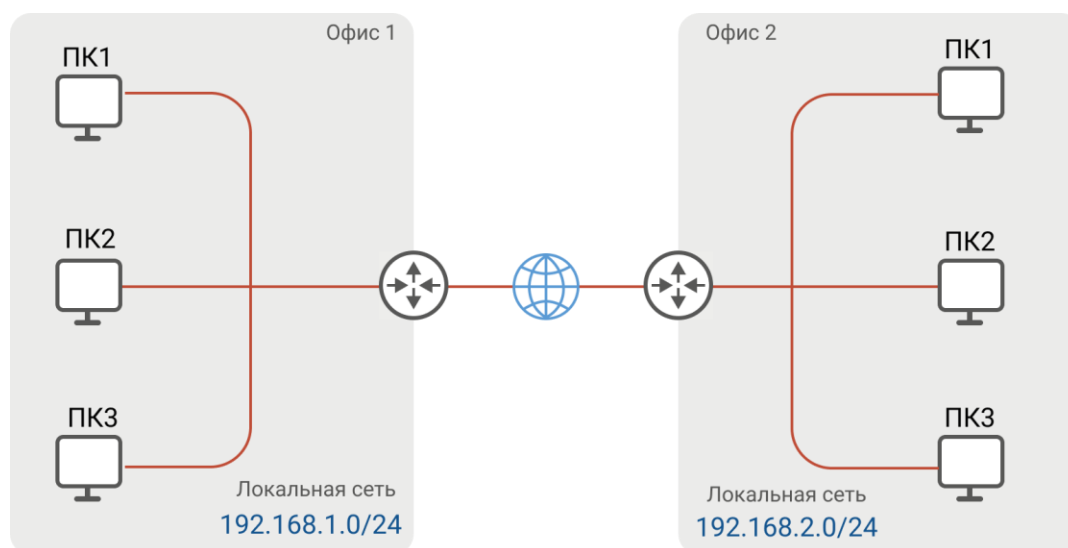


Рисунок 11. Схема сетей двух удаленных офисов

Требуется организовать защищенное взаимодействие пользователей в рамках своей локальной сети и с пользователями другой локальной сети. Предполагается, что в одном из офисов установлен сетевой ресурс, который останется открытым, но к которому должны иметь защищенный доступ пользователи из другого офиса.

Чтобы настроить защищенное взаимодействие между офисами:

- 1 В каждой локальной сети за шлюзом установите ViPNet Coordinator HW и создайте для него связи с компьютерами сети. Каждый ViPNet Coordinator HW будет выполнять в своей сети роль [координатора](#).

- 2 Для каждого ViPNet Coordinator HW задайте публичный IP-адрес. Это можно сделать централизованно в ViPNet ЦУС или Prime либо локально на ViPNet Coordinator HW:

```
hostname# inet ifconfig <интерфейс> address <IP-адрес> netmask <маска>
```

- 3 В каждой локальной сети установите программу ViPNet Client на компьютеры, которые должны быть защищены от несанкционированного доступа. Эти компьютеры будут выполнять в сети ViPNet роль **клиентов**.

При создании сетевых узлов ViPNet зарегистрируйте клиентов каждой сети за соответствующим координатором.

- 4 Для компьютера, который остается открытым (на него не устанавливается программа ViPNet Client), организуйте туннель. Для этого на каждом ViPNet Coordinator HW задайте адрес открытого компьютера в качестве туннелируемого адреса (см. [Организация туннелей между защищенными и открытыми узлами](#)).

В результате две локальные сети будут объединены в единую защищенную **сеть ViPNet**, состоящую из двух сегментов.

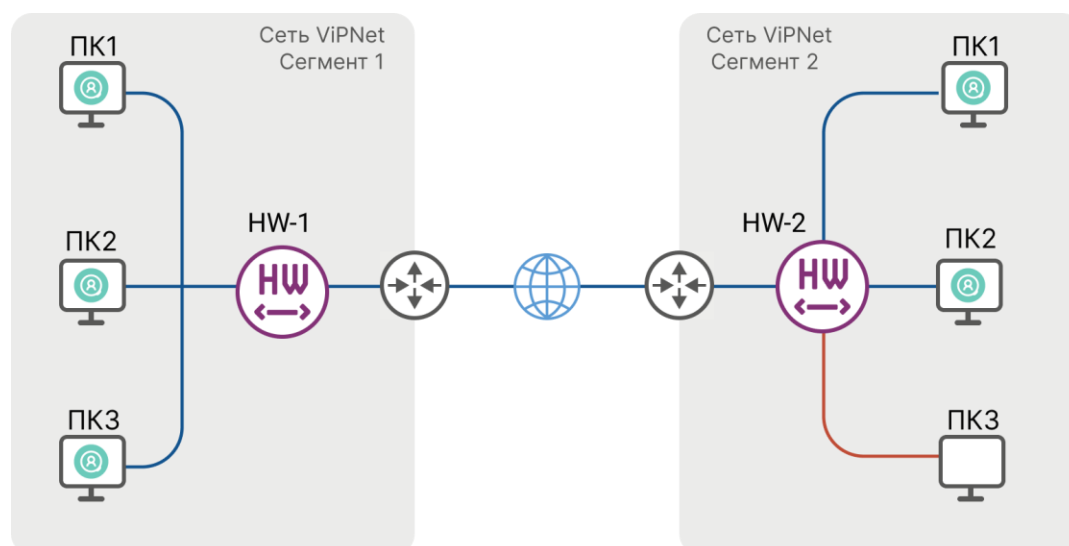


Рисунок 12. Объединение удаленных сегментов сети ViPNet

Использование ViPNet Coordinator HW в качестве защищенного интернет-шлюза

ViPNet Coordinator HW может выполнять функции защищенного интернет-шлюза. Эта технология позволяет организовать безопасное подключение к интернету отдельных узлов сети без их физического отключения от локальной сети и тем самым решить следующие задачи:

- предоставить пользователям доступ в интернет с рабочих мест без дополнительных затрат на создание и обслуживание специальной выделенной сети;
- обеспечить изоляцию локальной сети от узлов, работающих в интернете.

Технология защищенного интернет-шлюза основана на создании в локальной сети виртуального контура компьютеров, трафик которых при работе в сети интернет будет полностью изолирован от остальной локальной сети.

Чтобы использовать ViPNet Coordinator HW в качестве защищенного интернет-шлюза:

- 1 Установите на границе сети ViPNet Coordinator HW, для которого в ViPNet ЦУС или Prime была включена функция защищенного интернет-шлюза.
- 2 Выполните на ViPNet Coordinator HW настройки, приведенные в данном разделе ниже.
- 3 Установите на компьютеры пользователей программу ViPNet Client for Windows — эти компьютеры будут выполнять в сети ViPNet роль **клиентов**.

Те клиенты ViPNet, которым требуется разрешить доступ в интернет, должны иметь связь с ViPNet Coordinator HW. В настройках браузера этих клиентов должен быть указан IP-адрес и порт прокси-сервера ViPNet Coordinator HW (подробнее о настройке клиентов, которым разрешен доступ в интернет, см. в документе «ViPNet Client for Windows. Руководство пользователя»).

Полученная в результате конфигурация сети представлена на рисунке ниже. Клиенты, которым запрещен доступ в интернет, обозначены как группа А. Клиенты, которым разрешен доступ в интернет, обозначены как группа Б.

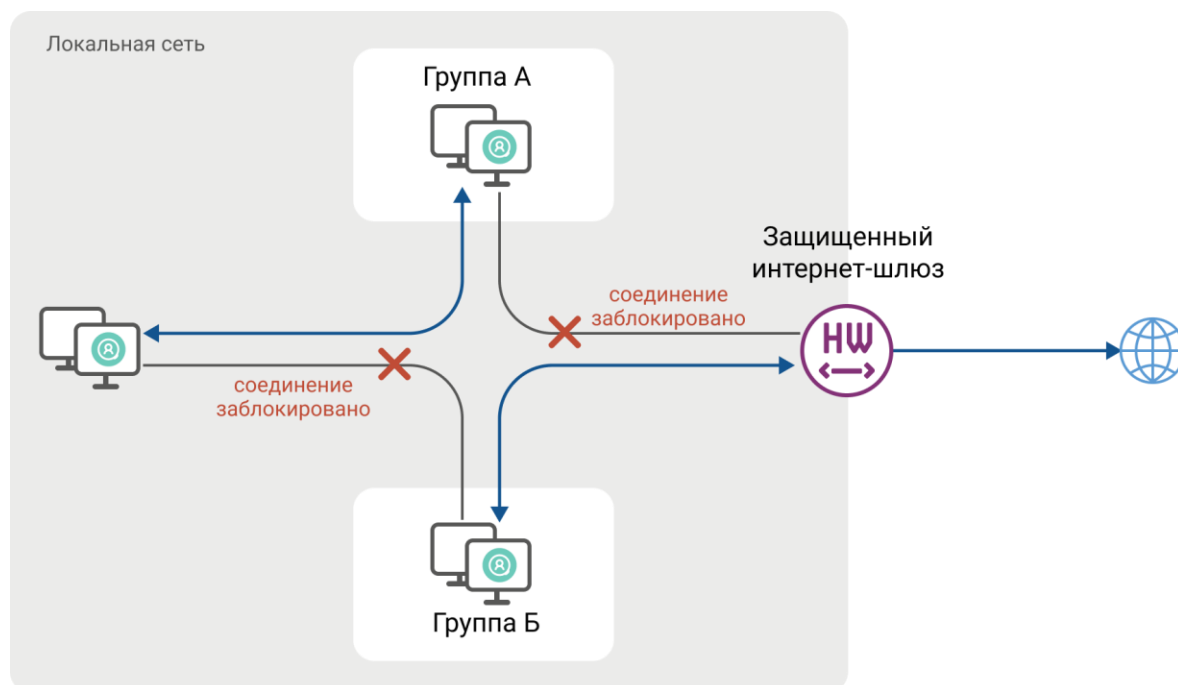


Рисунок 13. Использование ViPNet Coordinator HW в качестве защищенного интернет-шлюза

Защита сети обеспечивается путем разделения времени работы клиентов группы Б в сети интернет и в локальной сети. Клиенты, связанные с защищенным интернет-шлюзом, могут работать только в одном из следующих режимов:

- Работа в интернете. В этом режиме блокируется любое взаимодействие как с локальной сетью, так и с другими сетевыми узлами, кроме защищенного интернет-шлюза.

- Работа в локальной сети. В этом режиме блокируется любое взаимодействие с защищенным интернет-шлюзом, доступ в интернет невозможен.

В качестве защищенного интернет-шлюза ViPNet Coordinator HW выполняет следующие функции:

- запрещает доступ к ресурсам интернета для клиентов группы А;
- разрешает доступ к ресурсам интернета для клиентов группы Б;
- выполняет функции межсетевого экрана, который запрещает доступ в локальную сеть из интернета.

Для настройки ViPNet Coordinator HW, используемого в качестве защищенного интернет-шлюза (предполагается, что сетевые настройки соответствуют приведенным на рисунке выше):

- 1 Завершите работу прокси-сервера, если он запущен:

```
hostname# service http-proxy stop
```

- 2 Укажите IP-адрес и порт, на котором прокси-сервер будет принимать запросы от пользователей:

```
hostname# service http-proxy listen-address add eth0 3128
```

- 3 Укажите внешний IP-адрес ViPNet Coordinator HW, который используется для подключения прокси-сервера к интернету:

```
hostname# service http-proxy external-address set eth2
```

- 4 Добавьте сетевые фильтры и правила трансляции для работы прокси-сервера:

```
hostname# service http-proxy fw-rules apply
```

- 5 Включите автоматический запуск прокси-сервера при загрузке ViPNet Coordinator HW:

```
hostname# service http-proxy mode on
```

- 6 Запустите службу прокси-сервера:

```
hostname# service http-proxy start
```

- 7 При необходимости настройте другие параметры прокси-сервера (см. [Настройка прокси-сервера](#)).

- 8 Добавьте локальный фильтр открытой сети, запрещающий соединения со стороны внутреннего сегмента локальной сети:

```
hostname# firewall local add src interface eth0 dst @PrivateNetworkIP drop
```



Внимание! Обратите внимание на настройку маршрутизации на ViPNet Coordinator HW. Указывайте в качестве шлюза по умолчанию шлюз, который сообщит ваш интернет-провайдер. Если потребуется обеспечить маршрутизацию между сегментами локальной сети, используйте статические маршруты.

После произведенных настроек ViPNet Coordinator HW будет выполнять функции защищенного интернет-шлюза.

Настройка туннелирования на канальном уровне по технологии L2OverIP

Общее описание технологии L2OverIP

ViPNet Coordinator HW поддерживает технологию [L2OverIP](#), которая позволяет организовать защиту удалённых сегментов сети, использующих одно адресное пространство, на канальном уровне модели OSI. В результате сетевые узлы из разных сегментов смогут взаимодействовать друг с другом так, как будто они находятся в одном сегменте с прямой видимостью по MAC-адресам. Такая защита может потребоваться, например, для организации работы территориально распределённых ЦОДов (центров обработки данных), локальные сети которых объединены высокоскоростным каналом связи и представляют собой единый широковещательный домен.

Технология L2OverIP предполагает взаимодействие между узлами нескольких удалённых сегментов сети через ViPNet Coordinator HW, которые установлены на границах этих сегментов. ViPNet Coordinator HW перехватывает Ethernet-кадры, отправленные из его сегмента сети в другой, упаковывает их в IP-пакеты специального формата и передаёт эти IP-пакеты другому ViPNet Coordinator HW по защищённому каналу. ViPNet Coordinator HW, получивший IP-пакеты специального формата, извлекает из них исходные кадры и передаёт получателям в своем сегменте.

Сетевой интерфейс ViPNet Coordinator HW, к которому подключаются узлы сегмента сети, должен иметь класс:

- `access` со статическим IP-адресом для физических интерфейсов `eth`, виртуальных интерфейсов `vlan` или агрегированных интерфейсов `bond`;
- `trunk` для физических интерфейсов `eth` или агрегированных интерфейсов `bond`.

С помощью L2OverIP можно объединить несколько сегментов сети, в том числе VLAN:

- сегменты без VLAN, находящиеся за сетевыми интерфейсами класса `access`;
- виртуальная сеть VLAN из одного сегмента и виртуальная сеть VLAN из другого сегмента;
- все виртуальные сети VLAN из одного сегмента со всеми виртуальными сетями другого сегмента (для этого необходимо указать сетевые интерфейсы класса `trunk`, за которыми находятся виртуальные сети VLAN; кроме того, адреса виртуальных сетей VLAN должны совпадать);
- одна из виртуальных сетей VLAN сегмента и сегмент без VLAN, находящийся за сетевым интерфейсом класса `access`.

При использовании L2OverIP ViPNet Coordinator HW работает как виртуальный сетевой коммутатор, хранящий в памяти таблицу MAC-адресов сетевых узлов, от которых поступает сетевой трафик.

Каждому сегменту сети назначается свой номер порта, который задается в настройках L2OverIP. Порт, заданный для собственного сегмента сети, называется локальным. Порты, заданные на ViPNet Coordinator HW в других сегментах, называются удаленными. Трафик самого ViPNet Coordinator HW всегда относится к порту с номером 0.

Виртуальный коммутатор может по-разному обрабатывать одноадресные Ethernet-кадры с неизвестным MAC-адресом получателя: блокировать либо обрабатывать как многоадресные с рассылкой на все удаленные порты. Последняя возможность позволяет использовать L2OverIP для схемы, когда два ViPNet Coordinator HW объединены с помощью агрегированного канала и каждый из физических каналов работает через свою пару ViPNet Coordinator HW. Кадры, которыми обмениваются два узла из разных сегментов, могут в одном направлении постоянно пересылаться через одну пару ViPNet Coordinator HW, а в другом направлении — через другую. В этом случае адрес получателя кадров будет определяться как неизвестный, и для обработки пакетов многоадресной рассылки следует использовать режим `smart-broadcast`.

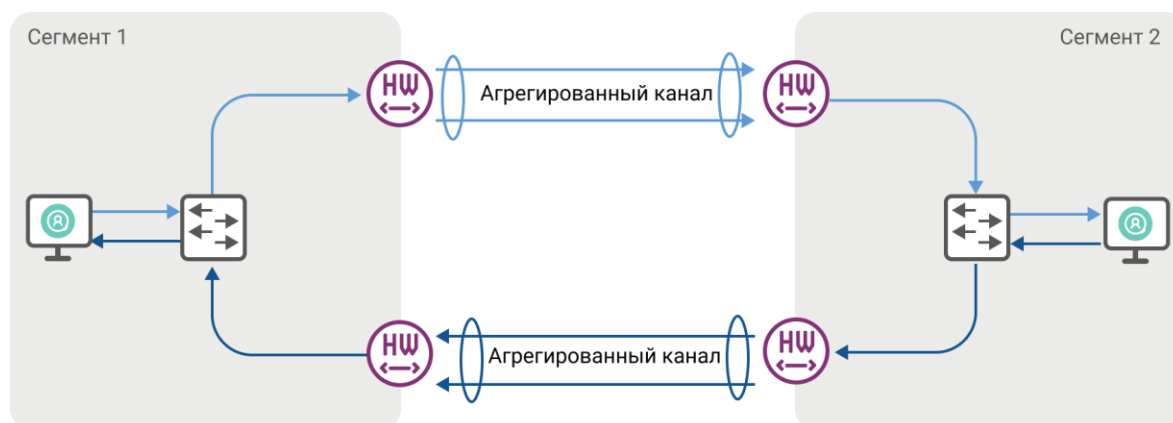


Рисунок 14. Схема работы сети с парами ViPNet Coordinator HW, объединенных с помощью агрегированных каналов

Рассылка Ethernet-кадров с неизвестным MAC-адресом получателя выполняется в соответствии с режимом обработки:

- кадры, принятые из локального порта, пересылаются на все удаленные порты и на порт с номером 0;
- кадры, принятые из удаленного порта, пересылаются на локальный порт и на порт с номером 0;
- кадры, принятые из порта с номером 0, в зависимости от выбранного режима либо блокируются, либо пересылаются на локальный порт и на все удаленные порты.

L2OverIP можно использовать в кластере горячего резервирования. Настройка L2OverIP выполняется на активном узле. Таблица MAC-адресов не передается пассивному узлу, и при переключении пассивного узла в активный режим его таблица пуста. Это приводит к увеличению времени конвергенции сети после смены активного узла — до тех пор, пока узлы из разных сегментов, взаимодействующие друг с другом, не отправят заново ARP-запросы.

Особенности реализации L2OverIP

- L2OverIP не поддерживается в исполнениях HW10 и HW50.
- Сегменты сети, объединяемые по L2OverIP, должны быть изолированы друг от друга. Сетевое взаимодействие между ними должно осуществляться только по каналам связи между ViPNet Coordinator HW, установленными на границах объединяемых сегментов.
- Использование беспроводных интерфейсов в L2OverIP не поддерживается.
- Текущая версия ViPNet Coordinator HW позволяет объединять не более 31 сегмента сети.
- Используйте не более 252 IP-адресов во всех объединяемых сегментах сети. Превышение этого количества может привести к серьезному снижению производительности сети в объединяемых сегментах. С увеличением числа IP-адресов в одном широковещательном домене значительно возрастает количество служебных пакетов, что приводит к снижению пропускной способности сети и увеличению нагрузки на ViPNet Coordinator HW.
- Для исполнения ViPNet Coordinator VA в настройках платформы виртуализации включите неразборчивый режим (Promiscuous Mode) для интерфейса, к которому привязан адаптер виртуальной машины.
- В L2OverIP не поддерживается автоматическое переключение между провайдерами при использовании MultiWAN. Это можно сделать вручную. При переключении между провайдерами перенастройте значение IP-адреса local-port в настройках L2OverIP на адрес интерфейса активного в данный момент времени провайдера.
- Поиск IP-пакетов, переданных по L2OverIP, в журнале регистрации IP-пакетов необходимо выполнять по признакам исходных IP-пакетов: IP-адрес источника и получателя, номер TCP- и UDP-порта, интерфейс.
- При настройке параметров удаленного сегмента сети, в зависимости от настройки видимости ViPNet Coordinator HW, установленного на границе этого сегмента, укажите реальный или виртуальный адрес. Если в процессе работы видимость поменялась, обязательно поменяйте его адрес в настройках (`iplir set l2overip remote-port <порт> <IP-адрес>`).
- Если вы меняете настройки удаленного сегмента сети (`iplir set l2overip remote-port`) при включенной функции, перезагрузите L2OverIP:
 - o `hostname# iplir set l2overip mode none`
 - o `hostname# iplir set l2overip mode switch`
- Если интерфейс, заданный для сегмента сети L2OverIP, используется туннелируемыми узлами, то трафик из внутренней сети до туннелируемых узлов будет блокироваться.

Чтобы пропускать трафик, добавьте разрешающее правило на координаторе, который туннелирует узел:

```
hostname# firewall local add src <IP-адрес туннелируемого узла> dst @local <протокол> pass
```

Организация защиты соединения между удаленными сегментами сети на канальном уровне модели OSI

При организации соединения между удаленными сегментами сети на канальном уровне каждый сегмент сети подключается к одному из сетевых интерфейсов ViPNet Coordinator HW, который установлен на границе сегмента. Если сегмент сети состоит из нескольких VLAN, они должны быть объединены с помощью коммутатора в транковый порт, к которому подключается один из сетевых интерфейсов ViPNet Coordinator HW.

Чтобы организовать защиту соединения сегментов, на каждом ViPNet Coordinator HW включите функцию L2OverIP и задайте ее параметры:

- 1 Укажите сетевой интерфейс, сегмент сети которого будет объединяться с другим сегментом с помощью функции L2OverIP:

```
hostname# iplir set l2overip interface <рабочий интерфейс>
```



Примечание. Если требуется объединить отдельную VLAN одного сегмента сети с другим сегментом сети, на ViPNet Coordinator HW, стоящем на границе сегмента с VLAN, укажите в качестве рабочего интерфейса виртуальный интерфейс VLAN в формате <физический интерфейс>.<номер виртуального интерфейса>.

- 2 Задайте параметры локального сегмента сети, указав уникальный номер порта и IP-адрес внешнего интерфейса:

```
hostname# iplir set l2overip local-port <номер порта> <IP-адрес внешнего интерфейса>
```

В качестве номера порта вы можете задать значение из интервала 1–31. Каждому сегменту необходимо назначить свой номер порта.

- 3 Задайте параметры удаленного сегмента сети, указав его номер порта и адрес видимости внешнего интерфейса (local-port) удаленного ViPNet Coordinator HW:

```
hostname# iplir set l2overip remote-port <номер порта> <IP-адрес видимости удаленного ViPNet Coordinator HW>
```

В зависимости от настройки видимости удаленного ViPNet Coordinator HW укажите реальный или виртуальный адрес.



Примечание. Вы можете удалить параметры удаленного сегмента:

```
hostname# iplir set l2overip remote-port <номер порта> delete
```

или изменить IP-адрес видимости:

```
hostname# iplir set l2overip remote-port <номер порта> <новый IP-адрес>
```

- 4 При необходимости измените время жизни MAC-адреса в таблице MAC-адресов виртуального коммутатора (см. [Общее описание технологии L2OverIP](#)) при отсутствии трафика, поступающего от этого адреса:

```
hostname# iplir set l2overip mac-ttl <время жизни MAC-адреса>
```

Время жизни MAC-адреса задается в секундах. Значение по умолчанию — 300 секунд. Задайте значение из интервала 60–86400.

- 5 Добавьте сетевые фильтры защищенной сети, разрешающие любые соединения по протоколу 97:

```
hostname# firewall vpn add src @local dst @any proto 97 pass
```

```
hostname# firewall vpn add src @any dst @local proto 97 pass
```

- 6 Включите функцию L2OverIP:

```
hostname# iplir set l2overip mode switch
```

- 7 Проверьте текущее состояние и настройки функции L2OverIP:

```
hostname# iplir show l2overip config
```

После настройки и включения функции L2OverIP на всех ViPNet Coordinator HW взаимодействие между узлами удаленных сегментов сети будет защищено на канальном уровне модели OSI.



Внимание! На двух ViPNet Coordinator HW, между которыми организовано соединение L2OverIP, IP-адреса локального и удаленного сегментов должны быть настроены симметричным образом. То есть на каждом ViPNet Coordinator HW в качестве IP-адреса удаленного сегмента должен быть указан IP-адрес видимости сетевого интерфейса, который задан на втором ViPNet Coordinator HW в качестве внешнего интерфейса локального сегмента.

Примеры организации защищенного соединения между удаленными сегментами сети с использованием технологии L2OverIP см. в разделе [Примеры настройки функции L2OverIP](#).

Примеры настройки функции L2OverIP

Настройка функции L2OverIP при отсутствии VLAN

В самом простом случае функция L2OverIP используется для объединения и защиты соединения двух удаленных сегментов сети. В каждом сегменте могут находиться как открытые, так и защищенные или туннелируемые узлы.

Каждый сегмент сети подключается к одному из интерфейсов ViPNet Coordinator HW, установленному на границе сегмента. Ниже приведена схема подключения, на которой два удаленных сегмента сети объединяются с помощью координаторов HW-1 и HW-2. По такой же схеме можно объединить больше двух сегментов сети, но не более 31.

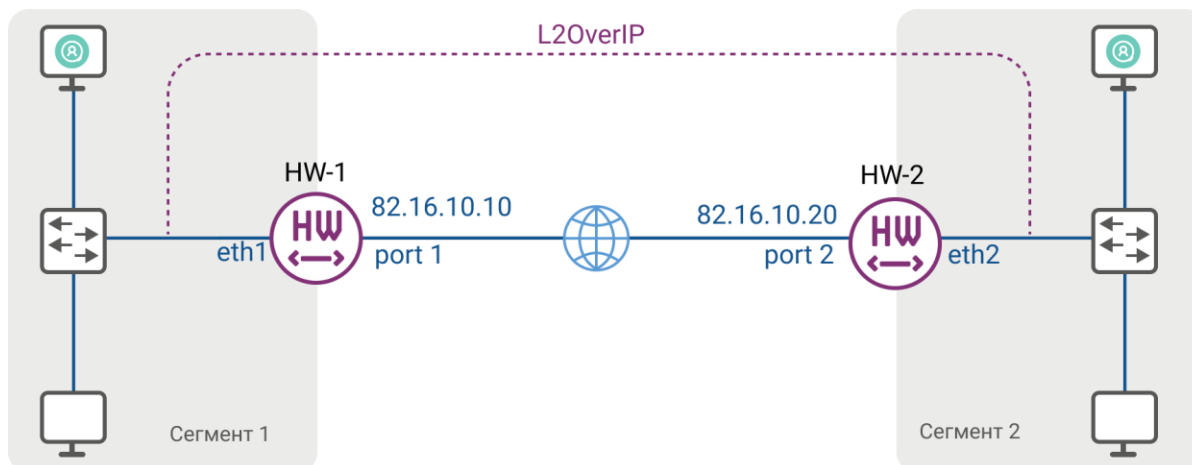


Рисунок 15. Защита соединения удаленных сегментов сети на канальном уровне модели OSI

Чтобы организовать защиту соединения сегментов, на каждом ViPNet Coordinator HW настройте функцию L2OverIP:

- 1 Укажите сетевой интерфейс, сегмент сети которого будет объединяться с другим сегментом с помощью функции L2OverIP:

- На HW-1:

```
hostname# iplir set l2overip interface eth1
```

- На HW-2:

```
hostname# iplir set l2overip interface eth2
```

Если сегментов больше двух, аналогичным образом укажите рабочий интерфейс на остальных ViPNet Coordinator HW.

- 2 Задайте параметры локального сегмента сети, указав уникальный номер порта и адрес внешнего интерфейса:

- На HW-1:

```
hostname# iplir set l2overip local-port 1 82.16.10.10
```

- На HW-2:

```
hostname# iplir set l2overip local-port 2 82.16.10.20
```

Если сегментов больше двух, укажите номер порта и адрес внешнего интерфейса на остальных ViPNet Coordinator HW. Каждому сегменту необходимо назначить свой номер порта.

- 3 Задайте параметры удаленного сегмента сети, указав его номер порта и актуальный адрес видимости удаленного ViPNet Coordinator HW:

- На HW-1:

```
hostname# iplir set l2overip remote-port 2 82.16.10.20
```

- На HW-2:

```
hostname# iplir set l2overip remote-port 1 82.16.10.10
```

В зависимости от настройки видимости удаленного ViPNet Coordinator HW укажите его реальный или виртуальный адрес.

Если сегментов больше двух, добавьте параметры всех удаленных сегментов и выполните аналогичную настройку на остальных ViPNet Coordinator HW.



Примечание. Вы можете удалить из настроек параметры удаленного сегмента:

```
hostname# iplir set l2overip remote-port <номер порта> delete
```

или поменять адрес видимости удаленного сегмента, если он изменился:

```
hostname# iplir set l2overip remote-port <номер порта> <новый адрес>
```

- 4 Добавьте сетевые фильтры защищенной сети, разрешающие любые соединения по протоколу 97:

```
hostname# firewall vpn add src @local dst @any proto 97 pass
```

```
hostname# firewall vpn add src @any dst @local proto 97 pass
```

- 5 Включите функцию L2OverIP:

```
hostname# iplir set l2overip mode switch
```

- 6 Убедитесь в корректности выполненных настроек функции L2OverIP:

```
hostname# iplir show l2overip config
```

После настройки и включения функции L2OverIP на всех ViPNet Coordinator HW взаимодействие между узлами удаленных сегментов сети будет защищено на канальном уровне модели OSI.

Настройка функции L2OverIP в случае использования VLAN

С помощью функции L2OverIP можно объединять сегменты сети, состоящие из нескольких виртуальных локальных сетей [VLAN](#).

Например, пусть требуется объединить два сегмента, в каждом из которых есть по два VLAN с одинаковыми номерами. В каждом сегменте виртуальные сети должны быть объединены с помощью коммутатора в транковый порт, к которому подключается один из интерфейсов ViPNet Coordinator HW. На рисунке ниже приведена схема подключения.

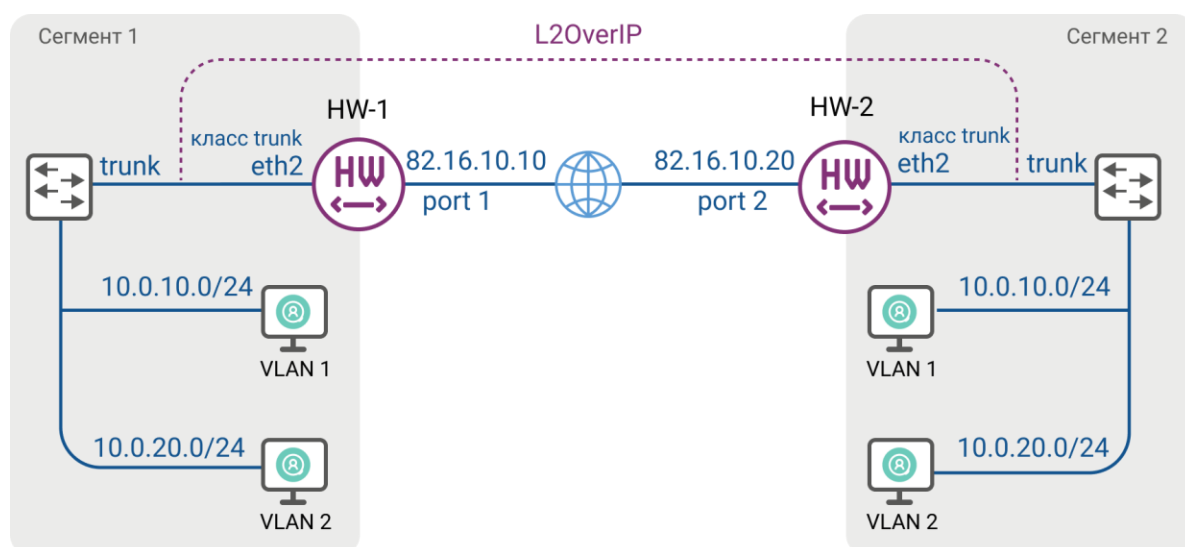


Рисунок 16. Схема объединения сегментов с VLAN

Для настройки функции L2OverIP в такой схеме на каждом ViPNet Coordinator HW:

- 1 Установите класс trunk для сетевого интерфейса eth2:

```
hostname# inet ifconfig eth2 class trunk
```

- 2 Укажите сетевой интерфейс eth2, сегмент сети которого будет объединяться с другим удаленным сегментом с помощью функции L2OverIP:

```
hostname# iplir set l2overip interface eth2
```

- 3 Укажите параметры локального сегмента сети:

- На HW-1:

```
hostname# iplir set l2overip local-port 1 82.16.10.10
```

- На HW-2:

```
hostname# iplir set l2overip local-port 2 82.16.10.20
```

- 4 Укажите параметры удаленного сегмента сети:

- На HW-1:

```
hostname# iplir set l2overip remote-port 2 82.16.10.20
```

- На HW-2:

```
hostname# iplir set l2overip remote-port 1 82.16.10.10
```

- 5 Добавьте сетевые фильтры защищенной сети, разрешающие любые соединения по протоколу 97:

```
hostname# firewall vpn add src @local dst @any proto 97 pass
```

```
hostname# firewall vpn add src @any dst @local proto 97 pass
```

- 6 Включите функцию L2OverIP:

```
hostname# iplir set l2overip mode switch
```

В результате взаимодействие между узлами удаленных сегментов в каждой виртуальной сети будет защищено на канальном уровне модели OSI.

Если в приведенном примере требуется объединить только одну виртуальную сеть одного сегмента с той же виртуальной сетью другого сегмента, то в качестве рабочего интерфейса для функции L2OverIP вместо физического интерфейса eth2 укажите виртуальный интерфейс для соответствующей VLAN (например, eth2.1).

Возможен вариант, когда в одном из сегментов есть виртуальные сети, а в другом — нет. На рисунке ниже приведен пример, когда требуется объединить одну виртуальную сеть одного сегмента с другим сегментом сети, в котором нет виртуальных сетей.

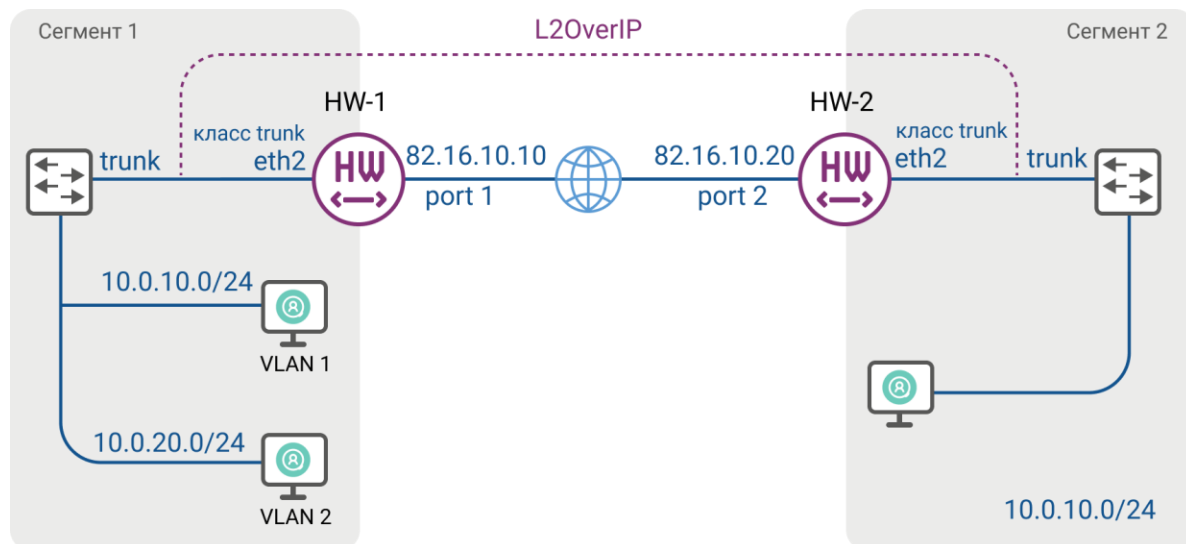


Рисунок 17. Схема объединения одним VLAN с сегментом без VLAN

В этом случае измените настройку следующим образом:

- Создайте виртуальные интерфейсы для VLAN только на HW-1.
- На HW-1 в качестве рабочего интерфейса для функции L2OverIP укажите виртуальный интерфейс для VLAN:

```
hostname# iplir set l2overip interface eth2.1
```

- На HW-2 в качестве рабочего интерфейса для функции L2OverIP укажите физический интерфейс, к которому подключен сегмент:

```
hostname# iplir set l2overip interface eth2
```

Настройка функции L2OverIP для ViPNet Coordinator VA

Для объединения и защиты соединения удаленных сегментов сети на канальном уровне модели OSI вы можете использовать не только аппаратные исполнения ViPNet Coordinator HW, но и исполнение ViPNet Coordinator VA, установленное на одной из поддерживаемых платформ виртуализации.

Рассмотрим случай, аналогичный описанному в разделе [Настройка функции L2OverIP при отсутствии VLAN](#), но в котором используются два координатора ViPNet Coordinator VA, развернутые в VMware vSphere ESXi.



Примечание. В этом примере приведены настройки для VMware vSphere ESXi. В других поддерживаемых платформах виртуализации необходимо задавать аналогичные настройки.

Для организации защищенного соединения L2OverIP по такой схеме перед выполнением настроек, приведенных в разделе [Настройка функции L2OverIP при отсутствии VLAN](#), в программе vSphere Client выполните:

- 1 Перейдите в раздел **Home > Inventory > Hosts and Clusters**.
- 2 В окне настроек виртуального коммутатора vSphere Standard Switch откройте вкладку **Security** и в списке **Promiscuous Mode** выберите пункт **Accept**. В результате будет выбран «неразборчивый» режим, при котором разрешен прием всех пакетов вне зависимости от того, кому они адресованы.
- 3 Убедитесь, что в списках **MAC Address Changes** и **Forged Transmits** также выбран пункт **Accept**.

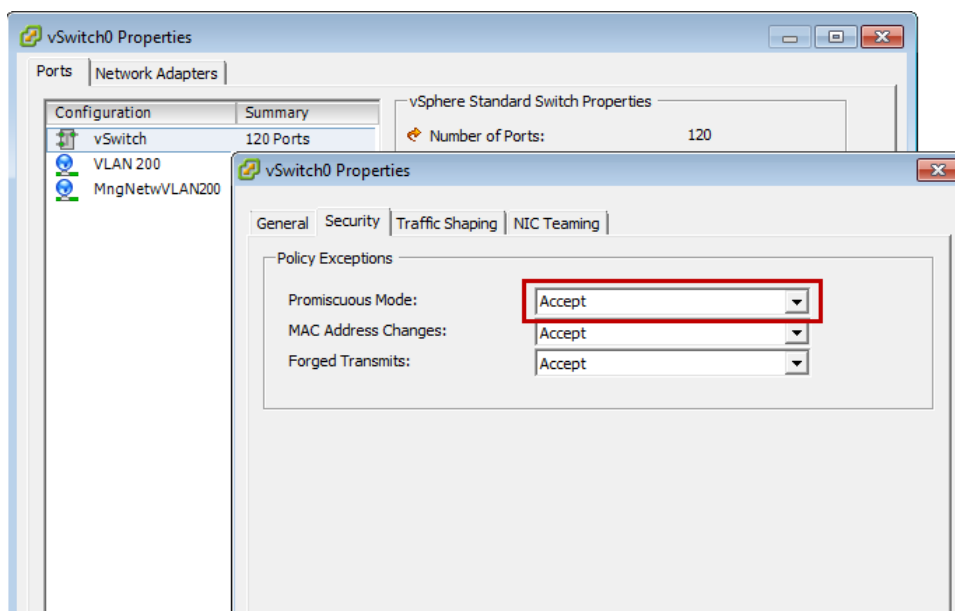


Рисунок 18. Установка неразборчивого режима на виртуальном коммутаторе



Примечание. Если вы используете распределенный виртуальный коммутатор vSphere Distributed Switch, задайте аналогичные настройки для этого коммутатора в разделе **Home > Inventory > Networking**.

В результате после задания параметров, приведенных в разделе [Настройка функции L2OverIP при отсутствии VLAN](#), защита соединения удаленных сегментов сети на канальном уровне модели OSI будет корректно работать.

4

Настройка сетевых фильтров

Основные принципы фильтрации трафика	103
Общие сведения о сетевых фильтрах	108
Группы объектов	110
Создание сетевого фильтра	116
Просмотр сетевых фильтров	123
Изменение сетевого фильтра	125
Удаление сетевого фильтра	126
Блокирование сайтов по доменным именам с помощью сетевых фильтров	127

Основные принципы фильтрации трафика

Виды трафика

Фильтрации подвергается весь трафик, который проходит через ViPNet Coordinator HW:

-  — незашифрованный (открытый) трафик между открытыми (или открытым и защищенным) узлами;
-  — зашифрованный (защищенный) трафик между узлами сети ViPNet.

Трафик может быть локальным или транзитным:

- локальный трафик — трафик, в котором ViPNet Coordinator HW является получателем или отправителем IP-пакетов;
- транзитный трафик — трафик, в котором ViPNet Coordinator HW не является ни получателем, ни отправителем IP-пакетов.

ViPNet Coordinator HW обрабатывает трафик, разделяя его на следующие виды:

- локальный незашифрованный трафик — между открытым узлом и ViPNet Coordinator HW. Обработывается локальными фильтрами открытой сети;
- локальный зашифрованный трафик — трафик между узлами сети ViPNet, при этом ViPNet Coordinator HW является получателем или отправителем IP-пакетов. Обработывается фильтрами защищенной сети;
- туннелируемый трафик — трафик между двумя открытыми (или открытым и защищенным) узлами, зашифрованный в пределах сети ViPNet. Обработывается фильтрами туннелируемых узлов;

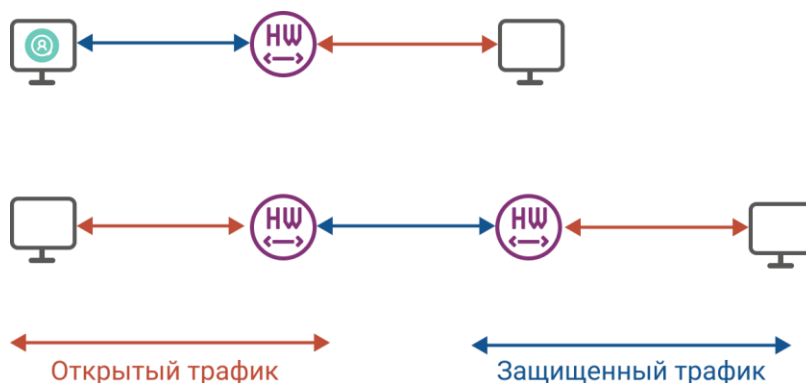


Рисунок 19. Туннелируемый трафик

- транзитный трафик — зашифрованный или незашифрованный трафик между узлами, проходящий через координатор, который не является ни отправителем, ни получателем IP-пакетов. Обрабатывается транзитными фильтрами открытой сети.

Принципы фильтрации

Принципы фильтрации трафика:

- Все входящие и исходящие открытые и защищенные IP-пакеты обрабатываются сетевыми фильтрами в порядке убывания приоритета этих фильтров.
- Если IP-пакет соответствует условию одного из фильтров, он пропускается или блокируется этим фильтром. При этом фильтры с более низким приоритетом не применяются.
- Если IP-пакет был пропущен одним из фильтров, ответные IP-пакеты (включая служебные ICMP-пакеты) в рамках текущего соединения будут пропускаться автоматически.
- Если IP-пакет не был обработан ни одним из фильтров, он блокируется фильтром по умолчанию.

ViPNet Coordinator HW обрабатывает входящий трафик следующим образом:

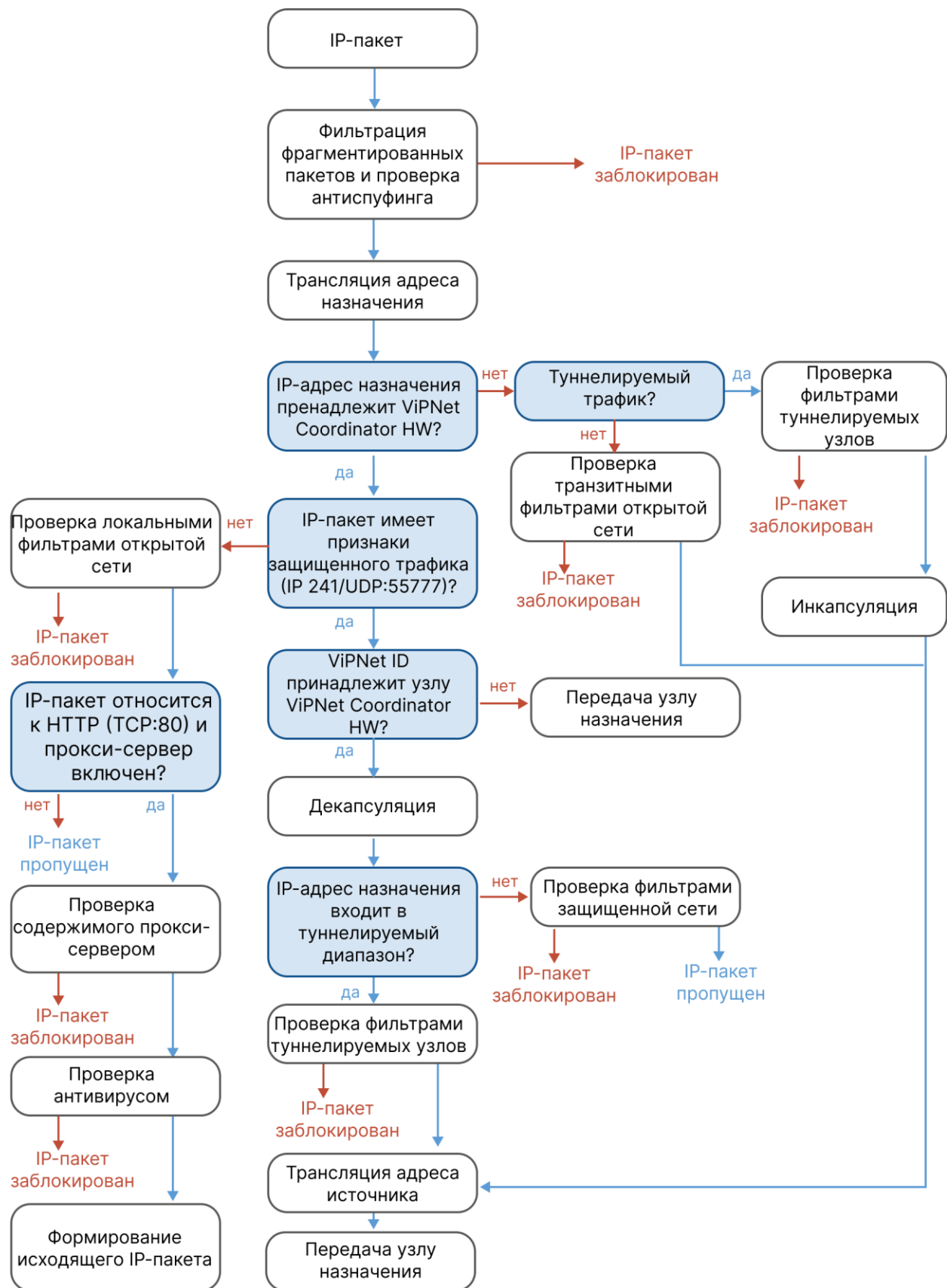


Рисунок 20. Схема фильтрации трафика

IP-пакет, поступивший на сетевой интерфейс ViPNet Coordinator HW, последовательно проходит ряд проверок:

- 1 IP-пакет проверяется на фрагментацию, если эта проверка включена. Если IP-пакет фрагментирован, он блокируется, иначе — передается на следующий шаг проверки.
- 2 IP-пакет обрабатывается правилами антиспуфинга, если он включен (см. [Настройка антиспуфинга](#)). В результате IP-пакет блокируется или передается на следующий шаг проверки.
- 3 В случаях, указанных ниже, выполняется трансляция IP-адреса назначения (см. [Трансляция адреса назначения](#)):
 - IP-пакет удовлетворяет условиям правила трансляции адресов (см. [Создание правила трансляции адресов](#)).
 - Включена трансляция прикладного протокола, к которому относится IP-пакет (см. [Настройка обработки прикладных протоколов](#)).
 - IP-пакет соответствует протоколу HTTP (TCP: 80) и прокси-сервер работает в «прозрачном» режиме (см. [Настройка прокси-сервера](#)). В этом случае с помощью трансляции адреса назначения транзитный трафик перенаправляется на ViPNet Coordinator HW и обрабатывается как локальный трафик открытой сети.
- 4 Если IP-пакет адресован ViPNet Coordinator HW и имеет признаки защищенного трафика (протокол IP/241 или протокол UDP и порт 55777 или другой порт, настроенный администратором), то в IP-пакете проверяется идентификатор узла ViPNet (ViPNet ID):
 - Если ViPNet ID не принадлежит ViPNet Coordinator HW, в IP-пакете преобразуются IP-адреса и порты в соответствии с имеющейся информацией о ближайшей к узлу назначения точке доступа. Затем IP-пакет отправляется в сеть для передачи узлу назначения.
 - Если ViPNet ID принадлежит ViPNet Coordinator HW, выполняется расшифровка и декапсуляция IP-пакета. Полученный IP-пакет (исходный IP-пакет) проверяется на принадлежность туннелируемому трафику:
 - Если IP-адрес назначения в исходном IP-пакете входит в туннелируемый диапазон ViPNet Coordinator HW, IP-пакет проверяется фильтрами туннелируемых узлов. В результате IP-пакет блокируется или отправляется на туннелируемый узел после трансляции адреса источника (см. [Трансляция адреса источника](#)).
 - Если IP-адрес назначения в исходном IP-пакете не входит в туннелируемый диапазон ViPNet Coordinator HW, IP-пакет проверяется фильтрами защищенной сети, после чего блокируется или пропускается.
- 5 Если IP-пакет адресован ViPNet Coordinator HW и не имеет признаков защищенного трафика, IP-пакет проверяется локальными фильтрами открытой сети. В результате IP-пакет блокируется или проходит следующие проверки:
 - 5.1 Если IP-пакет относится к протоколу HTTP (TCP: 80), он проверяется прокси-сервером на прикладном уровне. Если прокси-сервер выключен, IP-пакет пропускается.
 - 5.2 После проверки прокси-сервером выполняется антивирусная проверка IP-пакета, если она включена (см. [Настройка антивирусной проверки трафика](#)). В результате IP-пакет блокируется или формируется новый исходящий IP-пакет.

- 6 Если IP-пакет адресован не ViPNet Coordinator HW, а другому узлу, проверяется принадлежность IP-пакета туннелируемому трафику:
- Если IP-адрес отправителя входит в туннелируемый диапазон ViPNet Coordinator HW, а IP-адрес получателя входит в туннелируемый диапазон другого координатора или принадлежит защищенному узлу, IP-пакет проверяется фильтрами туннелируемых узлов. В результате IP-пакет блокируется или отправляется получателю после процедур инкапсуляции и трансляции адреса источника.
 - Если IP-адрес отправителя входит в туннелируемый диапазон ViPNet Coordinator HW, но при этом принадлежит защищенному узлу, то трафик не является туннелируемым и проверяется транзитными фильтрами открытой сети.
 - Если IP-пакет не принадлежит туннелируемому трафику, он проверяется транзитными фильтрами открытой сети. В результате IP-пакет блокируется или отправляется получателю после трансляции адреса источника.

Особенности при настройке фильтрации трафика на ViPNet Coordinator HW:

- TCP-порт 2047 и UDP-порт 2046 являются служебными портами ViPNet Coordinator HW, поэтому входящий открытый трафик, поступающий на эти порты, блокируется без возможности настройки.
- Изменения в конфигурации межсетевого экрана, влияющие на открытый транзитный трафик, начинают действовать сразу, в том числе для уже установленных ранее соединений.

Изменения в конфигурации межсетевого экрана, влияющие на другие виды трафика, будут действовать только для новых соединений. К уже установленным соединениям изменения не будут применены до истечения времени жизни соединения. Чтобы принудительно применить изменения ко всем соединениям, поочередно выполните:

```
hostname# vpn stop
hostname# vpn start
```

Общие сведения о сетевых фильтрах

Различаются сетевые фильтры для защищенного трафика, для открытого трафика (локального и транзитного) и для туннелируемого трафика. Они выполняют следующие функции:

- Фильтры открытой сети могут разрешать либо блокировать обмен IP-трафиком с [открытыми узлами](#).



Примечание. К [открытым узлам](#) также относятся компьютеры с ПО ViPNet Registration Point.

- Фильтры защищенной сети могут разрешать или блокировать обмен IP-трафиком с [защищенными узлами ViPNet](#), с которыми данный узел ViPNet Coordinator HW имеет связь.
- Фильтры для туннелируемого трафика могут разрешать или блокировать IP-пакеты, передаваемые между [туннелируемыми узлами](#) и узлами сети ViPNet, с которыми данный координатор имеет связь.



Внимание! Работа с фильтрами туннелируемых узлов возможна только при наличии лицензии на туннелирование хотя бы одного соединения.

Различаются следующие сетевые фильтры:

- Служебные фильтры. Создаются в ViPNet Coordinator HW автоматически. Имеют самый высокий приоритет и недоступны для редактирования.
- Фильтры, поступившие в составе политик безопасности (фильтры из управляющего приложения). Политика безопасности представляет собой набор параметров, регулирующих безопасность сетевого узла. Она формируется администратором сети ViPNet в ViPNet Policy Manager или Policy Management (модуль ViPNet Prime), может включать сетевые фильтры и правила трансляции сетевых адресов и рассылается на сетевые узлы ViPNet с помощью [транспортного сервера MFTP](#). При получении политики безопасности из ViPNet Policy Manager или Policy Management (модуль ViPNet Prime) она немедленно применяется на узле.

Фильтры, поступившие в составе политик безопасности, применяются после служебных фильтров. Недоступны для редактирования.

- Предустановленные фильтры и фильтры, заданные пользователем. Предустановленные фильтры разрешают только некоторые типы IP-пакетов. Для работы с какими-либо дополнительными сервисами пользователю необходимо создать свои фильтры.

Применяются после фильтров, поступивших в составе политик безопасности. Доступны для редактирования.

- Блокирующий фильтр по умолчанию.

Применяется после всех фильтров. Недоступен для редактирования.

Последовательность применения сетевых фильтров в порядке убывания приоритета:



Рисунок 21. Последовательность применения сетевых фильтров

Сетевые фильтры могут включать в себя следующие параметры:

- условие — адрес отправителя и получателя IP-пакетов, на которые действует фильтр, и протоколы, используемые для передачи этих IP-пакетов (например, TCP, UDP, ICMP);
- расписание применения фильтра — ежедневно, еженедельно или по календарю;
- действие, применяемое к IP-пакетам — пропускать или блокировать IP-пакеты, соответствующие условию фильтра.

Группы объектов

Группы объектов позволяют упростить процессы создания и изменения сетевых фильтров и правил [трансляции сетевых адресов](#) в ViPNet Coordinator HW. Каждая группа объединяет несколько объектов одного типа (например, IP-адреса или сетевые интерфейсы). Группы можно указывать в параметрах сетевого фильтра или правила трансляции сетевых адресов вместо перечисления отдельных объектов.

В зависимости от типа объединяемых объектов различаются следующие группы:

- Группа сетевых узлов ViPNet — содержит любую комбинацию узлов сети ViPNet, используется в фильтрах защищенной сети и фильтрах туннелируемых узлов.
- Группа IP-адресов — содержит любую комбинацию IP-адресов, диапазонов IP-адресов и доменных имен, используется в фильтрах открытой сети, фильтрах туннелируемых узлов и правилах трансляции сетевых адресов.
- Группа сетевых интерфейсов — содержит любую комбинацию сетевых интерфейсов, используется в фильтрах открытой сети и фильтрах туннелируемых узлов.
- Группа протоколов — содержит любую комбинацию сетевых протоколов и портов, используется в фильтрах открытой и защищенной сетей, фильтрах туннелируемых узлов и правилах трансляции сетевых адресов.
- Группа расписания — содержит любую комбинацию параметров, определяющих время действия сетевого фильтра, используется в фильтрах открытой и защищенной сетей, фильтрах туннелируемых узлов.

Каждая группа объектов относится к одному из следующих видов:

- [Системные группы объектов](#) — настроенные по умолчанию группы с фиксированными именами, которые могут использоваться для задания адресов отправителей и получателей IP-пакетов в сетевых [фильтрах](#) и правилах трансляции адресов (см. [Создание правила трансляции адресов](#)), а также при [создании пользовательских групп объектов](#). Такие группы объектов не отображаются в списках групп (см. [Просмотр групп объектов](#)), их нельзя ни изменить, ни удалить.
- Группы объектов из ViPNet Policy Manager или Policy Management (модуль ViPNet Prime) — группы, получаемые в составе политик безопасности, и используемые в соответствующих фильтрах (см. [Общие сведения о сетевых фильтрах](#)). Такие группы нельзя ни удалять, ни изменять, ни использовать для задания параметров пользовательских фильтров и групп объектов.
- Пользовательские группы объектов — группы, создаваемые пользователем на узле, а также несколько [групп, настроенных по умолчанию](#). Такие группы можно использовать для задания параметров сетевых [фильтров](#) и правил трансляции адресов (см. [Создание правила трансляции адресов](#)), а также при [создании других пользовательских групп объектов](#).

Системные группы объектов

В таблице ниже описаны доступные системные группы объектов и указано соответствие имен системных групп объектов ViPNet Coordinator HW в командном интерпретаторе и веб-интерфейсе.

Таблица 2. Системные группы объектов

Имя системной группы объектов	Описание и область применения группы
<code>allclients</code> (Все клиенты)	Все клиенты , с которыми у узла есть связь. Можно использовать в фильтрах защищённой сети и фильтрах туннелируемых узлов для задания адреса отправителя для входящих соединений узла или адреса получателя для исходящих соединений узла.
<code>allcoordinators</code> (Все координаторы)	Все координаторы , с которыми у узла есть связь. Можно использовать в фильтрах защищённой сети и фильтрах туннелируемых узлов для задания адреса отправителя для входящих соединений узла или адреса получателя для исходящих соединений узла.
<code>broadcast</code> (Широковещательные адреса)	Все широковещательные адреса. Можно использовать в фильтрах защищённой сети и фильтрах открытой сети для идентификации широковещательных адресов получателей. Использование других адресов совместно с этой группой недопустимо.
<code>local</code> (Мой узел)	Собственный узел ViPNet Coordinator HW. Можно использовать в фильтрах защищённой сети и фильтрах открытой сети для задания адреса отправителя для исходящих соединений узла или адреса получателя для входящих соединений узла. Использование других адресов совместно с этой группой недопустимо.
<code>remote</code> (Другие узлы)	Другие узлы (любые, кроме собственного). Можно использовать в фильтрах защищённой сети и фильтрах открытой сети для задания адреса отправителя для входящих соединений узла или адреса получателя для исходящих соединений узла. Использование других адресов совместно с этой группой недопустимо.
<code>tunneledip</code> (Туннелируемые IP-адреса)	Все IP-адреса, туннелируемые координатором . Можно использовать в фильтрах туннелируемых узлов для задания адреса отправителя для входящих соединений узла или адреса получателя для исходящих соединений узла.

Имя системной группы объектов	Описание и область применения группы
multicast (Групповые адреса)	Диапазон IP-адресов для групповой рассылки (224.0.0.0 – 239.255.255.255). Можно использовать в фильтрах открытой сети для задания адреса получателя. Использование других адресов совместно с этой группой недопустимо.

Кроме того, в сетевых фильтрах и правилах трансляции сетевых адресов может использоваться объект `any`, обозначающий любое значение, то есть:

- для локальных и транзитных фильтров открытой сети — все IP-адреса;
- для фильтров защищённой сети — все VPN-узлы;
- для фильтров туннелируемых узлов — все IP-адреса или все VPN-узлы;
- для протоколов — все протоколы;
- для расписаний — все время (постоянно).

Использование других условий совместно с объектом `any` недопустимо.

Пользовательские группы объектов по умолчанию

По умолчанию в ViPNet Coordinator HW настроены следующие пользовательские группы объектов:

- Группы IP-адресов:
 - `PrivateNetworkIP` (частные IP-адреса) — IP-адреса локальных сетей: 10.0.0.0/8; 172.16.0.0/12; 192.168.0.0/16.
 - `InternetIP` (публичные IP-адреса) — IP-адреса за исключением частных IP-адресов.
 - `HttpProxyUsers` — IP-адреса локальных сетей, которым разрешен доступ к прокси-серверу.



Внимание! Группа объектов `HttpProxyUsers` автоматически создается при добавлении сетевых фильтров и правил трансляции, необходимых для работы прокси-сервера, с помощью команды

```
hostname# service http-proxy fw-rules apply
```

- Группы протоколов (см. [Пользовательские группы протоколов по умолчанию](#)).
- Группы расписаний:
 - `Workdays` — рабочие дни недели (с понедельника по пятницу).

- o Weekends — выходные дни недели (суббота и воскресенье).

Создание пользовательской группы объектов

Чтобы создать группу объектов, выполните:

```
hostname# firewall <тип объектов> add name @<имя> <состав> [exclude <исключения>], где:
```

- параметр <тип объектов> может принимать одно из следующих значений:
 - o ip-object — IP-адреса или доменные имена;
 - o vpn-object — узлы сети ViPNet;
 - o interface-object — сетевые интерфейсы;
 - o service-object — протоколы;
 - o schedule-object — расписания;
- имя группы объектов должно быть уникальным, допустимые символы: a–z, A–Z, 0–9, _ и точка;
- в параметрах <состав> и <исключения> указываются объекты, входящие и не входящие в группу соответственно, причем:
 - o IP-адреса и идентификаторы узлов сети ViPNet указываются через запятую или в виде диапазона (подробнее см. в разделах [Адрес отправителя](#) и [Адрес получателя](#));
 - o сетевые интерфейсы разделяются пробелом, и перед именем каждого сетевого интерфейса необходимо указать слово `interface`;
 - o протоколы или расписания указываются в соответствии с синтаксисом, описанным в разделах [Протокол](#) и [Расписание](#);
 - o количество объектов, входящих в группу, не должно превышать 1000.



Примечание. Если при вводе команды допущена синтаксическая ошибка, то после выполнения такой команды появится сообщение с указанием места ошибки. Исправьте ошибку и выполните команду еще раз.

В результате будет создана группа объектов, и вы сможете использовать ее для [задания параметров сетевых фильтров](#) и правил трансляции сетевых адресов (см. [Создание правила трансляции адресов](#)).

Примеры создания групп объектов

- Чтобы создать группу IP-адресов, содержащую сегмент сети за исключением нескольких IP-адресов:

```
hostname# firewall ip-object add name @IP_group_1 110.35.14.0/24 exclude  
110.35.14.3, 110.35.14.13
```

- Чтобы создать группу расписания, содержащую выходные дни с 9 до 23 часов:

```
hostname# firewall schedule-object add name @weekend weekly sa su at 09:00-23:00
```

- Чтобы создать группу сетевых интерфейсов, содержащую интерфейсы eth0 и eth1:

```
hostname# firewall interface-object add name @intgroup interface eth0 interface eth1
```

Просмотр групп объектов

Чтобы просмотреть пользовательские группы объектов и группы объектов, полученные из ViPNet Policy Manager или Policy Management (модуль ViPNet Prime), выполните одну из команд:

- для просмотра всех групп объектов:

```
hostname> firewall object show
```

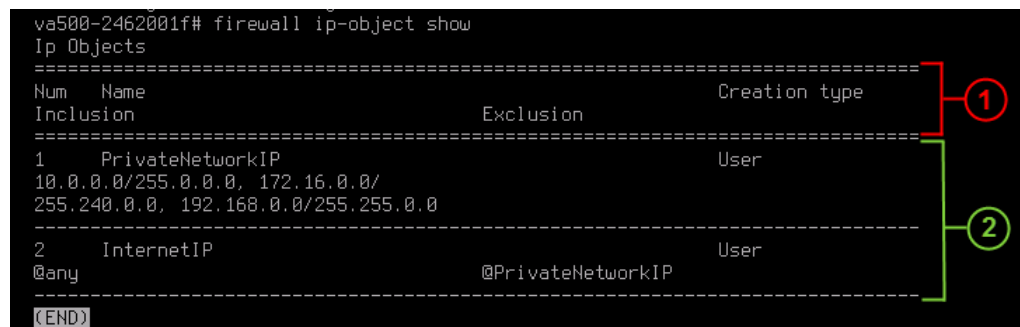
- для просмотра групп объектов определенного [типа](#):

```
hostname> firewall <тип объектов> show
```

Например, для просмотра всех групп IP-адресов:

```
hostname> firewall ip-object show
```

В результате будет выведена таблица, содержащая группы объектов.



```
va500-2462001f# firewall ip-object show
Ip Objects
=====
Num  Name                               Inclusion                               Exclusion                               Creation type
-----
1    PrivateNetworkIP                 10.0.0.0/255.0.0.0, 172.16.0.0/255.240.0.0, 192.168.0.0/255.255.0.0
2    InternetIP                        @any                                   @PrivateNetworkIP                      User
=====
(END)
```

Рисунок 22. Просмотр групп IP-адресов

Цифрами обозначены:

- 1 Заголовок таблицы, где:

- Num — порядковый номер группы объектов.
- Name — имя группы объектов.
- Creation type — вид группы объектов:
 - для групп из ViPNet Policy Manager или Policy Management (модуль ViPNet Prime) — Policy;
 - для пользовательских групп — User.
- Inclusion — состав группы объектов.
- Exclusion — исключения группы объектов.

- 2 Параметры групп объектов.

Удаление пользовательских групп объектов

Вы можете удалить пользовательскую группу объектов, если она не используется в других группах объектов, сетевых фильтрах или правилах трансляции сетевых адресов.

Для удаления пользовательской группы объектов выполните:

```
hostname# firewall object delete @<имя группы объектов>
```

При попытке удаления пользовательской группы объектов, используемой в другой группе объектов, сетевом фильтре или правиле трансляции сетевых адресов, появится сообщение об ошибке со списком всех групп, фильтров и правил, которые используют данную группу. Удалите эти группы, фильтры и правила, затем выполните команду для удаления группы еще раз.

Чтобы удалить группу сетевых узлов ViPNet с именем `range_<диапазон узлов>`, которая создается при конвертации правил защищенной сети во время обновления ViPNet Coordinator HW, сначала удалите все сетевые фильтры, в которых содержится эта группа.

Создание сетевого фильтра

Для каждого вида трафика создаются отдельные сетевые фильтры, которые хранятся в соответствующей таблице (см. [Просмотр сетевых фильтров](#)).

Чтобы создать сетевой фильтр:

```
hostname# firewall <тип> add [<номер>] [rule <имя>] src <адрес отправителя>  
dst <адрес получателя> [<протокол>] [<расписание>] <действие>
```

где:

- <тип> — тип фильтра:
 - local — локальный фильтр открытой сети;
 - forward — транзитный фильтр открытой сети;
 - vpn — фильтр защищенной сети;
 - tunnel — фильтр туннелируемых узлов;

Примечание. Если в ViPNet ЦУС или Prime запрещено использование туннелирования на узле ViPNet Coordinator HW, то на этом узле по умолчанию не будет фильтров туннелируемых узлов.



После получения разрешения на использование туннелирования создайте следующий фильтр:

```
hostname# firewall tunnel add src @any dst @any pass
```

-
- <номер> — порядковый номер сетевого фильтра в таблице, определяющий его приоритет. Чем меньше номер, тем выше приоритет фильтра (наиболее приоритетный фильтр имеет номер 1). Если заданный номер меньше максимального номера в таблице фильтров, номера фильтров, следующих за новым фильтром, автоматически увеличиваются на единицу (их приоритет понижается).
 - <имя> — имя сетевого фильтра. Если имя состоит из нескольких слов, разделенных пробелом, заключите его в кавычки (например, rule "number one").

Примечание.



- Не используйте в имени фильтра кириллицу. При срабатывании фильтра в системном журнале фиксируется событие. События, в описании которых есть символы на кириллице, не отображаются при [поиске по тексту сообщений](#).
 - При создании сетевого фильтра можно не указывать его номер. В этом случае фильтру будет присвоен номер, следующий за номером последнего фильтра в соответствующей таблице, и он будет иметь самый низкий приоритет.
 - Имя сетевого фильтра можно не указывать, в таком случае фильтр будет создан без имени.
-

- Параметры, определяющие процесс обработки IP-пакетов этим фильтром. Подробное описание этих параметров см. в соответствующих разделах:
 - [Адрес отправителя](#).
 - [Адрес получателя](#).
 - [Протокол](#).
 - [Расписание](#).
 - [Действие](#).



Внимание! Суммарное количество элементов для каждого правила в таблице сетевых фильтров не должно превышать 1000. Это ограничение контролируется программно, в случае превышения появится сообщение об ошибке и фильтр не применится.

В результате будет создан сетевой фильтр заданного типа. Вы можете просмотреть (см. [Просмотр сетевых фильтров](#)), изменить (см. [Изменение сетевого фильтра](#)) или удалить его (см. [Удаление сетевого фильтра](#)).



Примечание. Если при вводе команды допущена синтаксическая ошибка, то после выполнения такой команды появится сообщение с указанием места ошибки. Исправьте ошибку и выполните команду еще раз.

Примеры создания сетевых фильтров

- Чтобы разрешить отправку FTP-запросов со своего узла открытому узлу с адресом 192.168.2.4:

```
hostname# firewall local add src @local dst 192.168.2.4 service @FTP pass
```
- Чтобы создать локальный фильтр, блокирующий IP-пакеты, отправляемые узлом с адресом 192.168.30.1 через порт 2525 на порт 443 открытого узла с адресом 172.16.35.1 по протоколу TCP/IP:

```
hostname# firewall local add src 192.168.30.1 dst 172.16.35.1 tcp sport 2525 dport 443 drop
```
- Чтобы создать фильтр защищенной сети, разрешающий отправку IP-пакетов от защищенного узла с идентификатором 0x1a12000a на свой узел:

```
hostname# firewall vpn add src 0x1a12000a dst @local pass
```
- Чтобы создать фильтр, разрешающий отправку открытых транзитных IP-пакетов от узла с адресом 192.168.0.1 на узел с адресом 192.168.30.3 через координатор:

```
hostname# firewall forward add src 192.168.0.1 dst 192.168.30.3 pass
```
- Чтобы создать фильтр, разрешающий отправку IP-пакетов от туннелируемого узла с адресом 192.168.0.1 на защищенный узел с идентификатором 0x1234abab:

```
hostname# firewall tunnel add src 192.168.0.1 dst 0x1234abab pass
```
- Чтобы создать фильтр, разрешающий отправку IP-пакетов от туннелируемого узла с адресом 192.168.0.1 туннелируемому узлу с адресом 192.168.2.3:

```
hostname# firewall tunnel add src 192.168.0.1 dst 192.168.2.3 pass
```

Адрес отправителя

Адрес отправителя IP-пакетов является обязательным параметром сетевого фильтра и описывается лексемой:

```
src <адрес отправителя>
```

Возможные значения адреса отправителя:

- для локальных фильтров открытой сети:
 - IP-адрес или доменное имя узла;



Примечание. При задании адреса отправителя в виде доменного имени на DNS-сервер отправляется соответствующий запрос для разрешения данного имени. DNS-запись и время её жизни (TTL, «Time to live»), полученные в ответ на такой запрос, сохраняются в кеше. По истечении времени её жизни отправляется новый запрос на DNS-сервер для актуализации данных в кеше. Если полученные данные отличаются, происходит перезагрузка сетевых фильтров в драйвере ViPNet. Если разрешения указанного в сетевом фильтре доменного имени по каким-либо причинам не произошло, то такой фильтр не загружается в драйвер ViPNet до тех пор, пока в результате отправки повторных запросов на DNS-сервер не будет получен корректный ответ.

- диапазон IP-адресов узлов — два ограничивающие IP-адреса, разделенные дефисом. При этом значение второго IP-адреса (конца диапазона) должно быть больше первого (начала диапазона), например:

```
src 192.168.1.1-192.168.1.10
```
- маска подсети — IP-адрес подсети в формате классовой или бесклассовой адресации CIDR (Classless Inter-Domain Routing), например:

```
src 192.168.1.0/24
```

 или

```
src 192.168.1.0/255.255.255.0
```
- доменное имя узла, например:

```
src mydomain.ru
```



Примечание. При необходимости вы можете задать несколько IP-адресов и доменных имен отправителей IP-пакетов в одной лексеме, перечислив их через запятую, например:

```
src 192.168.30.2,192.169.1.1,mydomain.ru
```

- системная группа объектов `any` (любое значение), `local` (собственный узел) или `remote` (другие узлы) либо одна или несколько пользовательских групп IP-адресов. Имя группы необходимо дополнить символом «@», несколько групп указываются через запятую. Например:

```
src @IP_group_1,@IP_group_2
```



Внимание! Использование других адресов отправителей IP-пакетов совместно с системными группами объектов `any`, `local` и `remote` недопустимо.

- сетевой интерфейс собственного узла, через который будут проходить IP-пакеты, в следующем виде:

```
src interface {<имя интерфейса> | @<имя группы интерфейсов> | byip {<IP-адрес> | <диапазон IP-адресов> | <маска подсети>}}
```

- для транзитных фильтров открытой сети: то же, что для локальных фильтров открытой сети, но в таких фильтрах нельзя использовать системные группы объектов `local` и `remote`;
- для фильтров защищенной сети:
 - идентификатор узла сети ViPNet, например:

```
src 0x1a12000a
```

- системные группы объектов `any`, `allcoordinators`, `allclients`, `local` или `remote` либо пользовательские группы узлов ViPNet. Имя группы необходимо дополнить символом «@», несколько групп указываются через запятую. Например:

```
src @allclients,@allcoordinators
```

- идентификатор сети ViPNet, например:

```
src 0x1a12
```

Примечание. При задании адреса отправителя для фильтров защищенной сети:

- В одной лексеме можно указать несколько идентификаторов узлов или сетей ViPNet, пользовательских групп узлов ViPNet и системную группу `allcoordinators` или `allclients`, перечислив их через запятую. Например: `src @allcoordinators, 0x1a12000a`



- Нельзя указывать IP-адреса, доменные имена, маски адресов и сетевой интерфейс узла.
- Нельзя использовать другие адреса отправителей совместно с группами объектов `any`, `local` и `remote`.
- Фильтры защищенной сети используются только для локального защищенного трафика, поэтому если в качестве адреса получателя задано любое значение, кроме группы `local` или `broadcast`, то в качестве адреса отправителя может быть задана только группа `local`.

- для фильтров туннелируемых узлов: то же, что для локальных фильтров открытой сети и фильтров защищенной сети, но в таких фильтрах можно использовать только системные группы объектов `any`, `allcoordinators`, `allclients`, `tunneledip`.



Внимание! Если в фильтре туннелируемых узлов в качестве адреса отправителя указан адрес защищенного узла, то в качестве адреса получателя в этом фильтре должен быть указан адрес открытого узла, и наоборот.

Адрес получателя

Адрес получателя IP-пакетов является обязательным параметром сетевого фильтра. Синтаксис адреса получателя повторяет синтаксис [адреса отправителя](#), за исключением следующих особенностей:

- Адрес получателя описывается после адреса отправителя лексемой:
`dst <адрес получателя>`
- Для задания адреса получателя нельзя использовать сетевой интерфейс узла.
- Для задания адреса получателя, помимо пользовательских групп объектов, можно использовать следующие [системные группы объектов](#):
 - Для локальных фильтров открытой сети: `any`, `local`, `remote`, `broadcast`, `multicast`.



Внимание! Использование других адресов отправителей IP-пакетов совместно с системными группами объектов `any`, `local`, `remote`, `broadcast` и `multicast` недопустимо.

- Для фильтров защищенной сети: `any`, `allcoordinators`, `allclients`, `local`, `remote`, `broadcast`.



Примечание. Фильтры защищенной сети используются только для локального защищенного трафика, поэтому если в качестве адреса отправителя задано любое значение, кроме группы `local`, то в качестве адреса получателя может быть задана группа `local` или `broadcast`.

- Для фильтров туннелируемых узлов: `any`, `allcoordinators`, `allclients`, `tunneledip`.

Протокол

Протокол не является обязательным параметром сетевого фильтра.

Протоколы можно указывать, используя:

- Имена протоколов, написанные строчными буквами и разделенные пробелами, например:
`tcp udp icmp`
- В этом случае также можно задать дополнительные параметры для протоколов, например:
- для протоколов TCP и UDP: `sport` (порт или диапазон портов источника пакета) и (или) `dport` (порт или диапазон портов назначения пакета). При использовании обоих параметров сначала необходимо указать параметр `sport`, затем — параметр `dport`.
Например: `tcp sport 1024-65535 dport 1024;`

- о для протокола ICMP: только параметр `type` (тип пакета) либо параметр `type` вместе с параметром `code` (код пакета). Например: `icmp type 3 code 1`. Если параметр `code` не задан, то под условие будут подпадать все ICMP-пакеты указанного типа;
- номера протоколов. В этом случае для каждого протокола необходимо указать ключевое слово `proto` и его номер. Например, для протоколов TCP, UDP и ICMP: `proto 6 proto 17 proto 1`;



Внимание! При использовании номеров протоколов задать дополнительные параметры протоколов невозможно.

- пользовательские группы протоколов, в том числе заданные [по умолчанию](#), в виде:

`service @<имя группы>`

Например: `service @DNS`.

Расписание

Расписание задает время действия сетевого фильтра и не является его обязательным параметром. Описывается одной из следующих лексем:

- о `daily <чч:мм>-<чч:мм>` — фильтр действует ежедневно в течение заданного интервала времени. Время указывается в 24-часовом формате: `чч` — часы, `мм` — минуты.
- о `weekly [mo] [tu] [we] [th] [fr] [sa] [su] [at <чч:мм>-<чч:мм>]` — фильтр действует еженедельно в заданные дни недели:
 - `mo` — понедельник;
 - `tu` — вторник;
 - `we` — среда;
 - `th` — четверг;
 - `fr` — пятница;
 - `sa` — суббота;
 - `su` — воскресенье.
- о `calendar <дд.мм.гггг>-<дд.мм.гггг> [at <чч:мм>-<чч:мм>]` — фильтр действует в заданные даты и интервал времени. Дата указывается в следующем формате:
 - `дд` — день;
 - `мм` — месяц;
 - `гггг` — год.
- о `schedule @<имя группы объектов>` — фильтр действует по расписанию, описанному группой объектов соответствующего типа.

Для задания расписания можно использовать соответствующие пользовательские группы, в том числе заданные [по умолчанию](#).

Действие

Действие является обязательным параметром сетевого фильтра и определяет, как обрабатывается IP-пакет, соответствующий условию фильтра.

Действие описывается одной из следующих лексем:

- `pass` — пропускать IP-пакет;
- `drop` — блокировать IP-пакет.

Просмотр сетевых фильтров

Вы можете просмотреть сетевые фильтры, заданные на узле (см. [Общие сведения о сетевых фильтрах](#)). Для этого выполните одну из команд:

- для просмотра всех сетевых фильтров:

```
hostname> firewall rules show
```



Примечание. В результате выполнения команды `firewall rules show` отображаются не только сетевые фильтры, заданные на узле, но и правила трансляции сетевых адресов (см. [Просмотр, изменение, удаление правил трансляции сетевых адресов](#)).

- для просмотра сетевых фильтров определенного [типа](#):

```
hostname> firewall <тип> show
```

Например, для просмотра локальных фильтров открытой сети:

```
hostname> firewall local show
```

- для просмотра сетевых фильтров с конкретными параметрами:

```
hostname> firewall <тип> show <параметр>
```

указав в качестве параметра порядковый номер, имя, [адрес отправителя](#), [адрес получателя](#), [протокол](#) или [действие](#) сетевого фильтра. При необходимости вы можете указать несколько параметров.

Например, для просмотра локального фильтра открытой сети с несколькими адресами отправителей или одним адресом получателя:

```
hostname> firewall local show src 192.168.30.2,192.169.1.1,mydomain.ru dst 192.168.0.1
```

Внимание! Поиск сетевых фильтров для отображения выполняется по строгому совпадению с указанными параметрами. Например:



- если указан параметр `src 192.168.1.10`, отобразится фильтр, содержащий адрес `src 192.168.1.10`, но не отобразится фильтр, содержащий адрес `src 192.168.1.10-192.168.1.11`.
 - если указан параметр `rule Name`, отобразится фильтр с именем `Name`, но не отобразится фильтр с именем `Name1`.
-

В результате выполнения команды будет отображена таблица, содержащая те или иные сетевые фильтры. Пример таблицы локальных фильтров открытой сети представлен на рисунке ниже.

Local Rules:				
Num	Name	Option	Schedule	
Act	Protocol	Source	> Destination	
1			User	1
pass	@any	@any	> @any	
2	Allow DHCP Service		User	2
pass	udp:	@any	> @any	
	from 67			
	to 68			
3	Allow DHCP Service		User	
pass	udp:	@any	> @any	
	from 68			
	to 67			
4	Allow DHCP-Relay service		User	2
pass	udp:	@any	> @any	
	from 67			
	to 67			
5	ICMP redirect		User	
drop	icmp: 5	@any	> @any	

Рисунок 23. Просмотр локальных фильтров открытой сети

Цифрами на рисунке обозначены:

1 Заголовок таблицы, где:

- Num — порядковый номер сетевого фильтра, определяющий его приоритет.
- Name — имя фильтра.
- Option — категория сетевого фильтра: для фильтров из ViPNet Policy Manager или Policy Management (модуль ViPNet Prime) — Policy, для предустановленных фильтров и пользовательских фильтров — User, для фильтров системы защиты от сбоев — Failover.
- Schedule — расписание применения фильтра.
- Act — действие фильтра.
- Protocol — протокол.
- Source — адрес отправителя.
- Destination — адрес получателя.

2 Параметры сетевых фильтров, включающие:

- строку с порядковым номером, именем и категорией фильтра;
- одну или несколько строк, содержащих адрес отправителя, адрес получателя, протокол, расписание и действие фильтра.

Изменение сетевого фильтра

Вы можете изменить ранее созданный сетевой фильтр, добавив в него дополнительные значения параметров либо изменив его приоритет (порядковый номер в таблице сетевых фильтров соответствующего типа).

Чтобы добавить в сетевой фильтр адрес отправителя, адрес получателя, протокол или расписание:

1 Выполните команду:

```
hostname# firewall <тип> change append <номер> <параметры>
```

указав в качестве <параметры> [адрес отправителя](#), [адрес получателя](#), [протокол](#) или [расписание](#).

При необходимости вы можете указать несколько параметров.



Примечание. Если при вводе команды допущена синтаксическая ошибка, то после выполнения такой команды появится сообщение с указанием места ошибки. Исправьте ошибку и выполните команду еще раз.

2 Введите `y` для подтверждения команды. В результате новые значения параметров будут добавлены к уже имеющимся в сетевом фильтре.

Чтобы изменить приоритет сетевого фильтра:

```
hostname# firewall <тип> move rule <текущий номер> to <новый номер>
```

При этом номера фильтров, следующих за этим фильтром, автоматически увеличатся на единицу, то есть их приоритет понизится (см. [Просмотр сетевых фильтров](#)).

Пример добавления условия в сетевой фильтр

Для добавления в фильтр 1 RuleName1 src 192.168.1.1,2.2.2.2/24 dst @local drop **нового** адреса отправителя:

```
hostname# firewall local change append 1 src 192.168.3.3
```

Фильтр будет изменен следующим образом:

```
1 RuleName1 src 192.168.1.1,2.2.2.2/24,192.168.3.3 dst @local drop
```

Удаление сетевого фильтра

Чтобы удалить ненужный сетевой фильтр:

- Если вы знаете порядковый номер фильтра:

```
hostname# firewall <тип> delete <номер>
```

Например:

```
hostname# firewall local delete 7
```

- Если вы не знаете номер фильтра, то выполните поиск по известным вам параметрам:

```
hostname# firewall <тип> delete <параметр>,
```

где в качестве <параметр> можете указать имя, адрес отправителя, адрес получателя, протокол или действие. Вы можете указать несколько параметров. Например, для удаления локальных фильтров открытой сети с адресом отправителя 192.168.20.2:

```
hostname# firewall local delete src 192.168.20.2
```



Внимание! Поиск сетевых фильтров выполняется по строгому совпадению с указанными параметрами.

Будет выведен список подходящих сетевых фильтров:

- Если в результате поиска найдено несколько сетевых фильтров:
 - введите номер фильтра из столбца Num, чтобы удалить только его;
 - введите a, чтобы удалить все фильтры списка;
- Если найден один фильтр, подтвердите его удаление: введите y.

После удаления изменятся номера (приоритеты) фильтров. Номера фильтров с более низким приоритетом уменьшатся на единицу.

Блокирование сайтов по доменным именам с помощью сетевых фильтров

Для корректной работы запрещающих сетевых фильтров, в которых в качестве адресов отправителей или получателей заданы доменные имена:

- 1 Задайте для ViPNet Coordinator HW IP-адрес внешнего DNS-сервера (для примера используется IP-адрес 8.8.8.8):

```
hostname# inet dns forwarders add 8.8.8.8
```

- 2 Создайте разрешающие локальные фильтры для протокола DNS:

```
hostname# firewall local add 1 rule "Allow DNS 1" src 8.8.8.8 dst @local service @DNS pass
```

```
hostname# firewall local add 1 rule "Allow DNS 2" src @local dst 8.8.8.8 service @DNS pass
```

Эти фильтры разрешат обмен трафиком между ViPNet Coordinator HW и внешним DNS-сервером.

- 3 Создайте запрещающие сетевые фильтры для протокола DNS:

```
hostname# firewall forward add 1 rule "Deny DNS forward" src @any dst @any service @DNS drop
```

```
hostname# firewall tunnel add 2 rule "Deny DNS tunnel" src @any dst @any service @DNS drop
```

Эти фильтры запретят обмен трафиком между клиентами защищенной сети и любым DNS-сервером, кроме заданного для ViPNet Coordinator HW.

- 4 Задайте IP-адрес ViPNet Coordinator HW в качестве DNS-сервера для клиентов защищенной сети.

После этого вы можете создавать запрещающие сетевые фильтры для доменных имен сайтов, например:

```
hostname# firewall forward add src @any dst facebook.com drop
```

5

Настройка правил трансляции адресов

Трансляция сетевых адресов (NAT)	129
Создание правила трансляции адресов	132
Просмотр, изменение, удаление правил трансляции сетевых адресов	134
Примеры использования правил трансляции адресов	136

Трансляция сетевых адресов (NAT)

Трансляция сетевых адресов (NAT) — механизм преобразования IP-адресов одной сети в IP-адреса другой сети. Технология трансляции сетевых адресов описана в [RFC 2663](#) (IP Network Address Translator Terminology and Considerations).

ViPNet Coordinator HW может выполнять:

- Трансляцию адреса источника (маскарадинг (masquerading) или динамическую трансляцию).

В этом случае при прохождении через координатор IP-пакетов от отправителей с частными IP-адресами в них заменяется адрес отправителя на внешний (реальный) адрес координатора. При получении ответных пакетов в них подменяется IP-адрес получателя обратно на частный IP-адрес, и в таком виде пакет доставляется в частную сеть.

Используется для подключения локальной сети к интернету, когда количество узлов локальной сети превышает выданное поставщиком услуг интернета количество публичных IP-адресов. В результате локальные сети, использующие частные IP-адреса, получают доступ к ресурсам интернета.

- Трансляцию адреса назначения (перенаправление портов (port forwarding) или статическую трансляцию).

В этом случае IP-пакеты, приходящие из интернета на определенный порт внешнего адреса координатора, перенаправляются на указанный адрес внутренней сети путем подмены в них адреса получателя, а у ответных пакетов от узла внутренней сети подменяется адрес отправителя.

Используется для организации доступа к ресурсам локальной сети из интернета. В результате локальные сети, использующие частные адреса, могут быть доступны пользователям интернета по публичным IP-адресам.

- Одновременную трансляцию сетевых адресов источника и назначения.

Используется в сложных схемах маршрутизации трафика.

Трансляция сетевых адресов выполняется координатором, только если настроены соответствующие правила (см. [Создание правила трансляции адресов](#)). Координатор должен иметь как минимум два [сетевых интерфейса](#):

- внешний — имеет публичный IP-адрес и обеспечивает доступ в интернет;
- внутренний — имеет частный IP-адрес.



Внимание! Правила трансляции сетевых адресов, описанные в данном разделе, относятся только к открытому трафику. Для защищенного трафика действуют автоматические механизмы трансляции сетевых адресов, параметры которых не могут быть изменены.

Трансляция адреса назначения

Трансляция адреса узла назначения применяется для организации доступа из интернета к серверам локальной сети, не имеющим публичного IP-адреса. Правило трансляции адреса назначения ставит в соответствие частным IP-адресам локальных сетевых узлов публичный IP-адрес координатора. Таким образом, по публичному IP-адресу внешние пользователи могут получить доступ к ресурсам локальной сети.

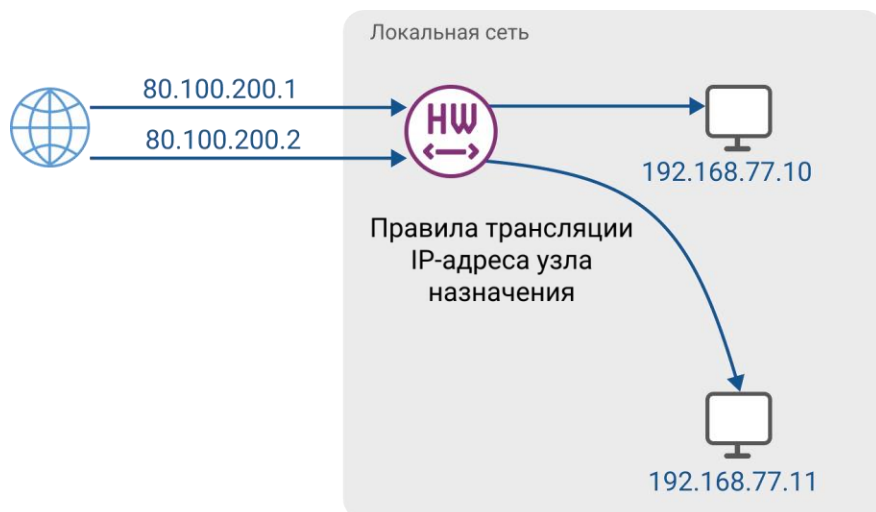


Рисунок 24. Доступ к внутренним ресурсам при помощи правил трансляции IP-адресов узлов назначения

Если для публичного IP-адреса координатора задано правило трансляции адреса назначения, то при обращении к этому IP-адресу из интернета будут выполняться преобразования:

- Во входящих IP-пакетах от внешнего узла координатор подменяет адрес получателя (публичный IP-адрес координатора) локальным IP-адресом в соответствии с правилом. Затем пакет передается через внутренний сетевой интерфейс на узел локальной сети, которому адресован пакет.
- При прохождении ответных IP-пакетов (в рамках уже созданной сессии) координатор производит обратную замену IP-адресов. Адрес отправителя (IP-адрес локального узла) подменяется публичным IP-адресом внешнего сетевого интерфейса координатора. Затем ответный пакет отправляется узлу в интернете.

Таким образом, при передаче в интернете IP-пакет выглядит так, будто отправитель и получатель этого пакета имеют публичные IP-адреса.



Внимание! При трансляции адреса узла назначения инициировать соединение может только внешний сетевой узел. Чтобы локальный сетевой узел мог также иметь доступ в интернет (двусторонний NAT), необходимо в дополнение к правилу трансляции адреса узла назначения задать также правило трансляции адреса источника (см. [Трансляция адреса источника](#)).

Трансляция адреса источника

Трансляция адреса источника предназначена для организации доступа узлов локальной сети в интернет. Правило трансляции адреса источника ставит в соответствие нескольким частным IP-адресам локальных сетевых узлов публичный IP-адрес координатора. Таким образом, узлы локальной сети могут устанавливать соединения с узлами в интернете от имени публичного IP-адреса координатора.



Рисунок 25. Организация доступа в интернет при помощи правила трансляции IP-адреса источника

Если на координаторе настроено правило трансляции адреса источника, то транзитные IP-пакеты, проходящие через координатор из локальной сети в интернет, будут преобразованы:

- В момент передачи IP-пакета из локальной сети в интернет координатор преобразует IP-адрес (или IP-адрес и порт) отправителя пакета для протоколов TCP и UDP. Для пакетов протокола ICMP преобразуется адрес отправителя, остальные параметры запоминаются. При этом частный IP-адрес отправителя заменяется на публичный IP-адрес внешнего сетевого интерфейса координатора, обеспечивающего доступ во внешнюю сеть. При дальнейшей передаче в интернете пакет имеет публичный IP-адрес отправителя. Номера портов отправителя (для протоколов TCP и UDP) и запоминаемые параметры (для протокола ICMP) пакетов имеют уникальные значения для всех исходящих IP-соединений внешнего сетевого интерфейса координатора. После преобразования пакет отправляется адресату в интернете.
- При прохождении ответных IP-пакетов ViPNet Coordinator HW производит обратное преобразование указанных параметров. То есть в момент передачи ответного IP-пакета координатор заменяет в нем IP-адрес получателя на частный IP-адрес узла локальной сети, которому адресован ответный пакет. Преобразование происходит на основании уникальных номеров портов, присвоенных исходящим пакетам (для протоколов TCP и UDP), и запоминаемых параметров исходящих пакетов (для протокола ICMP). Номера портов (для протоколов TCP и UDP) также преобразуются в свои истинные значения. Затем ответные пакеты передаются через внутренний сетевой интерфейс узлу локальной сети, которому адресован пакет.



Примечание. Для всех протоколов, кроме TCP, UDP и ICMP, преобразуются только IP-адреса. Для протоколов с частичным преобразованием трансляция адреса источника не будет работать, если несколько узлов локальной сети одновременно инициируют соединение с одним и тем же IP-адресом публичной сети.

Создание правила трансляции адресов

Правила трансляции адресов хранятся в соответствующей таблице (см. [Просмотр, изменение, удаление правил трансляции сетевых адресов](#)).



Примечание. Правила трансляции IP-адресов работают только при наличии транзитных фильтров [открытой сети](#), пропускающих трафик от узлов, к которым будет применяться трансляция адресов.

Чтобы создать правило трансляции адресов, выполните команду:

```
hostname# firewall nat add [<номер>] [rule <имя>] src <адрес отправителя>  
dst <адрес получателя> [<протокол>] [<расписание>] change {src <адрес отправителя> | dst  
<адрес получателя>}, указав следующие параметры:
```

- Порядковый номер правила в таблице, определяющий его приоритет. Чем меньше порядковый номер правила, тем выше его приоритет (наиболее приоритетное правило имеет номер 1). При создании правила с порядковым номером, меньшим максимального номера в таблице правил, номера правил, следующих за этим правилом, автоматически увеличиваются на единицу (то есть их приоритет понижается).
- Имя правила. Если имя состоит из нескольких слов, разделенных пробелом, заключите его в кавычки (например, rule "number one").



Примечание. При создании правила трансляции адресов можно не указывать его номер. В этом случае правилу будет присвоен номер, следующий за номером последнего правила в таблице, и оно будет иметь самый низкий приоритет.

Имя правила можно не указывать, правило будет создано без имени.

- Параметры, определяющие процесс обработки трафика правилом:
 - [адрес отправителя](#) и [адрес получателя](#) являются обязательными параметрами правила и задаются в виде IP-адреса или доменного имени узла, диапазона IP-адресов узлов, списка IP-адресов или доменных имен узлов, маски адресов подсети, доменного имени сети, одной или нескольких пользовательских групп IP-узлов;
 - [протокол](#) и [расписание](#) являются необязательными параметрами правила и задаются так же, как для сетевых фильтров;
 - действие правила в виде одной из лексем:
 - для трансляции адреса источника: change src {<адрес> | auto},
указав внешний адрес координатора, на который будет заменяться адрес отправителя пакетов либо параметр auto, чтобы адрес отправителя автоматически заменялся на публичный адрес внешнего сетевого интерфейса координатора.

- для трансляции адреса и/или порта назначения: `change dst <адрес>:[<порт>]`, указав адрес и/или порт узла локальной сети, которому будут перенаправляться пакеты.



Примечание. Если при вводе команды допущена синтаксическая ошибка, то после выполнения такой команды появится сообщение с указанием места ошибки. Исправьте ошибку и выполните команду еще раз.

В результате будет создано правило трансляции адресов. Вы можете просмотреть, изменить или удалить его (см. [Просмотр, изменение, удаление правил трансляции сетевых адресов](#)).



Примечание. Созданное правило трансляции сетевых адресов будет работать для новых соединений. Для уже установленных соединений правило будет работать после истечения времени жизни соединения. Чтобы принудительно применить правило трансляции сетевых адресов для всех соединений, выполните:

```
hostname# vpn stop
hostname# vpn start
```

Примеры создания правил трансляции адресов

- Чтобы при отправке пакета узлом с адресом 10.0.0.1 внешнему узлу с адресом 192.168.20.1 частный адрес отправителя пакета заменялся на публичный адрес внешнего сетевого интерфейса координатора, создайте правило трансляции адреса источника:

```
hostname# firewall nat add src 10.0.0.1 dst 192.168.20.1 change src auto
```

- Чтобы при отправке пакета внешним узлом с адресом mydomain.ru узлу с адресом 192.168.20.1 координатор подменял адрес получателя (публичный IP-адрес координатора) на локальный адрес, создайте правило трансляции адреса назначения:

```
hostname# firewall nat add src mydomain.ru dst 192.168.20.1 change dst 10.0.0.7
```

- Чтобы при отправке пакета внешним узлом с адресом mydomain.ru узлу с адресом 192.168.20.1 по протоколу TCP/IP через порт 8080 координатор подменял адрес получателя (публичный IP-адрес координатора) на локальный адрес, создайте правило трансляции адреса назначения:

```
hostname# firewall nat add src mydomain.ru dst 192.168.20.1 tcp dport 8080 change dst 10.0.0.7:8080
```

Просмотр, изменение, удаление правил трансляции сетевых адресов

Вы можете просмотреть, изменить или удалить правила трансляции сетевых адресов, заданные на ViPNet Coordinator HW.

Чтобы просмотреть правила, выполните одну из команд:

- для просмотра всех правил:

```
hostname> firewall nat show
```

- для просмотра правил с конкретными параметрами:

```
hostname> firewall nat show <параметр>,
```

указав в качестве параметра порядковый номер, имя, адрес отправителя, адрес получателя или протокол (см. [Создание правила трансляции адресов](#)). При необходимости вы можете указать несколько параметров.



Примечание. Поиск правил трансляции сетевых адресов выполняется по строгому совпадению с указанными параметрами.

В результате выполнения команды будет отображена таблица, содержащая правила трансляции сетевых адресов.

```
User:
=====
Num  Name      Option      Schedule
Act  Protocol  Source      > Destination
=====
1    NAT      @any        172.20.0.77  User
      Change: auto      > @any
=====
```

Рисунок 26. Просмотр правил трансляции сетевых адресов

Таблица содержит следующие столбцы:

- Num — порядковый номер правила, определяющий его приоритет.
- Name — имя правила.
- Option — категория правила: для правил из ViPNet Policy Manager или Policy Management (модуль ViPNet Prime) — Policy, для правил пользователя — User.
- Schedule — расписание применения правила.
- Act — действие правила (NAT).

- `Source` — адрес источника до или после трансляции.
- `Destination` — адрес назначения до или после трансляции.
- `Protocol` — протокол.



Внимание! Если из ViPNet Policy Manager или Policy Management (модуль ViPNet Prime) поступили правила трансляции сетевых адресов, в которых указан порт, но не указан протокол, они будут применены для протоколов TCP и UDP. Чтобы не открывать лишние порты, укажите протоколы для этих правил в ViPNet Policy Manager или Policy Management (модуль ViPNet Prime) и отправьте политики еще раз.

Изменение и удаление правил трансляции сетевых адресов выполняется аналогично изменению и удалению сетевых фильтров:

```
hostname# firewall nat change append <номер> <параметры>
```

```
hostname# firewall nat delete <параметры>
```

Подробнее см. в разделах [Изменение сетевого фильтра](#) и [Удаление сетевого фильтра](#).

Примеры использования правил трансляции адресов

Организация доступа пользователей к интернету

С помощью ViPNet Coordinator HW можно организовать доступ пользователей, имеющих частные адреса, к интернету. Такая задача обычно возникает, когда число выделенных организации публичных адресов меньше числа компьютеров локальной сети, которым необходим доступ к интернету. Для решения этой задачи используется [трансляция адресов источников](#), которая заключается в преобразовании частных адресов локальной сети в один публичный адрес.

На рисунке ниже представлена схема организации доступа пользователей локальной сети к интернету с помощью ViPNet Coordinator HW.

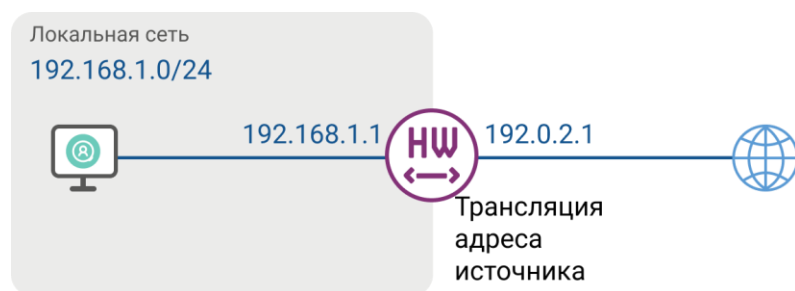


Рисунок 27. Организация доступа пользователей к интернету с помощью ViPNet Coordinator HW

На приведенной схеме локальная сеть имеет адресное пространство 192.168.1.0/24. Один интерфейс ViPNet Coordinator HW подключен к локальной сети и имеет частный адрес 192.168.1.1, другой интерфейс подключен к интернету и имеет публичный адрес 192.0.2.1.

Чтобы пользователи могли подключаться к интернету, на ViPNet Coordinator HW необходимо задать правило трансляции IP-адресов и сетевой фильтр транзитных IP-пакетов:

- 1 Создайте правило трансляции частных IP-адресов отправителей в IP-адрес внешнего сетевого интерфейса ViPNet Coordinator HW:

```
hostname# firewall nat add src 192.168.1.0/24 dst @InternetIP change src 192.0.2.1
```
- 2 Создайте один из следующих разрешающих транзитных фильтров открытой сети:
 - о чтобы пользователи могли работать в интернете с использованием любого протокола:

```
hostname# firewall forward add src 192.168.1.0/24 dst @InternetIP pass
```
 - о чтобы пользователи могли работать в интернете только по протоколу HTTP:

```
hostname# firewall forward add src 192.168.1.0/24 dst @InternetIP tcp dport 80 pass
```

В результате пользователи локальной сети будут иметь доступ к интернету.

Размещение общедоступного сервера в демилитаризованной зоне

В локальной сети наиболее уязвимыми для сетевых атак являются узлы (серверы), которые взаимодействуют с внешними системами или предоставляют различные сервисы внешним пользователям (почтовые, веб-серверы и так далее). В случае атаки на такие серверы под угрозой оказываются остальные компьютеры сети. Чтобы защитить локальную сеть, ее разбивают на сегменты, размещая серверы в демилитаризованной зоне (ДМЗ). Эту зону отделяют от остальной сети **межсетевым экраном**, который контролирует трафик между сегментами. При этом сервисы, предоставляемые серверами, доступны как внешним пользователям, так и пользователям локальной сети. В то же время локальная сеть недоступна внешним пользователям.

В качестве межсетевого экрана, отделяющего демилитаризованную зону от внутреннего сегмента сети, можно использовать ViPNet Coordinator HW. На рисунке ниже приведена схема сети с веб-сервером, размещенным в демилитаризованной зоне, и ViPNet Coordinator HW с тремя интерфейсами, установленным на границе локальной сети.

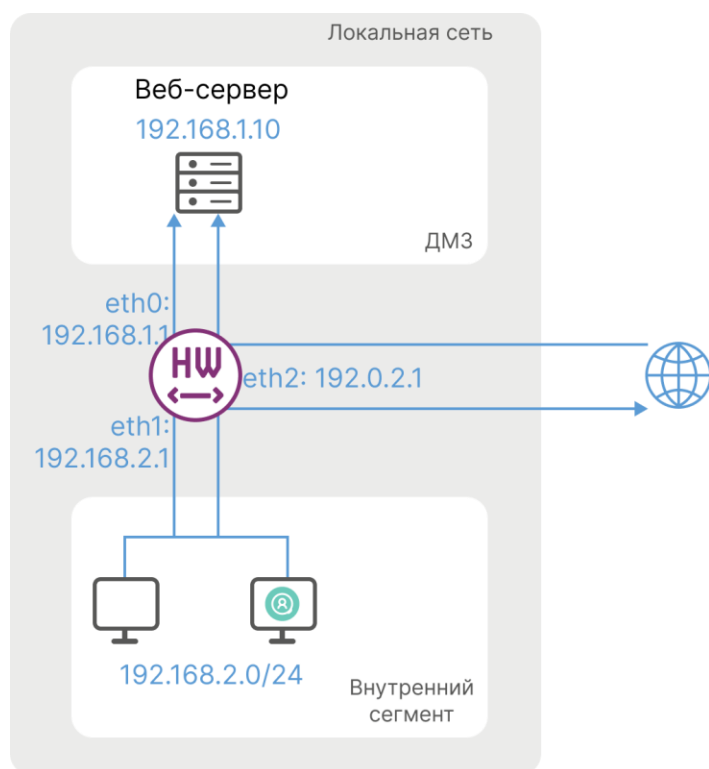


Рисунок 28. Использование ViPNet Coordinator HW для размещения общедоступного сервера в ДМЗ

Два интерфейса ViPNet Coordinator HW имеют частные IP-адреса и подключены к сегментам локальной сети, интерфейс `eth2` имеет публичный IP-адрес и подключен к интернету. Пусть заданы следующие сетевые настройки:

- компьютеры во внутреннем сегменте имеют адреса 192.168.2.0/24;
- веб-сервер имеет адрес 192.168.1.10;
- интерфейсы `eth0`, `eth1`, `eth2` имеют адреса 192.168.1.1, 192.168.2.1, 192.0.2.1 соответственно.

Предполагается, что в приведенной схеме допустимы инициативные соединения по следующим направлениям:

- из внутреннего сегмента в интернет;
- из внутреннего сегмента к веб-серверу, находящемуся в демилитаризованной зоне (ДМЗ);
- из интернета к веб-серверу.

Чтобы установление таких соединений было возможно, на ViPNet Coordinator HW необходимо создать сетевые фильтры транзитных IP-пакетов и правила трансляции IP-адресов:

1 Убедитесь, что TCP-туннель выключен:

```
hostname# iplir show tcptunnel-info
```

В результате выполнения команды будет указано, включен или выключен TCP-туннель. Если TCP-туннель включен, отключите его:

1.1 Завершите работу управляющей службы:

```
hostname# iplir stop
```

1.2 Откройте файл `iplir.conf` для редактирования:

```
hostname# iplir config
```

1.3 В секции `[misc]` параметру `tcptunnel_establish` присвойте значение `off`.

1.4 Сохраните изменения (**Ctrl+O**) и закройте редактор (**Ctrl+X**).

1.5 Запустите управляющую службу:

```
hostname# iplir start
```

2 Создайте транзитные фильтры открытой сети:

- фильтр, разрешающий соединения из внутреннего сегмента с любыми IP-адресами:

```
hostname# firewall forward add src 192.168.2.0/24 dst @any pass
```

- фильтр, разрешающий соединения с веб-сервером по протоколу HTTP с любых IP-адресов:

```
hostname# firewall forward add src @any dst 192.168.1.10 tcp dport 80 pass
```

Остальной транзитный трафик будет блокироваться сетевым фильтром, настроенным по умолчанию.

3 Создайте правила трансляции сетевых адресов:

- правило для организации доступа в интернет из внутреннего сегмента:

```
hostname# firewall nat add src 192.168.2.0/24 dst @InternetIP change src 192.0.2.1
```

- правило для организации доступа к веб-серверу из интернета:

```
hostname# firewall nat add src @InternetIP dst 192.0.2.1 tcp dport 80 change dst 192.168.1.10
```

В результате пользователи локальной сети и внешние пользователи смогут обращаться к веб-серверу по адресу 192.168.1.10. Пользователи локальной сети также будут иметь доступ в интернет, при этом их компьютеры будут защищены от атак из интернета.

Настройка доступа к узлам локальной сети с помощью дополнительных IP-адресов

Вы можете настроить доступ к сетевым узлам локальной сети, например к почтовому и веб-серверу, из интернета с помощью дополнительных IP-адресов на внешнем интерфейсе HW и правил статической трансляции адресов.

Чтобы настроить доступ:

- 1 На внешнем интерфейсе eth0 задайте дополнительные IP-адреса:

```
hostname# inet ifconfig eth0 address add 89.175.26.2 netmask 255.255.255.0
hostname# inet ifconfig eth0 address add 89.175.26.3 netmask 255.255.255.0
```

- 2 Создайте правила трансляции адреса получателя:

```
hostname# firewall nat add src @InternetIP dst 89.175.26.2 tcp dport 25 change dst 192.168.1.2:25
hostname# firewall nat add src @InternetIP dst 89.175.26.3 tcp dport 80 change dst 192.168.1.3:80
```

- 3 Создайте разрешающие сетевые фильтры:

```
hostname# firewall forward add src @InternetIP dst 192.168.1.2 tcp dport 25 pass

hostname# firewall forward add src @InternetIP dst 192.168.1.3 tcp dport 80 pass
```

Теперь почтовый сервер будет доступен из интернета по адресу 89.175.26.2:25, а веб-сервер — по адресу 89.175.26.3:80.

Обеспечение доступа удаленных клиентов к серверу по технологии PPTP VPN

ViPNet Coordinator HW позволяет обеспечить доступ удаленных клиентов по технологии PPTP VPN к серверу, установленному в локальной сети.

Пусть в локальной сети в качестве межсетевого экрана используется ViPNet Coordinator HW, за которым установлен PPTP-сервер. Требуется, чтобы с этим сервером могли взаимодействовать PPTP-клиенты, находящиеся вне сети. На рисунке ниже приведена схема взаимодействия PPTP-клиентов с PPTP-сервером через ViPNet Coordinator HW:

- интерфейс `eth0` ViPNet Coordinator HW подключен к локальной сети и имеет частный адрес 192.168.2.1;
- интерфейс `eth1` ViPNet Coordinator HW подключен к интернету и имеет публичный адрес 192.0.2.1;
- PPTP-сервер имеет в локальной сети адрес 192.168.2.10.

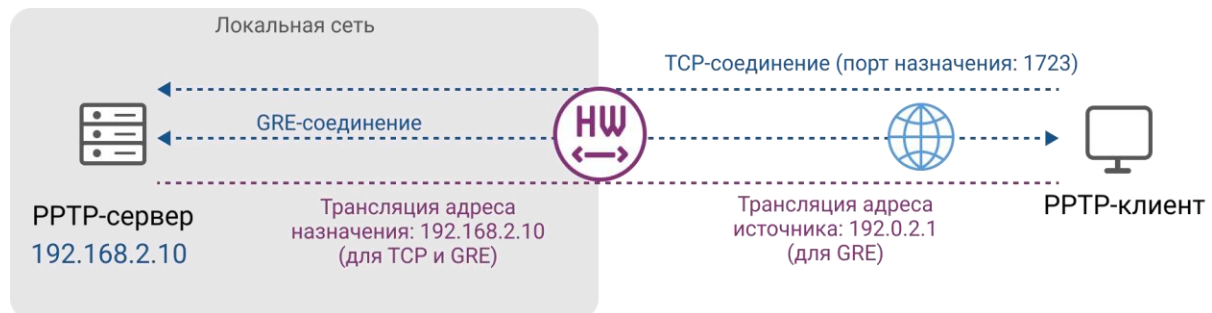


Рисунок 29. Схема взаимодействия PPTP-клиентов с PPTP-сервером через ViPNet Coordinator HW

В технологии PPTP VPN одновременно используются два соединения — TCP- и GRE-соединение. Когда PPTP-клиент обращается к PPTP-серверу, клиент инициирует TCP-соединение с портом назначения 1723. После этого PPTP-сервер инициирует соединение по IP-протоколу GRE, который инкапсулирует пакеты протокола PPTP в IP-пакеты. Через GRE-соединение происходит пересылка данных, а TCP-соединение используется как управляющий канал для поддержки GRE-соединения.

Таким образом, чтобы удаленные PPTP-клиенты могли взаимодействовать с PPTP-сервером, на ViPNet Coordinator HW необходимо задать правила трансляции адресов для протоколов TCP и GRE и транзитные фильтры открытой сети:

1 Добавьте правила трансляции адресов:

```
hostname# firewall nat add 1 src 192.168.2.10 dst @InternetIP service @GRE change src 192.0.2.1
hostname# firewall nat add 2 src @InternetIP dst 192.0.2.1 tcp dport 1723 change dst 192.168.2.10
hostname# firewall nat add 3 src @InternetIP dst 192.0.2.1 service @GRE change dst 192.168.2.10
```

2 Добавьте транзитные фильтры открытой сети:

```
hostname# firewall forward add 1 src 192.168.2.10 dst @InternetIP service @GRE pass
hostname# firewall forward add 2 src @InternetIP dst 192.168.2.10 tcp dport 1723 pass
hostname# firewall forward add 3 src @InternetIP dst 192.168.2.10 service @GRE pass
```


6

Тонкая настройка межсетевого экрана

Обработка фрагментированных пакетов	142
Настройка анτισпуфинга	144
Настройка дополнительных параметров межсетевого экрана	146

Обработка фрагментированных пакетов

Сетевые протоколы содержат механизмы управления размером IP-пакетов. ViPNet Coordinator HW при шифровании трафика добавляет к каждому IP-пакету служебный заголовок (выполняет инкапсуляцию), вследствие чего размер сформированного VPN-пакета может получиться больше заданного в MTU и потребует фрагментации. В свою очередь, некоторые сервисы не позволяют фрагментировать свои пакеты (например, видео-трафик), используя флаг Don't fragment (DF). В любом случае, обработка фрагментированных пакетов снижает пропускную способность сети (так как два пакета вместо одного суммарно длиннее, и их обработка требует больше ресурсов), поэтому в ViPNet Coordinator HW предусмотрены средства решения этой проблемы.

Установка TCP-соединения

На этапе установки соединения по протоколу TCP сервер и клиент обмениваются значениями MSS (максимальная длина сегмента), которые основаны на MTU их сетевых интерфейсов, и в итоге принимается минимальное из двух значений. В результате инкапсуляции IP-пакета ViPNet Coordinator HW учитывает размер, на который был увеличен IP-пакет после инкапсуляции, и уменьшает значение MSS с учетом этих данных.

Также вы можете настроить дополнительное уменьшение размера MSS с помощью параметра `mssdecrease` конфигурационного файла `iplir.conf` (подробнее см. в документе «Справочник команд и конфигурационных файлов», раздел «Файл `iplir.conf`»).

Обработка IP-пакетов с флагом Don't fragment (DF)

Рассмотрим случай, когда ViPNet Coordinator HW туннелирует сетевые узлы, которые отправляют через него IP-пакеты с размером, не превышающим размер MTU, заданный на внешнем сетевом интерфейсе ViPNet Coordinator HW. При шифровании трафика размер IP-пакета может превысить MTU, и в этом случае IP-пакет будет фрагментирован.

Если же такой IP-пакет имеет флаг DF, запрещающий фрагментацию, то, в зависимости от его размера, будет выполнен один из сценариев:

- Если размер IP-пакета после инкапсуляции не превышает MTU, то ViPNet Coordinator HW его пропустит и сбросит флаг DF (который не восстановится при последующей декапсуляции).
- Если размер IP-пакета после инкапсуляции превышает MTU, то ViPNet Coordinator HW его отбросит (при этом в журнале IP-пакетов появится запись о блокировке с кодом события 97) и отправит ICMP-сообщение о необходимости фрагментации IP-пакета (ICMP type 3, code 4) его отправителю. В зависимости от настройки отправителя IP-пакетов необходимо выполнить одно из действий:
 - Если отправитель IP-пакетов с флагом DF не получает (не обрабатывает) такие ICMP-сообщения, то ему необходимо самостоятельно уменьшить размер IP-пакетов.

- Если отправитель IP-пакетов с флагом DF принимает и корректно обрабатывает ICMP-сообщения ICMP type 3, code 4 (в соответствии с [RFC 4459](#)), то такой трафик будет пропущен.

Блокировка входящих фрагментированных пакетов

Фрагментированные IP-пакеты используются как средство сетевой атаки «отказ в обслуживании» (DDoS). В качестве меры защиты от такой атаки используется опция межсетевого экрана `block-fragmented-packets`, которая блокирует все входящие фрагментированные IP-пакеты на всех сетевых интерфейсах. Подробнее см. в разделе [Настройка дополнительных параметров межсетевого экрана](#).

Настройка антиспуфинга

В ViPNet Coordinator HW реализована функция антиспуфинга, то есть блокирования входящих IP-пакетов от отправителей, IP-адреса которых недопустимы на данном сетевом интерфейсе узла. Антиспуфинг работает как для открытого, так и для защищенного трафика. При этом открытые IP-пакеты, полученные в результате декапсуляции защищенных IP-пакетов, не проверяются антиспуфингом, так как уже не относятся к сети отправителя. Эти IP-пакеты, как и обычные открытые IP-пакеты, после проверки антиспуфингом обрабатываются сетевыми фильтрами (см. [Общие сведения о сетевых фильтрах](#)).

Основная задача антиспуфинга — защита от сетевых атак, называемых спуфингом. При спуфинге злоумышленник посылает на какой-либо компьютер IP-пакет, в котором в качестве адреса отправителя указан не адрес злоумышленника, а адрес какого-либо другого узла, которому разрешено соединение с данным координатором. Например, таким образом можно отправить открытый пакет из интернета через координатор, задав в качестве адреса отправителя адрес частной внутренней сети, которая также подключена к данному координатору. Правила антиспуфинга позволяют исключить такую возможность.

По умолчанию антиспуфинг в ViPNet Coordinator HW выключен. Для повышения уровня безопасности сети включите его:

- 1 Проверьте текущее состояние антиспуфинга:

```
hostname# iplir option get antispoofing
```

- 2 Включите антиспуфинг в ViPNet Coordinator HW:

```
hostname# iplir option set antispoofing on
```

При включении антиспуфинга на сетевом узле на основе таблицы маршрутизации автоматически задаются правила антиспуфинга.



Внимание! При использовании сложных схем маршрутизации на узле (с метриками маршрутов или асимметричными маршрутами) функция антиспуфинга может работать некорректно.

Правила антиспуфинга применяются для открытого транзитного трафика следующим образом:

- Для всех сетевых интерфейсов, кроме интерфейса, соответствующего маршруту по умолчанию, блокируются IP-пакеты, адреса отправителей которых не совпадают с адресами, маршрутизируемыми через данный интерфейс.
- Для сетевого интерфейса, соответствующего маршруту по умолчанию, блокируются IP-пакеты, адреса отправителей которых совпадают с зарегистрированными маршрутами других интерфейсов.

Пример настройки антиспуфинга

Пусть на сетевом узле используется следующая таблица маршрутизации.

Таблица 3. Пример таблицы маршрутизации на узле

Сетевой адрес	Маска подсети	Адрес шлюза	Сетевой интерфейс	Метрика
0.0.0.0	0.0.0.0	10.0.8.1	10.0.8.54	20
10.0.8.0	255.255.255.0	On-link	10.0.8.54	276
10.0.8.54	255.255.255.255	On-link	10.0.8.54	276
10.0.8.255	255.255.255.255	On-link	10.0.8.54	276
127.0.0.0	255.0.0.0	On-link	127.0.0.1	306
127.0.0.1	255.255.255.255	On-link	127.0.0.1	306
127.255.255.255	255.255.255.255	On-link	127.0.0.1	306
192.168.48.0	255.255.255.0	On-link	192.168.48.1	276
192.168.48.1	255.255.255.255	On-link	192.168.48.1	276
192.168.48.255	255.255.255.255	On-link	192.168.48.1	276
192.168.59.0	255.255.255.0	On-link	192.168.59.1	276
192.168.59.1	255.255.255.255	On-link	192.168.59.1	276
192.168.59.255	255.255.255.255	On-link	192.168.59.1	276
224.0.0.0	224.0.0.0	On-link	127.0.0.1	306
224.0.0.0	224.0.0.0	On-link	10.0.8.54	276
224.0.0.0	224.0.0.0	On-link	192.168.48.1	276
224.0.0.0	224.0.0.0	On-link	192.168.59.1	276
255.255.255.255	255.255.255.255	On-link	127.0.0.1	306
255.255.255.255	255.255.255.255	On-link	10.0.8.54	276
255.255.255.255	255.255.255.255	On-link	192.168.48.1	276
255.255.255.255	255.255.255.255	On-link	192.168.59.1	276

Рассматриваемый узел имеет четыре сетевых интерфейса, один из них с адресом 127.0.0.1 является сетевым интерфейсом «внутренней петли» (loopback).

При включении антиспуфинга на основе приведенной таблицы маршрутизации на узле сформируются следующие правила антиспуфинга:

- На сетевом интерфейсе с адресом 192.168.48.1 разрешены входящие пакеты только от узлов с IP-адресами 192.168.48.0/24.
- На сетевом интерфейсе с адресом 192.168.59.1 разрешены входящие пакеты только от узлов с IP-адресами 192.168.59.0/24.
- На сетевом интерфейсе с адресом 10.0.8.54 разрешены все входящие пакеты, кроме пакетов от узла с IP-адресами 192.168.48.0/24, 192.168.59.0/24.

Настройка дополнительных параметров межсетевого экрана

Чтобы посмотреть текущие настройки дополнительных параметров межсетевого экрана выполните:

```
hostname# iplir option get <параметр>
```

указав один из следующих параметров:

- `antispoofing` — включение или выключение антиспуфинга. Возможные значения:
 - `off` (по умолчанию) — антиспуфинг выключен.
 - `on` — антиспуфинг включен.
- `block-fragmented-packets` — включение или выключение блокирования входящих фрагментированных IP-пакетов, принимаемых по всем сетевым интерфейсам. Возможные значения:
 - `off` (по умолчанию) — фрагментированные пакеты пропускаются.
 - `on` — фрагментированные пакеты блокируются, в журнал регистрации IP-пакетов вносится соответствующая запись.
- `connection-ttl-ip` — время жизни соединений по протоколу IP при отсутствии активности в нем. Указывается в секундах, по умолчанию — 300 (5 минут).
- `connection-ttl-tcp` — время жизни соединений по протоколу TCP при отсутствии активности в нем. Указывается в секундах, по умолчанию — 1800 (30 минут).
- `connection-ttl-udp` — время жизни соединений по протоколу UDP при отсутствии активности в нем. Указывается в секундах, по умолчанию — 300 (5 минут).
- `max-connections` — максимальное количество параллельно установленных соединений. Значение данного параметра зависит от исполнения ViPNet Coordinator HW.

Таблица 4. Значения параметра `max-connections` для аппаратных платформ ViPNet Coordinator HW

Аппаратная платформа ViPNet Coordinator HW	Объем оперативной памяти	Значение по умолчанию	Максимальное значение
HW10 F1	1,8 Гбайт	150 000	150 000
HW50 N1, N2, N3, N4, A1	1,8 Гбайт	150 000	150 000
HW100 N1, N2, N3, Q1, Q2	2 Гбайт	150 000	150 000
HW1000 Q4, Q5, Q6	2 Гбайт	800 000	1 000 000
HW1000 Q7, Q8, Q9	4–16 Гбайт	800 000	1 000 000

Аппаратная платформа ViPNet Coordinator HW	Объем оперативной памяти	Значение по умолчанию	Максимальное значение
HW 1000 Q10	8 Гбайт	800 000	1 000 000
HW2000 Q4	4 Гбайт	2 500 000	3 000 000
HW2000 Q5	32 Гбайт	6 000 000	6 500 000
HW5000 Q1, Q2	8–64 Гбайт	6 000 000	6 500 000
VA100	2 Гбайт	150 000	150 000
VA500	2 Гбайт	400 000	500 000
VA1000	4 Гбайт	800 000	1 000 000
VA2000	8 Гбайт	2 500 000	3 000 000

Для изменения настройки какого-либо параметра межсетевого экрана в ViPNet Coordinator HW:

```
hostname# iplir option set <параметр> <значение>
```

7

Настройка обработки прикладных протоколов

О прикладных протоколах	149
Поддерживаемые прикладные протоколы	150
Настройка параметров обработки прикладных протоколов	151

О прикладных протоколах

Прикладные протоколы обеспечивают работу пользовательских сетевых сервисов, например, IP-телефонии, DNS, FTP. При использовании данных служб IP-адреса часто передаются в теле сообщения прикладного протокола. Эта особенность может повлиять на доступность сервисов в защищенной сети с [виртуальной адресацией](#) или [трансляцией сетевых адресов](#). Кроме того, некоторые прикладные протоколы, помимо управляющего соединения, открывают для передачи данных дополнительные соединения. Поскольку параметры таких соединений заранее неизвестны, невозможно создать сетевой фильтр, разрешающий только необходимый трафик.

ViPNet Coordinator HW может выполнять роль шлюза прикладного уровня, что позволяет решить перечисленные проблемы, обеспечивая:

- Подмену IP-адресов в теле сообщения прикладного протокола на виртуальные IP-адреса.
- Подмену IP-адреса в теле сообщения протокола FTP при трансляции адреса источника или назначения в транзитном FTP-трафике.
- Создание временных сетевых фильтров, пропускающих IP-пакеты дополнительных соединений, которые используются прикладным протоколом.



Примечание. Для разрешения управляющего соединения между узлами создайте сетевой фильтр вручную.

Вы можете настроить обработку прикладных протоколов для открытого и туннелируемого трафика. Список поддерживаемых прикладных протоколов см. ниже.

Поддерживаемые прикладные протоколы

В ViPNet Coordinator HW можно настроить обработку следующих прикладных протоколов:

- DNS — для доступа к сетевым узлам по доменным именам.
- FTP — для передачи файлов между FTP-клиентом и FTP-сервером.

Команды FTP-протокола EPRT и EPSV, используемые некоторыми FTP-клиентами, не поддерживаются ViPNet Coordinator HW.

- SIP — для установления соединений, включающих обмен мультимедийными данными между клиентами.

Не поддерживаются SIP-клиенты и телекоммуникационные серверы, использующие при подключении протокол TLS, «компактные поля» (compact headers) или сжатие данных в сообщениях SIP.

Для некоторых SIP-клиентов могут не обрабатываться такие дополнительные данные, как телефонные справочники, информация о доступности абонента и другие.

- H.323 — для передачи мультимедийных данных в IP-сетях (IP-телефония, видео- и аудиоконференции).

При использовании IP-телефонов Yate не поддерживается виртуальная видимость IP-адресов. RTCP-трафик между IP-телефонами Polycom может работать некорректно при виртуальной видимости IP-адресов.

- SCCP — для передачи сообщений между Skinny-клиентами (проводными и беспроводными IP-телефонами Cisco) и сервером голосовой почты (Cisco Unity Connection и Cisco Unified Communication Manager (CallManager)).

Передача мультимедийных данных с использованием протокола SCCP возможна, если сервер голосовой почты туннелируется координатором ViPNet Coordinator HW.



Примечание. ViPNet Coordinator HW также поддерживает обработку прикладного протокола Cisco NetMeeting для организация видеоконференций, но только при использовании видимости по реальным IP-адресам между клиентским ПО Cisco Meeting App в сети ViPNet и туннелируемым сервером Cisco Meeting Server.

Список поддерживаемых прикладных протоколов изменить нельзя.

Настройка параметров обработки прикладных протоколов

По умолчанию в ViPNet Coordinator HW включена обработка всех поддерживаемых прикладных протоколов.

Примечание.



- Трафик обрабатываемого прикладного протокола не должен проходить через трансляцию сетевых адресов. Это ограничение не распространяется на FTP-трафик. Для маршрутизации обратного трафика от туннелируемых узлов укажите координатор в качестве шлюза по умолчанию для таких узлов.
- Для туннелируемых узлов по умолчанию созданы фильтры, разрешающие передачу трафика по любым IP-адресам, протоколам и портам. Если вы настраиваете обработку прикладных протоколов в туннелируемом трафике, то в целях безопасности рекомендуем удалить эти фильтры и создать фильтры, разрешающие соединения между конкретными узлами по транспортным протоколам и портам, заданным в параметрах настройки прикладных протоколов.

Чтобы просмотреть настройки обработки прикладных протоколов:

```
hostname> alg show
```

В результате отобразится таблица:

```
H323 direct media: off
H323 direct h245: off
SIP direct media: on
```

SERVICE	PROTOCOL	PORTS	ON/OFF
FTP	TCP	21	ON
DNS	UDP	53	ON
H323	TCP	1720	ON
H323	UDP	1719	ON
SCCP	TCP	2000	ON
SIP	TCP	5060,5080	ON
SIP	UDP	5060,5080	ON

Рисунок 30. Просмотр параметров прикладных протоколов

где:

- Service — прикладной протокол.
- Protocol — транспортный протокол, используемый прикладным протоколом.
- Ports — порты, используемые прикладным протоколом.

- On/Off — состояние обработки: on — включена, off — выключена.

Чтобы изменить параметры обработки прикладного протокола:

```
hostname# alg module <прикладной протокол> process <транспортный протокол> <порт> on
```

где:

- <прикладной протокол> — dns, ftp, h323, sccp или sip;
- <транспортный протокол> — tcp или udp;

Выбор транспортного протокола поддерживается не для всех прикладных протоколов. Например, для FTP нельзя задать протокол UDP, а для DNS — TCP.

- <порт> — порт, диапазон портов либо список, состоящий из портов или их диапазонов, разделенных запятой.

Например, если для обработки данных, передаваемых по протоколу SIP, используются не только стандартные порты, но и дополнительные: 10000 и 21–65 для протокола UDP, добавьте их:

```
hostname# alg module sip process udp 10000,21-65 on
```

Заданные параметры обработки должны соответствовать параметрам, указанным в настройках соответствующих приложений (FTP-клиент, DNS-клиент, SIP-клиент и так далее).

Чтобы выключить обработку прикладного протокола:

```
hostname# alg module <прикладной протокол> process off
```



Внимание! После настройки обработки прикладных протоколов (включить или выключить обработку, изменить порт) необходимо перезапустить приложение (например, DNS-, FTP-, SIP-клиент) или устройство (например, SIP-телефон).

Особенности обработки прикладных протоколов DNS, FTP, SIP и H.323

Обработка протокола DNS

В сетях ViPNet приложения могут использовать виртуальные IP-адреса, реально не существующие в сети и уникальные на каждом сетевом узле, что позволяет избежать конфликтов при наличии пересекающихся адресов в разных сетях.

Чтобы узлы внутренней сети ViPNet могли обращаться к удалённым узлам ViPNet по доменным именам через туннелируемый DNS-сервер, требуются дополнительные настройки на DNS-сервере и узлах сети ViPNet:

- 1 На DNS-сервере настройте поддержку динамического изменения DNS-записей. Как правило, по умолчанию DNS-серверы не поддерживают эту настройку.
- 2 Для клиентов, к которым будут обращаться по доменному имени:

2.1 В ViPNet ЦУС или Prime укажите корпоративный DNS-сервер.

2.2 На самих клиентах настройте автоматическую регистрацию на DNS-сервере.

3 На туннелирующем координаторе [включите обработку протокола](#).

Обработка протокола FTP

При передаче файлов между клиентом и сервером по протоколу FTP устанавливается как минимум два TCP-соединения:

- для отправки команд на сервер и получения ответов от него — управляющее TCP-соединение;
- для передачи данных — дополнительное TCP-соединение.

TCP-соединения по протоколу FTP устанавливаются в следующем порядке:

- 1 Клиент создаёт управляющее TCP-соединение с сервером со своего порта из диапазона 1024-65535 на 21-й порт сервера.
- 2 В зависимости от выбранного режима передачи данных, клиент отправляет по управляющему соединению:
 - в активном режиме — номер порта, открытого клиентом для получения данных с сервера;
 - в пассивном режиме — запрос на номер порта сервера, к которому будет подключаться клиент.
- 3 Соединение устанавливается одним из двух способов:
 - в активном режиме — сервер создаёт дополнительное соединение для передачи данных на указанный клиентом порт;
 - в пассивном режиме — сервер по управляющему соединению передаёт клиенту номер порта, к которому клиент создаёт дополнительное соединение.

Таким образом, в активном режиме соединение для передачи данных создаёт сервер, а в пассивном — клиент.

Чтобы использовать протокол FTP:

- 1 Разрешите управляющее соединение. Для этого [создайте сетевой фильтр](#), разрешающий исходящее TCP-соединение на 21-й порт FTP-сервера.
- 2 При использовании активного режима передачи данных разрешите дополнительное соединение. Для этого [включите обработку протокола FTP](#), которая автоматически добавит необходимый сетевой фильтр.

Для разрешения дополнительного соединения в пассивном режиме работы настройки не требуются.

Обработка протокола SIP

При передаче мультимедийных данных по протоколу SIP создаётся управляющее и дополнительное соединение (RTP-сессия). Медиа-поток в дополнительных соединениях могут

передаваться между SIP-клиентами напрямую (т.н. режим Direct Call или Peer-to-Peer RTP) или через сервер телефонии.

Чтобы настроить связь между SIP-клиентами:

- 1 Убедитесь, что [включена обработка протокола SIP](#).
- 2 Если хотя бы один SIP-клиент является открытым узлом, убедитесь, что для всех защищённых узлов [настроена видимость по реальным IP-адресам](#).
- 3 [Создайте сетевые фильтры](#), разрешающие управляющие соединения. Сетевые фильтры должны пропускать трафик между SIP-клиентами и сервером телефонии по транспортным протоколам и портам, заданным в [параметрах обработки прикладных протоколов](#) (по умолчанию TCP и UDP с портами 5060 и 5080):
 - Если все SIP-клиенты туннелируются координаторами, создайте фильтры туннелируемых узлов.
 - Если хотя бы одно SIP-устройство является открытым узлом, создайте транзитные фильтры.



Примечание. Некоторые SIP-клиенты, например SIP-клиент компании Zoiper, могут устанавливать соединения TCP и UDP не через порт 5060, а через динамический порт, выбранный произвольно из определенного диапазона. Поэтому если хотя бы один из SIP-клиентов является открытым узлом, возможность использования динамических портов необходимо отключить в настройках SIP-клиента.

- 4 Настройте создание сетевых фильтров для разрешения RTP-сессий:
 - По умолчанию создаются фильтры, разрешающие RTP-трафик между SIP-клиентами и сервером телефонии. Для этого проверяется совпадение IP-адреса создаваемой RTP-сессии с одним из IP-адресов сигнальной сессии.
 - Если на сервере телефонии настроен режим передачи RTP-трафика между SIP-клиентами напрямую (без участия сервера телефонии), выполните:

```
hostname# alg module sip direct-media off
```

После этого разрешающие фильтры будут создаваться без проверки принадлежности RTP-сессии к сигнальной сессии.



Внимание! Сетевые фильтры, которые создаются для режима прямого соединения между SIP-клиентами, будут разрешать RTP-сессии между любыми IP-адресами. Поэтому не рекомендуется использовать этот режим, если ViPNet Coordinator HW размещён на границе защищаемой сети.

- 5 Чтобы настройки вступили в силу, перезапустите основные службы и драйверы:

```
hostname# vpn stop
```

```
hostname# vpn start
```

Обработка протокола H.323

В наборе протоколов H.323 используются сигнальные протоколы H.225 и H.245. Для передачи медиа-поток используется RTP. При включении обработки H.323 автоматически создаются временные сетевые фильтры, разрешающие сигнальные H.245-сессии и дополнительные RTP-сессии. При этом, чтобы разрешить сигнальные H.225-сессии, необходимо создать разрешающие сетевые фильтры вручную.

Терминалы H.323 (IP-телефоны) могут устанавливать связь друг с другом напрямую или с участием контроллера зоны (gatekeeper). В зависимости от наличия контроллера зоны и режимов его работы может потребоваться настройка создания временных сетевых фильтров, разрешающих H.245- и RTP-сессии.

Чтобы настроить связь между IP-телефонами по протоколу H.323:

- 1 Убедитесь, что [включена обработка протокола H.323](#).
- 2 При использовании IP-телефонов Yate [настройте видимость туннелируемых узлов по реальным IP-адресам](#).
- 3 [Создайте сетевые фильтры](#), разрешающие сигнальные H.225-сессии между H.323-устройствами. Сетевые фильтры должны пропускать трафик по транспортным протоколам и портам, заданным в [параметрах обработки прикладных протоколов](#) (по умолчанию UDP 1719 и TCP 1720):
 - Если H.323-устройства туннелируются координаторами, создайте фильтры туннелируемых узлов.
 - Если хотя бы одно H.323-устройство является открытым узлом, создайте транзитные фильтры.
- 4 Настройте автоматическое создание сетевых фильтров, разрешающих дополнительную сигнальную H.245-сессию, а также RTP-сессию:

В зависимости от настроек H.323-телефонии потоки сигнальных H.225- и H.245-сессий могут проходить через контроллер зоны или напрямую между IP-телефонами.

- Если пути прохождения H.225-потока и H.245-потока отличаются, например:

H.225: IP-телефон -> контроллер зоны -> IP-телефон

H.245: IP-телефон -> IP-телефон

RTP: IP-телефон -> IP-телефон

выполните команду:

```
hostname# alg module h323 direct-h245 off
```

- Если пути прохождения RTP-потока и родительского H.245-потока отличаются, например:

H.225: IP-телефон -> контроллер зоны -> IP-телефон

H.245: IP-телефон -> контроллер зоны -> IP-телефон

RTP: IP-телефон -> IP-телефон

выполните команду:

```
hostname# alg module h323 direct-media off
```

- Если H.225-, H.245- и RTP-сессии проходят между IP-телефонами напрямую, например:

H.225: IP-телефон -> IP-телефон

H.245: IP-телефон -> IP-телефон

RTP: IP-телефон -> IP-телефон

менять настройки по умолчанию не требуется (`direct-h245 on` и `direct-media on`).



Примечание. В некоторых IP-телефонах (например, Yate) отсутствует H.245-сессия. Если пути прохождения RTP-потока и родительского H.225-потока отличаются, выполните команду:

```
hostname# alg module h323 direct-media off
```

Иначе, оставьте настройки по умолчанию.

Подробнее о работе этих команд см. в документе «Справочник команд и конфигурационных файлов».

- 5 Чтобы настройки вступили в силу, перезапустите основные службы и драйверы:

```
hostname# vpn stop
```

```
hostname# vpn start
```


8

Настройка прокси-сервера

Общие сведения о работе прокси-сервера	158
Настройка основных параметров прокси-сервера	160
Настройка антивирусной проверки трафика	163
Настройка правил контент-фильтрации трафика	164

Общие сведения о работе прокси-сервера

Прокси-сервер ViPNet Coordinator HW обладает следующими возможностями:

- Кеширование данных для ускорения доступа пользователей к часто запрашиваемым ресурсам.
- «Прозрачный» режим работы, для которого не требуется дополнительная настройка приложений на рабочих местах пользователей.
- Контент-фильтрация HTTP-трафика.
- Проверка HTTP-трафика внешним антивирусом по протоколу ICAP.



Рисунок 31. Схема работы прокси-сервера ViPNet Coordinator HW

Порядок обработки запроса при обращении к интернет-ресурсу:

- 1 Запрос от клиента прокси-сервера поступает на ViPNet Coordinator HW:
 - Обработывается сетевыми фильтрами:
 - если сработал запрещающий сетевой фильтр, то передача IP-пакетов блокируется, в журнал IP-пакетов добавляется запись об этом событии (см. [Типы событий в журнале регистрации IP-пакетов](#));
 - если запрос не попал под действие ни одного из запрещающих сетевых фильтров, он передается на прокси-сервер.
 - Если включена контент-фильтрация трафика, то запрос обрабатывается ее правилами:
 - если ни одно правило не сработало, то применяется правило по умолчанию (блокирующее или разрешающее);
 - если сработало запрещающее правило по HTTP-методу или MIME-типу, то передача HTTP-трафика блокируется.
 - Если включена антивирусная проверка трафика, данные проверяются на наличие вирусов. Если пользователь попытается отправить файл, в котором был обнаружен вирус, он увидит соответствующее сообщение от антивируса и попытка отправки файла будет заблокирована. Запись об этом событии будет создана в системном журнале.
- 2 Если запрос не был заблокирован при контент-фильтрации трафика или при проверке антивирусом, он передается на интернет-ресурс.
- 3 Ответ от интернет-ресурса поступает на ViPNet Coordinator HW:

- Обрабатывается сетевыми фильтрами аналогично шагу 1.
- Если включена функция фильтрации содержимого трафика, то ответ обрабатывается ее правилами аналогично шагу 1.
- Если включена антивирусная проверка трафика, данные проверяются на наличие вирусов. Если пользователь попытается загрузить файл (в том числе открыть HTML-страницу), в котором был обнаружен вирус, он увидит сообщение от антивируса и попытка загрузки файла будет заблокирована. Запись об этом событии будет создана в системном журнале.

В случае успешного прохождения всех проверок HTTP-трафик от прокси-сервера передается пользователю.

Если вы хотите использовать встроенный прокси-сервер для защиты интернет-соединения пользователей вашей локальной сети, настройте основные параметры прокси-сервера. Вы также можете [включить антивирусную проверку](#) и [контент-фильтрацию трафика](#).

Режимы работы прокси-сервера:

- Обычный режим — в пользовательских приложениях, например в браузере, требуется настроить подключение: указать IP-адрес и порт прокси-сервера и исключить использование прокси-сервера для локальных адресов, в том числе адресов ViPNet Coordinator HW.
- «Прозрачный» режим — дополнительная настройка приложений не требуется. При этом пользователи не имеют возможности отказаться от использования прокси-сервера. На компьютерах пользователей в качестве шлюза по умолчанию необходимо указать IP-адрес ViPNet Coordinator HW, на котором запущен прокси-сервер.



Примечание. По умолчанию в «прозрачном» режиме работы прокси-сервера к DNS-серверу ViPNet Coordinator HW могут обращаться только защищенные узлы. Если необходимо предоставить доступ к DNS-серверу для каких-либо открытых узлов, на ViPNet Coordinator HW создайте сетевой фильтр:

```
hostname# firewall local add rule "AllowDNS" src <IP-адреса открытых узлов>
dst @local udp dport 53 pass
```

Настройка основных параметров прокси-сервера

- 1 Завершите работу прокси-сервера, если он запущен:

```
hostname# service http-proxy stop
```



Внимание! На запущенном прокси-сервере нельзя выполнить настройку параметров.

- 2 Укажите интерфейсы и порты, на которых прокси-сервер будет принимать запросы от пользователей:

```
hostname# service http-proxy listen-address add <интерфейс> <порт>
```



Внимание! Разрешено задавать номер порта, который не используется другими сервисами.

- 3 Укажите внешний интерфейс ViPNet Coordinator HW, который используется для подключения к интернету:

```
hostname# service http-proxy external-address set <интерфейс>
```

- 4 Если требуется, задайте размер кеша прокси-сервера:

```
hostname# service http-proxy cache <размер>
```

Размер кеша указывается в мегабайтах, максимальное значение и значение по умолчанию зависит от аппаратной платформы (подробнее см. описание команды в документе «Справочник команд и конфигурационных файлов»). Кеш используется для хранения копии данных, к которым часто обращаются пользователи.

- 5 Если требуется включите «прозрачный» режим работы прокси-сервера:

```
hostname# service http-proxy transparent-mode {on | off}
```

- 6 Добавьте сетевые фильтры и правила трансляции, необходимые для работы прокси-сервера:

```
hostname# service http-proxy fw-rules apply
```

При выполнении команды все необходимые сетевые фильтры и правила трансляции будут созданы автоматически.

При выполнении команды также автоматически создается пользовательская группа объектов `HttpProxyUsers`, содержащая IP-адреса локальных сетей, для которых разрешено использование прокси-сервера. По умолчанию в эту группу входят те же IP-адреса, что и в пользовательскую группу объектов `PrivateNetworkIP` (10.0.0.0/8; 172.16.0.0/12; 192.168.0.0/16).

Чтобы добавить или удалить IP-адреса из списка локальных сетей, которым разрешено использовать прокси-сервер, измените группу объектов `HttpProxyUsers`. Изменение

пользовательских групп объектов возможно только с помощью веб-интерфейса. Подробнее см. документ «Настройка с помощью веб-интерфейса», раздел «Создание и изменение групп объектов».



Примечание. Если для подключения к интернету ViPNet Coordinator HW использует маршрут с несколькими шлюзами (multi-hop route), для корректной работы прокси-сервера на ViPNet Coordinator HW создайте сетевой фильтр:

```
hostname# firewall local add src @local dst @InternetIP tcp dport 80 tcp dport 21 tcp dport 443 pass tcp dport 9443
```

-
- 7 Включите или выключите автоматический запуск прокси-сервера при загрузке ViPNet Coordinator HW:

```
hostname# service http-proxy mode {on | off}
```

- 8 Включите прокси-сервер:

```
hostname# service http-proxy start
```



Примечание. Для корректной работы прокси-сервера перед его запуском настройте и включите DNS-сервер (см. [Настройка DNS-сервера](#)).

Дополнительные команды для работы с прокси-сервером

- Просмотреть состояние и настройки прокси-сервера:

```
hostname# service http-proxy show
```

- Просмотреть список заданных интерфейсов и портов, на которых прокси-сервер принимает запросы от пользователей:

```
hostname# service http-proxy listen-address list
```

- Просмотреть заданный внешний интерфейс ViPNet Coordinator HW, который используется для подключения к интернету:

```
hostname# service http-proxy external-address show
```

- Удалить интерфейс, на который прокси-сервер принимает запросы от пользователей:

```
hostname# service http-proxy listen-address delete <интерфейс>
```

- Отключить «прозрачный» режим работы прокси-сервера:

```
hostname# service http-proxy transparent-mode off
```

- Удалить сетевые фильтры и правила трансляции, созданные автоматически командой `service http-proxy fw-rules apply`:

```
hostname# service http-proxy fw-rules delete
```

- Завершить работу прокси-сервера:

```
hostname# service http-proxy stop
```



Внимание! При завершении работы прокси-сервера фильтры и правила, созданные командой `service http-proxy fw-rules apply`, остаются активными (продолжают применяться). Удалите их:

```
hostname# service http-proxy fw-rules delete
```

- Сбросить текущие настройки прокси-сервера:

```
hostname# service http-proxy reset
```

Настройка антивирусной проверки трафика

Если на ViPNet Coordinator HW запущен прокси-сервер, вы можете настроить антивирусную проверку данных, передаваемых через него по HTTP в обоих направлениях: из интернета к пользователю и от пользователя в интернет, например при добавлении вложения в сообщение электронной почты с помощью веб-интерфейса. Проверка выполняется внешним антивирусом по протоколу [ICAP](#).



Внимание! Время на ICAP-сервере должно быть синхронизировано с ViPNet Coordinator HW.

1 Остановите работу прокси-сервера:

```
hostname# service http-proxy stop
```

2 Выключите антивирусную проверку трафика прокси-сервером:

```
hostname# service http-proxy antivirus mode off
```

3 Задайте IP-адрес ICAP-сервера внешнего антивируса для проверки входящего трафика:

```
hostname# service http-proxy antivirus server-url add respmod  
icap://192.168.1.45:1344/respmod
```

4 Задайте IP-адрес ICAP-сервера внешнего антивируса для проверки исходящего трафика:

```
hostname# service http-proxy antivirus server-url add reqmod  
icap://192.168.1.45:1344/reqmod
```

5 Создайте разрешающие фильтры межсетевого экрана для работы прокси-сервера:

```
hostname# service http-proxy fw-rules apply
```

6 Задайте режим пропуска трафика при недоступности ICAP-сервера:

```
hostname# service http-proxy antivirus bypass on
```

7 Включите антивирусную проверку трафика прокси-сервером:

```
hostname# service http-proxy antivirus mode on
```

8 Запустите прокси-сервер:

```
hostname# service http-proxy start
```

Настройка правил контент-фильтрации трафика

Прокси-сервер проверяет HTTP-трафик с помощью правил контент-фильтрации. Вы можете настроить блокировку трафика по HTTP-методу запроса к удаленному ресурсу, а также по MIME-типу файла или приложения.



Примечание. При блокировке трафика в браузере отображается веб-страница с соответствующим сообщением, а время блокировки отображается в формате UTC+0.

Контент-фильтрация HTTP-трафика включается автоматически при запуске прокси-сервера.

Для настройки контент-фильтрации:

- 1 Завершите работу прокси-сервера:

```
hostname# service http-proxy stop
```

- 2 Чтобы включить или выключить контент-фильтрацию трафика:

```
hostname# service http-proxy content-filter mode {on | off}
```

- 3 Задайте действие правила по умолчанию для запросов к HTTP-ресурсам:

```
hostname# service http-proxy content-filter default-request-action {pass | drop}
```

указав следующие параметры:

- o `pass` — правило по умолчанию пропускает запросы к HTTP-серверам, не попадающие под действие других правил;
- o `drop` — правило по умолчанию блокирует запросы к HTTP-серверам, не попадающие под действие других правил.

- 4 Задайте действие правила по умолчанию для ответов от HTTP-ресурсов:

```
hostname# service http-proxy content-filter default-reply-action {pass | drop}
```

указав следующие параметры:

- o `pass` — правило по умолчанию пропускает ответы от HTTP-серверов, не попадающие под действие других правил;
- o `drop` — правило по умолчанию блокирует ответы от HTTP-серверов, не попадающие под действие других правил.

- 5 Чтобы добавить правило контент-фильтрации трафика:

```
hostname# service http-proxy content-filter add [num <номер>] [rule <имя>] src <адрес  
отправителя> dst <адрес получателя> {command <HTTP-метод> | mime-type <mime-тип>}  
<действие>
```

указав следующие параметры:

- `<номер>` — порядковый номер правила в таблице, определяющий его приоритет. Чем меньше порядковый номер правила, тем выше его приоритет (наиболее приоритетное правило имеет номер 1). Если указанный номер правила меньше последнего номера в таблице, нумерация правил, следующих после нового правила, будет автоматически изменена (их номера будут увеличены на 1).
- `<имя>` — имя правила. Если имя состоит из нескольких слов, разделенных пробелом, заключите его в кавычки (например, `rule "number one"`).



Примечание. Порядковый номер и имя не являются обязательными параметрами правила фильтрации содержимого трафика. При создании правила без указания номера ему будет присвоен номер, следующий за номером последнего правила в таблице, и оно будет иметь самый низкий приоритет. При создании правила без указания имени будет создано правило без имени. Создание фильтра с порядковым номером, превышающим максимальный номер в таблице фильтров более чем на 1, запрещено.

- Параметры контент-фильтрации:

- Адрес отправителя.
- Адрес получателя.
- HTTP-метод.
- MIME-тип.
- Действие.

6 Чтобы изменить существующее правило контент-фильтрации:

```
hostname# service http-proxy content-filter change num <номер правила> [rule <имя правила>] src <адрес отправителя> dst <адрес получателя> {command <HTTP-метод> | mime-type <mime-тип>} <действие>
```

7 Чтобы удалить правило контент-фильтрации из таблицы правил:

```
hostname# service http-proxy content-filter delete {num <номер> | rule <имя>},
```

- `<номер>` — порядковый номер правила в таблице;
- `<имя>` — имя правила.

8 Чтобы изменить порядок правила контент-фильтрации в таблице правил:

```
hostname# service http-proxy content-filter move {num <номер> | rule <имя>} to <новый номер>,
```

где:

- `<номер>` — порядковый номер правила в таблице, определяющий его приоритет;
- `<имя>` — имя правила;
- `<новый номер>` — новый порядковый номер правила в таблице.

9 Чтобы просмотреть правила контент-фильтрации:

```
hostname# service http-proxy content-filter list
```

10 Запустите прокси-сервер:

```
hostname# service http-proxy start
```

Адрес отправителя

Адрес отправителя является обязательным параметром правила контент-фильтрации:

```
src <адрес отправителя>
```

Возможные значения адреса отправителя:

- IP-адрес, например:

```
src 192.168.1.36
```
- адрес подсети в формате классовой адресации, например:

```
src 192.168.1.0/24
```
- системная группа объектов `any`, например:

```
src @any
```



Внимание! В качестве отправителя всегда указывайте клиента прокси-сервера. Использование доменных имен не допускается.

Адрес получателя

Адрес получателя является обязательным параметром правила контент-фильтрации:

```
dst <адрес получателя>
```

Возможные значения адреса получателя:

- IP-адрес или доменное имя узла;
- системная группа объектов `any`, например:

```
dst @any
```



Внимание! Контент-фильтрация прокси-сервера работает на прикладном уровне модели OSI. Поэтому при использовании в качестве адреса получателя доменного имени узла фильтрация будет осуществляться только по доменному имени, а при использовании IP-адреса узла, только по IP-адресу, без взаимного разрешения «доменное имя — IP-адрес».

HTTP-метод

HTTP-метод не является обязательным параметром правила контент-фильтрации:

```
command <HTTP-метод>
```

Вы можете указать любой из методов протокола HTTP/1.0 (но только один):

- HEAD
- GET
- POST
- PUT
- DELETE
- CONNECT
- OPTIONS
- TRACE



Внимание! Вы можете указать HTTP-метод только в том случае, если в правиле контент-фильтрации трафика не был задан MIME-тип.

Правило с заданным HTTP-методом будет действовать в тех случаях, когда прокси-сервер получит от клиента запрос к удаленному HTTP-серверу с этим методом. Например, необходимо запретить заполнение и отправку форм на веб-портал `example.com`, доступный по протоколу HTTP. Для этого необходимо создать правило, которое будет блокировать метод POST для `example.com`:

```
hostname# service http-proxy content-filter add src @any dst example.com command POST drop
```

Чтобы полностью запретить доступ к этому portalу, создайте правило с методом GET:

```
hostname# service http-proxy content-filter add src @any dst example.com command GET drop
```

MIME-тип

Тип содержимого не является обязательным параметром правила контент-фильтрации:

```
mime-type <mime-тип>
```

Вы можете указать любой тип содержимого из поддерживаемых (см. [Поддерживаемые MIME-типы](#)), но только один.



Внимание! Вы можете указать MIME-тип только в том случае, если в правиле контент-фильтрации трафика не был задан HTTP-метод.

Правило с заданным MIME-типом будет блокировать трафик, передающий файлы этого типа. Например, чтобы заблокировать загрузку изображений формата PNG:

```
hostname# service http-proxy content-filter add src @any dst @any mime-type image/png drop
```



Примечание. MIME-тип файла может зависеть от настроек удаленного HTTP-сервера, с которого этот файл был загружен. То есть, файл будет иметь тот MIME-тип, который был назначен ему HTTP-сервером.

Действие

Действие является обязательным параметром правила контент-фильтрации:

- `pass` — пропускать HTTP-трафик;
- `drop` — блокировать HTTP-трафик.

9

Настройка сетевых служб

Настройка DHCP-сервера	170
Настройка DHCP-relay	172
Примеры настройки DHCP-сервера и DHCP-relay	174
Настройка DNS-сервера	184
Настройка NTP-сервера	186
Организация работы клиентов удаленного офиса с корпоративными DNS- и NTP-серверами	189
Настройка точки доступа сети Wi-Fi	191

Настройка DHCP-сервера

В состав ViPNet Coordinator HW входит DHCP-сервер. Он динамически назначает IP-адреса сетевым узлам (DHCP-клиентам). Также DHCP-сервер может передать дополнительные параметры — IP-адреса шлюза по умолчанию, DNS-, NTP-сервера и другие.

DHCP-сервер может обслуживать DHCP-клиентов, которые находятся:

- в одной сети с DHCP-сервером, в том числе в сегментах сети, объединенных по L2OverIP;
- в удаленных подсетях с помощью агента DHCP-relay.

Чтобы настроить DHCP-сервер:

- 1 Включите сетевой интерфейс, на котором будет работать DHCP-сервер:

```
hostname# inet ifconfig <интерфейс> up
```

- 2 Задайте этот интерфейс для работы DHCP-сервера:

```
hostname# inet dhcp server add interface <интерфейс>
```



Примечание. Можно указать Ethernet-, VLAN-, Wi-Fi или агрегированный интерфейс, у которого настроен статический IP-адрес.

Если вы указали wlan0, [настройте точку доступа к сети Wi-Fi](#).

- 3 Задайте диапазон IP-адресов, выделяемых клиентам:

```
hostname# inet dhcp server add range <начальный IP-адрес> <конечный IP-адрес>
{interface <интерфейс> | remote subnet <подсеть> mask <маска>}
```

Диапазон IP-адресов должен принадлежать сети интерфейса или заданной подсети (при указании `remote subnet`). IP-адрес интерфейса не должен входить в этот диапазон.



Примечание. DHCP-сервер может выделять клиентам любые диапазоны IP-адресов. Если локальная сеть маршрутизируется в интернет, используйте диапазоны, которые установлены стандартом для частных сетей: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16.

- 4 Задайте дополнительные параметры:

- IP-адрес шлюза по умолчанию:

```
hostname# inet dhcp server add router <IP-адрес> {interface <интерфейс> | remote
subnet <подсеть> mask <маска> | host <имя узла>}
```

IP-адрес шлюза должен принадлежать сети интерфейса и не должен входить в диапазон IP-адресов, выделяемых клиентам. В случае указания `remote subnet` IP-адреса шлюза должен принадлежать к указанной подсети `<subnet>`.

- IP-адрес DNS-сервера:

```
hostname# inet dhcp server add dns <IP-адрес> [{interface <интерфейс> | remote
subnet <подсеть> mask <маска> | host <имя узла>}]
```

- IP-адрес NTP-сервера:

```
hostname# inet dhcp server add ntp <IP-адрес> [{interface <интерфейс> | remote  
subnet <подсеть> mask <маска> | host <имя узла>}]
```

Например, чтобы задать адрес DNS-сервера 192.168.15.41 для клиентов другой сети с адресом 192.168.1.0/24:

- Задайте eth0 в качестве сетевого интерфейса, на котором DHCP-сервер будет обрабатывать запросы от агента DHCP-relay, работающего в удаленной подсети 192.168.1.0/24:

```
hostname# inet dhcp server add relay-interface eth0 remote subnet 192.168.1.0 mask  
255.255.255.0
```

- Задайте адрес DNS-сервера 192.168.15.41 для клиентов сети 192.168.1.0/24:

```
hostname# inet dhcp server add dns 192.168.15.41 remote subnet 192.168.1.0 mask  
255.255.255.0
```

5 Проверьте настройки DHCP-сервера:

```
hostname# inet show dhcp server
```

6 Включите автоматический запуск DHCP-сервера:

```
hostname# inet dhcp server mode on
```

По умолчанию автоматический запуск DHCP-сервера выключен.

7 Запустите DHCP-сервер:

```
hostname# inet dhcp server start
```

Чтобы посмотреть список клиентов DHCP-сервера:

```
hostname# inet show dhcp server lease [{last | all | full}], где:
```

- last — вывести текущий список клиентов DHCP-сервера (по умолчанию).
- all — вывести полный список клиентов DHCP-сервера с историей аренды IP-адресов.
- full — вывести полный список клиентов DHCP-сервера без форматирования вывода.

Если несколько клиентов получили одинаковые IP-адреса, возможно, у них совпадает идентификатор DHCP-клиента (UID). Чтобы избежать этого, в сетевых настройках клиентов задайте MAC-адрес в качестве идентификатора DHCP.

Настройка DHCP-relay

ViPNet Coordinator HW можно использовать в качестве агента [DHCP-relay](#). Он принимает от узлов DHCP-запросы и передает их DHCP-серверу. Ответы, полученные от DHCP-сервера, агент DHCP-relay перенаправляет сетевым узлам.



Внимание! Для приема и перенаправления DHCP-запросов через DHCP-relay необходимо, чтобы DHCP-сервер поддерживал Option 82 Suboption 5.

Агент DHCP-relay используется, когда сетевые узлы не могут обращаться напрямую к DHCP-серверу. Например, в разветвленной сети из нескольких подсетей, в которых нет своего DHCP-сервера.

Агент DHCP-relay может работать с несколькими локальными сетями. Это могут быть как сети, подключенные к разным физическим сетевым интерфейсам ViPNet Coordinator HW, так и виртуальные локальные сети, подключенные к одному физическому интерфейсу. Вы можете настроить несколько копий DHCP-relay, которые будут иметь различные слушающие, внешние интерфейсы и IP-адреса DHCP-серверов.

Узлы, которые будут обращаться к DHCP-серверу через ViPNet Coordinator HW, могут быть защищенными и открытыми. DHCP-сервер может быть защищенным, открытым или туннелируемым узлом.

Если агент DHCP-relay перенаправляет запросы на DHCP-сервер, который туннелируется другим координатором, настройте между координаторами видимость по реальным IP-адресам (см. [Настройка видимости узлов](#)). Если между ними настроена видимость по виртуальным адресам, ответы от DHCP-сервера будут отправляться на реальный адрес ViPNet Coordinator HW, выступающего в качестве агента DHCP-relay, и не будут доставлены.

Чтобы запустить DHCP-relay:

- 1 Задайте список сетевых интерфейсов, через которые DHCP-relay должен получать DHCP-запросы от клиентов. Чтобы добавить интерфейс в список:

```
hostname# inet dhcp relay [<номер копии>] add listen-interface <имя интерфейса>
```

Если номер копии процесса DHCP-relay не задан, то используется номер 1.

На каждом добавляемом сетевом интерфейсе должен быть задан статический IP-адрес, принадлежащий адресному пространству соответствующей подсети. Сетевой интерфейс, который используется для соединения с удаленным сервером может также получать IP-адрес по DHCP.

- 2 Задайте внешний DHCP-сервер для перенаправления DHCP-запросов клиентов и сетевой интерфейс ViPNet Coordinator HW для доступа к удаленному серверу:

```
hostname# inet dhcp relay [<номер копии>] add external-interface <имя интерфейса>  
server <IP-адрес внешнего DHCP-сервера>
```




Внимание! В случае если по событию DGD произойдет переключение внешних сетевых интерфейсов, то служба DHCP-relay, использующая один из этих интерфейсов в качестве внешнего, будет недоступна.

- 3 Укажите сетевой интерфейс ViPNet Coordinator HW, который должен использоваться для доступа к резервному DHCP-серверу, если он есть в сети:

```
hostname# inet dhcp relay [<номер копии>] add backup-interface <имя интерфейса> server  
<IP-адрес сервера>
```

- 4 Включите автоматический запуск DHCP-relay при загрузке ViPNet Coordinator HW:

```
hostname# inet dhcp relay mode on
```

По умолчанию автоматический запуск DHCP-relay выключен.

- 5 Запустите DHCP-relay:

```
hostname# inet dhcp relay [<номер копии>] start
```

- 6 Посмотрите настройки DHCP-relay и его текущее состояние:

```
hostname# inet show dhcp relay [<номер копии>]
```



Примечание. Чтобы впоследствии завершить работу DHCP-relay и одновременно сбросить все его настройки, а также выключить автоматический запуск при загрузке ViPNet Coordinator HW:

```
hostname# inet dhcp relay reset
```

Примеры настройки DHCP-сервера и DHCP-relay

Организация работы клиентов с локальным DHCP-сервером

В локальной сети небольшой организации, имеющей один офис, вы можете развернуть собственный DHCP-сервер с помощью ViPNet Coordinator HW:

- 1 Остановите DHCP-сервер, если он был запущен:

```
hostname# inet dhcp server stop
```

- 2 Подключите один из интерфейсов ViPNet Coordinator HW к локальной сети.
- 3 Установите на нем IP-адрес, принадлежащий адресному пространству локальной сети.
- 4 Включите интерфейс:

```
hostname# inet ifconfig <имя интерфейса> up
```

- 5 Задайте этот интерфейс для работы DHCP-сервера:

```
hostname# inet dhcp server add interface <имя интерфейса>
```

- 6 Задайте диапазон IP-адресов, выделяемых клиентам:

```
hostname# inet dhcp server add range <начальный IP-адрес> <конечный IP-адрес> interface  
<имя интерфейса>
```

Диапазон IP-адресов должен принадлежать сети интерфейса. При этом IP-адрес интерфейса не должен входить в этот диапазон.

- 7 Задайте срок (время аренды в секундах), на который клиентам предоставляются IP-адреса:

```
hostname# inet dhcp server add default-lease-time <время аренды>
```

- 8 Задайте IP-адрес шлюза по умолчанию, если его необходимо передавать клиентам:

```
hostname# inet dhcp server add router <IP-адрес> interface <имя интерфейса>
```

- 9 Задайте IP-адрес DNS-и NTP-сервера, если их необходимо передавать клиентам:

```
hostname# inet dhcp server add dns <IP-адрес> interface <имя интерфейса>
```

```
hostname# inet dhcp server add ntp <IP-адрес> interface <имя интерфейса>
```



Примечание. Вы можете задать другие параметры, передаваемые DHCP-сервером своим клиентам. Подробнее см. в документе «Справочник команд и конфигурационных файлов».

- 10 Проверьте текущие параметры DHCP-сервера:

```
hostname# inet show dhcp server
```

- 11 Включите автоматический запуск DHCP-сервера при загрузке ViPNet Coordinator HW:

```
hostname# inet dhcp server mode on
```

12 Запустите DHCP-сервер:

```
hostname# inet dhcp server start
```



Внимание! Вместе с IP-адресом DHCP-сервер предоставляет клиентам свой IP-адрес в качестве адресов DNS- и NTP-сервера. Поэтому клиенты будут отправлять запросы на разрешение имен и синхронизацию времени через соответствующие службы, запущенные на ViPNet Coordinator HW. Если эти службы не запущены, то клиенты не смогут работать с доменными именами и синхронизировать свое время.

Организация работы клиентов с удаленным DHCP-сервером

Рассмотрим пример:

- У организации есть несколько офисов, при этом в центральном установлен DHCP-сервер с адресом 172.16.1.10.
- В филиале имеется две подсети с адресными пространствами 10.0.0.0/24 и 10.0.1.0/24.
- Клиентам, находящимся в филиале, требуется динамически выделять IP-адреса, но устанавливать в филиале свой DHCP-сервер нецелесообразно.

Организуем взаимодействие клиентов филиала с DHCP-сервером с помощью ViPNet Coordinator HW в качестве агента DHCP-relay. При этом между филиалами и центральным офисом должны быть организованы каналы связи с прямой маршрутизацией.

Если агент DHCP-relay перенаправляет запросы на DHCP-сервер, который туннелируется другим координатором, то между ViPNet Coordinator HW, выступающим в качестве агента DHCP-relay, и координатором, туннелирующим DHCP-сервер, настройте видимость по реальным IP-адресам (см. [Настройка видимости узлов](#)). Если между этими координаторами настроена видимость по виртуальным адресам, ответы от DHCP-сервера будут отправляться на реальный адрес ViPNet Coordinator HW, выступающего в качестве агента DHCP-relay, и не будут доставлены

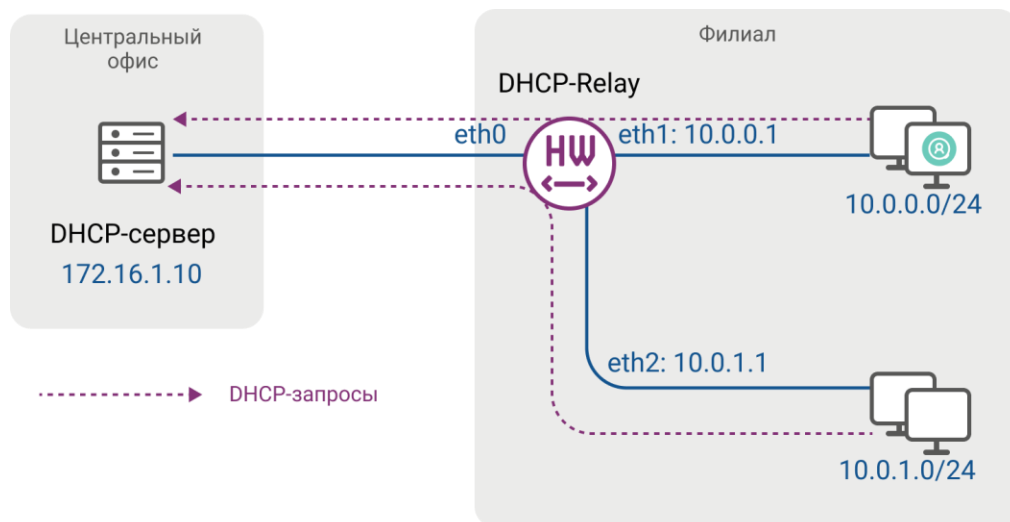


Рисунок 32. Схема использования ViPNet Coordinator HW в качестве агента DHCP-relay

Задайте на DHCP-сервере для каждой подсети диапазон выделяемых адресов и маршрут в сторону ViPNet Coordinator HW. На ViPNet Coordinator HW выполните:

- 1 Подключите eth1 к подсети 10.0.0.0/24, eth2 к подсети 10.0.1.0/24.

На каждом из этих интерфейсов задайте статический IP-адрес, принадлежащий адресному пространству соответствующей подсети (eth1 — 10.0.0.1, eth2 — 10.0.1.1).

- 2 Подключите eth0 к внешней сети.

- 3 Задайте eth1 и eth2 в качестве интерфейсов, принимающих от клиентов DHCP-запросы:

```
hostname# inet dhcp relay add listen-interface eth1
hostname# inet dhcp relay add listen-interface eth2
```

- 4 Задайте параметры для связи ViPNet Coordinator HW с внешним DHCP-сервером:

```
hostname# inet dhcp relay add external-interface eth0 server 172.16.1.10
```

- 5 Проверьте текущие параметры DHCP-relay:

```
hostname# inet show dhcp relay
```

- 6 Включите автоматический запуск DHCP-relay при загрузке ViPNet Coordinator HW:

```
hostname# inet dhcp relay mode on
```

По умолчанию автоматический запуск DHCP-relay выключен.

- 7 Запустите DHCP-relay:

```
hostname# inet dhcp relay start
```

Использование DHCP-сервера и DHCP-relay в разных сетях

Рассмотрим пример:

- HW-1 работает как DHCP-сервер.
- HW-2 и HW-3 работают как DHCP-relay.
- DHCP-сервер (HW-1), работает для следующих DHCP-клиентов:
 - клиенты сети 1 (192.168.2.10-20);
 - TFTP-сервер (192.168.3.20), передающий данные для тонкого клиента в сети 3;
 - защищенные узлы сети 2 (диапазон 172.16.0.10-20);
 - туннелируемые узлы сети 2 (диапазон 172.16.1.10-20);
 - тонкий клиент в сети 3, получающий данные от TFTP-сервера в сети 1.
- Для HW-1 настроена видимость по реальным IP-адресам с координаторами, работающими в качестве агентов DHCP-relay.

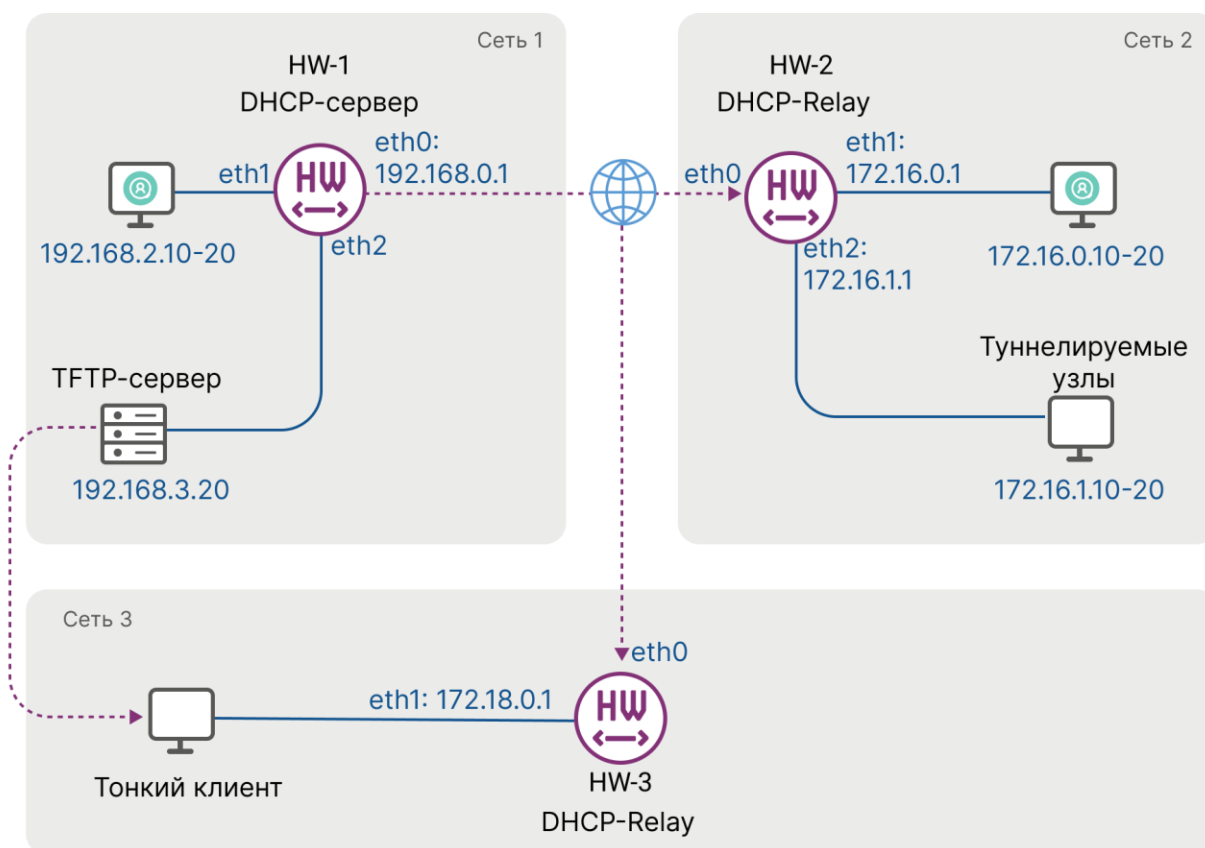


Рисунок 33. Использование DHCP-сервера и DHCP-relay в разных сетях

На HW-1 настройте DHCP-сервер:

- 1 Задайте рабочие сетевые интерфейсы DHCP-сервера:

- для сетей 2 и 3:

```
hostname# inet dhcp server add interface eth0
```

- для клиентов сети 1:

```
hostname# inet dhcp server add interface eth1
```

- для TFTP-сервера:

```
hostname# inet dhcp server add interface eth2
```

2 Добавьте подсеть защищенных узлов сети 2 в качестве удаленной подсети для DHCP-сервера:

```
hostname# inet dhcp server add relay-interface eth0 remote subnet 172.16.0.0 mask 255.255.255.0
```

3 Добавьте подсеть туннелируемых узлов сети 2 в качестве удаленной подсети для DHCP-сервера:

```
hostname# inet dhcp server add relay-interface eth0 remote subnet 172.16.1.0 mask 255.255.255.0
```

4 Задайте диапазоны IP-адресов, выделяемых DHCP-сервером:

- для клиентов сети 1:

```
hostname# inet dhcp server add range 192.168.2.10 192.168.2.20 interface eth1
```

- для защищенных узлов сети 2:

```
hostname# inet dhcp server add range 172.16.0.10 172.16.0.20 remote subnet 172.16.0.0 mask 255.255.255.0
```

- для туннелируемых узлов сети 2:

```
hostname# inet dhcp server add range 172.16.1.10 172.16.1.20 remote subnet 172.16.1.0 mask 255.255.255.0
```

5 Задайте IP-адрес шлюза по умолчанию:

- для сети 2:

```
hostname# inet dhcp server add router 172.16.0.1 remote subnet 172.16.0.0 mask 255.255.255.0
```

- для туннелируемых узлов сети 2:

```
hostname# inet dhcp server add router 172.16.1.1 remote subnet 172.16.1.0 mask 255.255.255.0
```

- для сети 3:

```
hostname# inet dhcp server add router 172.18.0.1 remote subnet 172.18.0.0 mask 255.255.255.0
```

6 Зарезервируйте IP-адрес 192.168.3.20 для TFTP-сервера:

```
hostname# inet dhcp server add host TFTP hardware <MAC-адрес TFTP-сервера> address 192.168.3.20 interface eth2
```

7 Задайте для тонкого клиента IP-адрес TFTP-сервера и путь к файлу с параметрами загрузки:

```
hostname# inet dhcp server add tftp 192.168.3.20 file <путь к файлу на TFTP-сервере> remote subnet 172.18.0.0 mask 255.255.255.0
```

8 Запустите DHCP-сервер:

```
hostname# inet dhcp server start
```

9 Включите автоматический запуск DHCP-сервера при загрузке ViPNet Coordinator HW:

```
hostname# inet dhcp server mode on
```

На HW-2 настройте DHCP-relay:

1 Задайте eth0 как внешний интерфейс DHCP-relay:

```
hostname# inet dhcp relay add external-interface eth0 server 192.168.0.1
```

2 Задайте eth1 как слушающий интерфейс для защищенных узлов:

```
hostname# inet dhcp relay add listen-interface eth1
```

3 Задайте eth2 как слушающий интерфейс для туннелируемых узлов:

```
hostname# inet dhcp relay add listen-interface eth2
```

4 Запустите DHCP-relay:

```
hostname# inet dhcp relay start
```

5 Включите автоматический запуск DHCP-relay при загрузке ViPNet Coordinator HW:

```
hostname# inet dhcp relay mode on
```

На HW-3 настройте DHCP-relay:

1 Задайте eth0 для связи с внешним DHCP-сервером:

```
hostname# inet dhcp relay add external-interface eth0 server 192.168.0.1
```

2 Задайте eth1 как принимающий запросы от клиентов к DHCP-серверу:

```
hostname# inet dhcp relay add listen-interface eth1
```

3 Запустите DHCP-relay:

```
hostname# inet dhcp relay start
```

4 Включите автоматический запуск DHCP-relay при загрузке ViPNet Coordinator HW:

```
hostname# inet dhcp relay mode on
```

Использование сторонних DHCP-серверов

Рассмотрим пример:

- В сети 1 находятся два туннелируемых HW-1 сторонних DHCP-сервера:
 - DHCP-сервер 1 (89.135.15.2).
 - DHCP-сервер 2 (210.11.0.2). DHCP-сервер 2 также выступает в роли VoIP-сервера для VoIP-телефонов в сети 1.
- Туннелируемые DHCP-серверы работают для следующих DHCP-клиентов сети 2:
 - DHCP-сервер 1 — для защищенных узлов (диапазон 192.168.0.100-200).
 - DHCP-сервер 2 — для туннелируемых VoIP-телефонов (диапазон 172.16.0.10-20). DHCP-сервер 2 также выступает в роли VoIP-сервера для VoIP-телефонов в сети 2.

- В сети 2 HW-2 работает как агент DHCP-relay для клиентов своей сети. На нем запущены две копии DHCP-relay, принимающие запросы от защищенных и туннелируемых узлов.
- Между координаторами настроена видимость по реальным IP-адресам.

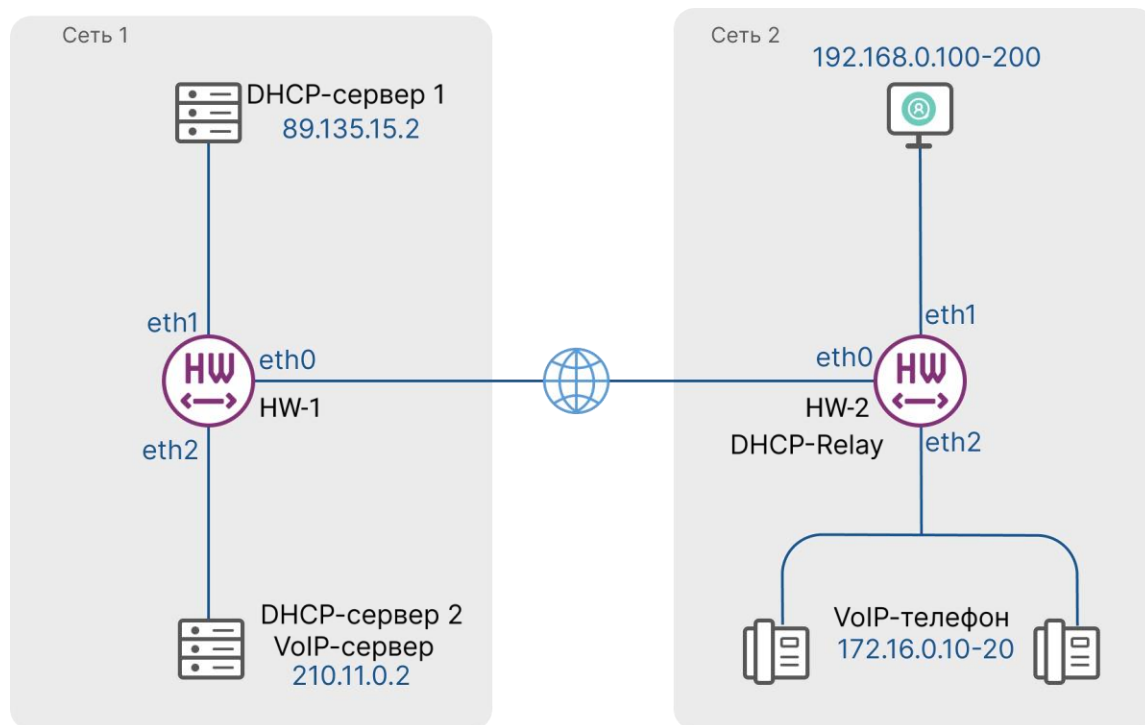


Рисунок 34. Использование сторонних DHCP-серверов

На HW-2:

1 Настройте DHCP-relay для защищенных узлов:

1.1 Задайте eth0 как внешний интерфейс DHCP-relay для защищенных узлов:

```
hostname# inet dhcp relay 1 add external-interface eth0 server 89.135.15.2
```

1.2 Задайте eth1 как слушающий интерфейс для защищенных узлов:

```
hostname# inet dhcp relay 1 add listen-interface eth1
```

2 Настройте DHCP-relay для туннелируемых узлов:

2.1 Задайте eth0 как внешний интерфейс DHCP-relay для туннелируемых узлов:

```
hostname# inet dhcp relay 2 add external-interface eth0 server 210.11.0.2
```

2.2 Задайте eth2 как слушающий интерфейс для туннелируемых узлов:

```
hostname# inet dhcp relay 2 add listen-interface eth2
```

3 Запустите DHCP-relay (обе копии):

```
hostname# inet dhcp relay 1 start
```

```
hostname# inet dhcp relay 2 start
```

4 Включите автоматический запуск DHCP-relay при загрузке ViPNet Coordinator HW:

```
hostname# inet dhcp relay mode on
```


Одновременное использование DHCP-сервера и DHCP-relay

Рассмотрим пример:

- В сети 1 находится туннелируемый HW-1 DHCP-сервер.
- В сети 2 находится HW-2, который работает как агент DHCP-relay для своих туннелируемых узлов и как DHCP-сервер для защищенных узлов.
- Между HW-2 и туннелируемым DHCP-сервером настроена видимость по реальным IP-адресам.

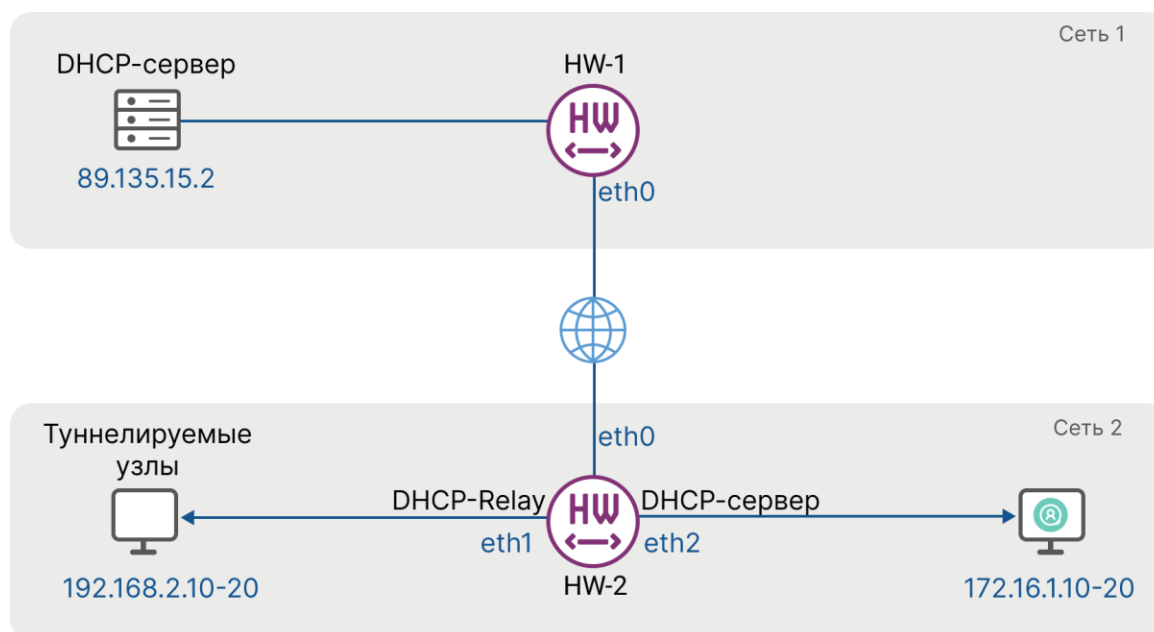


Рисунок 35. Одновременное использование DHCP-сервера и DHCP-relay

На HW-2:

- 1 Настройте DHCP-relay, принимающий данные от DHCP-сервера, стоящего за HW-1:
 - 1.1 Задайте eth0 как внешний интерфейс DHCP-relay:

```
hostname# inet dhcp relay add external-interface eth0 server 89.135.15.2
```
 - 1.2 Задайте eth1 как слушающий интерфейс для туннелируемых узлов:

```
hostname# inet dhcp relay add listen-interface eth1
```
 - 1.3 Запустите DHCP-relay:

```
hostname# inet dhcp relay start
```
 - 1.4 Включите автоматический запуск DHCP-relay при загрузке ViPNet Coordinator HW:

```
hostname# inet dhcp relay mode on
```
- 2 Настройте DHCP-сервер:
 - 2.1 Задайте рабочий сетевой интерфейс DHCP-сервера:

```
hostname# inet dhcp server add interface eth2
```

2.2 Задайте диапазон IP-адресов, выделяемых DHCP-сервером для защищенных узлов:

```
hostname# inet dhcp server add range 172.16.0.10 172.16.0.20 interface eth2
```

2.3 Запустите DHCP-сервер:

```
hostname# inet dhcp server start
```

2.4 Включите автоматический запуск DHCP-сервера при загрузке ViPNet Coordinator HW:

```
hostname# inet dhcp server mode on
```

Использование запасного DHCP-сервера

В этом примере рассматриваются две сети ViPNet, где HW-1 туннелирует два сторонних DHCP-сервера, один из которых является запасным, а HW-2 работает как агент DHCP-relay для клиентов своей сети:

- Между координаторами настроена видимость по реальным IP-адресам.
- В сети 1 туннелируемые DHCP-серверы, подключенные к HW-1, работают для защищенных узлов сети 1 (диапазон 192.168.0.10-20).
- На HW-2 для DHCP-relay настроен слушающий интерфейс, принимающий запросы от защищенных узлов, а также внешний интерфейс, настроенный на соединение как с основным, так и с запасным DHCP-серверами.

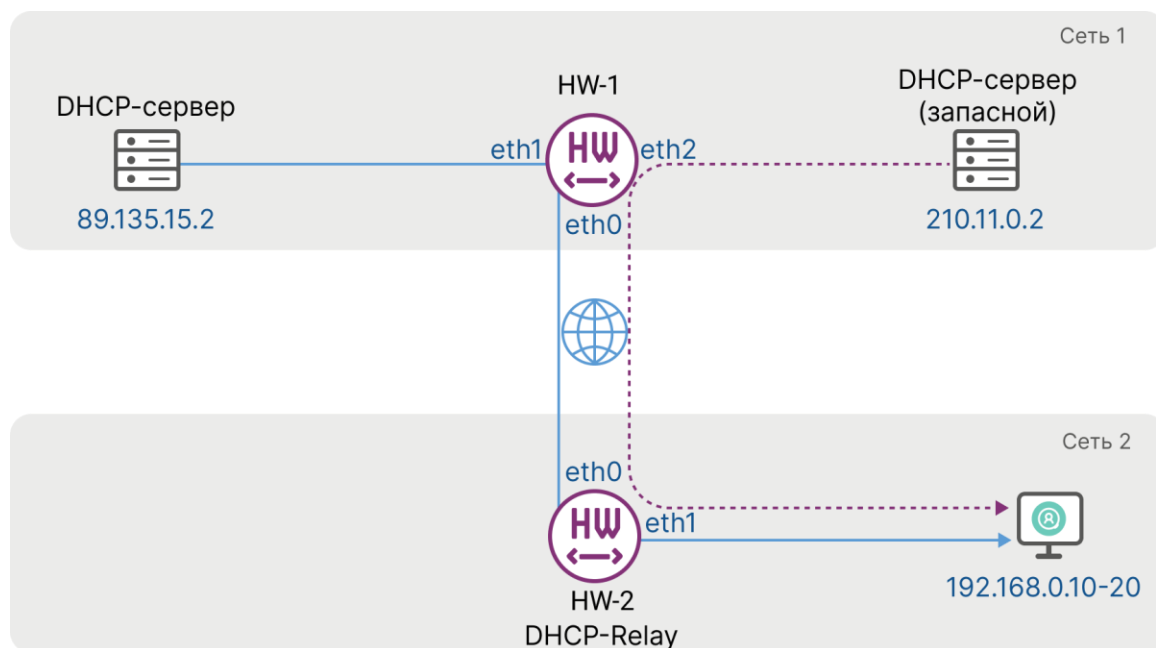


Рисунок 36. Использование DHCP-relay для запасного DHCP-сервера

На HW-2:

1 Задайте eth0 как внешний интерфейс DHCP-relay:

```
hostname# inet dhcp relay add external-interface eth0 server 89.135.15.2
```

2 Задайте eth0 как интерфейс DHCP-relay, подключенный к запасному DHCP-серверу:

```
hostname# inet dhcp relay add backup-interface eth0 server 210.11.0.2
```

3 Задайте eth1 как слушающий интерфейс для защищенных узлов:

```
hostname# inet dhcp relay add listen-interface eth1
```

4 Запустите DHCP-relay:

```
hostname# inet dhcp relay start
```

5 Включите автоматический запуск DHCP-relay при загрузке ViPNet Coordinator HW:

```
hostname# inet dhcp relay mode on
```

Настройка DNS-сервера

В состав ViPNet Coordinator HW входит локальный DNS-сервер, который преобразует доменные имена в IP-адреса в ответ на собственные запросы и на запросы других сетевых узлов (DNS-клиентов).

Локальный DNS-сервер перенаправляет поступающие к нему DNS-запросы на вышестоящие DNS-серверы и передает полученные ответы DNS-клиентам. По умолчанию локальный DNS-сервер преобразует доменные имена с использованием корневых DNS-серверов из домена `root-servers.net`. Этот список можно дополнить DNS-серверами пересылки (forwarders).



Внимание! В качестве DNS-серверов пересылки используйте доверенные DNS-серверы из сети ViPNet. Ответы на запросы DNS-клиентов, приходящие не от доверенных DNS-серверов, могут быть скомпрометированы.

Список дополнительных DNS-серверов:

- Можно получить от внешнего DHCP-сервера.

На сетевом интерфейсе должен быть установлен режим автоматического получения параметров от DHCP-сервера, а DHCP-сервер должен отправлять информацию о DNS-серверах.

Если режим DHCP установлен на нескольких интерфейсах ViPNet Coordinator HW, то результат обработки собственных DNS-запросов и запросов DNS-клиентов не определен, так как неизвестно, какой интерфейс будет включен первым и какой список DNS-серверов пересылки получит ViPNet Coordinator HW.

Настраивайте получение информации о DNS-серверах только на одном интерфейсе. Данные об адресах DNS-серверов поступают автоматически, откажитесь от их получения:

```
hostname# inet ifconfig <интерфейс> dhcp dns off
```

- Задать вручную.



Внимание! Если DNS-серверы пересылки недоступны, но при этом доступны корневые DNS-серверы, DNS-клиенты будут получать ответы на свои запросы с задержкой.

Команды для работы с локальным DNS-сервером

- Указать IP-адреса узлов и подсетей, узлам которых разрешены запросы к DNS-серверу:

```
hostname# inet dns clients add {<IP-адрес> | <IP-адрес/длина маски>}
```

По умолчанию DNS-запросы к серверу разрешены для любых узлов.

- Посмотреть список IP-адресов, которым разрешены DNS-запросы к серверу:

```
hostname# inet dns clients list
```

- Удалить IP-адреса из списка адресов, которым разрешены DNS-запросы к серверу:

```
hostname# inet dns clients delete {<IP-адрес> | <IP-адрес/длина маски>}
```

- Добавить IP-адрес DNS-сервера, на который ViPNet Coordinator HW будет перенаправлять DNS-запросы:

```
hostname# inet dns forwarders add <IP-адрес>
```

Если необходимо перенаправлять DNS-запросы для определенных зон, то добавьте для каждой такой зоны IP-адрес DNS-сервера пересылки с указанием параметра <for-zone>:

```
hostname# inet dns forwarders add <IP-адрес> [for-zone <имя зоны>]
```

- Посмотреть список DNS-серверов пересылки:

```
hostname# inet dns forwarders list
```

- Удалить DNS-сервер из списка:

```
hostname# inet dns forwarders delete <IP-адрес>
```

Для удаления из списка DNS-сервера пересылки, заданного для определенной зоны, укажите параметр <for-zone>:

```
hostname# inet dns forwarders delete <IP-адрес> [for-zone <имя зоны>]
```

Примечание. При добавлении двух и более DNS-серверов пересылки, серверы, добавленные без указания имени зоны, имеют приоритет и будут использоваться для перенаправления всех DNS-запросов вне зависимости от наличия серверов пересылки для отдельных зон. При этом появится сообщение:



Forward DNS address <адрес> resolves both named zone and all requests

При изменении конфигурации внешних DNS-серверов перезапустите локальный DNS-сервер:

```
hostname# inet dns stop
```

```
hostname# inet dns start
```

- Включить или выключить автоматический запуск DNS-сервера при загрузке ViPNet Coordinator HW:

```
hostname# inet dns mode {on | off}
```

Текущее состояние задано при установке справочников и ключей.

- Запустить или остановить DNS-сервер:

```
hostname# inet dns {start | stop}
```

- Посмотреть текущие параметры DNS-сервера:

```
hostname# inet show dns
```

Настройка NTP-сервера

В состав ViPNet Coordinator HW входит локальный NTP-сервер. Он используется для синхронизации времени на ViPNet Coordinator HW и на других сетевых узлах (NTP-клиентах).

По умолчанию локальный NTP-сервер синхронизирует время с публичными NTP-серверами из кластера `vniiftri.ru` при подключении к интернету.

Этот кластер серверов можно дополнить другими NTP-серверами (публичными или корпоративными). Это удобно, когда нет интернета или есть более предпочтительный NTP-сервер.



Примечание. При настройке исполнения ViPNet Coordinator VA, не устанавливайте одновременно синхронизацию времени на виртуальной платформе и ViPNet Coordinator VA.

Список дополнительных NTP-серверов:

- Можно получить от внешнего DHCP-сервера.

На сетевом интерфейсе должен быть установлен режим автоматического получения параметров от DHCP-сервера и DHCP-сервер должен отправлять информацию о NTP-серверах.

Если режим DHCP установлен на нескольких сетевых интерфейсах ViPNet Coordinator HW, то результат синхронизации времени на ViPNet Coordinator HW и на NTP-клиентах не определен, так как неизвестно, какой сетевой интерфейс будет включен первым и какой список дополнительных NTP-серверов получит ViPNet Coordinator HW.

Настраивайте получение адресов NTP-серверов только на одном интерфейсе. Данные об адресах NTP-серверов поступают автоматически, вы можете отказаться от их получения:

```
hostname# inet ifconfig <интерфейс> dhcp ntp off
```

- Сформировать вручную.

Список, полученный от DHCP-сервера, также можно редактировать вручную.

Дополнительный NTP-сервер может находиться в защищенной сети или за ее пределами.

- Если дополнительный NTP-сервер — защищенный узел, видимый по реальному адресу, то в файле `iplir.conf` в секции `[id]` для этого защищенного узла установите параметр `visibility` в значение `real`. Описание секции и параметра см. в документе «Справочник команд и конфигурационных файлов», в разделе «Файл `iplir.conf`».
- Если дополнительный NTP-сервер находится за пределами сети ViPNet, добавьте разрешающий фильтр открытой сети для входящего трафика от NTP-сервера:

```
hostname# firewall local add src @any dst @local udp dport 123 pass
```



Внимание! Использование NTP-серверов, расположенных за пределами сети ViPNet не безопасно.

Команды для работы с локальным NTP-сервером

- Добавить адрес дополнительного NTP-сервера:

```
hostname# inet ntp add {server | peer} {<IP-адрес> | <доменное имя>}, где
```

- `server` — если NTP-сервер только рассылает данные о времени.
- `peer` — если NTP-сервер рассылает и получает данные о времени.

- Отключить использование NTP-серверов, заданных по умолчанию, и использовать только дополнительные NTP-серверы:

```
hostname# inet ntp use-default off
```



Примечание. При отключении использования NTP-серверов по умолчанию добавьте не менее четырех дополнительных NTP-серверов. Иначе, если один из серверов выйдет из строя, у ViPNet Coordinator HW не будет достаточно информации, чтобы определить, какой из оставшихся серверов предоставляет правильное время.

- Удалить адрес дополнительного NTP-сервера:

```
hostname# inet ntp delete {server | peer} {<IP-адрес> | <доменное имя>}
```

- Посмотреть список NTP-серверов:

```
hostname# inet ntp list
```

- Включить автоматический запуск локального NTP-сервера при загрузке ViPNet Coordinator HW:

```
hostname# inet ntp mode on
```

Текущее состояние задано при установке справочников и ключей.

- Разрешить переход в изолированный (orphan) режим при потере соединения с внешними NTP-серверами:

```
hostname# inet ntp orphan on <stratum>, где
```

`<stratum>` — уровень внешнего NTP-сервера (от 1 до 10). Если со всеми внешними NTP-серверами меньше указанного уровня не удастся установить соединение в течение 5 минут, то локальный NTP-сервер переходит в изолированный режим. По умолчанию значение `<stratum>` равно 5. Значение `stratum` задается в настройках NTP-сервера.



Внимание! Разрешите переход локального NTP-сервера в изолированный режим, если ViPNet Coordinator HW используется в качестве NTP-сервера для внутренней сети. Иначе клиенты ViPNet Coordinator HW не будут получать информацию о точном времени при потере соединения ViPNet Coordinator HW с внешними NTP-серверами, так как в этом случае локальный NTP-сервер будет считаться недостоверным.

- Запустить или остановить NTP-сервер:

```
hostname# inet ntp {start | stop}
```



Внимание! NTP-серверы, указанные по доменному имени, будут доступны, только если запущен DNS-сервер.

Информация о попытках подключения к NTP-серверам записывается в системный журнал.

- Посмотреть текущие параметры и состояние локального NTP-сервера:

```
hostname# inet show ntp
```



Внимание! Если время на вышестоящем NTP-сервере будет изменено более чем на 1000 секунд вперед или назад, то это может привести к аварийному завершению работы локального NTP-сервера. Запустите NTP-сервер:

```
hostname# inet ntp start
```

Организация работы клиентов удаленного офиса с корпоративными DNS- и NTP-серверами

Рассмотрим пример, когда у организации есть несколько офисов, при этом в центральном офисе установлены корпоративные DNS- и NTP-серверы. Требуется, чтобы к этим серверам могли обращаться клиенты, находящиеся в удаленном офисе:

- 1 В сетевых настройках клиентов задайте IP-адрес ViPNet Coordinator HW в качестве DNS- и NTP-серверов. Клиенты удаленного офиса будут направлять запросы на разрешение доменных имен и на синхронизацию времени на ViPNet Coordinator HW.

- 2 На ViPNet Coordinator HW настройте обращение к корпоративному DNS-серверу:

- 2.1 Проверьте текущий список DNS-серверов пересылки:

```
hostname# inet dns forwarders list
```

По умолчанию DNS-запросы перенаправляются на корневые DNS-серверы из домена `root-servers.net`.

- 2.2 Добавьте IP-адрес корпоративного DNS-сервера:

```
hostname# inet dns forwarders add <IP-адрес>
```

- 2.3 Проверьте текущие параметры DNS-сервера:

```
hostname# inet show dns
```

- 2.4 Запустите DNS-сервер:

```
hostname# inet dns start
```

- 2.5 Включите автоматический запуск DNS-сервера при загрузке ViPNet Coordinator HW:

```
hostname# inet dns mode on
```

- 3 На ViPNet Coordinator HW настройте обращение к корпоративному NTP-серверу:

- 3.1 Проверьте текущий список NTP-серверов:

```
hostname# inet ntp list
```

- 3.2 Добавьте IP-адрес корпоративного NTP-сервера:

```
hostname# inet ntp add server {<IP-адрес> | <DNS-имя>}
```

- 3.3 Проверьте текущие параметры NTP-сервера:

```
hostname# inet show ntp
```

- 3.4 Запустите NTP-сервер:

```
hostname# inet ntp start
```

3.5 Включите автоматический запуск NTP-сервера при загрузке ViPNet Coordinator HW:

```
hostname# inet ntp mode on
```

Настройка точки доступа сети Wi-Fi



Примечание. Использование ViPNet Coordinator HW в качестве точки доступа Wi-Fi возможно только в исполнениях со встроенными адаптерами Wi-Fi: ViPNet Coordinator HW50 A, B на аппаратной платформе HW50 N2 и ViPNet Coordinator HW100 C на аппаратной платформе HW100 N2.

Если ViPNet Coordinator HW работает в режиме точки доступа, то он не сможет подключаться к другим сетям Wi-Fi.

Чтобы использовать ViPNet Coordinator HW в качестве точки доступа Wi-Fi:

- 1 Отключите сетевой интерфейс `wlan0`:

```
hostname# inet wifi mode off
```

- 2 Переключите `wlan0` в режим точки доступа:

```
hostname# inet wifi role access-point
```

- 3 Задайте способ аутентификации пользователей и имя вашей сети Wi-Fi:

- Аутентификация не требуется:

```
hostname# inet wifi access-point authentication open
```

Затем введите имя сети.

- Для аутентификации требуется использовать пароль. Вы можете выбрать режим аутентификации WPA-PSK или WPA2-PSK:

```
hostname# inet wifi access-point authentication {wpa-psk | wpa2-psk}
```

Затем введите имя сети и пароль. Заданный пароль сообщите пользователям вашей сети Wi-Fi.

- 4 Точка доступа поддерживает стандарты:

- IEEE 802.11b — 2,4 ГГц, скорость соединения до 11 Мбит/с.
- IEEE 802.11g — 2,4 ГГц, скорость соединения до 54 Мбит/с (используется по умолчанию).

Вы можете изменить стандарт связи:

```
hostname# inet wifi access-point hwmode {b | g}
```

- 5 Задайте номер используемого канала Wi-Fi:

```
hostname# inet wifi access-point channel <номер канала>
```

По умолчанию используется канал номер 1. Допустимые номера: 1–11.

- 6 Назначьте сетевому интерфейсу `wlan0` статический IP-адрес:

```
hostname# inet ifconfig wlan0 address <IP-адрес> netmask <маска подсети>
```

- 7 Включите сетевой интерфейс `wlan0`:

```
hostname# inet wifi mode on
```

- 8 Для связи между устройствами, которые подключены к сети Wi-Fi, и устройствами, подключенными к сети Ethernet, создайте [транзитные фильтры открытой сети](#), пропускающие [IP-пакеты между этими сетями](#).
- 9 Настройте параметры DHCP-сервера для интерфейса `wlan0` (см. [Настройка DHCP-сервера](#)).

10

Настройка маршрутизации

Общие сведения о маршрутизации	194
Принципы маршрутизации	195
Просмотр таблиц маршрутизации	199
Настройка статической маршрутизации	201
Настройка параметров маршрутов от DHCP/PPP-протокола	205
Настройка динамической маршрутизации	210
Настройка функции MultiWAN	225

Общие сведения о маршрутизации

ViPNet Coordinator HW поддерживает функцию маршрутизации IP-трафика в сетях со сложной структурой.

Маршрутизация — процесс выбора [маршрута следования IP-пакета](#), передаваемого в сети от одного узла другому. Он выбирается из подмножества маршрутов, заданных на ViPNet Coordinator HW в таблице маршрутизации.

Маршруты могут быть:

- Статические — добавляются в таблицу маршрутизации вручную. Удобно использовать в небольших сетях.
- Динамические — попадают в таблицу маршрутизации с помощью алгоритмов на основе информации о топологии и состоянии сети, предоставляемой протоколами маршрутизации. Удобно использовать в сетях с разветвлённой и неоднородной топологией. Все маршрутизаторы в сети должны использовать определённый протокол динамической маршрутизации, по которому они будут обмениваться информацией о доступных им сетях, автоматически строить доступные маршруты в каждую сеть и выбирать из них наилучшие.

Динамическая маршрутизация кроме автоматического формирования таблиц маршрутизации позволяет:

- Автоматически выбирать по определённым критериям лучший маршрут из нескольких доступных. В том числе, в случае сбоев других маршрутизаторов в сети.
- Организовать динамическую балансировку нагрузки передаваемого IP-трафика.

ViPNet Coordinator HW может маршрутизировать IP-трафик с использованием следующих видов маршрутов:

- статических, в том числе на основе маршрутов с несколькими шлюзами;
- полученных по протоколу DHCP и PPP;
- формируемых протоколом [OSPF](#).

Принципы маршрутизации

В ViPNet Coordinator HW для хранения маршрутов используются:

- Таблица маршрутизации по умолчанию (Main) — содержит:
 - статические маршруты, заданные вручную (Static);
 - маршруты в подсети, к которым подключены сетевые интерфейсы (Connected);
 - маршруты, полученные по протоколам DHCP, PPP, OSPF.
- Пользовательские таблицы маршрутизации — могут содержать только [статические маршруты, заданные вручную](#), и использоваться для создания политик маршрутизации (см. [Настройка функции MultiWAN](#)).

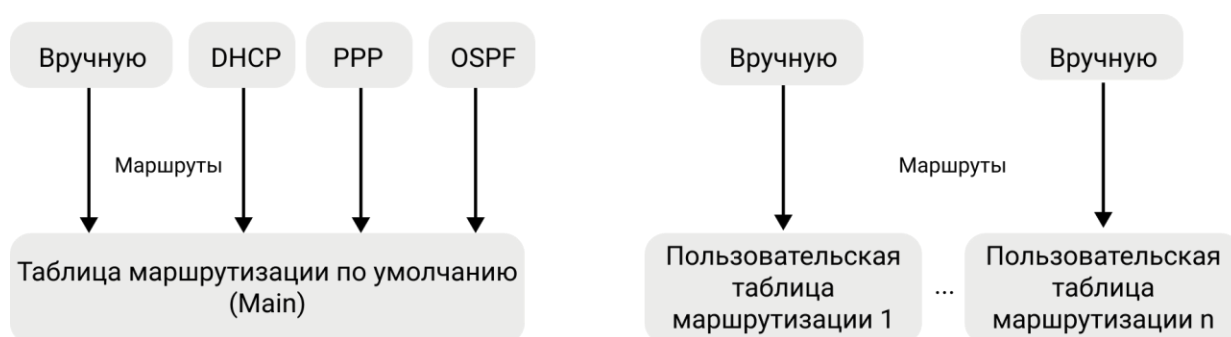


Рисунок 37. Формирование таблиц маршрутизации

В маршрутизации трафика участвуют не все маршруты, содержащиеся в таблице маршрутизации, а только лучшие маршруты в каждую сеть (подробнее см. «Формирование списка лучших маршрутов» ниже).

Характеристики маршрутов

- Источник получения маршрута (Connected, Static, PPP, DHCP, OSPF) — определяет, каким образом был получен маршрут.
- [Административная дистанция](#) — определяет приоритет маршрута от каждого источника. Учитывается, когда в таблице маршрутизации есть несколько маршрутов в одну и ту же сеть:
 - Чем меньше административная дистанция, тем более приоритетным считается маршрут.
 - Административная дистанция задаётся для каждого статического маршрута.
 - Для протокола PPP административная дистанция отдельно не задаётся и равна дистанции для протокола DHCP.
 - Административная дистанция статических маршрутов не может совпадать с дистанцией, заданной для маршрутов протокола OSPF.

Примечание. В ViPNet Coordinator HW для каждого типа маршрутов задаётся по умолчанию следующая административная дистанция:



- для статических маршрутов — 10;
- для маршрутов протоколов DHCP и PPP — 70;
- для маршрутов протокола OSPF — 110.

Для первых двух типов маршрутов вы можете менять значение административной дистанции, для маршрутов последнего типа менять это значение нельзя.

- **Метрика** — определяет приоритет маршрутов протоколов DHCP, PPP и OSPF. Чем меньше метрика, тем выше приоритет маршрута.
 - Для маршрутов протокола DHCP метрики указываются вручную, причем на каждом сетевом интерфейсе, на котором включен режим DHCP и настроен параметр получения маршрутов от DHCP-сервера. Таким образом, если на разные сетевые интерфейсы будут получены одинаковые DHCP-маршруты в одну и ту же сеть, то будет выбран маршрут с наименьшей метрикой.

Для PPP-протокола также может быть задана метрика вручную в том случае, если ViPNet Coordinator HW подключен к нескольким сетям, одной из которых является сеть 3G, а в другой развёрнут DHCP-сервер. Если метрики для DHCP/PPP-протоколов не были заданы, то используется метрика по умолчанию.

Для динамических маршрутов, формируемых протоколом OSPF, метрики формируются самим протоколом, поэтому их задавать не требуется.

- **Вес** — определяет долю IP-трафика, который будет проходить по маршруту через указанный шлюз. Используется только в статических маршрутах и задаётся для шлюза. Позволяет настроить балансировку передаваемого IP-трафика между несколькими шлюзами, когда часть IP-пакетов направляется на один шлюз, а часть — на другой (см. [Настройка балансировки IP-трафика](#)). Поэтому вес задаётся при создании нескольких статических маршрутов с одинаковым IP-адресом назначения и разными шлюзами.

Политики маршрутизации

Выбор, по какой таблице маршрутизировать трафик, определяется активной политикой маршрутизации. В качестве активной политики вы можете выбрать:

- Политику маршрутизации по умолчанию — выбрана автоматически после инициализации ViPNet Coordinator HW. Политика использует маршруты из таблицы маршрутизации по умолчанию (Main). Настройка политики недоступна.
- Одну из пользовательских политик маршрутизации — создаются пользователем. С помощью таких политик вы можете разделить IP-трафик по признакам и маршрутизировать его по разным таблицам маршрутизации (см. [Настройка функции MultiWAN](#)).

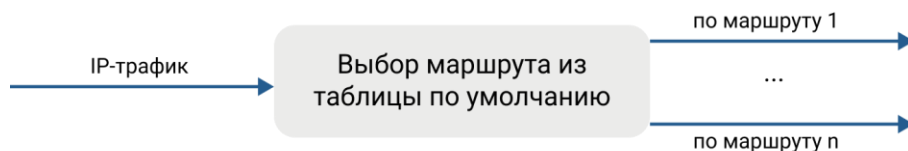


Рисунок 38. Маршрутизация при политике по умолчанию

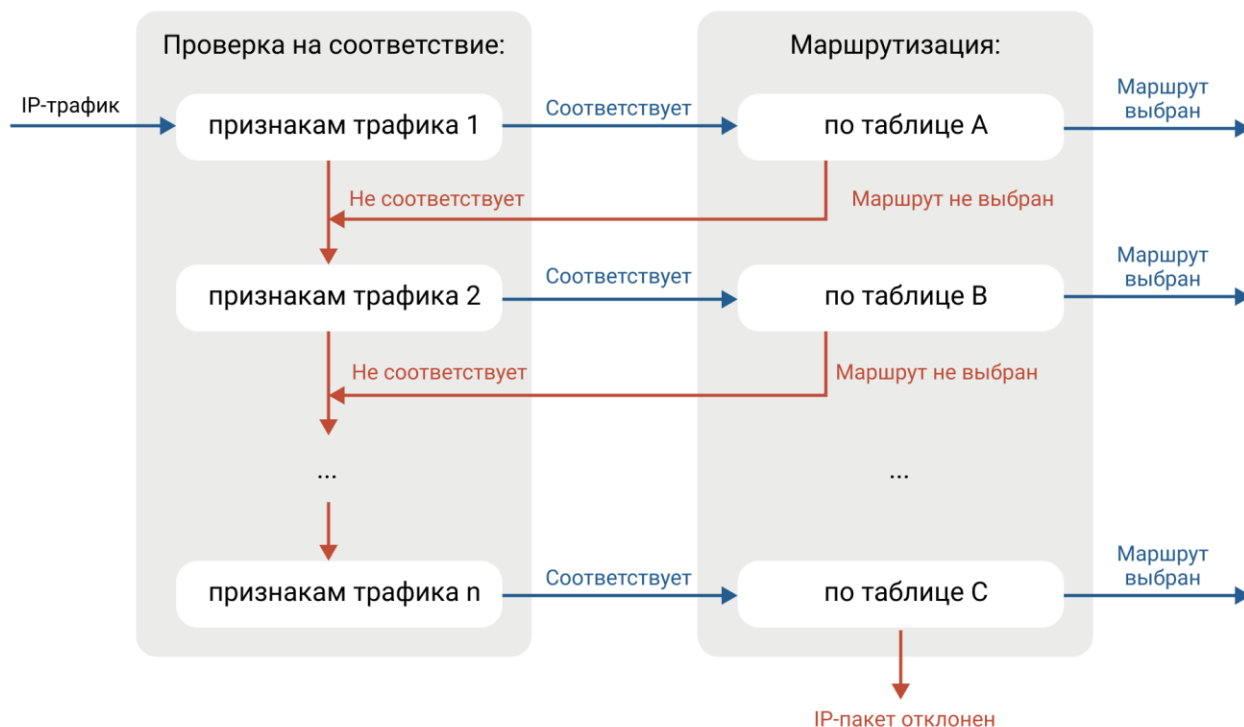


Рисунок 39. Маршрутизация при пользовательской политике

Пользовательская политика маршрутизации состоит из правил маршрутизации. Для каждого правила задаются:

- Приоритет — определяет в каком порядке правила будут применяться к трафику.
- Признаки трафика — задают условия, которым должен соответствовать трафик. Если признаки не указаны, правило распространяется на весь трафик.
- Таблица маршрутизации, по которой должен обрабатываться трафик, соответствующий заданным признакам трафика. Можно указать пользовательскую таблицу или таблицу по умолчанию.

Принципы маршрутизации при использовании пользовательской политики:

- IP-пакет последовательно проходит проверку правилами маршрутизации в соответствии с их приоритетом.
- Если IP-пакет соответствует всем признакам, заданным в правиле, он будет обработан по указанной таблице маршрутизации.
- Если IP-пакет не соответствует хотя бы одному из признаков правила, он передаётся на проверку следующему по приоритету правилу маршрутизации.

Формирование списка лучших маршрутов

В маршрутизации участвуют только лучшие статические и динамические маршруты в каждую сеть. Остальные маршруты в таблице игнорируются. Лучшие маршруты определяются в каждой таблице маршрутизации по следующим правилам:

- Из нескольких маршрутов в одну сеть лучшим признаётся маршрут с меньшей административной дистанцией, даже если маршруты получены из разных источников (Static, DHCP или OSPF). Например, имеется два маршрута в сеть 10.0.5.0 с маской 255.255.255.0. Для первого маршрута задана административная дистанция — 10, для второго — 30. В результате в маршрутизации будет участвовать первый маршрут. Если дистанции маршрутов, полученных из одного источника, равны, будут участвовать оба маршрута.
- Из нескольких маршрутов с разными префиксами (например, 10.0.0.0/8 и 10.0.0.0/16) лучшими признаются все маршруты.

Выбор маршрута из списка лучших

В процессе маршрутизации для каждого IP-пакета выбирается маршрут из списка лучших маршрутов. При наличии нескольких маршрутов в нужную сеть выполняются правила:

- Из нескольких лучших динамических маршрутов в одну сеть будет выбран маршрут с наименьшей метрикой, заданной протоколом динамической маршрутизации. Если метрики равны, будет выбран один из маршрутов в соответствии со стратегией маршрутизации ESMR (Equal-cost multi-path routing).
- Из нескольких лучших маршрутов в одну сеть с разным префиксом (маской подсети), будет выбран маршрут с наиболее длинным префиксом.
- Маршруты от одного источника (Static, DHCP, OSPF) в одну и ту же сеть с одинаковыми метриками (или административными дистанциями в случае статических маршрутов) при маршрутизации объединяются в один маршрут с несколькими шлюзами (multi-hop route). При просмотре такие маршруты отображаются в виде одного маршрута с несколькими шлюзами:

```
10.100.2.0/24 [30/23] (weight 1) via 10.1.30.202, eth1
                (weight 1) via 10.1.31.201, eth2
```



Внимание! Если хотя бы один из шлюзов объединённого маршрута multi-hop route выйдет из строя, работоспособность маршрута не гарантируется.

Просмотр таблиц маршрутизации

Чтобы просмотреть список всех маршрутов в таблице маршрутизации по умолчанию (Main):

```
hostname> inet show routing
```

```
hw-va-16310045# inet show routing
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel, D - DHCP/PPP,
       > - selected route, * - FIB route

S> 0.0.0.0/0 [10/0] (weight 1) via 10.0.1.4 inactive
   *                (weight 1) via 10.0.2.1, eth2
   *                (weight 1) via 10.0.3.3, eth1
S>* 10.0.1.0/24 [10/0] via 10.0.2.1, eth2
O  10.0.2.0/24 [110/10] is directly connected, eth2, 01w0d20h
C>* 10.0.2.0/24 is directly connected, eth2
C>* 10.0.3.0/24 is directly connected, eth1
S>* 10.0.4.0/24 [10/0] via 10.0.3.3, eth1
O  10.0.5.0/24 [110/10] is directly connected, eth0, 01w0d20h
C>* 10.0.5.0/24 is directly connected, eth0
O  10.1.30.0/24 [110/20] via 10.0.5.5, eth0, 01w0d20h
S>* 10.1.30.0/24 [10/0] via 10.0.5.5, eth0
O>* 10.100.1.0/24 [110/20] via 10.0.5.5, eth0, 01w0d20h
O>* 10.100.2.0/24 [110/20] via 10.0.5.5, eth0, 01w0d20h
C>* 127.0.0.0/8 is directly connected, lo
S>* 192.168.0.0/16 [10/0] (weight 1) via 10.0.3.3, eth1
   *                (weight 1) via 10.0.2.1, eth2
hw-va-16310045#
```

Рисунок 40. Просмотр таблицы маршрутизации

Цифрами на рисунке обозначены:

- 1 Статический маршрут по умолчанию. Выбран как лучший. Объединяет три маршрута, поэтому включает в себя три шлюза. Каждому шлюзу назначен вес, поэтому в процессе маршрутизации будет осуществляться балансировка нагрузки.



Примечание. Если маршрут по умолчанию не задан, отображается соответствующее предупреждение. В этом случае [создайте статический маршрут по умолчанию](#).

- 2 Маршрут в подсеть 10.0.3.0 с маской 255.255.255.0, к которой подключен один из интерфейсов ViPNet Coordinator HW. Выбран как лучший.
- 3 Маршруты в подсети 10.100.1.0 и 10.100.2.0 с маской 255.255.255.0, полученные по протоколу OSPF. Выбраны как лучшие.
- 4 Статический маршрут в подсеть 192.168.0.0 с маской 255.255.0.0. Выбран как лучший. Объединяет два маршрута, поэтому включает в себя два шлюза. Каждому шлюзу назначен вес, поэтому в процессе маршрутизации будет осуществляться балансировка нагрузки.

Каждый раз перед списком маршрутов выводится список возможных атрибутов, которые могут присваиваться для маршрутов, и их расшифровка. В таблице ниже приведены пояснения по данным атрибутам.

Таблица 5. Атрибуты маршрутов

Атрибут	Расшифровка в командном интерпретаторе	Пояснение
C	connected	Маршрут в подсеть, к которой подключен один из интерфейсов ViPNet Coordinator HW.
S	static	Статический маршрут.
O	OSPF	Маршрут протокола OSPF.
D	DHCP/PPP	Маршрут, предоставленный DHCP/PPP/PPPoE-сервером.
>	selected route	Лучший маршрут. Отображается, если в таблице есть несколько маршрутов в одну и ту же сеть с разной административной дистанцией .
*	FIB route	Маршрут, признанный лучшим и участвующий в маршрутизации.



Примечание. В таблице маршрутизации могут присутствовать другие атрибуты, кроме описанных в таблице выше. Эти атрибуты зарезервированы для других протоколов маршрутизации и в текущей версии ViPNet Coordinator HW не используются.

Чтобы просмотреть определённые типы маршрутов из таблицы маршрутизации по умолчанию:

```
hostname> inet show routing {static table default | dhcp | ospf}
```

где:

- `static table default` — просмотр статических маршрутов;
- `dhcp` — просмотр маршрутов протоколов DHCP и PPP;
- `ospf` — просмотр маршрутов протокола OSPF.

Чтобы просмотреть содержимое пользовательской таблицы маршрутизации:

```
hostname> inet show routing static table {<имя таблицы> | <номер таблицы>}
```

где:

- `<имя таблицы>` — имя пользовательской таблицы маршрутизации;
- `<номер таблицы>` — номер пользовательской таблицы маршрутизации.

Настройка статической маршрутизации

Если ViPNet Coordinator HW функционирует в сети, конфигурация которой не меняется или в которой используется минимальное количество путей для доставки IP-пакетов, то вы можете настроить маршрутизацию на ViPNet Coordinator HW с помощью статических маршрутов. Статическая маршрутизация может потребоваться, когда нет возможности использовать маршрутизаторы, работающие по протоколам динамической маршрутизации, или в следующих случаях:

- Чтобы направлять часть IP-трафика по конкретным маршрутам (например, есть требование направлять IP-трафик по самому надежному каналу связи) или через конкретный шлюз.
- Нужен статический маршрут по умолчанию, если работа по протоколам динамической маршрутизации невозможна.
- Чтобы распределять передачу IP-трафика по нескольким маршрутам.



Внимание! В текущей версии ViPNet Coordinator HW не поддерживается асимметричная маршрутизация. Убедитесь, что пакеты проходят от источника до пункта назначения и обратно через координатор. При необходимости добавьте необходимые маршруты до узла или сети назначения или настройте динамическую трансляцию адресов.

Создание статических маршрутов

Вы можете добавить статические маршруты в таблицу маршрутизации по умолчанию или создать пользовательские таблицы маршрутизации. Пользовательские таблицы используются для создания политик маршрутизации.

- Чтобы добавить **маршрут по умолчанию** в таблицу маршрутизации по умолчанию:

```
hostname# inet route add default next-hop <IP-адрес шлюза> [distance <1-255> [weight <1-255>]]
```

- Чтобы добавить статический маршрут в таблицу маршрутизации по умолчанию:

```
hostname# inet route add <IP-адрес назначения> [netmask <маска>] next-hop <IP-адрес шлюза> [distance <1-255> [weight <1-255>]]
```



Совет. Если вам требуется добавить один статический маршрут с несколькими шлюзами, создайте несколько статических маршрутов в одну и ту же сеть с одинаковой административной дистанцией, указав в каждом по одному шлюзу. В результате эти маршруты будут просуммированы и объединены в один с несколькими шлюзами. Как правило, такие маршруты требуются для балансировки нагрузки, поэтому в маршрутах также должен быть задан вес шлюзам (см. ниже).

- Чтобы добавить маршрут в пользовательскую таблицу маршрутизации (см. [Настройка функции MultiWAN](#)):

```
hostname# inet route add {<IP-адрес назначения> [netmask <маска>] | default} next-hop
<IP-адрес шлюза> table <номер таблицы> [name <имя таблицы>] [distance <1-255> [weight
<1-255>]]
```

Если указанной таблицы маршрутизации не существует, она будет создана с заданными именем и номером.

Во всех командах вы можете задать необязательные параметры: `distance` — административную дистанцию маршрута и `weight` — вес шлюза.

Административная дистанция задаётся, чтобы назначить приоритет маршруту. Если вы не укажете значение административной дистанции, то оно будет задано автоматически: `distance=10`. Если вы укажете административную дистанцию, и её значение будет совпадать со значением административной дистанции, заданной протоколам динамической маршрутизации, появится сообщение, что маршрут не может быть добавлен. В этом случае создайте маршрут с другим значением административной дистанции.

Вес задается в маршруте для шлюза. Его требуется задавать в том случае, если создаётся несколько маршрутов в одну сеть с разными шлюзами, между которыми требуется производить балансировку нагрузки. Подробнее см. раздел [Настройка балансировки IP-трафика](#).

Если маршрут, добавленный в таблицу маршрутизации по умолчанию, будет определён как лучший, то он будет участвовать в маршрутизации IP-трафика.

Для использования пользовательских таблиц маршрутизации создайте политику маршрутизации и активируйте её (см. [Настройка функции MultiWAN](#)).

Удаление статических маршрутов

Вы можете удалить маршрут по умолчанию или статический маршрут:

- Чтобы удалить [маршрут по умолчанию](#) из таблицы маршрутизации по умолчанию:

```
hostname# inet route delete default next-hop <IP-адрес шлюза>
```

- Чтобы удалить статический маршрут из таблицы маршрутизации по умолчанию:

```
hostname# inet route delete <IP-адрес назначения> [netmask <маска подсети>] next-hop
<IP-адрес шлюза>
```

- Чтобы удалить маршрут из пользовательской таблицы маршрутизации:

```
hostname# inet route delete {<IP-адрес назначения> [netmask <маска подсети>] | default}
next-hop <IP-адрес шлюза> table <номер таблицы> [name <имя таблицы>]
```

Если при вводе команды вы не указали маску подсети или IP-адрес шлюза, и в процессе проверки будет установлено, что в таблице маршрутизации есть несколько маршрутов в указанную сеть, то на экран будет выведен список этих маршрутов. Для удаления всех выбранных маршрутов введите `y` и нажмите **Enter**. Аналогичная ситуация возникнет при удалении маршрута с несколькими шлюзами.

После удаления маршрута будет выбран новый лучший маршрут в ту же сеть.



Примечание. Если требуется удалить все статические маршруты, включая все маршруты по умолчанию, воспользуйтесь командой:

```
hostname# inet route clear [table {<номер таблицы> | name <имя таблицы>}]
```

Настройка балансировки IP-трафика

Если вы хотите ускорить процесс отправки IP-пакетов в сеть назначения, вы можете создать несколько статических маршрутов в данную сеть через разные шлюзы и настроить балансировку IP-трафика между этими маршрутами. Это позволит в процессе разных TCP-соединений отправлять часть IP-пакетов по одному маршруту, часть — по другому, что повысит скорость передачи IP-пакетов в сеть назначения. Балансировка IP-трафика настраивается с помощью одного из параметров маршрутов — *веса* (*weight*). При этом все маршруты, которые будут участвовать в балансировке IP-трафика, должны иметь одинаковую административную дистанцию, то есть иметь одинаковый приоритет.



Примечание. При маршрутизации маршруты в одну сеть назначения и с одинаковой дистанцией объединяются в один маршрут с несколькими шлюзами (*multi-hop route*). При просмотре таблицы маршрутизации они отображаются как один маршрут с несколькими шлюзами.

Для настройки балансировки IP-трафика:

- 1 **Создайте нужное количество маршрутов** с одинаковым адресом назначения и с разными шлюзами.
- 2 **Задайте каждому маршруту одинаковую административную дистанцию.**
- 3 **Задайте в каждом маршруте вес шлюзу.** IP-трафик будет распределяться между шлюзами в соответствии с соотношением их весов. Например:
 - о если вы создаёте 2 маршрута и в каждом маршруте вес шлюза равен 1, то IP-трафик будет разделён между этими маршрутами поровну, то есть 50% IP-трафика будет передаваться по одному маршруту, 50% — по второму;
 - о если вы создаёте 3 маршрута, и в первом маршруте вес шлюза равен 1, во втором — вес шлюза равен 2, в третьем — вес шлюза равен 3, то по второму маршруту будет передаваться в два раза больше IP-трафика, чем по первому, по третьему — в три раза больше, чем по первому.

Пример команд для создания статических маршрутов с весом шлюзов равным 1:

```
hostname# inet route add 10.0.5.0 netmask 255.255.255.0 next-hop 10.0.1.1 distance 20 weight 1
```

```
hostname# inet route add 10.0.5.0 netmask 255.255.255.0 next-hop 10.0.4.3 distance 20 weight 1
```



Примечание. Если в процессе создания маршрута вы не укажете вес шлюза, то шлюзу автоматически будет назначен вес равный 1. Вы не можете задать вес равный 0. Балансировка IP-трафика будет осуществляться только если созданные маршруты признаны лучшими.

Просмотр статических маршрутов

Чтобы просмотреть список всех статических маршрутов, содержащихся в таблице маршрутизации по умолчанию и в пользовательских таблицах маршрутизации:

```
hostname> inet show routing static
```

В списке для каждого маршрута указаны: IP-адрес назначения, маска сети, шлюз, административная дистанция маршрута и вес шлюза.

Настройка параметров маршрутов от DHCP/PPP-протокола

Если на каком-либо сетевом интерфейсе ViPNet Coordinator HW включен режим DHCP и настроено автоматическое получение маршрутов (см. [Настройка сетевых интерфейсов Ethernet](#)), то в процессе маршрутизации будут использоваться эти маршруты. В этом случае настройте дополнительные параметры:

- 1 [Настройте административную дистанцию для протокола DHCP](#). Это позволит определить приоритет маршрутов DHCP-сервера среди всех маршрутов в таблице маршрутизации по умолчанию (статических маршрутов, маршрутов протокола OSPF, если такие есть).
- 2 Если режим DHCP включен на нескольких сетевых интерфейсах, [настройте метрики протокола DHCP](#) на каждом сетевом интерфейсе, на котором включен этот режим. Это позволит определить приоритет среди маршрутов DHCP-сервера в одну и ту же сеть, полученных с разных сетевых интерфейсов.

Если ViPNet Coordinator HW подключен, например, к сети 3G и получает информацию для маршрута по умолчанию от PPP-сервера провайдера (см. [Подключение к мобильной сети](#)), и при этом он также подключен к другой сети и получает в ней маршрутную информацию от DHCP-сервера, то в этом случае задайте метрику для PPP-протокола (см. [Настройка метрики для маршрутов PPP-протокола](#)). Это позволит определить, какой маршрут по умолчанию (сформированный на основе шлюза по умолчанию от PPP-сервера или предоставленный DHCP-сервером) является приоритетным. Административная дистанция для этого протокола не задается и равна дистанции для протокола DHCP.

Настройка административной дистанции для маршрутов DHCP-протокола

Если на ViPNet Coordinator HW настроено взаимодействие с DHCP-сервером, от которого поступают маршруты, а также настроена статическая маршрутизация (см. [Настройка статической маршрутизации](#)) или динамическая маршрутизация, то задайте административную дистанцию для маршрутов протокола DHCP. Административная дистанция определит приоритет данных маршрутов среди всех остальных, и в случае, если будет обнаружено несколько маршрутов в одну и ту же сеть из разных источников, будет выбран тот маршрут, у которого выше приоритет. Чем меньше значение [административной дистанции](#), тем выше приоритет маршрута.

В настройках ViPNet Coordinator HW вы можете:

- Задать административную дистанцию для маршрутов, поступающих от DHCP-сервера:

```
hostname# inet dhcp client route-distance <1-255>
```

Эта дистанция будет определять приоритет всех маршрутов DHCP-сервера, независимо от того, на какой сетевой интерфейс они поступили.

- Задать административную дистанцию для маршрутов по умолчанию вместе с общей административной дистанцией для всех маршрутов DHCP-сервера:

```
hostname# inet dhcp client route-distance <1-255> [default-route <1-255>],
```

где default-route <1-255> — значение административной дистанции для маршрутов по умолчанию.

Эта дистанция будет определять приоритет только маршрутов по умолчанию. Если дистанция не задана, то у маршрутов по умолчанию дистанция и приоритет будет таким же, как и у других маршрутов DHCP-сервера.

При необходимости перед или после настройки административной дистанции вы можете посмотреть, на каких сетевых интерфейсах включен режим DHCP и какая административная дистанция задана. Подробнее см. раздел [Просмотр настроек DHCP в режиме клиента](#).

Настройка метрики для маршрутов DHCP-протокола

Если на ViPNet Coordinator HW режим DHCP включен на нескольких сетевых интерфейсах, то может сложиться ситуация, когда с этих интерфейсов поступят разные маршруты от DHCP-сервера в одну и ту же сеть. В этом случае требуется определить, какой маршрут является лучшим, чтобы использовать его при маршрутизации трафика. Параметром, позволяющим выбрать наилучший маршрут, выступает метрика маршрута. Её можно задать на каждом сетевом интерфейсе, на котором включен режим DHCP, чтобы определить приоритет маршрутов DHCP-сервера на разных сетевых интерфейсах. Чем меньше значение метрики, тем выше приоритет маршрута. Если на разных сетевых интерфейсах заданы одинаковые метрики, то поступившие маршруты в одну и ту же сеть будут объединены в один маршрут с несколькими шлюзами.

В ViPNet Coordinator HW вы можете задать или удалить метрику для маршрутов DHCP-сервера на конкретном сетевом интерфейсе (далее — специфичная метрика). Если специфичная метрика не задана, то вместо неё используется метрика по умолчанию.

- Чтобы добавить специфичную метрику на сетевом интерфейсе:

```
hostname# inet ifconfig <имя интерфейса> dhcp route-metric <1-255>
```

Примечание. Задать специфичную метрику для протокола DHCP вы можете на сетевых интерфейсах следующих типов:



- Ethernet (см. [Настройка сетевых интерфейсов Ethernet](#)).
- VLAN.
- Wi-Fi (wlan) (см. [Подключение к сети Wi-Fi](#)).
- Агрегированный интерфейс.

Если режим DHCP включен только на одном сетевом интерфейсе, то настройка специфичной метрики не требуется, поскольку для группы маршрутов в рамках одного сетевого интерфейса она всегда будет одинаковой.

- Чтобы удалить метрику, заданную на сетевом интерфейсе:

```
hostname# inet ifconfig <имя интерфейса> dhcp route-metric none
```

После удаления метрики на этом интерфейсе будет использоваться метрика по умолчанию.

При необходимости перед или после настройки метрик вы можете посмотреть, на каких сетевых интерфейсах включен режим DHCP и какие метрики заданы. Подробнее см. раздел [Просмотр настроек DHCP в режиме клиента](#).

Настройка метрики для маршрутов PPP-протокола

Если ViPNet Coordinator HW подключен к мобильной сети (см. [Подключение к мобильной сети](#)), то при первом соединении с сервером провайдера он получает от него по протоколу PPP IP-адрес шлюза по умолчанию, на основе которого создается [маршрут по умолчанию](#).



Примечание. Шлюз по умолчанию поступает на ViPNet Coordinator HW от PPP-сервера только, если была выполнена команда `hostname# inet usb-modem set route on` (см. [Подключение к мобильной сети](#)).

Если кроме подключения к мобильной сети ViPNet Coordinator HW также подключен к другим сетям, в которых адреса и маршруты распространяется DHCP-сервером, то он также будет получать маршруты по умолчанию по протоколу DHCP. Если требуется, чтобы маршрут по умолчанию PPP-протокола имел приоритет выше, чем маршруты по умолчанию DHCP-протокола, и использовался в процессе маршрутизации, задайте метрику для протокола PPP (далее — специфичную метрику). Причем она должна быть меньше метрики DHCP-протокола на каждом сетевом интерфейсе, на которых они заданы. Если специфичная метрика не будет задана, то вместо нее будет использоваться метрика по умолчанию (см. [Изменение метрики по умолчанию для маршрутов протоколов DHCP и PPP](#)). В этом случае маршруты по умолчанию PPP- и DHCP-протокола будут иметь одинаковый приоритет.

- Чтобы добавить специфичную метрику для протокола PPP:

```
hostname# inet usb-modem set route-metric <1-255>
```

Примечание. Данную операцию вы можете выполнить только при следующих условиях:



- включено использование модема;
- разрешено применение маршрута по умолчанию от PPP-сервера провайдера.

При невыполнении данных условий появится сообщение, что метрика не может быть задана. О том, как включить использование модема и настроить применение маршрута по умолчанию, см. в разделе [Подключение к мобильной сети](#).

- Чтобы удалить специфичную метрику для протокола PPP:

```
hostname# inet usb-modem set route-metric none
```

После удаления метрики для протокола PPP будет использоваться метрика по умолчанию.

Изменение метрики по умолчанию для маршрутов протоколов DHCP и PPP

Если для маршрутов протоколов DHCP и PPP не задавались специфичные метрики, то для определения приоритета маршрутов будет использоваться метрика по умолчанию (первоначально — 70).

Чтобы изменить метрику:

```
hostname# inet dhcp client route-default-metric <1-255>
```

Просмотр настроек DHCP в режиме клиента

Вы можете просмотреть следующие параметры DHCP в режиме клиента:

- административная дистанция, которая задана для маршрутов, поступающих от DHCP-сервера (см. [Настройка административной дистанции для маршрутов DHCP-протокола](#));
- метрика по умолчанию (см. [Настройка метрики для маршрутов DHCP-протокола](#));
- сетевые интерфейсы, на которых включен режим DHCP;
- специфичные метрики на сетевых интерфейсах, если такие заданы;
- разрешение на автоматическое получение IP-адресов, адресов DNS- и NTP-серверов.

Чтобы посмотреть указанные настройки:

```
hostname> inet show dhcp client
```

В результате выполнения команды указанные настройки будут отражены в следующем виде:

```
hw50-2b3b0014# inet show dhcp client
Administrative distance for DHCP/PPP routes: 70
Default metric for DHCP/PPP routes: 70
```

Interface	DHCP	Routes	Metric	DNS	NTP
eth0	no	yes	default	yes	yes
eth1	no	yes	default	yes	yes
eth2	no	yes	default	yes	yes

```
hw50-2b3b0014#
```

Рисунок 41. Просмотр настроек DHCP-режима

Просмотр маршрутов DHCP/PPP-протокола

Чтобы просмотреть список маршрутов, полученных от DHCP/PPP-сервера:

```
hostname> inet show routing dhcp
```

```
hw-va-16310046# inet show routing dhcp
DHCP routes currently available: (* = RIB route)
>* 0.0.0.0/0 [35/23] via 10.1.30.5, eth1
>* 10.100.1.0/24 [30/23] via 10.1.30.201, eth1
>* 10.100.2.0/24 [30/23] via 10.1.30.202, eth1
*
via 10.1.30.202, eth1
hw-va-16310046#
```

Рисунок 42. Просмотр маршрутов DHCP-сервера

В списке для каждого маршрута указаны: атрибут, IP-адрес назначения, маска сети, шлюз и имя сетевого интерфейса, на который поступил маршрут. Если маршрут определён как лучший маршрут в сеть, перед ним указан атрибут «*».

Настройка динамической маршрутизации



Внимание! Исполнения HW10 и HW50 не поддерживают динамическую маршрутизацию.

Если в вашей сети поддерживается работа по протоколу [OSPF](#), то вы можете настроить на ViPNet Coordinator HW динамическую маршрутизацию. Данные настройки позволят автоматически создавать таблицы маршрутизации на основе маршрутов, формируемых по данному протоколу, и распространять их между другими маршрутизаторами в рамках одной сети.

Настройка динамической маршрутизации по протоколу OSPF имеет смысл в том случае, если в сети, в которой установлен ViPNet Coordinator HW, одновременно:

- используются другие OSPF-маршрутизаторы, и они напрямую связаны с ViPNet Coordinator HW;
- используется протокол OSPF версии 2;
- поддерживается многоадресное вещание (multicast);
- по требованиям безопасности вашей организации разрешена передача IP-трафика по протоколу OSPF между сетевыми узлами.



Внимание! Настройку динамической маршрутизации должен выполнять опытный администратор, понимающий принципы работы протоколов динамической маршрутизации, в том числе протокола OSPF. Приведенная в руководстве информация носит справочный характер.

Общие сведения о протоколе OSPF

Протокол OSPF — внутренний шлюзовой протокол (Interior Gateway Protocol, IGP), используется для распространения данных маршрутизации внутри одной [автономной системы](#).

Реализация протокола OSPF в ViPNet Coordinator HW имеет ограничения: из [RFC 2328](#) не поддерживаются:

- изменение пропускной способности reference-bandwidth — cost, auto-cost;
- настройка метрик маршрутов;
- перераспределение (redistribute) маршрутов, кроме заданных статически и полученных по DHCP;

- настройка параметров взаимодействия соседних маршрутизаторов — HelloInterval и DeadInterval;
- другие типы областей (area type), кроме backbone и standard;
- настройка типа сети (network types), — задан тип Broadcast;
- фильтрация по политике импорта и экспорта маршрутов;
- виртуальные каналы (Virtual Links).

OSPF-маршрутизация работает следующим образом:

- 1 OSPF-маршрутизаторы устанавливают связь друг с другом посредством многоадресной (multicast) рассылки.
- 2 OSPF-маршрутизаторы начинают обмениваться сообщениями типа LSA (Link State Advertisements), в которых передают информацию о состоянии каналов связи с помощью однонаправленной (unicast) рассылки.
- 3 При получении LSA-сообщения информация из него заносится в базу данных (link state database). Если LSA-сообщение содержит новую информацию о канале связи по сравнению с той, которая содержится в базе данных, то оно передаётся [маршрутизаторам-соседям](#).
- 4 На каждом маршрутизаторе по алгоритму Дейкстры (SPF algorithm) вычисляется множество кратчайших маршрутов ко всем сетям назначения и их [стоимость](#). Таким образом, каждый OSPF-маршрутизатор имеет собственное представление о состоянии каналов связи и топологии сети, но при этом все маршрутизаторы используют одну базу данных состояний каналов связи для вычисления кратчайших маршрутов.

Межсетевая среда, в которой находятся OSPF-маршрутизаторы, представляется как совокупность [областей маршрутизации](#), соединённых друг с другом через некоторую базовую область.

Выделяют следующие типы областей маршрутизации:

- Базовая или область 0 (backbone area) — с ней должны быть соединены все остальные области [автономной системы](#).
- Стандартная (standard area) — граничит как с другими областями, так и с другими автономными системами.
- Стандартная тупиковая (stub area) — граничит только с другими областями.
- Полностью тупиковая (totally stubby area) — граничит только с одной областью.
- Не полностью тупиковая (not-so-stubby area) — стандартная тупиковая область, которая может взаимодействовать с другими автономными системами с помощью пограничных маршрутизаторов (ABR).

Каждый OSPF-маршрутизатор входит в определённую область маршрутизации и обменивается информацией только с маршрутизаторами, входящими в эту же область. При этом обмен идет не напрямую, а через посредника — маршрутизатор, выбранный главным в этой области.

Информация между разными областями передаётся через маршрутизатор, который располагается на границе этих областей. Если автономная система взаимодействует с другой автономной системой, то информация передаётся через маршрутизатор, который располагается на границе

систем. Такой подход позволяет уменьшить объём рассылаемых LSA-сообщений, тем самым уменьшить нагрузку на маршрутизаторы и повысить скорость построения таблиц маршрутизации.

Выделяют следующие типы маршрутизаторов:

- Внутренний (internal router, IR) — все его сетевые интерфейсы находятся в одной области.
- Назначенный (designated router, DR) — выбран главным среди всех внутренних маршрутизаторов в одной области.
- Резервный назначенный (backup designated router, BDR) — берёт на себя функции главного в случае, если главный маршрутизатор вышел из строя.
- Межобластной граничный (area border router, ABR) — соединяет несколько областей OSPF одной автономной системы.
- Граничный маршрутизатор автономной системы (autonomous system boundary router, ASBR) — граничит с другой автономной системой, работающей по другому протоколу динамической маршрутизации (IGRP, EIGRP, IS-IS, RIP, BGP, Static).



Примечание. Для проверки подлинности маршрутов, полученных с других устройств по протоколу OSPF, мы рекомендуем настроить аутентификацию на маршрутизаторах.

Протокол OSPF поддерживает два способа аутентификации:

- простая аутентификация на пароле (simple password authentication). Одинаковый пароль настраивается на интерфейсах, через которые передаётся информация по протоколу OSPF. Пароль передаётся в открытом виде.
- криптографическая аутентификация (cryptographic authentication). Общий секретный ключ и идентификатор ключа настраиваются на интерфейсах, через которые передаётся информация по протоколу OSPF. Ключ передаётся в зашифрованном виде (используется алгоритм хэширования md5).

Аутентификация настраивается для соседних маршрутизаторов. По умолчанию протокол OSPF не использует аутентификацию.

Настройка параметров динамической маршрутизации по протоколу OSPF

В процессе настройки маршрутизации по протоколу OSPF вам потребуется:

- включить использование протокола OSPF;
- указать сети, к которым подключен ViPNet Coordinator HW и которые будут участвовать в маршрутизации по протоколу OSPF, и [области маршрутизации](#);
- создать ряд сетевых фильтров.

При настройке динамической маршрутизации по протоколу OSPF учитывайте:

- В случае сбоя на одном из маршрутов для переключения на альтернативный маршрут может потребоваться около 10 секунд после того, как будет обнаружено отсутствие ответного трафика на исходящий трафик от координатора, который потерял связь с удаленным координатором.
- Если ваш ViPNet Coordinator HW непосредственно связан с другими координаторами ViPNet, рекомендуется в файле `iplir.conf` в секциях `[id]`, соответствующих этим координаторам, указать в параметре `accessiplist` метрику 1. В этом случае при наличии нескольких альтернативных маршрутов защищенный трафик всегда будет передаваться по кратчайшему маршруту — через соседний координатор ViPNet.

Вы также можете настроить [перераспределение маршрутов](#) по протоколу OSPF.

Настройка протокола OSPF

- 1 Включите использование протокола:

```
hostname# inet ospf mode on
```

- 2 Укажите сеть, в которой происходит обмен информацией по протоколу OSPF:

```
hostname# inet ospf network add <IP-адрес назначения> netmask <маска сети> area <area-ID>
```

К данной сети должен быть подключен сетевой интерфейс ViPNet Coordinator HW.

С помощью параметра `area <area-ID>` задайте область маршрутизации, в которую входит ViPNet Coordinator HW (см. [Общие сведения о протоколе OSPF](#)).

Указать область маршрутизации можно:

- в десятичном формате 0-4294967295;
- в формате A.B.C.D, где A, B, C, D — целое число от 0 до 255. Данный способ можно использовать, чтобы в качестве идентификатора указать адрес сети, в которой будет осуществляться обмен информацией.

В интерфейсе будут отображены оба представления.



Примечание. Вы можете указать не более 128 сетей, в которых происходит обмен информацией по протоколу OSPF.

- 3 Для защиты от получения ложной информации от других устройств по протоколу OSPF, настройте аутентификацию. По умолчанию обмен информацией по протоколу OSPF происходит без аутентификации.



Внимание! Настройки аутентификации на маршрутизаторах, которые обмениваются информацией, должны быть одинаковыми.

- 3.1 Включите аутентификацию в выбранной области маршрутизации:

- Аутентификация на простом пароле:

```
hostname# inet ospf area <area-ID> auth on pswd
```
- Криптографическая аутентификация:

```
hostname# inet ospf area <area-ID> auth on md5,
```

где <area-ID> — область маршрутизации.

3.2 Установите пароль (при аутентификации на пароле) или добавьте ключ (при криптографической аутентификации) на интерфейсе, за которым находится область маршрутизации:

- Установка пароля:

```
hostname# inet ospf interface <интерфейс> password
```
- Добавление ключа:

```
hostname# inet ospf interface <интерфейс> keys add <keyid>
```



Примечание. На каждом интерфейсе вы можете создать не более 255 ключей с уникальным keyid и только 1 пароль.

4 Создайте на ViPNet Coordinator HW сетевые фильтры:

- Для каждого соседнего OSPF-маршрутизатора в области, в которой находится ViPNet Coordinator HW, разрешите входящий однонаправленный (unicast) и многоадресный (multicast) IP-трафик по протоколу OSPF от всех соседних OSPF-маршрутизаторов, с которыми взаимодействует ViPNet Coordinator HW:

```
hostname# firewall local add 1 rule "Allow OSPF unicast" src <IP-адрес OSPF-маршрутизатора> dst @local service @OSPF pass
```

```
hostname# firewall local add 2 rule "Allow OSPF multicast" src <IP-адрес OSPF-маршрутизатора> dst @multicast service @OSPF pass
```



Примечание. Если вы обновили ViPNet Coordinator HW с версии 4.1 и ниже, то системный объект @OSPF вам требуется создать вручную (см. [Создание пользовательской группы объектов](#)). В фильтрах вместо service @OSPF вы можете указать proto 89.

- Разрешите исходящий трафик ViPNet Coordinator HW по протоколу OSPF на любой IP-адрес назначения:

```
hostname# firewall local add 3 rule "Allow OSPF outgoing" src @local dst @any service @OSPF pass
```

- Если между координаторами, которые участвуют в обмене по протоколу OSPF, установлена связь в управляющем ПО (в том числе между координатором и ViPNet xFirewall), разрешите трафик протокола OSPF в защищенной сети:

```
hostname# firewall vpn add 1 rule "Allow OSPF outgoing" src @local dst @any service @OSPF pass
```

```
hostname# firewall vpn add 2 rule "Allow OSPF incoming" src @any dst @local service @OSPF pass
```

Подробнее о создании сетевых фильтров см. в разделе [Создание сетевого фильтра](#).



Внимание! Если вы не настроите указанные фильтры, то ViPNet Coordinator HW не сможет работать по протоколу OSPF (см. [Общие сведения о протоколе OSPF](#)).

Другие команды для настройки OSPF

- Удалить подсеть, для которой прекращено использование протокола OSPF:

```
hostname# inet ospf network delete <IP-адрес назначения> netmask <маска подсети> area <area-ID>
```

- Отключить аутентификацию для области маршрутизации:

```
hostname# inet ospf area <area-ID> auth off
```

- Удалить пароль или ключ для аутентификации:

- Удалить пароль:

```
hostname# inet ospf interface <интерфейс> password remove
```

- Удалить один ключ:

```
hostname# inet ospf interface <интерфейс> keys remove <keyid>
```

- Удалить все ключи, добавленные для интерфейса:

```
hostname# inet ospf interface <интерфейс> keys remove all
```

- Отказаться от использования протокола OSPF для маршрутизации:

```
hostname# inet ospf mode off
```

Настройка перераспределения маршрутов

Протокол OSPF может перераспределять маршруты, заданные статически или полученные по протоколу DHCP. Это может понадобиться, если в подсетях разветвленной сети используются разные виды маршрутов.

Рассмотрим пример:

- Главный офис и филиал входят в одну автономную систему. Используют маршрутизацию по протоколу OSPF.
- Сеть партнера является другой автономной системой. Использует статическую маршрутизацию (Static). На маршрутизаторе настроены статические маршруты для сетей главного офиса и филиала.
- HW-1 расположен на границе автономных областей и является граничным маршрутизатором.
- На HW-1 настроены статические маршруты в сеть партнера.

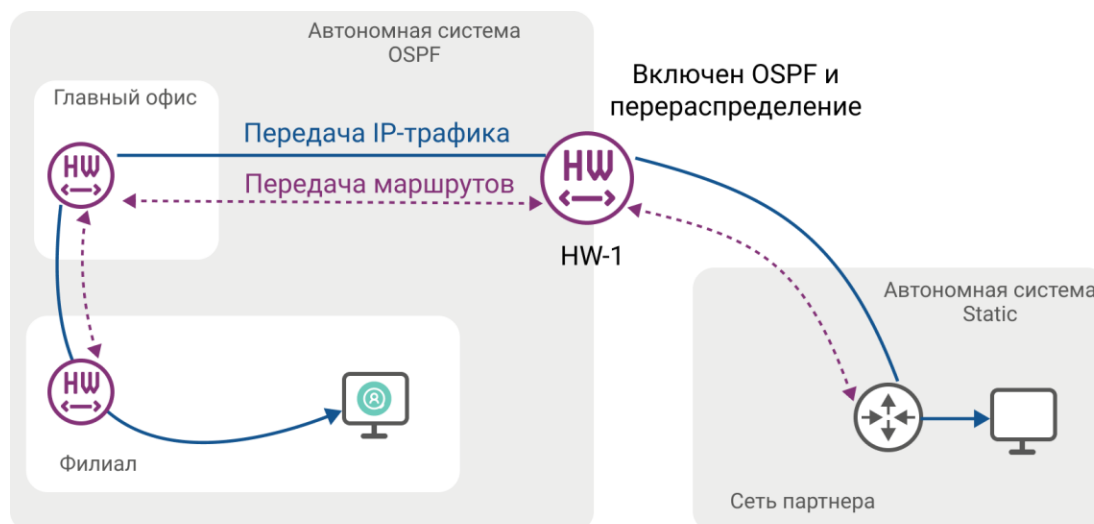


Рисунок 43. Перераспределение маршрутов

Чтобы узлы сети филиала могли обмениваться IP-трафиком с узлами сети партнера, вместо настройки статических маршрутов на оборудовании сетей главного офиса и филиала включите перераспределение статических маршрутов на HW-1:

```
hostname# inet ospf redistribute add static
```

HW-1 передаст статические маршруты в автономную систему офиса и узлы сети филиала смогут обмениваться IP-трафиком с узлами сети партнера.

Примечание.



- Статический маршрут по умолчанию не перераспределяется.
- Если в сети партнера используются динамические маршруты от DHCP-сервера, перераспределение включается командой `inet ospf redistribute add dhcp`. Также на граничном маршрутизаторе должно быть настроено получение маршрутов от DHCP-сервера сети партнера.

Просмотр настроек протокола OSPF

Вы можете просмотреть следующие параметры протокола OSPF:

- статус использования протокола OSPF: включен или выключен;
- список сетей, в которых ViPNet Coordinator HW выполняет маршрутизацию по протоколу OSPF (IP-адреса, маски и области);
- статус перераспределения маршрутов: доступно или недоступно.

Чтобы посмотреть указанные настройки, выполните команду:

```
hostname> inet ospf show configuration
```

Настройка протокола OSPF в сетях со множественным доступом

ViPNet Coordinator HW выполняет фильтрацию трафика между узлами сегмента локальной сети и внешними сетями. Вы можете сделать сегмент локальной сети частью сети со множественным доступом, входящей в область с маршрутизацией по протоколу OSPF. Когда в сети со множественным доступом становится больше двух маршрутизаторов, необходимо, для сокращения служебного трафика между маршрутизаторами, выбрать среди них **назначенный маршрутизатор DR** и **резервный назначенный маршрутизатор BDR**. Выбор роли DR или BDR происходит на основании параметров приоритета `priority` или идентификатора `router-id` маршрутизаторов:

- маршрутизатор с наивысшим значением приоритета становится DR;
- маршрутизатор с наивысшим значением приоритета из оставшихся становится BDR;
- маршрутизатор с приоритетом 0 не может стать DR или BDR;
- если у маршрутизаторов одинаковые приоритеты, то маршрутизатор с наивысшим идентификатором становится DR, а с наивысшим из оставшихся — BDR;
- появление в сети со множественным доступом нового маршрутизатора с более высоким приоритетом, чем у текущего DR, не влияет на роли текущих DR и BDR;
- процедура выбора нового DR или BDR запускается, если текущий DR или BDR выходят из строя.

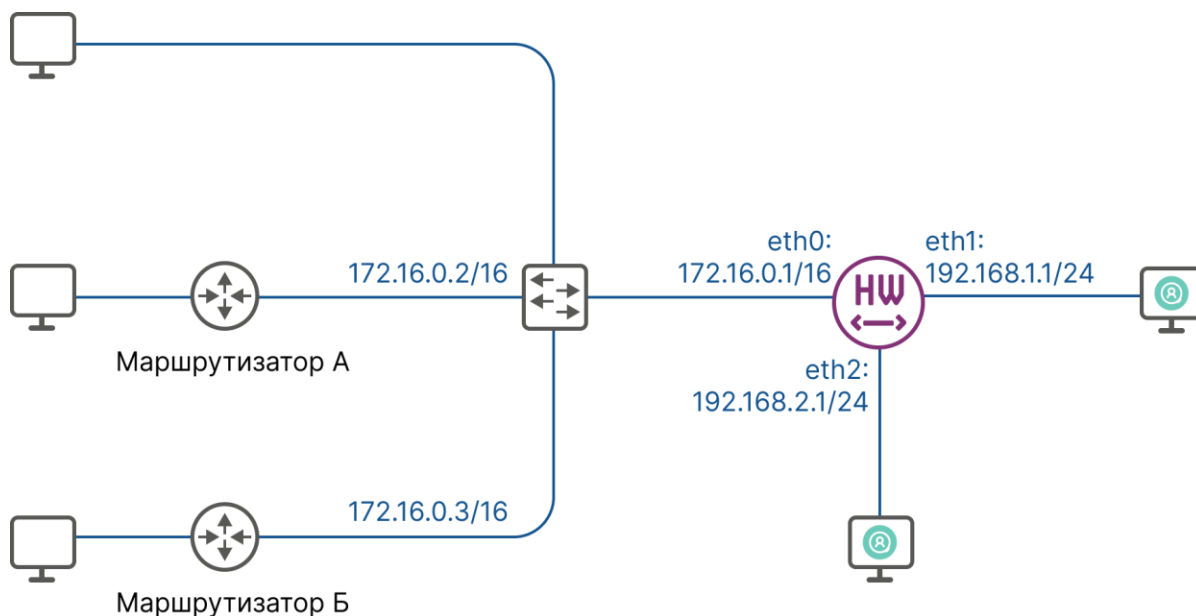


Рисунок 44. ViPNet Coordinator HW в сети со множественным доступом

Приоритет является параметром интерфейса ViPNet Coordinator HW. По умолчанию на всех интерфейсах ViPNet Coordinator HW установлен приоритет — 1. Вы можете изменить значение приоритета интерфейса ViPNet Coordinator HW на целое число из диапазона от 0 до 255:

```
hostname# inet ospf priority <значение приоритета> interface <имя интерфейса>
```

Идентификатор ViPNet Coordinator HW представляет собой число в формате адреса протокола IPv4, которое однозначно идентифицирует ViPNet Coordinator HW в пределах одной автономной области. По умолчанию идентификатор равен максимальному значению IP-адреса из назначенных на интерфейсах ViPNet Coordinator HW. Вы можете изменить значение идентификатора:

```
hostname# inet ospf router-id <адрес IPv4>
```



Внимание! Если в сети со множественным доступом у двух маршрутизаторов будут одинаковые идентификаторы `router-id`, то такие маршрутизаторы не смогут установить отношения соседства.

Чтобы посмотреть текущие значения параметров `priority` и `router-id` ViPNet Coordinator HW:

```
hostname# inet ospf show configuration
```

Просмотр состояния каналов связи между маршрутизаторами

Кроме настроек протокола OSPF вы также можете просмотреть информацию о состоянии каналов связи между всеми маршрутизаторами, работающими по протоколу OSPF. Она содержится в базе данных (link state database) (см. [Общие сведения о протоколе OSPF](#)) и синхронизируется на всех OSPF-маршрутизаторах. Именно на ее основе вычисляются динамические маршруты протокола OSPF. Для просмотра информации базы данных состояний каналов связи:

```
hostname> inet ospf show database
```

```

hw-16310045# inet ospf show database

      OSPF Router with ID (10.0.3.2)

      Router Link States (Area 0.0.0.0)

Link ID      ADV Router    Age Seq#         CkSum Link count
10.0.3.2     10.0.3.2      155 0x80000017d 0x590a 2
10.1.30.5    10.1.30.5     220 0x80000029f 0x3fa0 2

      Net Link States (Area 0.0.0.0)

Link ID      ADV Router    Age Seq#         CkSum
10.0.5.5     10.1.30.5     751 0x800000263 0x3541

      AS External Link States

Link ID      ADV Router    Age Seq#         CkSum Route
10.0.1.0     10.1.30.5     1551 0x800000182 0x9061 E2 10.0.1.0/24 [0x0]
10.0.2.0     10.1.30.5     1001 0x800000183 0x836c E2 10.0.2.0/24 [0x0]
10.0.3.0     10.1.30.5     210 0x800000182 0x7a75 E2 10.0.3.0/24 [0x0]
10.0.4.0     10.1.30.5     341 0x800000182 0x6f7f E2 10.0.4.0/24 [0x0]
10.100.1.0   10.1.30.5     911 0x80000029d 0xe1ad E2 10.100.1.0/24 [0x0]
10.100.2.0   10.1.30.5     180 0x80000029f 0xe0aa E2 10.100.2.0/24 [0x0]
192.168.0.0  10.1.30.5     821 0x800000182 0x6c27 E2 192.168.0.0/16 [0x0]

hw-16310045#

```

Рисунок 45. Просмотр базы данных состояний связи

Просмотр информации о соседних маршрутизаторах

В составлении маршрутов по протоколу OSPF участвуют все OSPF-маршрутизаторы, находящиеся в области маршрутизации [ViPNet Coordinator HW](#). При необходимости вы можете просмотреть сведения об OSPF-маршрутизаторах, которые являются соседними для [ViPNet Coordinator HW](#):

```
hostname> inet ospf show neighbour
```

В результате выполнения команды отобразится информация обо всех соседних OSPF-маршрутизаторах:

По каждому маршрутизатору отображается следующая информация:

- уникальный идентификатор OSPF-маршрутизатора, под которым он известен другим маршрутизаторам при работе по протоколу OSPF;
- приоритет маршрутизатора;
- тип маршрутизатора (см. [Общие сведения о протоколе OSPF](#)) (в случае, приведенном на рисунке, маршрутизатор-сосед является назначенным, на что указывает значение DR в поле State);
- интервал простоя маршрутизатора, по истечении которого он будет считаться неактивным (выключенным);
- IP-адрес маршрутизатора, по которому он доступен для обмена информацией по протоколу OSPF;
- IP-адрес активного сетевого интерфейса маршрутизатора;

- другие параметры.

Просмотр маршрутов, сформированных по протоколу OSPF

Чтобы просмотреть список динамических маршрутов, сформированных по протоколу OSPF:

```
hostname> inet show routing ospf
```

В результате отобразятся списки маршрутов в трех разделах:

В каждом из разделов отображаются следующие маршруты:

- OSPF network routing table — маршруты, которые были сформированы по OSPF-протоколу в той [области маршрутизации](#), к которой подключен ViPNet Coordinator HW. Если ViPNet Coordinator HW является межобластным маршрутизатором (ABR) (см. [Общие сведения о протоколе OSPF](#)), то в этом разделе будут присутствовать маршруты из всех областей, которые он объединяет.
- OSPF router routing table — маршруты до [соседних OSPF-маршрутизаторов](#), которые находятся в области маршрутизации ViPNet Coordinator HW и выполняют перераспределение маршрутов (см. [Настройка перераспределения маршрутов](#)).
- OSPF external routing table — маршруты в другие автономные системы, поступившие при перераспределении от соседних OSPF-маршрутизаторов.

Настройка функции L2OverIP для обеспечения работоспособности протоколов динамической маршрутизации

Технология L2OverIP также может использоваться для объединения удаленных сегментов сети с целью организации между ними динамической [маршрутизации](#) по протоколу [OSPF](#).

Для динамической маршрутизации по протоколу OSPF должно выполняться следующее условие: работа по протоколу OSPF должна поддерживаться на всем пути следования IP-пакетов. В сетях общего пользования, в частности, в сети интернет, протокол OSPF не поддерживается. В связи с этим динамическая маршрутизация между несколькими сегментами сети, объединенными интернетом, не сможет осуществляться. Решить данную проблему можно путем объединения сегментов сети с помощью технологии L2OverIP. В результате маршрутизация IP-пакетов будет производиться как бы в одном сегменте сети, а не в двух, разделенных сетью интернет.

Например, пусть требуется объединить два сегмента сети, в каждом из которых создана разветвленная структура узлов с несколькими маршрутизаторами. Каждый сегмент сети

подключается к одному из интерфейсов ViPNet Coordinator HW, установленному на границе сегмента. На всех маршрутизаторах обоих сегментов поддерживается работа по протоколу OSPF.

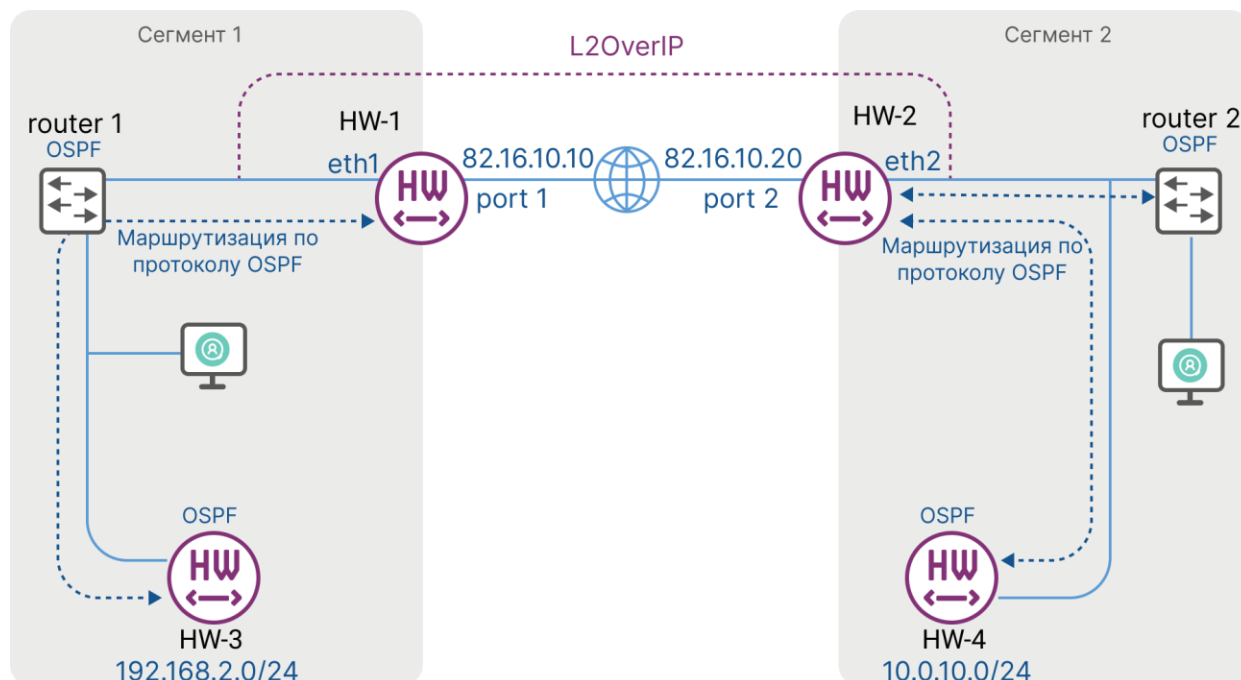


Рисунок 46. Схема работы L2OverIP и динамической маршрутизации по протоколу OSPF

Чтобы организовать маршрутизацию по протоколу OSPF в рассмотренном примере:

- 1 Настройте параметры L2OverIP на координаторах HW-1 и HW-2 (см. [Настройка параметров L2OverIP](#)).
- 2 Настройте работу по протоколу OSPF на маршрутизаторах router1 и router2 и координаторах HW-3, HW-4 (см. [Настройка протокола OSPF](#)).



Примечание. На HW-1 и HW-2, установленных на границах сегментов, настройка протокола OSPF не требуется, поскольку в данном сценарии эти координаторы не участвуют в маршрутизации IP-трафика.

Настройка параметров L2OverIP

Для настройки параметров L2OverIP:

- 1 Укажите сетевой интерфейс, сегмент сети которого будет объединяться с другим сегментом с помощью функции L2OverIP:
 - На HW-1:

```
hostname# iplir set l2overip interface eth1
```
 - На HW-2:

```
hostname# iplir set l2overip interface eth2
```

Если сегментов больше двух, аналогичным образом укажите рабочий интерфейс на остальных ViPNet Coordinator HW.

- 2 Задайте параметры локального сегмента сети, указав уникальный номер порта и адрес внешнего интерфейса:

- На HW-1:

```
hostname# iplir set l2overip local-port 1 82.16.10.10
```

- На HW-2:

```
hostname# iplir set l2overip local-port 2 82.16.10.20
```

- 3 Задайте параметры удаленного сегмента сети, указав его номер порта и актуальный адрес видимости удаленного ViPNet Coordinator HW:

- На HW-1:

```
hostname# iplir set l2overip remote-port 2 82.16.10.20
```

- На HW-2:

```
hostname# iplir set l2overip remote-port 1 82.16.10.10
```

В зависимости от настройки видимости удаленного ViPNet Coordinator HW укажите его реальный или виртуальный адрес.



Внимание! Если в процессе работы вы поменяли видимость удаленного узла, обязательно поменяйте его адрес в настройках.

- 4 На каждом координаторе добавьте сетевые фильтры защищенной сети, разрешающие любые соединения по протоколу 97:

```
hostname# firewall vpn add src @local dst @any proto 97 pass
```

```
hostname# firewall vpn add src @any dst @local proto 97 pass
```

- 5 На каждом координаторе включите функцию L2OverIP:

```
hostname# iplir set l2overip mode switch
```

- 6 На каждом координаторе убедитесь в корректности выполненных настроек функции L2OverIP:

```
hostname# iplir show l2overip config
```

Настройка протокола OSPF

Чтобы настроить работу по протоколу OSPF на координаторах HW-3, HW-4, выполните:

- 1 На каждом координаторе включите использование протокола OSPF:

```
hostname# inet ospf mode on
```

- 2 Укажите на координаторах сеть, в которой происходит обмен информацией по протоколу OSPF:

- На HW-3:

```
hostname# inet ospf network add 192.168.2.0 netmask 255.255.255.0 area 1
```

- На HW-4:

```
hostname# inet ospf network add 10.0.10.0 netmask 255.255.255.0 area 1
```



Примечание. В случае, рассмотренном на примере, все координаторы, работающие по протоколу OSPF, принадлежат одной области маршрутизации.

3 На каждом координаторе создайте следующие сетевые фильтры открытой сети:

- Фильтры, разрешающие входящий однонаправленный IP-трафик (unicast) и многоадресный IP-трафик (multicast) по протоколу OSPF от соседних OSPF-маршрутизаторов, с которыми взаимодействует координатор.

Если политика безопасности вашей организации не запрещает прохождение служебного IP-трафика по протоколу OSPF (IP:89) от любого узла, то создайте следующие фильтры:

```
hostname# firewall local add 1 rule "Rule 1" src @any dst @local service @OSPF pass
hostname# firewall local add 2 rule "Rule 2" src @any dst @multicast service @OSPF pass
```

Если создание фильтров, пропускающих служебный IP-трафик по протоколу OSPF от любых узлов запрещено, создайте аналогичные фильтры для каждого OSPF-маршрутизатора, который является соседним для координатора.

Таблица 6. Фильтры для входящего OSPF-трафика на конкретном координаторе

Имя координатора	Название	Источник	Назначение	Протокол	Действие
HW-3	Rule 1	<IP-адрес router1>	@local	IP:89	pass
	Rule 2	<IP-адрес router1>	@multicast	IP:89	pass
HW-4	Rule 1	<IP-адрес router1>	@local	IP:89	pass
	Rule 2	<IP-адрес router1>	@multicast	IP:89	pass
	Rule 3	<IP-адрес router2>	@local	IP:89	pass
	Rule 4	<IP-адрес router2>	@multicast	IP:89	pass

- Фильтр, разрешающий исходящий IP-трафик с координатора по протоколу OSPF на любой IP-адрес назначения:

```
hostname# firewall local add 1 rule "Rule 1" src @local dst @any service @OSPF pass
```



Примечание. На маршрутизаторах router1 и router2 настройка протокола OSPF должна быть выполнена по тому сценарию, который они поддерживают.

В результате сегменты сети будут объединены по технологии L2OverIP и между ними будет организована динамическая маршрутизация по протоколу OSPF.



Внимание! В случае сбоя на одном из маршрутов для переключения на альтернативный маршрут может потребоваться около 10 секунд после того, как будет обнаружено отсутствие ответного трафика на исходящий трафик от координатора, который потерял связь с удаленным координатором.

Настройка функции MultiWAN

Если для доступа ViPNet Coordinator HW в интернет используется несколько шлюзов (провайдеров), то вы можете настроить политики маршрутизации для различных случаев. Например, для распределения нагрузки между каналами связи или для резервирования каналов при использовании нескольких интернет-провайдеров на случай отказа одного из них (чтобы избежать потерь трафика).

Для этих целей в ViPNet Coordinator HW реализована технология MultiWAN, позволяющая настроить поведение ViPNet Coordinator HW в случае недоступности одного из интернет-шлюзов. Эта технология представлена в ViPNet Coordinator HW следующими функциями:

- Пользовательские таблицы маршрутизации (см. [Создание статических маршрутов](#)). С их помощью вы можете задать различные маршруты для различных ситуаций. Например, основную таблицу маршрутизации и дополнительную таблицу маршрутизации на случай, если шлюз основного провайдера станет недоступен и потребуется перенаправить трафик через шлюз альтернативного провайдера.
- Политики маршрутизации. С их помощью вы можете задать условия, при которых трафик будет перенаправляться по заданным таблицам маршрутизации.
- Проверка состояния шлюзов (Dead Gateway Detection, DGD). С помощью этой функции вы можете задавать выполнение политики маршрутизации в зависимости от доступности или недоступности [шлюза](#).

Особенности настройки

- Не добавляйте в пользовательские таблицы маршрутизации маршруты, противоречащие маршрутам в таблице по умолчанию.
- Не используйте динамические маршруты, получаемые по протоколам DHCP и OSPF при настройке маршрутизации для резервирования канала доступа в интернет (при использовании DHCP отключите автоматическое получение маршрутов: `hostname# inet ifconfig <интерфейс> dhcp route off`). Динамические маршруты записываются в таблицу по умолчанию и будут задействованы с соответствующим приоритетом. Не используйте канал с динамическими маршрутами в качестве резервного канала.
- Добавляйте маршруты только в пользовательские таблицы маршрутизации, а таблицу по умолчанию изменяйте лишь в крайнем случае.
- Перед активацией политики маршрутизации убедитесь, что она настроена корректно и не нарушит связь ViPNet Coordinator HW с узлом, с которого вы настраиваете ViPNet Coordinator HW.
- Не используйте одинаковые IP-адреса для проверки шлюзов DGD и для параметров `testip` при работе в кластере — это может привести к циклической перезагрузке обоих узлов кластера.

- После добавления проверки шлюза DGD по протоколу TCP и портам 80 и 443 необходимо создать соответствующий разрешающий сетевой фильтр: источник — **Мой узел ViPNet**, назначение — IP-адрес шлюза, порты назначения 80 и 443.

Резервирование каналов доступа в интернет

В этом примере сеть ViPNet имеет выход в интернет через двух провайдеров:

- основного (шлюз GW-1), 89.135.15.2;
- запасного (шлюз GW-2), 210.11.0.2.

Чтобы в случае потери соединения со шлюзом одного провайдера переключаться на другой шлюз, настроим резервирование каналов доступа в интернет с помощью MultiWAN.



Рисунок 47. Резервирование каналов доступа в интернет

Для настройки резервирования каналов доступа:

- 1 Удалите маршруты по умолчанию из таблицы по умолчанию:

```
hostname# inet route delete default
```

- 2 Добавьте маршруты по умолчанию для шлюзов GW-1 и GW-2:

```
hostname# inet route add default next-hop 89.135.15.2 table 1024 name GW-1
hostname# inet route add default next-hop 210.11.0.2 table 1025 name GW-2
```

- 3 Создайте политики маршрутизации, по которым трафик будет маршрутизироваться через шлюз GW-2, когда недоступен шлюз GW-1:

- 3.1 Добавьте более приоритетные правила для таблицы по умолчанию:

```
hostname# inet policy rule add route-via-GW-2-policy 1024 table default
hostname# inet policy rule add route-via-GW-1-policy 1024 table default
```

- 3.2 Добавьте менее приоритетные правила для пользовательских таблиц:

```
hostname# inet policy rule add route-via-GW-2-policy 1025 table name GW-2
hostname# inet policy rule add route-via-GW-1-policy 1025 table name GW-1
```



Примечание. Приоритет у правил по таблице по умолчанию должен быть выше, чем у общих правил маршрутизации, применяющихся для всего трафика.

4 Запустите службу DGD:

```
hostname# inet dgd mode on
```

5 Настройте проверку доступности шлюзов провайдеров GW-1 и GW-2:

```
hostname# inet dgd next-hop add GW-1 address 89.135.15.2 check icmp
```

```
hostname# inet dgd next-hop add GW-2 address 210.11.0.2 check icmp
```

6 Создайте правила службы DGD, по которым действует политика маршрутизации route-via-GW-1-policy, если доступен шлюз провайдера GW-1:

```
hostname# inet dgd rule add GW-1-up match-next-hop GW-1 up
```

```
hostname# inet dgd rule add GW-1-up action-priority 1 service route command policy active route-via-GW-1-policy
```

7 Создайте правила службы DGD, по которым действует политика маршрутизации route-via-GW-2-policy, если доступен шлюз провайдера GW-2, а шлюз провайдера GW-1 недоступен:

```
hostname# inet dgd rule add GW-1-down match-next-hop GW-1 down
```

```
hostname# inet dgd rule add GW-1-down match-next-hop GW-2 up
```

```
hostname# inet dgd rule add GW-1-down action-priority 1 service route command policy active route-via-GW-2-policy
```

8 Создайте правила службы DGD, по которым действует политика маршрутизации по умолчанию и создается запись в системном журнале, если недоступны шлюзы обоих провайдеров:

```
hostname# inet dgd rule add all-down match-next-hop GW-1 down
```

```
hostname# inet dgd rule add all-down match-next-hop GW-2 down
```

```
hostname# inet dgd rule add all-down action-priority 2 service log command All-Routers-Down
```

9 Перезапустите службу DGD для применения новых правил:

```
hostname# inet dgd mode off
```

```
hostname# inet dgd mode on
```

Так вы обеспечите маршрутизацию трафика между двумя сетями ViPNet даже в случае недоступности одного из провайдеров.

В качестве резервного канала связи можно использовать встроенный модем (исполнения со встроенным модемом см. в документе «Описание аппаратных платформ»).

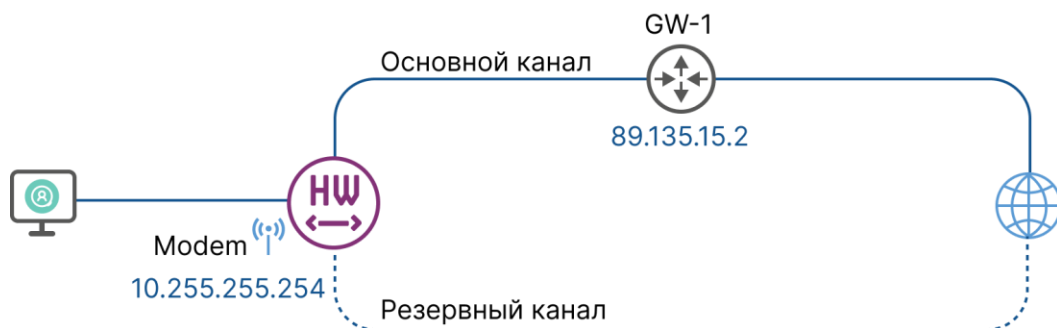


Рисунок 48. Резервирование каналов доступа в интернет с помощью встроенного модема

В этом случае для работы сценария резервирования, описанного выше, требуется выполнить дополнительные действия:

- 1 Отключите модем и получение маршрутов:

```
hostname# inet usb-modem mode off
hostname# inet usb-modem route off
```

- 2 Включите модем:

```
hostname# inet usb-modem mode on
```

- 3 Создайте таблицу маршрутизации для модема:

```
hostname# inet route add default next-hop 10.255.255.254 table 1026 name ModemTable
```

- 4 Выполните действия по настройке резервирования каналов доступа в интернет, как описано в сценарии выше. Используйте назначенный адрес шлюза (в примере 10.255.255.254) вместо адреса шлюза провайдера GW-2 и созданную таблицу маршрутизации (ModemTable) вместо таблицы маршрутизации GW-2.



Примечание. При ручном отключении модема (с помощью команды `inet usb modem mode off`), проверка состояния шлюза модемного интерфейса будет работать некорректно. В этом случае отключите проверку состояния шлюза модемного интерфейса:

```
hostname# inet dgd next-hop add <имя модемного шлюза> address <IP-адрес модемного шлюза> test-address <тестовый IP-адрес модемного шлюза> no-check icmp
```

Обеспечение связи между прокси-сервером и его клиентами, находящимися в разных сетях

В этом сценарии рассматривается схема, в которой защищенные клиенты выходят в интернет через прокси-сервер, расположенный в другой сети и стоящий за координатором этой сети.

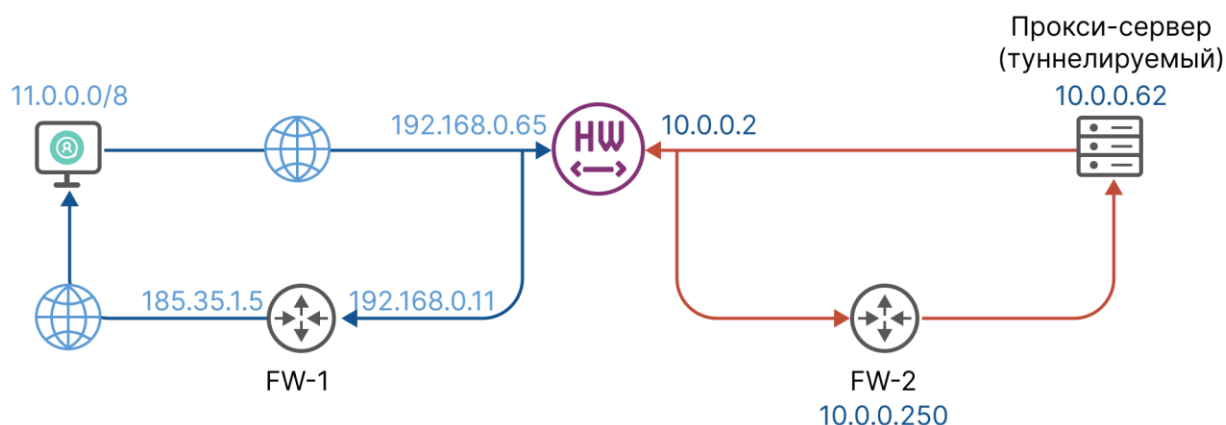


Рисунок 49. Связь между туннелируемыми и защищенными узлами

В этой схеме необходимо настроить туннелирование всех IP-адресов интернета, чтобы прокси-сервер, расположенный за ViPNet Coordinator HW, мог получать запросы на доступ к ресурсам интернет от защищенных клиентов и отправлять на них ответы.

Дальнейшее описание схемы приведено со следующими условиями:

- используется диапазон виртуальных IP-адресов по умолчанию (11.0.0.0/8);
- видимость сетевых узлов задана как «автоматическая» или «с помощью виртуальных IP-адресов» (соответствующие параметры `auto` и `virtual` секции `[visibility]` конфигурационного файла `iplir.conf`). При этом следует учитывать, что если задана автоматическая видимость сетевых узлов, то необходимо исключить ситуацию, когда защищенному сетевому узлу назначается IP-адрес из того же диапазона, которому принадлежит IP-адрес координатора, так как в этом случае видимость сетевых узлов изменится на «с помощью реальных IP-адресов» и трафик между защищенными узлами и прокси-сервером перестанет обрабатываться настроенными правилами маршрутизации.

Для настройки этой схемы:

- 1 В ViPNet ЦУС или Prime для координатора задайте диапазоны туннелируемых IP-адресов (ресурсы):

```

1.0.0.1-10.0.0.0
13.0.0.0-172.16.0.0
172.31.255.255-192.168.0.0
192.168.255.255-255.255.255.254

```

- 2 На координаторе создайте маршрут по умолчанию до внутреннего сетевого интерфейса маршрутизатора FW-1:

```
hostname# inet route add default next-hop 192.168.0.11
```

- 3 Создайте таблицу маршрутизации до прокси-сервера через маршрутизатор FW-2:

```
hostname# inet route add default next-hop 10.0.0.250 table 1025 name PROXY-DEFAULT
```

- 4 Создайте правило политики маршрутизации для IP-пакетов с защищенных узлов по ранее созданной таблице маршрутизации PROXY-DEFAULT:

```
hostname# inet policy rule add ClientToProxy 1024 match address from 11.0.0.0/8
```

```
hostname# inet policy rule add ClientToProxy 1024 table name PROXY-DEFAULT
```

- 5 Создайте правило политики маршрутизации для обработки остального трафика по таблице маршрутизации по умолчанию (default):

```
hostname# inet policy rule add ClientToProxy 1025 table default
```

- 6 Активируйте политику маршрутизации:

```
hostname# inet policy active ClientToProxy
```

- 7 На маршрутизаторе FW-2 и прокси-сервере создайте маршрут от координатора до виртуальных IP-адресов клиентов (11.0.0.0/8). Например, если вы используете набор утилит `iproute2`, маршрут следует добавить так:

```
hostname# ip route add 11.0.0.0/8 via 10.0.0.2
```

В результате трафик от защищенных узлов до туннелируемых узлов будет поступать на координатор, расшифровываться и после обработки в соответствии с заданной политикой маршрутизации отправляться на прокси-сервер (или внутренние ресурсы) через маршрутизатор FW-2. Ответные IP-пакеты от прокси-сервера (или внутренних ресурсов) будут поступать на координатор, шифроваться и затем отправляться на защищенные узлы через маршрутизатор FW-1.

11

Настройка параметров безопасности

Изменение пароля пользователя	232
Изменение способа аутентификации пользователя	233
Подключение к веб-интерфейсу по HTTPS	235
Просмотр информации об установленных ключах	239
Обновление справочников и ключей	240
Добавление резервного набора персональных ключей (РНПК)	243
Обновление ключей при истечении срока их действия	245
Контроль целостности файлов ViPNet Coordinator HW	247

Изменение пароля пользователя

Пароль пользователя ViPNet Coordinator HW назначается администратором сети ViPNet. После первичной установки справочников и ключей рекомендуется его изменить. Это дополнительно повысит безопасность, поскольку пароль не будет известен администратору сети ViPNet.



Внимание! Пароль пользователя ViPNet Coordinator HW можно изменить только локально, как описано ниже. Смена пароля пользователя с помощью программы ViPNet ЦУС или Prime невозможна.

Чтобы изменить пароль пользователя:

- 1 Выполните команду:

```
hostname# admin passwd
```
- 2 Введите текущий пароль пользователя или пароль администратора сетевого узла. При вводе пароля вводимые символы не отображаются
- 3 Введите новый пароль пользователя и подтвердите его. Появится сообщение об успешном изменении пароля пользователя.

В результате с помощью нового пароля будут перешифрованы ключи пользователя, в том числе [резервный набор персональных ключей](#) при наличии.



Совет. Мы рекомендуем запомнить пароль, назначенный администратором сети ViPNet. Он может потребоваться вам при установке нового [дистрибутива ключей](#) или РНПК, поскольку они будут защищены этим паролем и без него не смогут быть установлены на ViPNet Coordinator HW. Если вы забудете пароль, запросите его у администратора сети ViPNet вместе с дистрибутивом или РНПК.

Изменение пароля пользователя в кластере горячего резервирования

При работе ViPNet Coordinator HW в режиме кластера горячего резервирования пароль пользователя не синхронизируется между узлами кластера. Чтобы изменить пароль:

- 1 Остановите службу `failoverd` на пассивном узле кластера, а затем на активном:

```
hostname# failover stop
```
- 2 Измените пароль на пассивном узле кластера, а затем на активном, как указано в сценарии выше.
- 3 Включите службу `failoverd` сначала на пассивном, а затем на активном узле:

```
hostname# failover start
```

Изменение способа аутентификации пользователя

Способ аутентификации (см. [Аутентификация пользователя на ViPNet Coordinator HW](#)) первоначально задается администратором сети в ViPNet ЦУС. При использовании ViPNet Prime в качестве управляющего ПО, ViPNet Coordinator HW поддерживает только способ аутентификации «Пароль».

Информация о заданном способе аутентификации на ViPNet Coordinator HW попадает в составе справочников при первом развертывании дистрибутива ключей. Чтобы посмотреть, какой способ аутентификации используется на ViPNet Coordinator HW, выполните:

```
hostname# iplir show authentication-type
```

В процессе использования ViPNet Coordinator HW вы можете изменить способ аутентификации «Пароль» на «Устройство».

Внимание!



- После изменения способа аутентификации на «Устройство» вы не сможете изменить его обратно на «Пароль».
- Изменить способ аутентификации можно только при локальном подключении к ViPNet Coordinator HW.
- При работе ViPNet Coordinator HW в режиме [кластера горячего резервирования](#) изменить способ аутентификации достаточно на активном узле кластера. Настройки будут переданы на пассивный узел при следующей синхронизации.

Чтобы изменить способ аутентификации на «Устройство»:

- 1 Обратитесь к администратору сети ViPNet с запросом изменить способ аутентификации пользователя ViPNet Coordinator HW на «Устройство».

Администратор сети после изменения способа аутентификации должен выслать на ViPNet Coordinator HW обновление справочников и ключей и предоставить вам защищенный USB-носитель (токен) с персональным ключом пользователя.



Внимание! В текущей версии ViPNet Coordinator HW для аутентификации используются только защищенные USB-носители (токены) Рутокен Lite производства компании «Актив». На носителе не должны содержаться другие контейнеры ключей. Иначе персональный ключ пользователя не будет найден и работа с ViPNet Coordinator HW будет невозможна.

-
- 2 Проверьте, что на ViPNet Coordinator HW принято поступившее обновление справочников и ключей:

```
hostname# iplir show keys-upgrade-log
```

- 3 Выполните команду:

```
hostname# admin authentication-type-token
```

4 В ответ на предложение изменить способ аутентификации для повышения безопасности In version 4.2, you can enhance security by two-factor authentication. As you switch to the two-factor authentication, you will not be able to return to the previous settings [Yes/No] введите Yes.

5 Подключите защищенный USB-носитель (токен) с персональным ключом.

6 Введите ПИН доступа к подключенному устройству в ответ на сообщение Enter PIN code.

Если ПИН верен, появится сообщение об успешном изменении способа аутентификации.

Способ аутентификации будет изменен. Теперь пользователю потребуется подключать USB-носитель (токен) с персональным ключом и вводить ПИН.

Подключение к веб-интерфейсу по HTTPS

По умолчанию для подключения к веб-интерфейсу ViPNet Coordinator HW используется протокол HTTPS и самоподписанный сертификат, который генерируется при инициализации ViPNet Coordinator HW.

При использовании самоподписанного сертификата ViPNet Coordinator HW может появляться сообщение **Подключение не защищено**, так как браузер не может проверить, принадлежит ли используемый сертификат координатору.

Для подключения к веб-интерфейсу вы можете:

- Издать сертификат в доверенном удостоверяющем центре (далее — УЦ).

Браузер при каждом подключении будет проверять, является ли сертификат доверенным. Это защитит от подмены сертификата.

- Добавить самоподписанный сертификат в исключения.

Вы сможете подключиться к веб-интерфейсу ViPNet Coordinator HW, но браузер не сможет проверять подлинность сертификата.

Способ зависит от браузера. Например, для Google Chrome при первом подключении к веб-интерфейсу нажмите **Подробнее** > **Перейти на сайт**.

- Добавить самоподписанный сертификат в браузер как доверенный.

Браузер сможет проверять подлинность сертификата и предупреждать о его подмене.

Метод снижает защищенность компьютера, с которого происходит подключение к ViPNet Coordinator HW, так как все сертификаты, подписанные самоподписанным сертификатом ViPNet Coordinator HW, также будут доверенными.

Издание сертификата в доверенном УЦ

При подключении по HTTPS вместо самоподписанного сертификата вы можете использовать сертификат, выданный доверенным УЦ. Это позволит избежать потенциальных угроз безопасности.



Примечание. Вы можете создать запрос на сертификат и импортировать сертификат через веб-интерфейс. Подробнее см. в документе «Настройка с помощью веб-интерфейса», раздел «Работа с сертификатами».

При использовании сертификата, импортированного через веб-интерфейс, начните настройку с шага 6.

1 Создайте запрос на сертификат:

```
hostname# service cert request create name <запрос> bits <2048|3072|4096> digest {sha256|sha384|sha512} [subj <subject>]
```

Длину ключа, алгоритм хэширования и дополнительные параметры задайте в соответствии с требованиями вашей организации.

В процессе выполнения команды будут запрошены личные данные, необходимые для создания сертификата (страна, город, организация и т.д.).

После выполнения команды будет создан файл запроса и файл с закрытым ключом.

2 Подключите к ViPNet Coordinator HW USB-носитель и экспортируйте файл запроса:

```
hostname# service cert request export <запрос>
```

3 Отправьте файл запроса в аккредитованный УЦ (например, GlobalSign, VeriSign и др.).

Примечание. При выпуске сертификатов учитывайте:



- Сертификаты должны быть подписаны с использованием алгоритма RSA.
- Поддерживается импорт сертификатов в кодировке DER и Base64.
- УЦ должен использовать алгоритм подписи SHA-2.

4 Получите из УЦ сертификат, скопируйте его на USB-носитель и подключите носитель к ViPNet Coordinator HW.

5 Импортируйте сертификат в локальное хранилище ViPNet Coordinator HW:

```
hostname# service cert import
```

6 Установите сертификат для использования службой webUI при доступе к ViPNet Coordinator HW по протоколу HTTPS:

```
hostname# webui https-cert <имя файла сертификата>
```

7 Если вы работали по протоколу HTTP, установите HTTPS для доступа к веб-интерфейсу ViPNet Coordinator HW:

```
hostname# webui protocol https
```

8 Вы можете изменить TCP-порт (по умолчанию — 8080):

8.1 Измените TCP-порт:

```
hostname# webui port <номер TCP-порта>
```

8.2 В фильтре Allow ViPNet WebGUI измените номер TCP-порта:

8.2.1 Удалите существующий сетевой фильтр:

```
hostname# firewall vpn delete tcp dport 8080
```

8.2.2 Добавьте новый сетевой фильтр:

```
hostname# firewall vpn add <номер> rule "Allow ViPNet WebGUI" src @any dst @local tcp dport <номер порта> pass
```

8.3 Перезапустите службу webUI:

```
hostname# webui restart
```

Если на шаге 8 вы изменили TCP-порт по умолчанию — в адресной строке браузера укажите:

`https://<IP-адрес ViPNet Coordinator HW>:<номер TCP-порта>.`

Добавление самоподписанного сертификата в браузер как доверенного

Если у вас нет возможности выпустить сертификат в доверенном УЦ, вы можете добавить самоподписанный сертификат в браузер как доверенный. Метод снижает защищенность компьютера, с которого происходит подключение к ViPNet Coordinator HW, так как все сертификаты, подписанные самоподписанным сертификатом ViPNet Coordinator HW, также будут доверенными.

- 1 Перевыпустите самоподписанный сертификат с указанием реальных и виртуальных IP-адресов и доменных имен (подробное описание команды см. в «Справочник команд и конфигурационных файлов»):

```
hostname# webui recreate-cert [bits <длина ключа> digest {sha256 | sha384 | sha512}  
span <срок действия> [subj <параметр>]]
```



Примечание. Если вы используете виртуальные IP-адреса:

Чтобы подключение было защищено, каждый раз при смене виртуальных IP-адресов перевыпускайте самоподписанный сертификат и добавляйте его в доверенные.

В зависимости от браузера, который вы используете, могут понадобиться дополнительные атрибуты сертификата. Например, для браузеров на базе Chromium необходимо указать параметр `subjectAltName`.

- 2 Экпортируйте сертификат из браузера.

Способ экспорта сертификата зависит от браузера. Пример для браузера Google Chrome:

- 2.1 Нажмите **Не защищено** > **Сертификат недействителен**.
- 2.2 В окне просмотра сертификата выберите вкладку **Подробнее** и нажмите **Экспорт**.
- 2.3 Выберите место для сохранения сертификата и нажмите **Сохранить**.

- 3 Добавьте сертификат в хранилище доверенных корневых центров сертификации.

Способ добавления и хранилище зависят от операционной системы. Пример для ОС Windows 10:

- 3.1 Запустите консоль управления:

3.1.1 Откройте диалоговое окно **Выполнить** (Win + R).

3.1.2 В строке **Открыть** введите `mmc` и нажмите **ОК**.

- 3.2 Добавьте оснастку сертификатов:

3.2.1 Нажмите **Файл** > **Добавить или удалить оснастку**.

3.2.2 В списке **Доступные оснастки** выберите **Сертификаты** и нажмите **Добавить**.

3.2.3 В окне **Оснастка диспетчера сертификатов** выберите **учетной записи компьютера** и нажмите **Далее**.

3.2.4 В окне **Выбор компьютера** выберите **локальным компьютером** и нажмите **Готово**.

3.2.5 В окне **Добавление и удаление оснасток** нажмите **ОК**.

- 3.3 Добавьте сертификат в доверенные:

3.3.1 На панели справа выберите **Сертификаты (локальный компьютер) > Доверенные корневые центры сертификации > Сертификаты**.

3.3.2 Нажмите правой кнопкой мыши на **Сертификаты** и выберите **Все задачи > Импорт**.

Откроется мастер импорта сертификатов.

3.3.3 В мастере импорта выберите самоподписанный сертификат ViPNet Coordinator HW.

3.4 Закройте консоль управления (MMC) и перезапустите браузер.

Перевыпуск самоподписанного сертификата

Если для подключения к веб-интерфейсу вы используете самоподписанный сертификат, то его можно перевыпустить, например, если закончился срок его действия:

```
hostname# webui recreate-cert [bits <длина ключа> digest {sha256 | sha384 | sha512} span  
<срок действия> [subj <параметр>]]
```

Длину ключа, алгоритм хэширования и дополнительные параметры задайте в соответствии с требованиями вашей организации. Подробное описание команды смотрите в документе «Справочник команд и конфигурационных файлов» раздел «webui recreate-cert».

Если вы выполните команду без параметров, будут использоваться значения, аналогичные используемым на этапе инициализации ViPNet Coordinator HW:

- длина ключа — 2048;
- алгоритм хэширования — sha256;
- срок действия — 1825 дней;
- имя субъекта — имя узла.



Внимание! После перевыпуска самоподписанный сертификат назначается действующим для подключения к веб-интерфейсу ViPNet Coordinator HW по протоколу HTTPS.

Если до этого служба веб-интерфейса работала по протоколу HTTP, установите для нее протокол HTTPS: `hostname# webui protocol https`

Подключение по HTTP

Чтобы подключаться к веб-интерфейсу ViPNet Coordinator HW по HTTP, выполните:

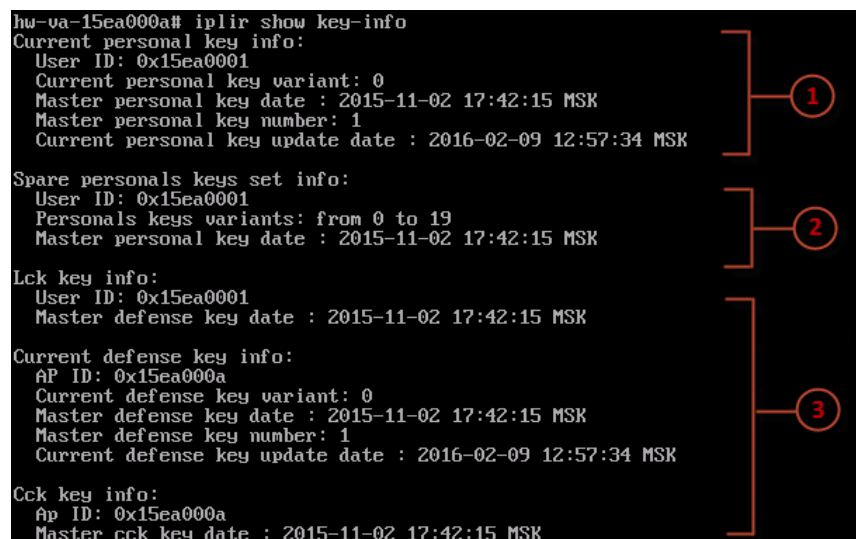
```
hostname# webui protocol http
```

Просмотр информации об установленных ключах

Информация об установленных ключах может потребоваться вам перед [добавлением РНПК](#) на ViPNet Coordinator HW — чтобы убедиться, что РНПК отсутствует на узле. Остальная информация может быть востребована сотрудниками технического сопровождения при возникновении проблем с ключевой системой ViPNet Coordinator HW (например, после некорректного обновления справочников и ключей на узле).

Чтобы посмотреть информацию о ключах, выполните:

```
hostname> iplir show key-info
```



```
hw-va-15ea000a# iplir show key-info
Current personal key info:
  User ID: 0x15ea0001
  Current personal key variant: 0
  Master personal key date : 2015-11-02 17:42:15 MSK
  Master personal key number: 1
  Current personal key update date : 2016-02-09 12:57:34 MSK

Spare personals keys set info:
  User ID: 0x15ea0001
  Personals keys variants: from 0 to 19
  Master personal key date : 2015-11-02 17:42:15 MSK

Lck key info:
  User ID: 0x15ea0001
  Master defense key date : 2015-11-02 17:42:15 MSK

Current defense key info:
  AP ID: 0x15ea000a
  Current defense key variant: 0
  Master defense key date : 2015-11-02 17:42:15 MSK
  Master defense key number: 1
  Current defense key update date : 2016-02-09 12:57:34 MSK

Cck key info:
  Ap ID: 0x15ea000a
  Master cck key date : 2015-11-02 17:42:15 MSK
```

Рисунок 50. Просмотр информации о ключах

Цифрами на рисунке обозначены:

- 1 Информация о текущем [персональном ключе пользователя](#).
- 2 Информация о [РНПК](#).



Примечание. В этом списке может быть указано `Spare key set is not present`. Это означает, что РНПК отсутствует на узле. О том, как добавить РНПК на ViPNet Coordinator HW, см. в разделе [Добавление резервного набора персональных ключей \(РНПК\)](#).

- 3 Информация о [ключях узла](#).

Обновление справочников и ключей

Удаленное обновление из управляющего ПО

Если в ViPNet ЦУС или Prime была изменена структура сети ViPNet или свойства сетевых узлов, требуется передать эту информацию на сетевые узлы. Для этого администратор ViPNet ЦУС или Prime отправляет на сетевые узлы новые справочники и ключи. На ViPNet Coordinator HW обновления применяются автоматически.



Примечание. Если обновления справочников и ключей поступили на ViPNet Coordinator HW в незашифрованном транспортном конверте, или конверт не проходит проверку подлинности, обновления не будут применены. При этом на узел с ViPNet ЦУС или Prime будет отправлена [транспортная квитанция](#) с символом «!» в начале имени файла.

Локальное обновление

Если у вас нет возможности обновить справочники и ключи удаленно, вы можете сделать это локально с помощью дистрибутива ключей. Для этого:

- 1 Получите от администратора сети ViPNet новый дистрибутив ключей (файл *.dst) и пароль от него.
- 2 Запишите дистрибутив ключей на внешний носитель (USB или CD).



Примечание. Если РНПК включен в новый дистрибутив ключей, он будет обновлен на ViPNet Coordinator HW.

Чтобы обновить справочники и ключи локально на ViPNet Coordinator HW:

- 1 Выполните команду:

```
hostname# admin keys update
```
- 2 Подтвердите процесс обновления справочников и ключей.
Будут остановлены все службы ViPNet.
- 3 Выберите способ обновления:
В строке `Would you like to install keys from USB or CD storage device? [u/c] or press ctrl+C to abort` введите:
 - o u — с USB-носителя.
 - o c — с CD-диска.
- 4 Подключите USB-носитель или CD-привод к ViPNet Coordinator HW; вставьте CD-диск в привод.

Если на USB-носителе или CD-диске обнаружен только один файл *.dst, он будет выбран автоматически.

Если обнаружено несколько файлов *.dst, введите номер файла из списка и нажмите **Enter**.

5 Введите пароль к файлу.

После обновления справочников и ключей сессия пользователя завершится. Для продолжения работы войдите повторно.



Внимание! Пароли пользователя и администратора могли поменяться. Пароль пользователя будет совпадать с паролем дистрибутива ключей, который вы использовали для обновления.

Уточните пароли у администратора сети ViPNet.

Локальное обновление в режиме кластера горячего резервирования

Порядок обновления ключей и справочников на ViPNet Coordinator HW в режиме кластера горячего резервирования зависит от пароля пользователя:

1 Если пароль пользователя совпадает с паролем дистрибутива ключей:

Обновите ключи и справочники на активном узле кластера горячего резервирования. На пассивном узле ключи и справочники обновятся автоматически.

2 Если пароль пользователя не совпадает с паролем дистрибутива ключей:

Примечание. Пароль пользователя может отличаться от пароля дистрибутива ключей, если:



- после получения дистрибутива, использованного для первичной инициализации, в УКЦ был изменен пароль пользователя узла;
- на ViPNet Coordinator HW был изменен пароль пользователя командой `admin passwd`;
- в качестве управляющего ПО используется ViPNet Prime.

2.1 Остановите службу failoverd на пассивном узле:

```
hostname# failover stop
```

2.2 Выключите интерфейс синхронизации на пассивном узле:

```
hostname# inet ifconfig <интерфейс> down
```



Примечание. Имя интерфейса синхронизации пассивного узла вы можете посмотреть на активном узле в конфигурационном файле `failover.ini` в секции `[sendconfig]`.

2.3 Выполните локальное обновление ключей и справочников на пассивном узле кластера.

2.4 Выполните локальное обновление ключей и справочников на активном узле кластера.

2.5 Включите интерфейс синхронизации на пассивном узле:

```
hostname# inet ifconfig <интерфейс> up
```

2.6 Запустите службу failoverd на пассивном узле:

```
hostname# failover start
```

Добавление резервного набора персональных ключей (РНПК)

Когда требуется добавлять РНПК

В процессе эксплуатации сети ViPNet администратор может выполнять обновление ключей узлов в случае их компрометации и при смене мастер-ключа персональных ключей. В процессе данных операций меняется персональный ключ пользователя — берется новый ключ из [резервного набора персональных ключей \(РНПК\) пользователя](#) и с его помощью выполняется перешифрование всех остальных ключей. В итоге при централизованном обновлении ключи на узел отправляются зашифрованными на новом персональном ключе. Чтобы ключи могли быть приняты, на узле должен присутствовать РНПК. Иначе расшифрование ключей будет невозможно, и их придется устанавливать вручную. Таким образом, РНПК, будучи заранее переданным пользователю, позволяет в указанных случаях дистанционно обновить ключи без необходимости высылать пользователю новый персональный ключ по скомпрометированному каналу.

Как правило, на ViPNet Coordinator HW файл с РНПК попадает в составе [дистрибутива ключей](#). Но может возникнуть ситуация, когда его может не быть на узле. Например, если вы для восстановления работоспособности узла или обновления справочников и ключей повторно развернули дистрибутив ключей, в котором не было РНПК. В этом случае вы можете добавить файл с РНПК на ViPNet Coordinator HW вручную.

Порядок добавления РНПК

Внимание! Добавить РНПК на ViPNet Coordinator HW вы можете только при использовании ViPNet Administrator в качестве управляющего ПО.

Если на ViPNet Coordinator HW зарегистрировано несколько пользователей, убедитесь, что идентификатор пользователя User ID совпадает для персональных ключей пользователя и для РНПК:



- Выполните команду `iplir show key-info`.
- Сравните параметр User ID в секциях `Current personal key info` и `Spare personals keys set info`.

Если параметр в секциях не совпадает, добавьте РНПК для пользователя, ключи которого установлены на ViPNet Coordinator HW.

Для добавления [резервного набора персональных ключей](#) на ViPNet Coordinator HW:

- 1 Проверьте, что на ViPNet Coordinator HW отсутствует файл с РНПК (*.pk):

```
hostname# iplir show key-info
```

При отсутствии файла с РНПК на узле будет выдана следующая информация:

Spare key set is not present.

- 2 Запросите РНПК у администратора сети ViPNet. Администратор должен передать вам файл с РНПК на отдельном внешнем устройстве лично в руки либо другим доверенным способом. Файл РНПК на устройстве будет защищен паролем пользователя ViPNet Coordinator HW, который задан в ViPNet ЦУС.



Внимание! Если в процессе использования ViPNet Coordinator HW вы изменяли пароль пользователя и не помните пароль, заданный администратором сети ViPNet в ViPNet ЦУС, запросите его у администратора. Без данного пароля вы не сможете добавить файл с РНПК на узел.

- 3 Выполните команду:

```
hostname# admin add spare keys
```

Будет дополнительно выполнена проверка наличия файла с РНПК. Если файл отсутствует, будет автоматически предложено указать внешнее устройство, на котором содержится файл РНПК.

- 4 В зависимости от того, на каком внешнем устройстве вам был передан файл с РНПК:

- подключите USB-носитель к ViPNet Coordinator HW;
- вставьте CD (DVD)-диск в оптический привод ViPNet Coordinator HW.

После этого подтвердите подключение.

Будет произведен поиск файлов с РНПК и проверка идентификатора пользователя в каждом найденном файле:

- если подходящих файлов нет, появится соответствующее сообщение;
- если файлы найдены, и идентификатор пользователя в файлах и на узле совпадает, то появится список файлов для выбора.

- 5 Введите номер файла из предложенного списка и нажмите **Enter**.

- 6 Введите пароль пользователя ViPNet Coordinator HW, которым защищен файл с РНПК, и нажмите **Enter**. Пароль к РНПК будет отличаться от пароля пользователя, если вы меняли последний.

При успешном вводе пароля файл РНПК будет сохранен на ViPNet Coordinator HW и перешифрован на парольном ключе, отличном от пароля пользователя. Это сделано для того, чтобы в дальнейшем при обращении к РНПК не требовалось вводить пароль каждый раз.

Обновление ключей при истечении срока их действия

Персональный ключ пользователя, который входит в состав ключей пользователя, а также ключ защиты ключей обмена, который входит в состав ключей узла, имеют ограниченный срок действия — 12 месяцев. ViPNet Coordinator HW проверяет срок действия этих ключей с момента их установки или после обновления. За месяц, неделю и по истечении срока действия ключей появляется соответствующее сообщение. Кроме этого, информация об истечении ключей добавляется в системный журнал.

Как правило, персональный ключ и ключ защиты ключей обмена устанавливаются и обновляются на узле одновременно, поэтому даты начала и окончания срока их действия чаще всего будут совпадать.

При истечении срока действия ключей никакие ограничения на работу ViPNet Coordinator HW не накладываются. Но настоятельно рекомендуется обновить ключи, для чего следует обратиться к администратору сети ViPNet.

Чтобы обновить ключи:

1 Посмотрите информацию об установленных ключах:

1.1 Выполните команду:

```
hostname# iplir show key-info
```

1.2 В выводе команды посмотрите и запомните значения параметров `key variant` и `key number` соответствующих ключей.

1.3 Если в качестве управляющего ПО вы используете ViPNet Administrator, убедитесь, что на узле присутствует резервный набор персональных ключей (РНПК).



Внимание! При отсутствии РНПК на ViPNet Coordinator HW обновление ключей не сможет быть выполнено корректно. Кроме этого, по завершении такого обновления ключей потребуются повторная установка справочников и ключей для восстановления работоспособности ViPNet Coordinator HW.

2 После вашего обращения администратор сети ViPNet должен:

2.1 Поднять вариант ключей пользователя при обновлении персонального ключа, ключей узла — при обновлении ключа защиты ключей обмена.

2.2 После изменения варианта ключей сформировать и выслать обновления ключей на ViPNet Coordinator HW. Если удаленное обновление невозможно, администратор должен передать новый дистрибутив ключей и пароль от него.

- 2.3 В случае обновления персонального ключа — перезаписать персональный ключ на защищенный USB-носитель (токен), если пользователь ViPNet Coordinator HW производит аутентификацию с помощью устройства.
- 3 На ViPNet Coordinator HW примите поступившие обновления ключей или [обновите ключи самостоятельно](#).
- 4 Проверьте правильность установки ключей:
 - 4.1 Выполните команду:

```
hostname# iplir show key-info
```
 - 4.2 Проверьте, что в выводе команды поменялись значения параметров `key variant` и `key number` соответствующих ключей. Если значения не поменялись, обратитесь к администратору сети.

Контроль целостности файлов

ViPNet Coordinator HW

Чтобы не было сбоев в работе ПО ViPNet Coordinator HW, целостность критически важных файлов регулярно проверяется встроенными скриптами. Контролируются следующие компоненты:

- ядро Linux и образы модулей ViPNet Coordinator HW, используемые при загрузке ПО;
- системные службы и библиотеки;
- конфигурационные файлы служб ViPNet Coordinator HW;
- [справочники и ключи ViPNet](#).

Указанные файлы защищены контрольными суммами, которые хранятся в файлах с расширением *.crg. Проверка целостности заключается в вычислении текущей контрольной суммы файла и ее сравнения с контрольной суммой из файла *.crg.

Каждый файл *.crg содержит также собственную контрольную сумму. Проверка контрольной суммы файла *.crg выполняется перед проверкой целостности остальных файлов, контрольная сумма которых в нем указана.

Проверка целостности файлов автоматически выполняется в следующих случаях:

- Установка ПО ViPNet Coordinator HW.
- Обновление ПО ViPNet Coordinator HW.
- Загрузка ПО ViPNet Coordinator HW.
- Периодическая проверка с интервалом 12 часов (в этом случае конфигурационные файлы ViPNet Coordinator HW, а также справочники и ключи ViPNet не проверяются).



Примечание. При каждой периодической проверке в системный журнал заносится запись, содержащая время и результаты этой проверки. Чтобы просмотреть информацию о последней периодической проверке, выполните:

```
hostname# admin show check integrity status
```

Если команда выполнена до первой периодической проверки целостности файлов, то появится сообщение об отсутствии результатов проверки целостности (No integrity check information).

Кроме того, вы можете проверить целостность конфигурационных файлов вручную:

```
hostname# machine self-test
```

До начала тестирования работа всех служб будет автоматически завершена, а после успешного окончания тестирования — автоматически восстановлена.



Примечание. Команду проверки целостности конфигурационных файлов нельзя использовать при удаленном подключении по SSH.

Нарушением целостности файлов также считается отсутствие хотя бы одного файла конфигурации или файла с контрольными суммами.

При несовпадении контрольных сумм в ходе периодической проверки ViPNet Coordinator HW автоматически перезагружается.

При несовпадении контрольных сумм в ходе проверки во время установки, обновления или загрузки ПО, а также в ходе проверки, выполняемой вручную, выводится сообщение о нарушении целостности файла. Затем выполняется одно из следующих действий:

- Если для искаженного файла имеется резервная копия, файл будет автоматически восстановлен. При этом будет выведено соответствующее сообщение. При восстановлении файла ViPNet Coordinator HW может быть перезагружен.
- Если резервная копия искаженного конфигурационного файла не найдена, будет выведено соответствующее сообщение. После этого службы, отвечающие за работу в сети ViPNet, будут остановлены и сетевые интерфейсы будут выключены.



Совет. Если у вас есть файл экспорта справочников, ключей и настроек с расширением *.vbe, попробуйте восстановить систему, импортировав этот файл (подробнее см. раздел [Резервное копирование и восстановление настроек](#)). Если восстановить работу ViPNet Coordinator HW не удалось, обратитесь в службу поддержки ИнфоТеКС.

- Если при загрузке обнаружено нарушение целостности ядра Linux или одного из образов модулей ViPNet Coordinator HW, и соответствующая резервная копия не найдена, то будет выведено соответствующее сообщение и загрузка ViPNet Coordinator HW завершится. Для выключения ViPNet Coordinator HW нажмите любую клавишу.

12

Настройка системных параметров

Настройка даты и времени	250
Установка серийного номера	251
Настройка расписания перезагрузки	252
Настройка параметров использования файла подкачки	253
Установка режима повышенной производительности	254
Управление копиями конфигурации VPN	255
Резервное копирование и восстановление настроек	259
Использование резервного раздела в исполнениях ViPNet Coordinator HW с двумя накопителями	263
Сброс настроек ViPNet Coordinator HW к заводским	264

Настройка даты и времени



Внимание! Если системные дата и время заданы неверно, защищенные соединения с другими узлами ViPNet могут блокироваться. Рекомендуем настраивать синхронизацию системного времени по протоколу NTP.

Настройка даты и времени вручную:

1 Посмотрите текущие дату и время:

```
hostname# machine show date
```

2 Чтобы изменить часовой пояс:

```
hostname# machine set timezone {<Континент/Пояс> | <UTC>}
```

Название континента и часового пояса должны начинаться с прописной буквы. Например, чтобы установить часовой пояс Москвы:

```
hostname# machine set timezone Europe/Moscow
```

Чтобы посмотреть список всех часовых поясов, введите команду без параметра.

3 Чтобы изменить дату и время:

```
hostname# machine set date <YYYY-MM-DD> <hh:mm:ss>,
```

где YYYY — год, MM — месяц, DD — день, hh — часы, mm — минуты, ss — секунды.

Например:

```
hostname# machine set date 2024-02-22 12:00:00
```

```
Do you really wish to change the date?
```

```
Continue? [Yes/No]: Yes
```

Настраивайте системную дату, время и часовой пояс только на активном узле кластера.

Установка серийного номера

Начиная с версии 4.5.6 ПО ViPNet Coordinator HW поддерживает серийные номера устройства для исполнений на аппаратных платформах. Для новых устройств номер устанавливается на производстве и не может быть изменен. Для устройств с ПО, обновленных пользователем, номер рекомендуется установить вручную в соответствии с указанным на этикетке устройства.



Примечание. Серийный номер указан на этикетке infotecs, которая находится на корпусе устройства, а также записан в формуляр. Если серийный номер был утерян обратитесь в службу поддержки ИнфоТеКС.

Серийный номер может быть длиной 8-9 символов и содержать:

- Арабские цифры: 0–9.
- Специальный символ – .

Серийный номер имеет формат XXX-XXXXX(X).

Используйте серийный номер:

- При обращении в службу поддержки ИнфоТеКС.
- Для инвентаризации оборудования: серийный номер доступен по SNMP (OID .1.3.6.1.4.1.10812.1.10.1. в [VIPNET-MIB](#)).

Чтобы проверить серийный номер:

```
hostname# version

Product: ViPNet Coordinator HW

Platform: HW1000C

Serial number: Not set

License: HW1000C

Software version: 4.5.8-4550
```

Если номера нет, вы можете установить его:

```
hostname# machine serial <серийный номер>
```



Примечание. Если вы ввели неверный серийный номер выполните команду еще раз.

Настройка расписания перезагрузки

- 1 Включите автоматическую перезагрузку по расписанию:

```
hostname# machine reboot-schedule on
```

По умолчанию перезагрузка по расписанию выключена.

- 2 Выполните одну из команд:

- Если необходимо настроить перезагрузку в определенный день недели:

```
hostname# machine reboot-schedule time <время> day <день>
```

Вы можете указать только один день недели (sunday, monday, tuesday, wednesday, thursday, friday или saturday).

- Если необходимо настроить ежедневную перезагрузку:

```
hostname# machine reboot-schedule time <время>
```

Время указывается в формате hh:mm, где hh — часы (24-часовой формат), mm — минуты.

Чтобы просмотреть текущие настройки автоматической перезагрузки:

```
hostname> machine reboot-schedule show
```

Чтобы перезагрузить ViPNet Coordinator HW вручную:

```
hostname> machine reboot
```

Настройка перезагрузки в кластере

Расписание перезагрузки (дата и время) не синхронизируется между узлами кластера.

Настройку автоматической перезагрузки выполняйте в следующем порядке:

- 1 Настройте расписание перезагрузки на активном узле кластера.
- 2 Дождитесь, когда данные на активном и пассивном узле синхронизируются (по умолчанию — 60 секунд).
- 3 Подключитесь к пассивному узлу и проверьте настройки автоматической перезагрузки. Автоматическая перезагрузка должна быть включена.
Если настройки не синхронизировались, проверьте их чуть позже.
- 4 Настройте время и дату перезагрузки на пассивном узле. Чтобы не допустить одновременной перезагрузки обоих узлов кластера, устанавливайте время перезагрузки на узлах с разницей не менее 30 минут.

Настройка параметров использования файла подкачки

Чтобы увеличить производительность ViPNet Coordinator HW за счет оптимизации скорости его процессов, настройте использование на нем [файла подкачки](#).



Внимание! Использование файла подкачки в исполнениях с одним дисковым накопителем невозможно.

Чтобы настроить параметры файла подкачки:

1. Задайте размер файла подкачки:

```
hostname# machine swap set <size>,
```

где <size> — размер файла подкачки в Мбайтах.

Если будет задан размер файла подкачки, превышающий размер доступного пространства на диске, появится соответствующее сообщение.



Внимание! После создания файла подкачки на диске должно остаться не менее 256 Мбайт свободного пространства.

2. Включите использование файла подкачки:

```
hostname# machine swap mode on
```

3. Посмотрите сведения об использовании оперативной памяти и файле подкачки:

```
hostname# machine show memory
```

Установка режима повышенной производительности

При обработке одного сетевого подключения можно настроить повышенную скорость обработки и передачи защищенного трафика.

По умолчанию ViPNet Coordinator HW работает в многопоточном режиме, который нацелен на быструю обработку нескольких сетевых соединений. Потоки шифрования в этом случае разбиты на несколько групп.

При включении режима повышенной производительности происходит перераспределение ресурсов ViPNet Coordinator HW: все аппаратные прерывания (IRQ) от сетевых адаптеров выполняются одним ядром процессора, а шифрование выполняется оставшимися ядрами.



Внимание! При включенном режиме в случае обработки большого количества подключений скорость обработки и передачи защищенного трафика будет снижена.

Режим повышенной производительности можно включить на следующих платформах:

- ViPNet Coordinator HW100 N1, N2, N3;
- ViPNet Coordinator HW1000 Q10;
- ViPNet Coordinator HW2000 Q4, Q5;
- ViPNet Coordinator HW5000 Q1, Q2.

Для включения режима повышенной производительности:

```
hostname# iplir set performance-mode single
```

Режим повышенной производительности применится после перезагрузки ViPNet Coordinator HW. Вы можете согласиться на перезагрузку во время задания режима, или отложить ее.

Для просмотра установленного режима:

```
hostname# iplir show performance-mode
```

В случае если при включении режима вы не выполнили перезагрузку ViPNet Coordinator HW, отобразится как текущее, так и планируемое состояние после перезагрузки.

Управление копиями конфигурации VPN

Виды копий конфигурации VPN

Копия конфигурации VPN — совокупность настроек ViPNet Coordinator HW. С помощью нее вы можете вернуться к другому состоянию ViPNet Coordinator HW (например, к состоянию до выполнения настроек или до приема обновленных справочников и ключей).

Существуют два вида копий конфигурации VPN:

- **частичная** — включает в себя следующие конфигурационные файлы ПО ViPNet Coordinator HW: `iplir.conf` и все файлы с именем, соответствующим маске `iplir.conf-<интерфейс или группа интерфейсов>`, `failover.ini`, `mftp.conf`;
- **полная** — включает в себя все файлы частичной копии конфигурации VPN, а также сетевые фильтры, [справочники](#) и [ключи](#).

Копии конфигурации VPN могут создаваться следующими способами:

- Автоматически (перед обновлением ПО или справочников и ключей). В этом случае по умолчанию создаются частичные копии конфигурации VPN. Вы можете изменить вид копии конфигурации VPN, создаваемой автоматически, или выключить автоматическое создание копий конфигурации VPN, отредактировав значение параметра `confsave` в секции `[upgrade]` в файле `mftp.conf`.

Кроме того, с помощью параметра `maxautosaves` этой же секции вы можете задать максимальное число автоматически созданных копий конфигурации VPN, которое может храниться на ViPNet Coordinator HW. По умолчанию может храниться не более 10 автоматически созданных копий конфигурации VPN.

Описание параметров секции `[upgrade]` см. в документе «Справочник команд и конфигурационных файлов», в разделе «Секция `[upgrade]`».

- Вручную по команде. В этом случае всегда создаются только полные копии конфигурации VPN.

Создание копии конфигурации VPN



Примечание. По умолчанию перед обновлением ПО или справочников и ключей автоматически создается частичная копия конфигурации VPN (см. [Виды копий конфигурации VPN](#)). Вы можете изменить эту настройку в секции `[upgrade]` файла `mftp.conf` (подробнее см. в документе «Справочник команд и конфигурационных файлов»).

Чтобы вручную создать копию текущей конфигурации VPN:

1 Завершите работу служб `iplircfg`, `failoverd` и `mftpd`:

```
hostname# iplir stop
hostname# failover stop
hostname# mftp stop
```

2 Выполните команду:

```
hostname# admin config save <имя>
```

и укажите имя копии конфигурации VPN, используя только буквы латинского алфавита, цифры, символы «-» и «_».

3 Если копия конфигурации VPN с выбранным именем уже существует, будет предложено ее перезаписать:

```
Config with name '<имя копии конфигурации VPN>' (version <версия ViPNet Coordinator HW>)
already exists! Overwrite it?
```

Выполните одно из действий:

- Чтобы подтвердить перезапись, введите `y` и нажмите **Enter**.
- Чтобы создать конфигурацию с другим именем, введите `n` и нажмите **Enter**.

4 Запустите службы `iplircfg`, `failoverd` и `mftpd`:

```
hostname# iplir start
hostname# failover start
hostname# mftp start
```

Просмотр списка копий конфигурации VPN

Чтобы просмотреть список ранее созданных копий конфигурации VPN:

```
hostname# admin config list
```

В результате отобразится список, каждая строка которого содержит следующую информацию:

- имя копии;
- версия ViPNet Coordinator HW, в которой была создана копия конфигурации VPN (после слова `version`);
- вид копии конфигурации VPN — полная (`full`) или частичная (`part`);
- дата и время создания копии конфигурации VPN (после слова `saved`).

Если копия конфигурации VPN была создана автоматически, то ее имя начинается со слова `autosave` и содержит дату создания. Например: `autosave-2019-11-12`.

Если копия конфигурации VPN уже использовалась для восстановления настроек ViPNet Coordinator HW, дата и время использования отображаются после даты и времени создания копии

конфигурации VPN (после слова `loaded`). Если копия конфигурации VPN ни разу не использовалась, после даты и времени создания будет указано `never loaded`.

Восстановление настроек из копии конфигурации VPN

- 1 Завершите работу всех служб и драйверов ViPNet Coordinator HW:

```
hostname# iplir stop
hostname# failover stop
hostname# mftp stop
```

- 2 Выполните команду по загрузке конфигурации:

```
hostname# admin config load <имя копии конфигурации> [<версия ViPNet Coordinator HW>],
```

указав имя копии конфигурации VPN и версию ViPNet Coordinator HW, в которой она была создана. Если имя копии конфигурации VPN состоит из нескольких слов, заключите его в кавычки.

- 3 Если текущая версия ViPNet Coordinator HW выше версии ViPNet Coordinator HW, в которой была создана выбранная для восстановления настроек копия конфигурации VPN, подтвердите возврат к более ранней версии.

- 4 Будет предложено сохранить все текущие настройки ViPNet Coordinator HW:

```
Save current configuration [Yes/No]?
```

- Чтобы сохранить все текущие настройки, введите `y` и нажмите **Enter**. Затем укажите имя соответствующей полной копии конфигурации VPN.
- Чтобы не сохранять текущие настройки ViPNet Coordinator HW, введите `n` и нажмите **Enter**. Для подтверждения действия введите фразу `Yes, do as I say`.

- 5 Дождитесь завершения процесса восстановления настроек ViPNet Coordinator HW.

- 6 Запустите службы и драйверы ViPNet Coordinator HW:

```
hostname# iplir start
hostname# failover start
hostname# mftp start
```

Теперь вы можете продолжить работу с ViPNet Coordinator HW.

Удаление копии конфигурации VPN

Чтобы удалить ненужные копии конфигурации VPN:

```
hostname# admin config delete <имя копии конфигурации> [<версия ViPNet Coordinator HW>],
```

указав имя копии конфигурации VPN и версию ПО ViPNet Coordinator HW, в которой она была создана.

Чтобы не писать имя копии конфигурации VPN полностью или удалить сразу несколько копий конфигурации VPN, имена которых содержат определенные символы, вместо имени укажите маску имени, используя символ `*`.

Например, чтобы удалить все копии конфигурации VPN, созданные автоматически в августе 2020 года:

```
hostname# admin config delete rollback-2020-08-*
```

Если несколько копий конфигурации VPN имеют имя, соответствующие указанной маске, то будет отображено количество таких копий конфигурации VPN:

```
<количество копий конфигурации VPN> configs match that pattern. Delete? [Yes/No]
```

Выполните одно из действий:

- Чтобы подтвердить удаление всех копий конфигурации VPN, введите `y` и нажмите **Enter**.
- Чтобы отменить удаление всех копий конфигурации VPN, введите `n` и нажмите **Enter**.

Чтобы удалить копии конфигурации VPN, созданные в определенной версии VPN-компонента ViPNet Coordinator HW, укажите номер этой версии. Например:

```
hostname# admin config delete rollback-2020-08-* 4.2.0
```

Резервное копирование и восстановление настроек

Назначение экспорта и импорта справочников, ключей и настроек

Вы можете использовать экспорт и импорт справочников, ключей и настроек ViPNet Coordinator HW в следующих ситуациях:

- Сохранение и восстановление данных на ViPNet Coordinator HW в случае некорректного обновления ПО.
- Перенос справочников, ключей и настроек с одного ViPNet Coordinator HW на другой, например, в случае замены неисправного экземпляра. Это позволяет упростить подготовку нового ViPNet Coordinator HW к работе, так как не требуется выполнять установку справочников и ключей и необходимые настройки вручную.

В этом случае оба ViPNet Coordinator HW должны иметь одинаковые исполнение и тип лицензии. Кроме этого, по завершении импорта координатор, с которого переносятся данные, должен быть отключен от сети или на нем должны быть удалены справочники и ключи. Это связано с тем, что в сети ViPNet не допускается функционирование нескольких узлов с одинаковыми справочниками и ключами.

В состав экспортируемой информации входят следующие данные:

- файлы, содержащие ключи шифрования для связанных с ViPNet Coordinator HW узлов;
- [резервный набор персональных ключей](#) (файл *.pk, при наличии);
- файлы, содержащие информацию о связанных с ViPNet Coordinator HW узлах (справочники);
- настройки защищенной сети (файл `iplir.conf`);
- настройки сетевых интерфейсов (IP-адрес, маска подсети, файл `iplir.conf`-<интерфейс>);
- настройки маршрутизации и статические маршруты, если такие созданы;
- настройки часового пояса;
- сетевые фильтры и правила трансляции адресов;
- настройки параметров обработки прикладных протоколов;
- настройки системы защиты от сбоев (файл `failover.ini`);
- настройки транспортного модуля (файл `mftp.conf`);
- настройки протоколирования;
- настройки дополнительных сервисов (DHCP-сервера и других);

- настройки взаимодействия с ИБП;
- настройки туннелирования канального уровня (L2OverIP);
- текущие ключи для соединений по протоколу SSH2.

Экспорт справочников, ключей и настроек

Экспорт справочников, ключей и настроек вы можете выполнить только при непосредственном доступе к ViPNet Coordinator HW одним из следующих способов:

- На ноутбук, подключенный через сетевой кабель Ethernet и технологический адрес, по протоколу TFTP.
- На USB-носитель при подключении через обычную консоль (с использованием монитора и клавиатуры) или COM-консоль (с использованием ноутбука). Перед началом экспорта этим способом отформатируйте USB-носитель в одну из поддерживаемых файловых систем: FAT32, ext2, ext3 или ext4.



Внимание! Экспортировать справочники, ключи и настройки в удаленной SSH-сессии запрещено.

На время выполнения экспорта с использованием ноутбука (после ввода команды) автоматически изменяются настройки сетевых интерфейсов ViPNet Coordinator HW: на интерфейсе Ethernet1 устанавливается технологический IP-адрес 169.254.241.1, все остальные интерфейсы ViPNet Coordinator HW выключаются. Восстановление настроек интерфейсов ViPNet Coordinator HW производится только при успешном завершении экспорта.



Примечание. Чтобы получить работоспособный файл экспорта, предварительно проведите регламентное тестирование ViPNet Coordinator HW: `hostname# machine self-test`.

Для экспорта справочников, ключей и настроек:

1 Выполните одно из действий:

- Если вы экспортируете на USB-носитель, завершите работу служб `iplircfg` и `mftpd`:

```
hostname# iplir stop
```

```
hostname# mftp stop
```
- Если вы экспортируете на ноутбук, выгрузите драйверы и завершите работу всех служб ViPNet Coordinator HW:

```
hostname# vpn stop
```

2 Если вы экспортируете справочники, ключи и настройки с пассивного узла, остановите на нем службу `failoverd`:

```
hostname# failover stop
```


3 Выберите место сохранения файла экспорта:

hostname# admin export keys binary-encrypted {tftp | usb}, где:

- o tftp — экспорт на ноутбук по протоколу TFTP.
- o usb — экспорт на USB-носитель.

При успешном выполнении команды появится сообщение, что файл экспорта сформирован и сохранен в каталоге /tmp/vipnet:

Configuration file will be saved to /tmp/vipnet/<имя файла экспорта>



Примечание. Имя файла экспорта формируется по шаблону <название ViPNet Coordinator HW>-%Y-%m-%d-%H-%M-%S.vbe. По умолчанию <название ViPNet Coordinator HW> включает <платформа ViPNet Coordinator HW>-<идентификатор узла>.

Файл экспорта автоматически зашифровывается на текущем пароле пользователя.

4 Перенесите сформированный файл экспорта на ноутбук или USB-носитель.

Для переноса файла на ноутбук установите вручную на сетевом интерфейсе ноутбука технологический IP-адрес 169.254.241.5, после чего выполните на ноутбуке команду:

```
tftp -i 169.254.241.1 get <имя файла экспорта>
```

Для переноса файла на USB-носитель:

4.1 Подключите USB-носитель к ViPNet Coordinator HW или компьютеру, на котором развернут виртуальный образ ViPNet Coordinator VA в ответ на предложение:

```
Put <имя файла экспорта> file onto USB drive.
```

```
Insert USB drive with at least 64KB free space and press Enter.
```

4.2 Появится список обнаруженных устройств и сообщение Select target partition [1-1] or 0 to abort.

Введите цифру, соответствующую USB-носителю в списке, и нажмите **Enter**.

4.3 При успешной записи файла экспорта на USB-носитель появится сообщение с разрешением извлечь устройство:

```
You may remove the USB drive.
```

5 Выполните одно из действий:

- o Если вы экспортировали на USB-носитель, запустите службы iplircfg и mftpd:

```
hostname# iplir start
```

```
hostname# mftp start
```

- o Если вы экспортировали на ноутбук, загрузите драйверы и запустите все службы ViPNet Coordinator HW:

```
hostname# vpn start
```

6 Если вы экспортировали справочники, ключи и настройки с пассивного узла, запустите на нем службу failoverd:

```
hostname# failover start
```

Импорт справочников, ключей и настроек

Импорт справочников, ключей и настроек требуется выполнять в следующих случаях:

- Для восстановления данных при некорректном обновлении ПО на ViPNet Coordinator HW (см. [Обновление программного обеспечения](#)).
- Для загрузки справочников, ключей и настроек с одного ViPNet Coordinator HW на другой (например, при замене аппаратной платформы).

Импорт справочников, ключей и настроек происходит так же, как первичная установка ключей (см. документ «Подготовка к работе»).

Перед началом импорта убедитесь в наличии файла *.vbe, полученного в результате экспорта справочников, ключей и настроек на действующем ViPNet Coordinator HW. Вместе с файлом *.vbe вам должен быть предоставлен пароль, на котором он зашифрован. Данный пароль потребуется ввести при выполнении импорта данных из файла *.vbe.



Примечание. При экспорте файл *.vbe автоматически зашифровывается на текущем пароле пользователя.

Кроме этого, если импорт справочников, ключей и настроек выполняется на ViPNet Coordinator HW, на котором уже установлены справочники и ключи, то удалите их перед началом импорта:

```
hostname# admin keys remove
```

Если не удалить уже загруженные справочники и ключи, импорт данных будет невозможен.



Примечание. При удалении справочников и ключей удаляется информация об используемом способе аутентификации пользователя ViPNet Coordinator HW. Если до удаления справочников и ключей использовался способ аутентификации по устройству, то после удаления и импорта данных из файла *.vbe будет доступен только способ аутентификации по паролю. Для возврата способа аутентификации по устройству следует вручную изменить способ аутентификации. Подробнее см. раздел [Настройка параметров безопасности](#).

Использование резервного раздела в исполнениях ViPNet Coordinator HW с двумя накопителями



Примечание. Данная функция не предусмотрена в исполнениях ViPNet Coordinator HW с одним дисковым накопителем (ViPNet Coordinator HW10 и 50).

Кроме того, данная функция доступна только на устройствах, на которых первоначально было установлено ПО ViPNet Coordinator HW версии не ниже 4.1.1. При обновлении ПО с более ранней версии данная функция не добавляется.

Чтобы быстро восстанавливать критически важные файлы ViPNet Coordinator HW, целостность которых регулярно проверяется (см. [Контроль целостности файлов ViPNet Coordinator HW](#)):

- Файлы копируются в резервный раздел (`/mnt/reserv`) незагрузочного диска ViPNet Coordinator HW.
- Поврежденные файлы автоматически восстанавливаются из резервных копий.

Копирование файлов в резервный раздел выполняется в следующих случаях:

- Успешная проверка целостности файлов при первоначальной установке и обновлении ПО либо при выполнении команды `machine self-test`.
- Успешная проверка целостности при загрузке ViPNet Coordinator HW.
- Обновление справочников и ключей.
- Изменение пароля пользователя.



Примечание. В первом случае копируются все конфигурационные файлы, а в остальных — только часть файлов.

Если два ViPNet Coordinator HW работают в режиме [кластера горячего резервирования](#), то резервные копии файлов регулярно копируются с активного узла на пассивный. При изменении справочников и ключей, копии справочников и ключей обновляются как на активном, так и на пассивном узле.

Сброс настроек ViPNet Coordinator HW к заводским

Есть ряд ситуаций, при которых вы не сможете аутентифицироваться на ViPNet Coordinator HW, например:

- истечение срока действия ключей или лицензии в тот момент, когда нет связи с защищенной сетью. Без связи с защищенной сетью выслать обновления невозможно. Связь может быть потеряна, например, при несовпадении времени на ViPNet Coordinator HW и на узле центра управления.
- некорректное обновление справочников и ключей или некорректная смена мастер-ключа;
- утрата пароля администратором;

Чтобы продолжить работу, вы можете сбросить настройки ViPNet Coordinator HW к заводским. При этом удаляются все пользовательские настройки:

- файлы, содержащие ключи шифрования для связанных с ViPNet Coordinator HW узлов;
- резервный набор персональных ключей (файл *.pk, при наличии);
- файлы, содержащие информацию о связанных с ViPNet Coordinator HW узлах (справочники);
- настройки защищенной сети (файл iplir.conf);
- настройки сетевых интерфейсов (IP-адрес, маска подсети, файл iplir.conf-`<интерфейс>`);
- настройки маршрутизации и статические маршруты, если такие созданы;
- настройки часового пояса;
- сетевые фильтры и правила трансляции адресов;
- настройки параметров обработки прикладных протоколов;
- настройки системы защиты от сбоев (файл failover.ini);
- настройки транспортного сервера MFTP (файл mftp.conf);
- настройки протоколирования;
- настройки дополнительных сервисов (DHCP-сервера и других);
- настройки туннелирования канального уровня (L2OverIP);
- текущие ключи для соединений по протоколу SSH2.



Внимание! При сбросе настроек к заводским необходимо заново развернуть справочники и ключи. Для восстановления работоспособности ViPNet Coordinator HW после сбоев проводите резервное копирование данных (см. [Резервное копирование и восстановление настроек](#)).

Для сброса настроек ViPNet Coordinator HW к заводским:

1 Включите ViPNet Coordinator HW:

- для виртуальных исполнений используйте кнопки управления виртуальными машинами;
- для аппаратных исполнений используйте кнопку питания.

2 В появившемся окне загрузки выберите **Factory reset** и нажмите **Enter**:

- Для виртуальных исполнений выберите **Factory reset**.



Рисунок 51. Окно загрузки. Виртуальные исполнения

- Для аппаратных исполнений (кроме HW10) выбор зависит от типа подключения монитора:
 - при подключении через видеовыход VGA/HDMI выберите **Factory reset**;
 - при подключении через COM-порт выберите **Factory reset/Serial console**.

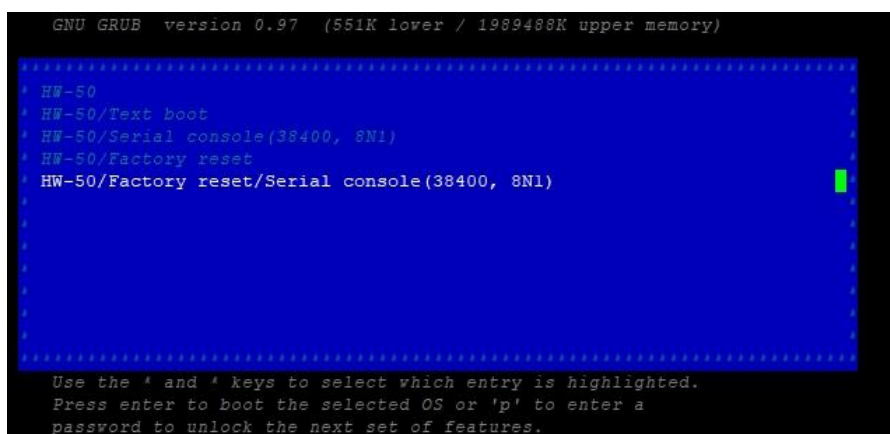


Рисунок 52. Окно загрузки. Аппаратное исполнение

- Для исполнения HW10 нажмите клавишу 2. Все остальные введенные значения рассматриваются как стандартная загрузка.

```
1) HW10/Text boot
2) HW10/Factory reset
```

Рисунок 53. Окно загрузки. Исполнение HW10

- 3 Введите `Delete` для подтверждения удаления справочников и ключей. Начнется процесс удаления справочников и ключей и сброс к заводским настройкам.



Примечание. Отменить сброс к заводским настройкам вы можете в течение 30 секунд после его начала. Для этого нажмите **Ctrl+C**.

- 4 Нажмите **Enter** для подтверждения перезагрузки ViPNet Coordinator HW.
Настройки будут сброшены к заводским.

13

Обеспечение отказоустойчивости

Настройка системы защиты от сбоев	268
Организация электропитания от ИБП	290

Настройка системы защиты от сбоев

Назначение системы защиты от сбоев

Система защиты от сбоев предназначена для контроля работоспособности ПО ViPNet Coordinator HW и создания отказоустойчивого решения на базе двух ViPNet Coordinator HW. Система может работать в одиночном режиме или в режиме кластера.

Параметры работы системы защиты от сбоев задаются в файле `failover.ini`. Подробнее см. «Справочник команд и конфигурационных файлов», раздел «Файл `failover.ini`».

Работа системы защиты от сбоев в одиночном режиме

По умолчанию система защиты от сбоев работает в одиночном режиме и обеспечивает:

- контроль собственной работоспособности;
- контроль работоспособности запущенных служб и драйверов ViPNet Coordinator HW, ведение статистики использования системных ресурсов;
- контроль сбоев при обработке IP-пакетов VPN-драйвером.

В состав системы защиты от сбоев входят watchdog-драйвер `itcswd` и служба `failoverd`. При загрузке системы сначала запускается watchdog-драйвер, а затем служба `failoverd`, которая запускает остальные службы и драйверы. Watchdog-драйвер `itcswd` и служба `failoverd` работают в фоновом режиме.

Watchdog-драйвер контролирует работоспособность службы `failoverd`. Если в настройках включен механизм регистрации служб и драйверов, `failoverd` периодически посылает драйверу сообщения для подтверждения своей работоспособности. Если по истечении определенного времени от службы не будет получено ни одного сообщения, драйвер инициирует перезагрузку системы. Подробнее см. в документе «Справочник команд и конфигурационных файлов», в описании секции `[misc]` файла `failover.ini`.

Служба `failoverd` контролирует работоспособность остальных служб и драйверов ViPNet Coordinator HW. В процессе своей работы каждая из них периодически посылает службе `failoverd` сообщения для подтверждения своей работоспособности. Если по истечении определенного времени от какой-либо службы не будет получено ни одного сообщения или от VPN-драйвера будет получено сообщение с кодом ошибки обработки IP-пакетов, служба `failoverd` считает, что произошел сбой и пытается завершить работу этой службы. Если завершить работу корректно не удастся, служба `failoverd` производит принудительный перезапуск. Если после пяти перезапусков подряд, работу контролируемой службы восстановить не удалось, она считается неработоспособной, и `failoverd` инициирует перезапуск системы.

Работа системы защиты от сбоев в режиме кластера горячего резервирования

Внимание! В режиме кластера горячего резервирования не поддерживаются:



- интерфейс Wi-Fi;
- 3G/4G-модем;
- TCP-туннель.

Перед переключением в режим кластера горячего резервирования отключите указанные интерфейсы и функции.

Исполнение HW10 не поддерживает работу в кластере горячего резервирования.

Система защиты от сбоев позволяет объединить два ViPNet Coordinator HW с одинаковой аппаратной платформой в кластер горячего резервирования. Кластер состоит из двух узлов:

- активный узел — выполняет функции координатора ViPNet;
- пассивный узел — находится в режиме ожидания.

В случае нарушения работоспособности активного узла, пассивный узел переключается в активный режим и выполняет функции сбойного узла, а сбойный узел переходит в пассивный режим.

При описании кластера используются следующие понятия:

- Резервируемый интерфейс — сетевой интерфейс узла кластера, который резервируется на другом узле кластера.
- Интерфейс кластера — резервируемый интерфейс активного узла кластера.

Каждый интерфейс кластера имеет один IP-адрес, по которому кластер доступен для других узлов. Этот IP-адрес всегда принадлежит узлу, работающему в данный момент в активном режиме. Резервируемые интерфейсы узла, работающего в пассивном режиме, не имеют IP-адресов, по которым другие узлы могли бы установить соединение с кластером. При переключении узлов кластера, резервируемые интерфейсы узла, который переходит в активный режим, получают IP-адреса интерфейсов кластера.

- Интерфейс синхронизации — сетевой интерфейс узла кластера, который подключен к интерфейсу другого узла кластера, и используется для передачи данных между узлами кластера.

Наличие канала передачи данных между узлами кластера позволяет пассивному узлу содержать актуальные конфигурационные файлы, журналы, справочники и ключи на момент переключения в активный режим. Для этого активный узел кластера периодически передает указанные данные на пассивный узел. Пропускная способность канала передачи данных должна быть не меньше 100 Мбит/с (рекомендуемое значение 1 Гбит/с).

Через интерфейс синхронизации передаются:

- Пакеты синхронизации, содержащие информацию о текущем режиме работы каждого узла кластера. Обмен пакетами синхронизации позволяет избежать ситуации, когда оба узла кластера работают в активном режиме (например, в случае практически одновременного запуска).

- Копии принятых и готовых к отправке MFTP-конвертов.
- Копии конфигурационных файлов и журналов. Перечень файлов, подлежащих передаче, задается в секции [sendconfig] конфигурационного файла failover.ini (подробнее см. в документе «Справочник команд и конфигурационных файлов»). При низкой пропускной способности канала вы можете выключить необходимость синхронизации журналов (параметр journals) или уменьшить их размер.
- Справочники и ключи.



Примечание. Конфигурационный файл `iplir.conf-<интерфейс>` и журнал регистрации IP-пакетов интерфейса синхронизации с активного узла кластера на пассивный не передаются.

Каждая группа данных передается путем копирования во временный каталог с последующей проверкой, что все файлы группы успешно переданы. Если интерфейс синхронизации подключен не напрямую, настройте защиту на канальном уровне.



Примечание. Для настройки защиты данных на канальном уровне можно установить дополнительное оборудование на границах защищаемых сетей, например ViPNet Coordinator HW или ViPNet L2-10G.

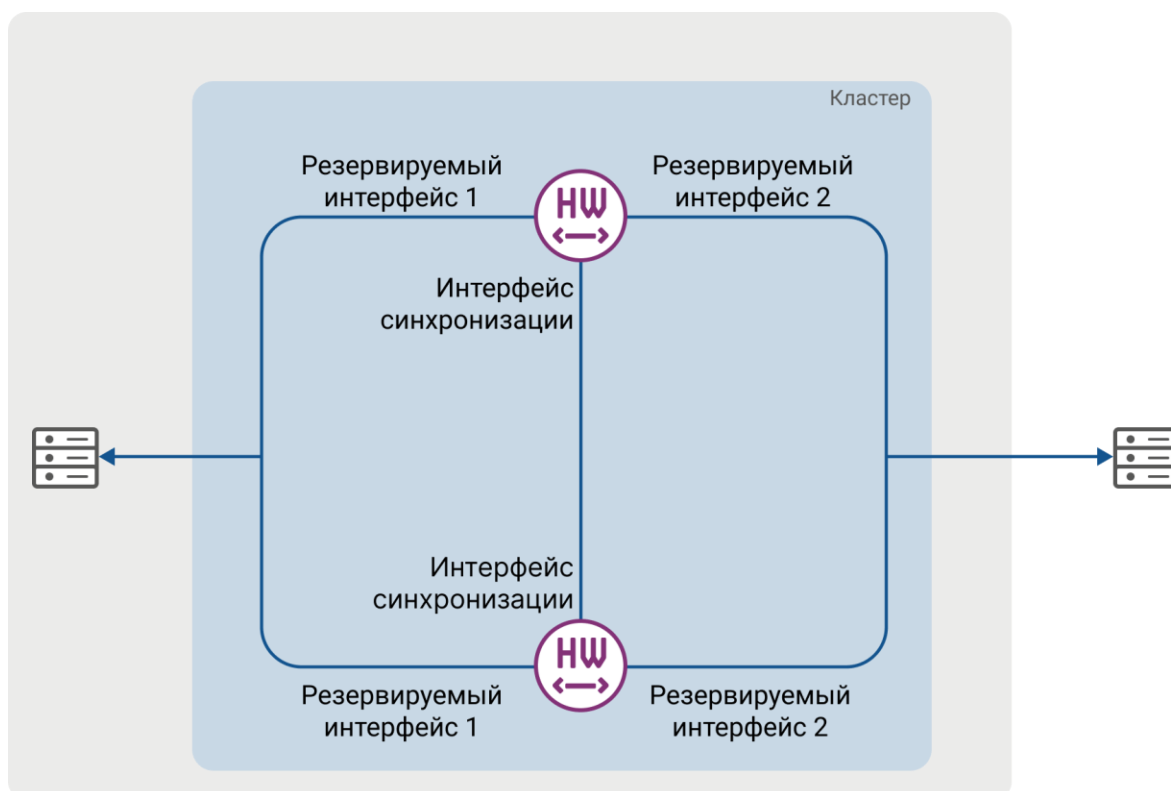


Рисунок 54. Схема кластера

В процессе работы кластера пассивный узел контролирует работоспособность активного узла кластера, а активный узел контролирует работоспособность своих интерфейсов.

Алгоритм проверки работоспособности активного узла кластера:

- 1 Пассивный узел периодически проверяет присутствие в сети интерфейсов с IP-адресами активного узла кластера. Для этого используются ARP-запросы в режиме [Duplicate address detection](#). Если нет возможности обеспечить прохождение запросов за указанное время, увеличьте время ожидания ответа на ARP-запрос (параметр `timeout`).



Внимание! Увеличение времени ожидания ответа на ARP-запрос приведет к увеличению времени перехода пассивного узла в активный режим.

- 2 Если ни один из резервируемых интерфейсов активного узла не отвечает на запросы, счетчик сбоев увеличивается на единицу. Если хотя бы один интерфейс ответил на запрос, счетчик сбоев сбрасывается.
- 3 При достижении счетчиком порогового значения активный узел считается неработоспособным, и пассивный узел переходит в активный режим.

Примечание. Пассивный узел также следит за состоянием интерфейса синхронизации. Это позволяет обнаружить потерю питания на активном узле и быстрее переключить кластер.



- 1 При потере сигнала на интерфейсе синхронизации пассивный узел проверяет присутствие в сети интерфейсов с IP-адресами активного узла кластера.
 - 2 Если ни один из резервируемых интерфейсов активного узла не отвечает, пассивный узел переходит в активный режим сразу, без использования счетчика сбоев.
-

Алгоритм проверки работоспособности интерфейсов активного узла кластера:

- 1 Активный узел периодически анализирует IP-трафик, проходящий через каждый из своих резервируемых интерфейсов.
- 2 Если за определенный период времени не зафиксировано ни одного отправленного и принятого IP-пакета, через каждый интерфейс посылаются эхо-запросы заданным объектам сети (например, маршрутизаторам).

Подробный алгоритм проверки см. в разделе [Типовая схема организации кластера](#).

- 3 При отсутствии ответов на эхо-запросы, счетчик отказов для соответствующего интерфейса увеличивается на единицу. Если получен хотя бы один ответ или зафиксирован трафик на интерфейсе, счетчик сбрасывается.
- 4 При достижении счетчиком порогового значения интерфейс считается неработоспособным, и активный узел переключается в пассивный режим, а пассивный узел становится активным.



Примечание. При потере физического соединения на интерфейсе активного узла кластера, для которого настроен параметр `testdevice` (см. [Развертывание кластера горячего резервирования](#)), переключение узлов происходит сразу, без проверки трафика.

Для опроса IP-адресов активного узла и обмена данными между активным и пассивным узлом на обоих узлах кластера автоматически создаются разрешающие сетевые фильтры.

Развертывание кластера горячего резервирования

Перед развертыванием кластера горячего резервирования:

- 1 Убедитесь, что оба ViPNet Coordinator HW, из которых будет организован кластер, имеют одинаковую аппаратную платформу и версию ПО (версию ПО вы можете узнать с помощью команды: `hostname# version`). Чтобы обновить версию ПО см. [Обновление программного обеспечения](#).

Примечание. Если вы развертываете кластер на базе ViPNet Coordinator VA, оба узла должны иметь одинаковые параметры эмулируемого аппаратного обеспечения: количество процессоров, ОЗУ и сетевых интерфейсов.

Если вы развертываете кластер на базе ViPNet Coordinator VA, развернутого на VMware vSphere ESXi, в окне настроек виртуальной машины на вкладке **Security** в пунктах:



- **MAC address changes**
- **Forged transmits**

выберите пункт **Accept** и нажмите **OK**.

Также исполнение ViPNet Coordinator VA не поддерживает использование виртуальных MAC-адресов для сетевых интерфейсов кластера.

- 2 Отключите от локальной сети оба ViPNet Coordinator HW, которые будут входить в состав кластера. Это позволит избежать проблем в работе локальной сети и рассылке справочно-ключевой информации в сети ViPNet.
- 3 Установите на оба ViPNet Coordinator HW одинаковые справочники и ключи одним из следующих способов:
 - Установите один дистрибутив ключей (файл *.dst) на оба ViPNet Coordinator HW (подробнее см. в документе «Подготовка к работе»). Если на одном из ViPNet Coordinator HW уже установлены справочники и ключи, то сначала удалите их с помощью команды `hostname# admin keys remove`.
 - Перенесите справочники и ключи с одного ViPNet Coordinator HW на другой. Подробнее см. разделы [Экспорт справочников, ключей и настроек](#) и [Импорт справочников, ключей и настроек](#).

- 4 Если вы разворачиваете кластер на исполнениях HW50, HW100 или ViPNet Coordinator VA, то убедитесь, что для них в ViPNet ЦУС или Prime задана роль (функция) в соответствии с таблицей:

```
hostname# version features list
```

Таблица 7. Роли для использования в кластере

Исполнение	Роль для использования в кластере
Coordinator HW50	Failover100
Coordinator HW100	
Coordinator VA100	Failover100
Coordinator VA500	Failover500
Coordinator VA1000	Failover1000
Coordinator VA2000	Failover2000

В списке функциональных модулей, который выведет команда, должен присутствовать модуль `failover-cluster`. Если же этого модуля нет, обратитесь к администратору сети ViPNet для назначения роли (включения функции) сетевому узлу ViPNet Coordinator HW и последующего обновления справочников ключей на обоих узлах кластера ViPNet Coordinator HW.

- 5 Выберите сетевые интерфейсы, которые будут использоваться в качестве резервируемых интерфейсов. Вы также можете использовать виртуальные или VLAN интерфейсы.
- 6 Убедитесь, что в настройках резервируемых интерфейсов заданы статические IP-адреса, установлен класс `access` (см. [Настройка сетевых интерфейсов Ethernet](#)), и данные интерфейсы включены. Назначение IP-адресов по протоколу DHCP не допускается.
- 7 На каждом ViPNet Coordinator HW выберите сетевой интерфейс, который будет использоваться в качестве интерфейса синхронизации. Соедините интерфейсы синхронизации обоих ViPNet Coordinator HW и настройте их. Для этого задайте статические IP-адреса, установите класс `access` и включите данные интерфейсы. IP-адреса интерфейсов синхронизации должны принадлежать одной сети.



Внимание! IP-адрес интерфейса синхронизации на каждом из узлов кластера не должен совпадать с IP-адресами защищенных узлов, иначе трафик, идущий на эти адреса, будет блокироваться. Если необходимо использовать одинаковые адреса, для разрешения конфликта рекомендуется настроить видимость соответствующих защищенных узлов по виртуальным адресам (см. [Настройка виртуальных IP-адресов](#)).

- 8 Выберите один или несколько стабильных объектов сети (например, маршрутизатор), которые будут использоваться для проверки работоспособности интерфейсов активного узла кластера.



Внимание! Настройку сетевых служб необходимо выполнять после развертывания кластера и запуска системы защиты от сбоев. Настройка выполняется на активном узле кластера.

Для развертывания кластера выполните следующие действия на каждом из ViPNet Coordinator HW:

1 Завершите работу службы системы защиты от сбоев:

```
hostname# failover stop
```

2 Откройте файл `failover.ini` для редактирования:

```
hostname# failover config edit
```

3 Для каждого резервируемого интерфейса задайте следующие параметры в секции `[channel]` (если для нужного интерфейса секция отсутствует, создайте ее):



Внимание! Для каждой пары резервируемых интерфейсов, подключенных к одной сети, значения параметров секции `[channel]` должны совпадать. То есть секции `[channel]` должны заполняться одинаково на обоих узлах кластера.

- o `device` — имя сетевого интерфейса.

На обоих узлах кластера для доступа к одной и той же сети должны использоваться сетевые интерфейсы с одинаковыми именами.

- o `activeip` — IP-адрес с префиксом маски, который присваивается сетевому интерфейсу, когда узел работает в активном режиме.
- o `testip` — IP-адрес маршрутизатора или другого стабильного объекта сети, находящегося в одной подсети с IP-адресом из параметра `activeip`, к которому будут посылаться эхо-запросы для проверки работоспособности интерфейса.

При необходимости для каждого резервируемого интерфейса вы можете указать несколько параметров `testip`. В этом случае сбоем интерфейса будет считаться ситуация, когда ни от одного из заданных IP-адресов не будет получено ответа.

Примечание. В качестве параметра `testip` не рекомендуется задавать адрес интерфейса «внутренней петли» (loopback), например, `127.0.0.1`, так как в этом случае реальной проверки работоспособности сетевого интерфейса не производится.



Если планируется использование кластера горячего резервирования совместно с DGD, то в каждой секции `[channel]` используйте IP-адреса всех проверяемых шлюзов. В этом случае система защиты от сбоев будет контролировать активность на интерфейсе, а DGD обеспечит работу только через доступный шлюз.

Например, чтобы эхо-запросы отправлялись на IP-адреса `192.168.100.34` и `192.168.100.25`, добавьте строки:

```
testip = 192.168.100.34
```

```
testip = 192.168.100.25
```



Внимание! Если в кластере используется функция L2OverIP (см. [Настройка туннелирования на канальном уровне по технологии L2OverIP](#)) и требуется проверять состояние его интерфейса, то в качестве тестового IP-адреса для этого интерфейса необходимо задать адрес из локального сегмента, а также выключить блокирование одноадресных Ethernet-кадров с неизвестным MAC-адресом получателе командой `iplir set l2overip unsolicited-frames broadcast`. Настоятельно не рекомендуется указывать для рабочего интерфейса L2OverIP тестовый IP-адрес в удаленном сегменте.

- o `testdevice` — имя сетевого интерфейса, для которого необходимо контролировать состояние соединения. При потере физического соединения на указанном интерфейсе будет выполнено переключение узлов кластера.

В качестве сетевого интерфейса в параметре `testdevice` вы можете указать только физический интерфейс. Не допускается задавать виртуальные, агрегированные или VLAN интерфейсы, а также `localhost`. При этом вы можете указать физический интерфейс, который задан как `trunk` интерфейс для VLAN.

Параметр `testdevice` необязательный. В каждой секции `[channel]` вы можете задать только один параметр `testdevice`.

Внимание!



- Не изменяйте значение параметра `testdevice`, когда служба `failoverd` запущена в режиме кластера горячего резервирования. Иначе это будет воспринято системой как потеря физического соединения и приведет к переключению узлов кластера.
- В параметре `testdevice` указывайте только те сетевые интерфейсы, состояния которых действительно должны контролироваться для работоспособности кластера.

-
- o `ident` — текстовая строка, идентифицирующая сетевой интерфейс. Для интерфейсов, подключенных к одинаковым сетям, параметры `ident` должны совпадать. По этому параметру система защиты от сбоев определяет интерфейсы, которые выполняют одинаковые функции на узлах кластера.
 - o `checkonlyidle` — способ проверки доступности сетевых узлов через интерфейс.

Особенности настройки см. в разделе [Настройка параметра checkonlyidle](#).

Возможные значения:

- `yes` (по умолчанию) — проверять доступность сетевых узлов, только если на интерфейсе нет сетевой активности.

Если за период опроса через интерфейс не проходят входящие или исходящие пакеты, на IP-адреса из `testip` отправляются эхо-запросы.

- `no` — всегда проверять доступность сетевых узлов.

Эхо-запросы на IP-адреса из `testip` отправляются раз в период опроса.

Особенности использования:

- Узлы кластера переключаются, если нет ответа на эхо-запросы. Время ожидания ответа определяется параметром `timeout` секции `[network]`.
 - Период опроса определяется параметром `checktime` секции `[network]`.
- `usevirtualmac` — использование виртуальных MAC-адресов для сетевых интерфейсов кластера, заданных в данной секции `[channel]`. Возможные значения:
- `yes` — использовать виртуальные MAC-адреса для сетевых интерфейсов;
 - `no` (по умолчанию) — не использовать виртуальные MAC-адреса для сетевых интерфейсов.

Включение данной функции позволит ускорить восстановление связи с координатором после переключения узлов кластера. Виртуальные MAC-адреса для сетевых интерфейсов кластера генерируются на основе значения параметра `virtualmacprefix` в секции `[network]` (см. описание параметра в документе «Справочник команд и конфигурационных файлов»).

Примечание.



- Для работы этой функции необходимо настроить коммутационное оборудование таким образом, чтобы разрешить использование двух MAC-адресов для одного Ethernet-порта. Убедитесь, что настройки `port-security` коммутационного оборудования не блокируют смену MAC-адресов сетевых интерфейсов кластера.
 - Виртуальные MAC-адреса не могут назначаться для агрегированных (например, `bond1`) и виртуальных (например, `eth0:1`) сетевых интерфейсов.
-

Перед выполнением следующего шага, вернитесь к началу текущего шага и настройте следующий резервируемый интерфейс.

4 В секции `[misc]` задайте значение параметра `syncconnections`:

- `yes` — синхронизация соединений при работе в кластере включена. В этом случае при переключении узлов кластера соединения, открытые на активном узле до момента переключения, продолжают работу на пассивном узле после его переключения в активный режим. При этом не будут синхронизированы следующие типы соединений:
- открытые сессии разрешения доменных имен;
 - защищенные соединения по TCP-туннелям;
 - локальные соединения (открытые и защищенные соединения, в которых ViPNet Coordinator HW является источником или назначением);
 - соединения, использующие прикладные протоколы (см. [Настройка обработки прикладных протоколов](#)).

После переключения узлов кластера связь по указанным типам соединений может быть прервана.

- `no` (по умолчанию) — синхронизация соединений при работе в кластере выключена.

Для работы функции синхронизации соединений данный параметр должен иметь значение `yes` на обоих узлах кластера.

- 5 В секции `[sendconfig]` задайте следующие параметры для синхронизации настроек между узлами кластера:

- o `device` — имя интерфейса синхронизации.
- o `activeip` — IP-адрес интерфейса синхронизации другого узла кластера.

- 6 При необходимости измените другие параметры секции `[sendconfig]` и `[network]` файла `failover.ini`. Описание параметров секций см. в документе «Справочник команд и конфигурационных файлов».

- 7 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).

- 8 Переведите систему защиты от сбоев в режим кластера горячего резервирования:

```
hostname# failover config mode cluster
```

- 9 Подключите оба ViPNet Coordinator HW к локальной сети.

- 10 Убедитесь, что настройки коммутационного оборудования соответствуют требованиям раздела [Работа кластера горячего резервирования совместно с коммутационным оборудованием](#).

- 11 Проверьте связь с каждым IP-адресом, заданным в параметре `testip` секции `[channel]` файла `failover.ini`:

```
hostname# inet ping <IP-адрес>
```



Внимание! Отсутствие связи со всеми IP-адресами, указанными в параметрах `testip` хотя бы одной секции `[channel]`, может привести к циклическому переключению узлов кластера после запуска системы защиты от сбоев.

- 12 Запустите систему защиты от сбоев:

```
hostname# failover start
```

ViPNet Coordinator HW, на котором служба системы защиты от сбоев будет запущена первым, начнет работать как активный узел кластера. Вы можете принудительно задать режим работы каждого узла кластера при запуске службы:

```
hostname# failover start {active | passive}
```

Если активный узел кластера уже запущен, перед запуском службы на втором узле кластера, сначала проверьте доступность активного узла:

```
hostname# failover show active-mac-address
```

В случае успешной проверки будет выведен список активных сетевых интерфейсов кластера, описанных параметрами `device` и `activeip` секции `[channel]` конфигурационного файла `failover.ini`, и соответствующих им MAC-адресов, например:

```
Interface IP-address MAC-address
-----
eth0 10.0.2.10 38:D5:47:C9:DA:4C
```

eth1 10.0.3.10 C9:DA:4C:C9:DA:4C



Внимание! Не изменяйте параметры сетевых интерфейсов, задействованных в работе кластера, если служба `failoverd` запущена в режиме кластера горячего резервирования. Иначе это может вызвать сбой в работе кластера.

После запуска системы защиты от сбоев на обоих узлах кластера, кластер начнет свою работу. При этом на каждом узле кластера будут автоматически созданы нередатируемые сетевые фильтры, необходимые для корректной работы системы защиты от сбоев с выбранными параметрами.

Примечание. Система защиты от сбоев не контролирует работоспособность интерфейса синхронизации. Чтобы проверить его состояние, выполните команду:



```
hostname# failover show info
```

Если на экран выводится информация о состоянии обоих узлов кластера, значит интерфейсы работают исправно. Иначе, проверьте физическое соединение кабеля с интерфейсом.

Настройка параметра `checkonlyidle`

Параметр `checkonlyidle` определяет способ проверки доступности сетевых узлов через интерфейс на активном узле кластера. Подробнее см. в документе «Справочник команд и конфигурационных файлов», раздел «Файл `failover.ini`» > «Секция `[channel]`».

Особенности использования:

- При `checkonlyidle=yes`:
 - снижается нагрузка на кластер и сетевые узлы, с которыми проверяется доступность, уменьшается количество служебного трафика;
 - проверяется только сетевая активность на интерфейсе, поэтому устройства с IP-адресами из `testip` должны быть напрямую подключены к резервируемым интерфейсам кластера.
- При `checkonlyidle=no`:
 - повышается нагрузка на кластер и увеличивается количество служебного трафика;
 - контролируется доступность узлов в сети, а не состояние резервируемого интерфейса.

Многоуровневая сеть

Рассмотрим схему организации сети, в которой интерфейс активного узла кластера подключен не напрямую к устройству с IP-адресом из `testip`:

- Если `checkonlyidle=yes`, проверяется сетевая активность интерфейса (учитываются и входящие, и исходящие пакеты).

При потере соединения между SW1 и SW3 узлы кластера не переключатся, так как между активным узлом и SW1 есть трафик.

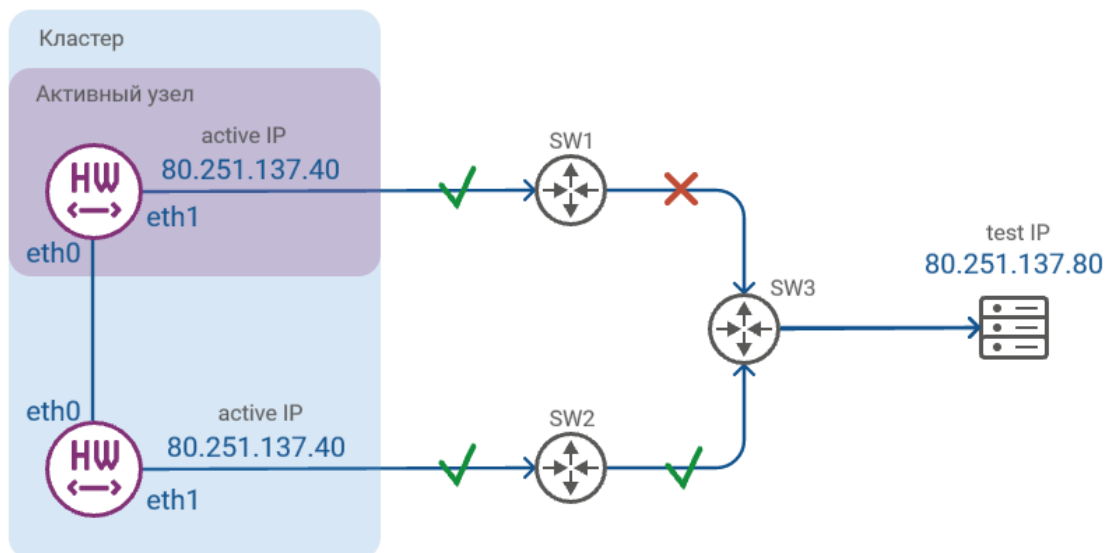


Рисунок 55. Узлы кластера не переключаются из-за сетевой активности на интерфейсе eth1

Доступность сетевых узлов из testip будет проверяться только при потере соединения между активным узлом и SW1.

- Если checkonlyidle=no, проверяется доступность сетевых узлов из testip.

Узлы кластера переключаются при потере соединения с устройствами с IP-адресами из testip.

Доступность узлов проверяется раз в период опроса.

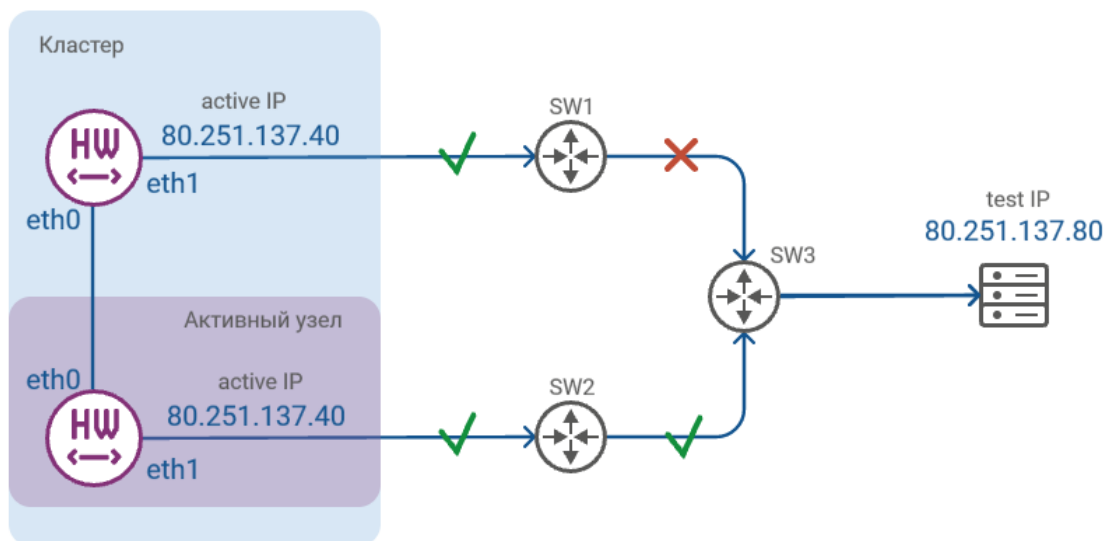


Рисунок 56. Узлы кластера переключаются при недоступности устройства из testip

Агрегированное соединение

При резервировании агрегированных интерфейсов используйте checkonlyidle=no.

Для агрегированных интерфейсов не отслеживается состояние соединений на подчинённых интерфейсах. Статус up подчинённых интерфейсов указывает только на их административное состояние.

Даже при потере соединения на eth1 и eth2 агрегированный интерфейс считается активным. Узлы кластера не переключаются даже при прямом подключении агрегированного интерфейса к сетевому устройству с IP-адресом из testip.

При назначении агрегированному интерфейсу класса trunk (для обработки трафика из нескольких VLAN) VLAN-интерфейсы наследуют состояние родительского агрегированного интерфейса.

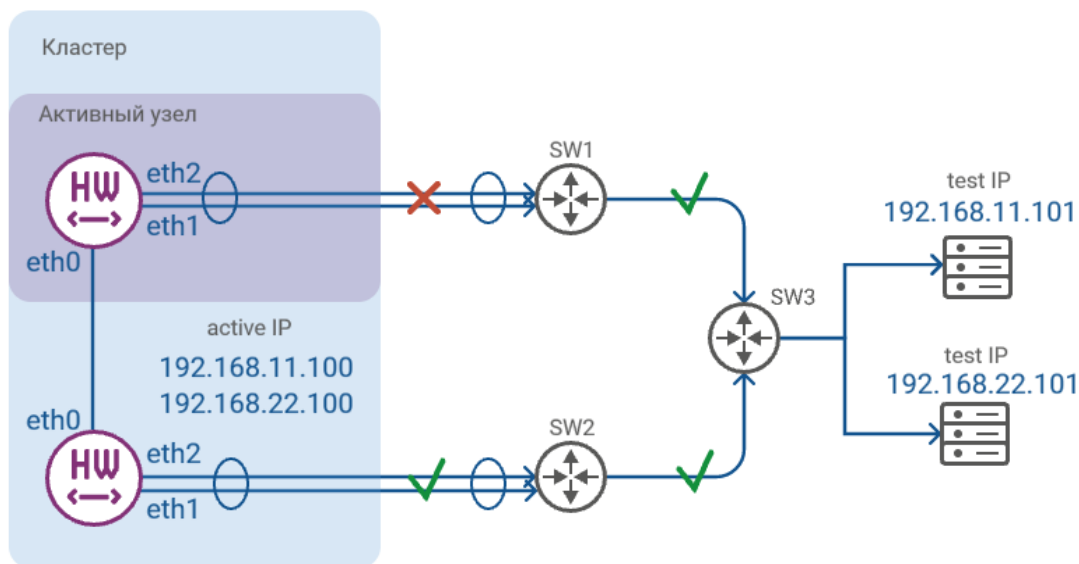


Рисунок 57. Узлы кластера не переключаются при потере соединения на агрегированном интерфейсе

Типовая схема кластера

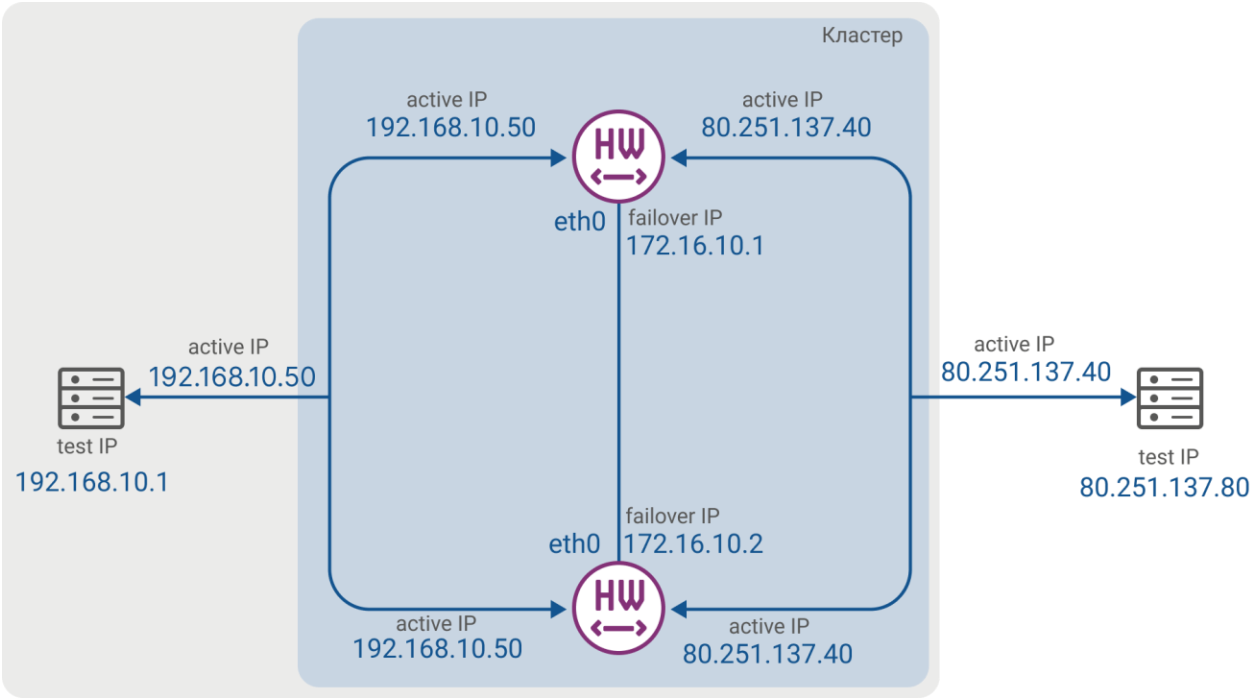


Рисунок 58. Типовая схема кластера

Таблица 8. Настройки параметров кластера

Настройки HW-1	Настройки HW-2
[channel]	[channel]
device = eth1	device = eth1
activeip = 192.168.10.50/24	activeip = 192.168.10.50/24
testip = 192.168.10.1	testip = 192.168.10.1
testdevice = eth1	testdevice = eth1
ident = internal	ident = internal
checkonlyidle = yes	checkonlyidle = yes
[channel]	[channel]
device = eth2	device = eth2
activeip = 80.251.137.40/24	activeip = 80.251.137.40/24
testip = 80.251.137.80	testip = 80.251.137.80
testdevice = eth2	testdevice = eth2
ident = external	ident = external
checkonlyidle = yes	checkonlyidle = yes
usevirtualmac = yes	usevirtualmac = yes
[misc]	[misc]
syncconnections = yes	syncconnections = yes
[network]	[network]

Настройки HW-1	Настройки HW-2
<code>checktime = 10</code>	<code>checktime = 10</code>
<code>timeout = 2</code>	<code>timeout = 2</code>
<code>activeretries = 3</code>	<code>activeretries = 3</code>
<code>channelretries = 3</code>	<code>channelretries = 3</code>
<code>synctime = 5</code>	<code>synctime = 5</code>
<code>fastdown = yes</code>	<code>fastdown = yes</code>
<code>virtualmacprefix = 39</code>	<code>virtualmacprefix = 39</code>
[sendconfig]	[sendconfig]
<code>device = eth0</code>	<code>device = eth0</code>
<code>activeip = 172.16.10.2</code> (соответствует failover IP HW-2)	<code>activeip = 172.16.10.1</code> (соответствует failover IP HW-1)

Активный узел кластера в реальном времени контролирует наличие физического соединения на сетевом интерфейсе, указанном в параметре `testdevice`. В случае потери соединения на этом интерфейсе переключение узлов кластера происходит в течение трех секунд.

Параллельно применяется следующий алгоритм работы активного узла:

- 1 Каждые `checktime` секунд проверяется работоспособность каждого из указанных в конфигурационном файле интерфейсов.
- 2 Если параметр `checkonlyidle` установлен в значение `yes`, то проверяется сетевая активность на интерфейсе.
- 3 Если в течение `checktime` через интерфейс не прошло ни одного входящего или исходящего пакета, через этот интерфейс посылаются эхо-запросы на узел с адресом `testip`, и в течение `timeout` ожидаются ответы на эти запросы.
- 4 Если ни одного ответа не приходит, то счетчик сбоев этого интерфейса увеличивается на единицу, и посылаются новые эхо-запросы. Если на эхо-запрос приходит хотя бы один ответ, то счетчик сбоев интерфейса обнуляется. В случае обнуления счетчиков сбоев всех интерфейсов происходит возврат к началу цикла проверки (ожидание в течение `checktime` и так далее). При достижении счетчиком сбоев какого-либо интерфейса значения `channelretries` интерфейс считается неработоспособным, и происходит переключение узлов кластера.



Примечание. Чтобы эхо-запросы на узел с адресом `testip` посылались каждые `checktime` секунд, установите параметр `checkonlyidle` в значение `no`.

Максимальное время, необходимое системе защиты от сбоев для определения неработоспособности интерфейса, вычисляется по формуле:

$$\text{checktime} + (\text{timeout} * \text{channelretries})$$

Алгоритм работы пассивного узла:

- 1 Каждые `checktime` секунд посылаются ARP-запросы в режиме DAD на узел с адресом `activeip`.
- 2 По истечении `timeout` проверяется наличие положительного ответа для каждого `activeip` в системной ARP-таблице и делается вывод о работоспособности соответствующего интерфейса активного узла.
- 3 Если ни от одного интерфейса не был получен ответ, счетчик сбоев (один для всех интерфейсов) увеличивается. Если хотя бы от одного интерфейса был получен ответ, счетчик сбоев обнуляется. В случае достижения счетчиком сбоев значения `activeretries`, производится переход узла в активный режим.

Максимальное время, проходящее с момента неактивности (сбоя, перезагрузки) активного узла до обнаружения пассивным узлом этого факта, вычисляется по формуле:

`checktime + (timeout * activeretries)`

Развертывание кластера совместно с DHCP-сервером и DHCP-relay

- 1 Разверните кластер (см. [Развертывание кластера горячего резервирования](#)) с учетом следующего требования: IP-адрес каждого резервируемого интерфейса, на котором должен быть настроен DHCP-сервер или DHCP-relay, должен совпадать с IP-адресом, установленным в параметре `activeip` секции `[channel]` конфигурационного файла `failover.ini` для соответствующего интерфейса кластера.
- 2 На активном узле настройте параметры работы DHCP-сервера (см. [Настройка DHCP-сервера](#)) и DHCP-relay (см. [Настройка DHCP-relay](#)).
- 3 Убедитесь, что параметры DHCP-сервера и DHCP-relay успешно переданы на пассивный узел. Для этого на пассивном узле выполните:

```
hostname> inet show dhcp server
hostname> inet show dhcp relay [<номер копии>]
```

- 4 Запустите службы DHCP-сервера и DHCP-relay на активном узле:

```
hostname# inet dhcp server start
hostname# inet dhcp relay [<номер копии>] start
```

- 5 Включите автоматический запуск DHCP-сервера и DHCP-relay:

```
hostname# inet dhcp server mode on
hostname# inet dhcp relay mode on
```

- 6 Переключите режим работы пассивного узла кластера в активный. Для этого перезагрузите активный узел:

```
hostname> machine reboot
```

- 7 На узле кластера, ставшем активным, проверьте параметры DHCP-сервера и DHCP-relay:

```
hostname> inet show dhcp server
```

```
hostname> inet show dhcp relay [<номер копии>]
```

Запуск и завершение работы системы защиты от сбоев

Чтобы запустить систему защиты от сбоев на одиночном узле, выполните:

```
hostname# failover start
```

Чтобы запустить систему защиты от сбоев на узле кластера, выполните:

- в режиме активного узла:

```
hostname# failover start active
```

- в режиме пассивного узла:

```
hostname# failover start passive
```



Внимание! Если при запуске системы защиты от сбоев не был указан параметр `active` или `passive`, то она будет запущена в том режиме, в котором находилась до завершения работы.

Перед запуском системы защиты от сбоев в активном или пассивном режиме убедитесь в том, что другой узел кластера работает в противоположном режиме. Запуск обоих узлов кластера в активном режиме вызовет конфликт IP-адресов.

Чтобы завершить работу системы защиты от сбоев, выполните:

```
hostname# failover stop
```

Переключение кластера горячего резервирования в одиночный режим работы

Чтобы переключить кластер горячего резервирования в одиночный режим:

- 1 На каждом из узлов кластера выполните команду:

```
hostname# failover config mode single
```

и подтвердите действие (введите `y` и нажмите **Enter**).

- 2 Отключите один из узлов кластера от локальной сети, чтобы избежать конфликтов при получении справочно-ключевой информации.
- 3 Убедитесь, что IP-адреса сетевых интерфейсов второго узла кластера заданы верно, так как при переводе кластера в одиночный режим параметры сетевых интерфейсов, заданные в файле `failover.ini`, игнорируются.
- 4 В зависимости от дальнейших планов по использованию кластера выполните:

- Если вы больше не планируете использовать кластер, удалите ключи с узла, который был отключен от локальной сети:

```
hostname# admin keys remove
```

Если вы хотите в дальнейшем использовать этот ViPNet Coordinator HW как одиночный узел, установите на нем новый дистрибутив ключей (см. документ «Подготовка к работе»).



Внимание! Для узлов кластера, которые в дальнейшем будут использоваться как одиночные узлы, необходимо использовать разные дистрибутивы ключей.

- Если вы планируете возобновить работу кластера:
 - Выполните необходимые настройки для развертывания кластера (см. [Развертывание кластера горячего резервирования](#)). В частности, убедитесь, что в конфигурационном файле `failover.ini` заданы IP-адреса всех сетевых интерфейсов кластера.
 - Подключите к локальной сети отключенный узел кластера.

Замена сбойного узла кластера на новый без прерывания трафика

В случае сбоя активного узла кластера он переключится в пассивный режим. После этого замените сбойный узел кластера на новый. Для замены узла кластера без прерывания трафика, проходящего через кластер:

- 1 Убедитесь, что новый ViPNet Coordinator HW, который будет включен в состав кластера, отключен от локальной сети.
- 2 Выполните [экспорт справочников, ключей и настроек активного узла кластера](#).

Примечание. Для настройки нового узла кластера также потребуется:

- IP-адрес интерфейса синхронизации, который был установлен на сбойном узле;
- IP-адрес интерфейса синхронизации активного узла кластера.



Эту информацию вы можете получить на активном узле кластера.

Чтобы узнать IP-адрес интерфейса синхронизации сбойного узла:

```
hostname# failover show config
```

IP-адрес указан в секции `[sendconfig]` в параметре `activeip`.

Чтобы узнать IP-адрес интерфейса синхронизации активного узла:

```
hostname# inet show interface <имя интерфейса>
```

- 3 Импортируйте полученный файл экспорта (*.vbe) на новый узел кластера (см. [Импорт справочников, ключей и настроек](#)).
- 4 После перезагрузки нового узла кластера завершите работу системы защиты от сбоев:

```
hostname# failover stop
```

- 5 Задайте IP-адрес интерфейса синхронизации, который был установлен на сбойном узле кластера:

```
hostname# inet ifconfig <интерфейс> address <IP-адрес> netmask <маска>
```

где:

- o <интерфейс> — имя сетевого интерфейса;
- o <IP-адрес> — IP-адрес;
- o <маска> — маска подсети.

- 6 Откройте конфигурационный файл `failover.ini` для редактирования:

```
hostname# failover config edit
```

- 7 В конфигурационном файле `failover.ini` замените значение параметра `activeip` секции `[sendconfig]` на IP-адрес интерфейса синхронизации активного узла кластера.

- 8 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).

- 9 Если на новом узле кластера включена функция плановой перезагрузки, измените время перезагрузки:

```
hostname# machine reboot-schedule time <время> [day <день>]
```

Время перезагрузки пассивного и активного узлов кластера должно отличаться (не менее чем на 30 минут).

- 10 Выключите новый узел кластера:

```
hostname# machine halt
```

- 11 Подключите новый узел кластера к локальной сети, а интерфейсы синхронизации обоих узлов кластера соедините между собой.

- 12 Включите новый узел кластера.

- 13 Убедитесь, что пассивный узел кластера стал доступен. Для этого на активном узле кластера выполните команду:

```
hostname# failover show info
```

Просмотр информации о работе системы защиты от сбоев

Текущее состояние системы защиты от сбоев

Просмотр информации о текущем состоянии системы защиты от сбоев узла :

```
hostname> failover show info
```

В результате выводится следующая информация об узле:

- версии ПО ViPNet Coordinator HW и системы защиты от сбоев, установленных на узле (versions);
- идентификатор и имя узла (ID);
- локальное время на узле (workstation time);
- режим работы системы защиты от сбоев узла (failover mode):
 - single — одиночный режим работы;
 - active — активный режим работы в составе кластера горячего резервирования;
 - passive — пассивный режим работы в составе кластера горячего резервирования;
- время непрерывной работы системы защиты от сбоев (failover uptime);
- общая загруженность процессора в процентах (total cpu);
- общий объем памяти в килобайтах (total memory);
- общий объем свободной памяти в килобайтах (available memory);
- сведения о текущем состоянии служб ViPNet Coordinator HW:
 - initializing — загружается;
 - works — работает;
 - stopped — работа завершена;
 - unknown — неизвестно (например, если произошел сбой в работе службы, её перезапустили и пока нет данных о ее состоянии);
- сведения о ресурсах процессора, используемых каждой из служб, в процентах (failover cpu, iplir cpu и так далее).



Примечание. В случае просмотра информации о текущем состоянии системы защиты от сбоев на узле, работающем в составе кластера горячего резервирования, статистика использования системных ресурсов и текущее состояние служб ViPNet Coordinator HW отображаются для обоих узлов кластера (в столбце `local` — для узла, на котором выполнена команда, в столбце `remote` — для второго узла кластера, см. пример ниже)

Например, если служба `failoverd` работает в одиночном режиме, выводится следующая информация:

```

Versions: ViPNet 4.5.8(150), daemon 1.5 (1)
Workstation configured for ID 15EA000B (Coordinator_2)
The workstation works in a single mode of protection against failures
Workstation time (utc: 1406806848) Wed Jul 31 15:40:48 2024
failover mode    * single
failover uptime  * 0d 0:14
total cpu        * 3%
total memory     * 2055488 kB
free memory      * 1915324 kB
failover state   * works
failover cpu     * 1%
```

```

iplir state      * works
iplir cpu        * 4%
mftp state       * works
mftp cpu         * 1%
webgui state     * works
webgui cpu       * 0%

```

Если служба failoverd работает в режиме кластера горячего резервирования, выводится следующая информация:

```

Versions: ViPNet 4.5.8 (30), daemon 1.5 (1)
Workstation configured for ID 29A0022 (Coordinator_1)
Workstation works in a cluster mode of protection against failures
Workstation time (utc: 1204638024) Thu Jul 27 17:35:30 2024

                                * local                                * remote
failover mode                  * active                              * passive
failover uptime                 * 3d 5:26                      * 0d 0:00
total cpu                       * 80%                          * 0%
total memory                    * 2044104 kB                   * 2044104 kB
free memory                     * 1672360 kB                   * 1672360 kB
failover state                  * works                        * works
failover cpu                    * 7%                          * 0%
iplir state                     * works                        * works
iplir cpu                      * 0%                          * 0%
mftp state                      * works                        * works
mftp cpu                       * 66%                         * 0%
webgui state                    * works                        * stopped
webgui cpu                     * 0%                          * 0%

```

Если служба failoverd не работает, выводится только информация о его режиме. Например:

```
The failover system is in the single mode
```

Журнал переключений

События о работе системы защиты от сбоев в режиме кластера регистрируются службой failoverd в журнале переключений. Среди таких событий:

- <BOOT> — запуск службы failoverd при загрузке операционной системы;
- <P_START> — ручной запуск службы failoverd в режиме пассивного узла;
- <A_START> — ручной запуск службы failoverd в режиме активного узла;
- <SWITCH> — переход узла из пассивного в активный (или наоборот).

Чтобы просмотреть записи в журнале переключений узла, выполните:

```
hostname> failover view <дд.мм.гггг[.чч.мм.сс]> <дд.мм.гггг[.чч.мм.сс]>,
```

указав начало и конец интересующего интервала времени. Например:

```
failover view 08.03.2024.00.00.00 23.03.2024.16.37.00
```

Выводится следующая информация:

- версии ПО ViPNet Coordinator HW и службы failoverd;
- идентификатор и имя узла;
- указание на работу системы защиты от сбоев в режиме кластера;
- локальные дата и время на узле;
- список записей за выбранный интервал времени (дата, время, идентификатор и описание события).

Данная информация выводится в следующем формате:

```
View journal of failover switching
Versions: ViPNet 4.5.8 (30), daemon 1.5 (1)
Workstation configured for ID 29A0022 (Coordinator_HW)
The workstation works in a cluster mode of protection against failures
Workstation time (utc: 1174916969) Thu Mar 27 17:35:30 2024

09 Mar 2024 12:51:42      <P_START> Start failover daemon in passive mode
22 Mar 2024 12:27:27      <A_START> Start failover daemon in active mode
22 Mar 2024 14:10:35      <A_START> Start failover daemon in active mode
22 Mar 2024 15:30:46      <BOOT> Boot the system
23 Mar 2024 11:09:07      <SWITCH> Switch server from passive mode to active mode
```

Если за указанный интервал времени не произошло ни одного события, выводится сообщение:

```
There are no records in journal of switchings
```

Работа кластера с коммутационным оборудованием

Настройки коммутационного оборудования (коммутаторы и маршрутизаторы), к которому подключается кластер, могут влиять на его работу. Поэтому при их совместном использовании убедитесь, что на коммутационном оборудовании:

- пропускаются эхо-запросы ICMP с IP-адресов активного узла до всех узлов с IP-адресами, заданными в параметрах `testip` секции `[channel]` конфигурационного файла `failover.ini`, и ответы на них;
- отключена функция Proxy ARP и пропускаются ARP-запросы от пассивного узла до активного узла и ответы на них.

Эти рекомендации касаются всех сетевых интерфейсов ViPNet Coordinator HW, для которых в файле `failover.ini` существует секция `[channel]`.

Организация электропитания от ИБП

Примечание.



- Для исполнений ViPNet Coordinator VA и HW10 данная функция не поддерживается.
 - В настоящее время поддерживаются только ИБП фирмы APC, подключаемые через USB-порт (например, APC Smart-UPS).
-

Чтобы обеспечить электропитание ViPNet Coordinator HW в случае отключения электроэнергии или перебоев в электросети, подключите его к [источнику бесперебойного питания](#). Тогда при отключении электроэнергии:

- 1 ViPNet Coordinator HW начинает питаться от аккумуляторной батареи ИБП.
- 2 Если заряд аккумуляторной батареи достигает критически низкого уровня (этот уровень зависит от следующих параметров, заданных на ИБП: `battery.charge`, `battery.charge.low`, `battery.runtime` и `battery.runtime.low`):
 - 2.1 ИБП взаимодействует с ViPNet Coordinator HW по интерфейсному кабелю и посылает сигнал об истощении батареи.
 - 2.2 ViPNet Coordinator HW получает сигнал и начинает выключаться. При этом на ИБП передается информация о том, что для выключения ViPNet Coordinator HW потребуется не менее 180 секунд. Получив эти данные, ИБП корректирует расход оставшегося заряда батареи таким образом, чтобы ViPNet Coordinator HW успел штатно завершить работу.

В результате ViPNet Coordinator HW успевает выключиться до полного разряда батареи и потери данных не происходит.
- 3 Если подача электроэнергии восстанавливается до достижения батареей критически низкого уровня заряда, ViPNet Coordinator HW снова начинает питаться от электросети.

В общем случае от одного ИБП могут питаться несколько устройств. Устройство, к которому подключен интерфейсный кабель ИБП, является главным (master), остальные — подчиненными (slave). Главное устройство взаимодействует непосредственно с ИБП и отвечает за своевременное оповещение по сети подчиненных устройств. Оно должно быть доступно подчиненным по одному из своих IP-адресов.

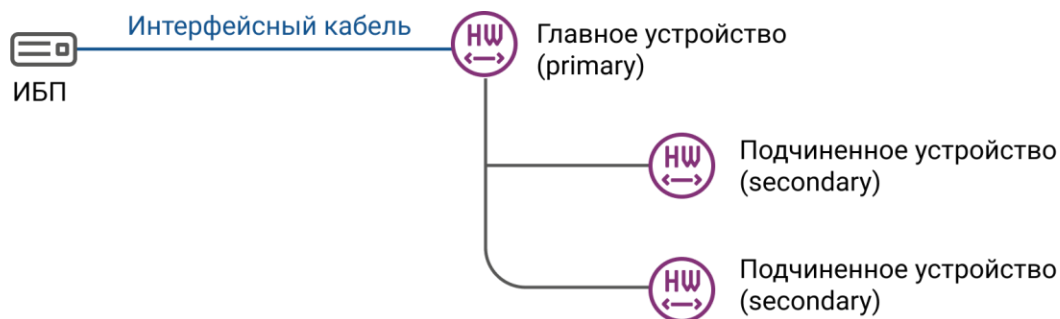


Рисунок 59. Схема подключения ViPNet Coordinator HW к ИБП

Чтобы настроить взаимодействие нескольких ViPNet Coordinator HW с ИБП, необходимо на каждом ViPNet Coordinator HW выполнить ряд команд, соответствующих его роли (master или slave).

На ViPNet Coordinator HW, который будет выступать в роли master:

- 1 Подключите к ViPNet Coordinator HW интерфейсный кабель ИБП.

- 2 Включите взаимодействие ViPNet Coordinator HW с ИБП:

```
hostname# ups set monitoring on
```

- 3 Установите на ViPNet Coordinator HW режим master:

```
hostname# ups set mode master
```

- 4 Запустите взаимодействие ViPNet Coordinator HW с ИБП:

```
hostname# ups start
```

- 5 Проверьте взаимодействие ViPNet Coordinator HW с ИБП:

```
hostname# ups show status
```

Если взаимодействие установлено, отобразится информация о текущем состоянии ИБП.



Внимание! Если вы переподключили интерфейсный кабель ИБП (то есть отсоединили его от ViPNet Coordinator HW, а затем снова подключили), то для восстановления взаимодействия ViPNet Coordinator HW с ИБП выполните последовательно команды `ups stop` и `ups start`.

На ViPNet Coordinator HW, который будет выступать в роли slave:

- 1 Включите взаимодействие ViPNet Coordinator HW с ИБП:

```
hostname# ups set monitoring on
```

- 2 Установите на ViPNet Coordinator HW режим slave:

```
hostname# ups set mode slave <master_IP>
```

ViPNet Coordinator HW, выступающий в роли master, должен быть доступен с данного ViPNet Coordinator HW по этому IP-адресу.

При подключении к ИБП кластера горячего резервирования в качестве IP-адреса мастера укажите адрес интерфейса синхронизации первого ViPNet Coordinator HW.

- 3 Запустите взаимодействие ViPNet Coordinator HW с ИБП:

```
hostname# ups start
```

4 Проверьте взаимодействие ViPNet Coordinator HW с ИБП:

```
hostname# ups show status
```

Если взаимодействие установлено, отобразится информация о текущем состоянии ИБП, подключенного к ViPNet Coordinator HW в роли master.

Если к ИБП подключается кластер горячего резервирования, то один ViPNet Coordinator HW будет главным, а другой — подчиненным. При этом никакие другие устройства к ИБП подключать нельзя. Для связи между двумя ViPNet Coordinator HW кластера используйте канал между интерфейсами синхронизации, так как на нем IP-адреса неизменны при переходе ViPNet Coordinator HW из активного режима кластера в пассивный и наоборот.

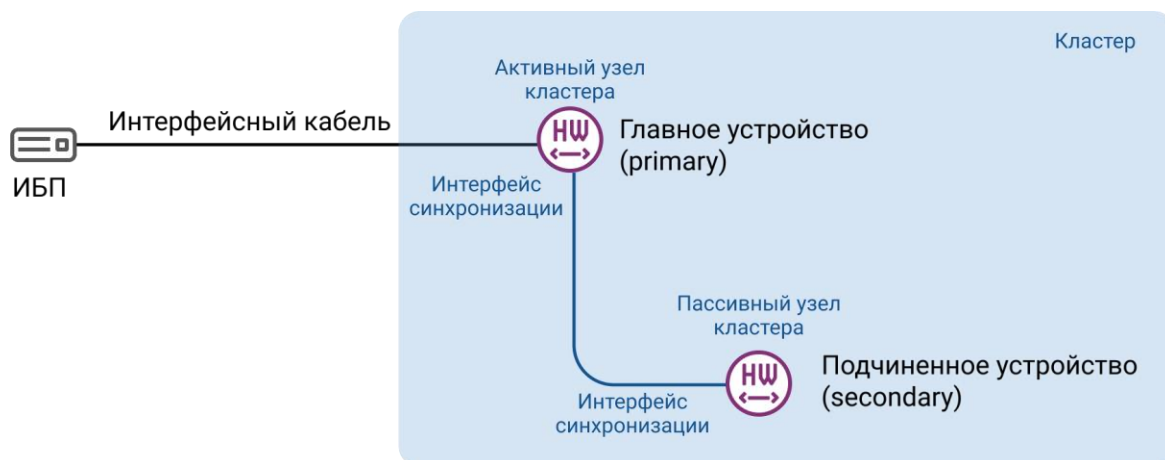


Рисунок 60. Схема подключения кластера к ИБП

14

Настройка транспортного сервера MFTP

Назначение транспортного сервера MFTP	294
Настройка протоколирования событий транспортного сервера MFTP	295
Настройка прямой маршрутизации между сетями ViPNet	296

Назначение транспортного сервера MFTP

Транспортный сервер MFTP предназначен для передачи транспортных конвертов MFTP между связанными узлами сети ViPNet. Сервер обеспечивает удаленное обновление на узлах сети:

- Справочников и ключей.
- Программного обеспечения ViPNet.
- Политик безопасности из ViPNet Policy Manager или Policy Management (модуль ViPNet Prime).

Транспортный сервер MFTP реализован в ViPNet Coordinator HW в виде службы `mftpd`.

Транспортные конверты MFTP представляют собой файлы с данными, которыми обмениваются между собой приложения, входящие в состав ПО ViPNet. Размер конверта не может превышать 2 Гбайт. Транспортный сервер передает конверты в соответствии с адресами получателей, указанными в заголовках конвертов, по каналу MFTP: узел-отправитель устанавливает TCP-соединение с узлом-получателем, проводится взаимная аутентификация узлов и передаются конверты.



Примечание. В исполнениях ViPNet Coordinator HW10, HW50 транспортный сервер MFTP выполняет обновление справочников, ключей, ПО ViPNet и политик безопасности только для собственного узла; конверты MFTP, адресованные другим узлам сети, не обрабатываются.

Поэтому при формировании структуры сети на ViPNet Coordinator HW в исполнениях ViPNet Coordinator HW10, HW50 не должны регистрироваться клиенты ViPNet.

Настройка транспортного сервера MFTP выполняется в конфигурационном файле `mftp.conf`. Подробнее о параметрах, содержащихся в файле `mftp.conf`, см. в документе «Справочник команд и конфигурационных файлов».

Настройка протоколирования событий транспортного сервера MFTP

Служба `mftpd` записывает информацию о событиях, происходящих в процессе работы транспортного сервера MFTP, в системный журнал (см. [Работа с системным журналом](#)).

Чтобы изменить параметры записи событий:

- 1 Завершите работу службы `mftpd`:

```
hostname# mftp stop
```

- 2 Откройте файл `mftp.conf` для редактирования:

```
hostname# mftp config
```

- 3 В секции `[debug]` при необходимости измените значения следующих параметров:

- o `debuglevel` — уровень важности событий, записываемых в журнал. Возможные значения: от -1 до 4 (по умолчанию 3). Чем выше уровень важности, тем более подробная информация записывается в журнал. Значение параметра -1 выключает ведение журнала (при этом некоторые важные системные события по-прежнему будут выводиться в журнал).

Уровень важности 4 используется только для диагностики возможных проблем и должен быть включен по рекомендации специалистов ИнфоТеКС. Не задавайте его для постоянного использования.

- o `debuglogfile` — источник информации, выводимой в журнал, в формате: `syslog:<facility.level>`, где:
 - `facility` — процесс, формирующий информацию. Возможные значения: `kern` (ядро), `user` (пользовательские программы) или `daemon` (службы).
 - `level` — уровень важности информации. Возможные значения: `err` (ошибка), `info` (информационное сообщение) или `debug` (отладочная информация).

Значение параметра `debuglogfile` по умолчанию — `syslog:daemon.debug`.

- 4 Сохраните изменения (**Ctrl+O**) и закройте редактор (**Ctrl+X**).

- 5 Запустите службу `mftpd`:

```
hostname# mftp start
```

Настройка прямой маршрутизации между сетями ViPNet

Если между двумя сетями ViPNet установлено межсетевое взаимодействие, то по умолчанию транспортные конверты MFTP от клиентов одной сети к клиентам другой сети проходят через шлюзовые координаторы сетей и обрабатываются на них транспортными серверами MFTP. Для снижения нагрузки на шлюзовые координаторы можно настроить прямую маршрутизацию между координаторами своей и доверенной сетей. В этом случае конверты будут обрабатываться транспортными серверами MFTP на координаторах сетей и не будут обрабатываться на шлюзовых координаторах.

Настройка прямой маршрутизации транспортных пакетов выполняется в конфигурационном файле `mftp.conf` в секциях `[channel]`. Для настройки необходимо:

- 1 Включить канал обмена с ViPNet Coordinator HW другой сети.
- 2 Указать идентификатор ViPNet Coordinator HW другой сети в параметре `transit`.

Настройка прямой маршрутизации без обработки транспортных конвертов на шлюзовых координаторах



Внимание! Для настройки такой схемы необходимо установить связь в управляющем ПО между координаторами доверенных сетей (HW-1 и HW-2).

Рассмотрим пример настройки прямой маршрутизации:

- У координаторов HW-1 и HW-2 установлена связь в управляющем ПО.
- Координаторы GW-1 и GW-2 выполняют функции шлюзовых координаторов.

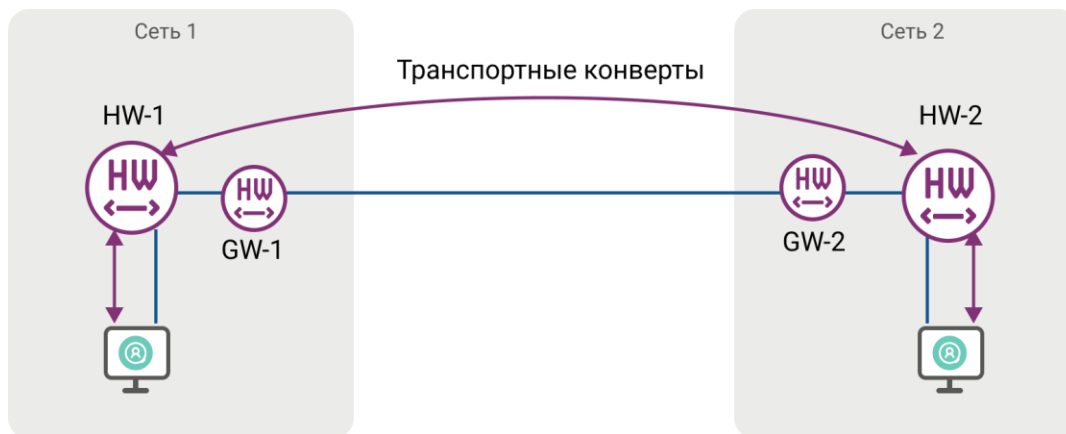


Рисунок 61. Прямая маршрутизация в обход иллюзовых координаторов

Для настройки прямой маршрутизации на координаторах HW-1 и HW-2:

- 1 Завершите работу службы mftpd:

```
hostname# mftp stop
```
- 2 Откройте файл для редактирования:

```
hostname# mftp config
```
- 3 Укажите идентификатор ViPNet Coordinator HW другой сети в параметре transit:
 - 3.1 На координаторе HW-1 в секции [channel] для канала обмена с HW-2 добавьте параметр transit и укажите его значение:

```
transit = <идентификатор HW-2>
```
 - 3.2 На координаторе HW-2 в секции [channel] для канала обмена с HW-1 добавьте параметр transit и укажите его значение:

```
transit = <идентификатор HW-1>
```

Таблица 9. Параметры секции [channel] файла mftp.conf

Настройки на HW-1	Настройки на HW-2
<code>id = <идентификатор HW-2></code>	<code>id = <идентификатор HW-1></code>
<code>name = HW_2</code>	<code>name = HW_1</code>
<code>type = mftp</code>	<code>type = mftp</code>
<code>off_flag = no</code>	<code>off_flag = no</code>
<code>call_flag = yes</code>	<code>call_flag = yes</code>
<code>ip = <IP-адрес HW-2></code>	<code>ip = <IP-адрес HW-1></code>
<code>call_timeout = -1</code>	<code>call_timeout = -1</code>
<code>last_port = 5000</code>	<code>last_port = 5000</code>
<code>last_call = 0</code>	<code>last_call = 0</code>
<code>last_err = 0</code>	<code>last_err = 0</code>
<code>transit = <идентификатор HW-2></code>	<code>transit = <идентификатор HW-1></code>

4 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).

5 Запустите службу `mftpd`:

```
hostname# mftp start
```

Настройка маршрутизации без обработки транспортных пакетов на шлюзовом координаторе своей сети

Рассмотрим пример настройки маршрутизации транспортных конвертов от клиентов из первой сети во вторую, без обработки на шлюзовом координаторе GW-1. Координаторы GW-1 и GW-2 выполняют функции шлюзовых координаторов. Для настройки такой схемы устанавливать связь в управляющем ПО между координаторами HW-1 и HW-2 необязательно.

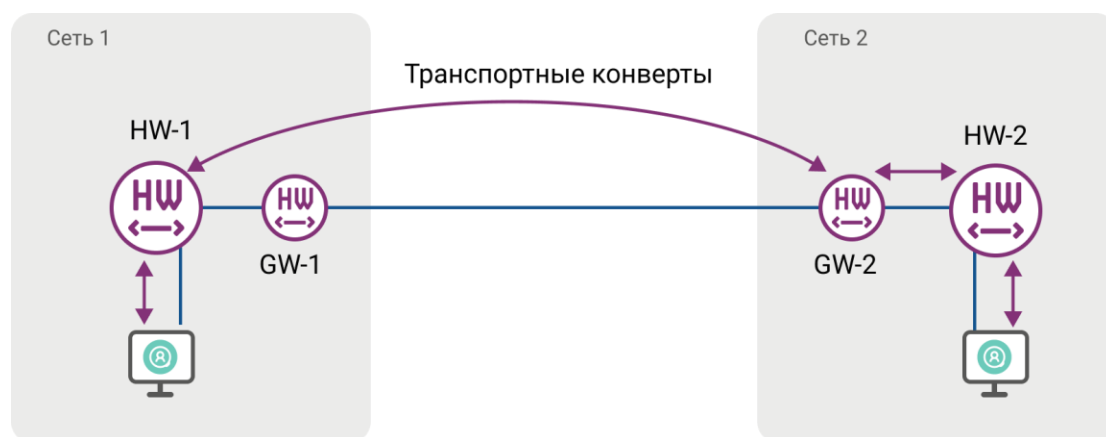


Рисунок 62. Прямая маршрутизация в обход одного шлюзового координатора

Для настройки прямой маршрутизации измените параметры секций `[channel]` координатора HW-1:

1 Завершите работу службы `mftpd`:

```
hostname# mftp stop
```

2 Откройте файл для редактирования:

```
hostname# mftp config
```

3 В секциях `[channel]` укажите параметры `off_flag`, `call_flag` и `transit`:



Внимание! Если между координаторами не установлена связь в управляющем ПО, секции `[channel]` для канала обмена с HW-2 не будет в файле `mftp.conf`. Чтобы настроить маршрутизацию укажите параметры только для канала обмена с GW-2.

Таблица 10. Параметры секции [channel] файла mftp.conf

Секция [channel] для канала обмена с HW-2	Секция [channel] для канала обмена со шлюзом GW-2
id = <идентификатор HW-2>	id = <идентификатор GW-2>
name = HW_2	name = GW_2
type = mftp	type = mftp
off_flag = yes	off_flag = no
call_flag = no	call_flag = yes
ip = <IP-адрес HW-2>	ip = <IP-адрес GW-2>
call_timeout = -1	call_timeout = -1
last_port = 5000	last_port = 5000
last_call = 0	last_call = 0
last_err = 0	last_err = 0
transit = <идентификатор GW-2>	transit = <идентификатор GW-2>

4 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).

5 Запустите службу mftpd:

```
hostname# mftp start
```

15

Обновление программного обеспечения

Порядок обновления ПО	301
Особенности обновления ПО	302
Проверка ЭП файла обновления	304
Проведение обновления ПО	308
Проверка корректности обновления	312

Порядок обновления ПО

Для обновления программного обеспечения ViPNet Coordinator HW выполните все действия из приведенной ниже таблицы в предложенном порядке.



Внимание! Обновите лицензию вашей сети перед обновлением ViPNet Coordinator HW.

Таблица 11. Порядок действий при обновлении ПО ViPNet Coordinator HW

Действие	Ссылка
Ознакомьтесь с особенностями обновления ПО ViPNet Coordinator HW и следуйте указанным рекомендациям.	Особенности обновления ПО
Если вы планируете обновить ПО локально, проверьте ЭП файла обновления	Проверка ЭП файла обновления
Выполните обновление ПО локально или удаленно.	Удаленное обновление ПО Локальное обновление ПО
В случае обновления ПО ViPNet Coordinator HW на кластере горячего резервирования следуйте специальным рекомендациям.	Обновление ПО на кластере горячего резервирования
Проверьте, корректно ли прошло обновление.	Проверка корректности обновления
Для исполнения ViPNet Coordinator VA: Если на виртуальной платформе Microsoft Hyper-V вы используете устаревший тип сетевого адаптера, то в настройках виртуальной платформы смените тип сетевого адаптера с Legacy Network Adapter на Network Adapter . Иначе скорость соединения будет ограничена 100 Мбит/с.	

Особенности обновления ПО

- Не используйте для обновления версию ниже, чем у вас установлено, это может привести к неполадкам в работе координатора. Начиная с версии 4.3.2 такие попытки заблокированы.
- Обновление ПО ViPNet Coordinator HW до версии 4.5.8 возможно с версий не ниже 4.3.2 (4.3.3 для виртуализированного исполнения). Если вы обновляйтесь с более ранних версий, сначала выполните обновление до версии 4.3.2 (4.3.3).
- При обновлении ПО с версии 4.5.8 на более новую версию ручная проверка ЭП файла обновления не требуется. Она будет выполняться автоматически в ViPNet Coordinator HW. Для этого необходимо наличие на ViPNet Coordinator HW актуального списка отозванных сертификатов (далее — CRL). Срок действия CRL — 1 год. Если на момент обновления ПО срок действия CRL уже истечёт, обновление не выполнится. Чтобы избежать этого, получите от представителей ИнфоТеКС актуальный CRL и загрузите его на ViPNet Coordinator HW до обновления ПО с помощью команды: `service cert import`.
- Если выполнено обновление ViPNet Coordinator HW с версии 4.3.2, проверьте параметры маршрутизации. Подробнее см. [Настройка статической маршрутизации](#).
- При обновлении до версии 4.5.8 в файл `mftp.conf` в секцию `[journal]` добавится новый параметр `max_log_space`. Этот параметр отвечает за максимальный объем дискового пространства, доступного для сохранения текстовых файлов, в которые выгружается информация из журнала конвертов. Подробнее см. документ «Справочник команд и конфигурационных файлов» описание параметра в секции `[journal]` файла `mftp.conf`.
- В версии 4.x фиксированные альтернативные каналы больше не поддерживаются. Теперь альтернативные каналы задаются с помощью множественных адресов доступа и их метрик. Секция `[channels]` и параметр `group` в секциях `[id]` файла `iplir.conf` больше не используются. Если раньше вы использовали фиксированные альтернативные каналы доступа для каких-либо узлов, после обновления настройте их новым способом — с помощью функции MultiWAN. Примеры настройки см. в разделе [Настройка функции MultiWAN](#). Подробнее о принципах назначения метрик см. в разделе [Настройка IP-адресов доступа к узлу и их приоритета](#).
- При обновлении ViPNet Coordinator HW с версии 4.2.4 на более позднюю изменится диапазон портов, используемых для динамической трансляции IP-адресов. Для применения новых значений перезагрузите ViPNet Coordinator HW.
- Начиная с версии 4.2.4, ViPNet Coordinator HW поддерживает доступ к координаторам по доменным именам (подробнее см. в документе «Справочник команд и конфигурационных файлов», описание параметра `fqdn` в секции `[id]` файла `iplir.conf`). Если ранее в [ViPNet Центр управления сетью](#) были заданы доменные имена координаторов, то после обновления ViPNet Coordinator HW до версии 4.2.4 и выше с более ранних версий выполните:
 - В ViPNet ЦУС или Prime измените заданные ранее доменные имена координаторов (например, вместо `Coordinator1` задайте имя `Coordinator11`) и отправьте на ViPNet Coordinator HW обновление справочников и ключей.

- В ViPNet ЦУС или Prime верните прежние доменные имена координаторов (например, вместо Coordinator11 задайте имя Coordinator1) и отправьте на ViPNet Coordinator HW обновление справочников и ключей.
- При обновлении ViPNet Coordinator HW с версии 4.2.1 до версии 4.2.4 и выше вместо параметра `tunnel_local_network` в секции `[misc]` файла `iplir.conf` используется параметр `tunnel_local_networks` в секциях `[id]` координаторов, связанных с ViPNet Coordinator HW. При этом старый параметр `tunnel_local_network` удаляется, а его значение присваивается новому параметру `tunnel_local_networks`.
- При обновлении ViPNet Coordinator HW до версии 4.3.0 из конфигурационного файла `failover.ini` будут удалены параметры `afterifconf`, `beforeifconf` и `passiveip`, так как они больше не используются.

Проверка ЭП файла обновления



Примечание. При обновлении ПО с версии 4.5.8 на более новую версию ручная проверка ЭП файла обновления не требуется. Она будет выполняться автоматически в ViPNet Coordinator HW. При обновлении ПО до версии 4.5.8 с более ранних версий проверьте ЭП файла обновления согласно данному разделу.

Ниже представлен пример проверки ЭП с помощью модуля File Unit программы ViPNet PKI Client.

Предварительные действия

- 1 Получите доверенным способом от представителей ИнфоТеКС следующие файлы (либо используйте те файлы, которые были получены ранее):
 - корневые сертификаты ключа проверки электронной подписи ИнфоТеКС (*.crt);
 - списки отозванных сертификатов (*.crl).
- 2 Получите любым способом:
 - файл обновления ПО ViPNet (*.lzh);
 - электронную подпись файла обновления (*.sig).
- 3 Скачайте с официального сайта ИнфоТеКС по протоколу HTTPS [программу ViPNet PKI Client](#) и установите ее.

Установка корневого сертификата и списка отозванных сертификатов

- 1 Запустите модуль File Unit программы ViPNet PKI Client и нажмите  **Настройки:**

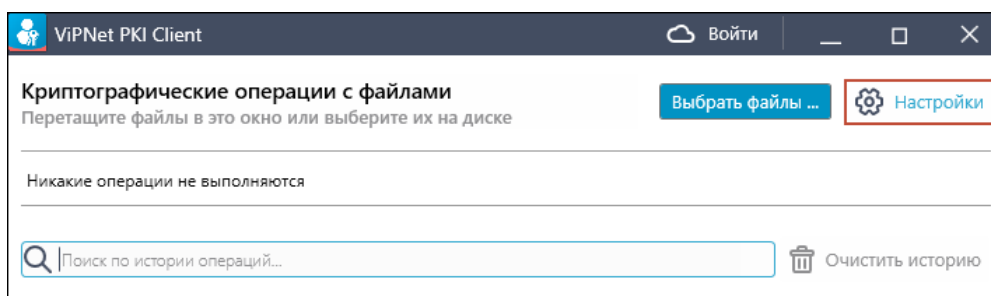


Рисунок 63. Настройка ViPNet PKI Client

- 2 Перейдите в раздел **Сертификаты** и нажмите  **Добавить сертификаты или CRL:**

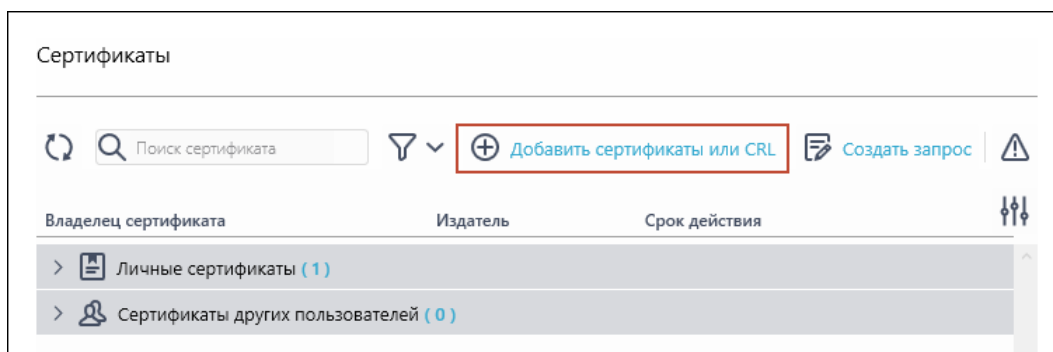


Рисунок 64. Импорт корневого сертификата и списка отозванных сертификатов

- 3 Выберите файлы корневых сертификатов (*.crt, *.cer) и файлы списков отозванных сертификатов (*.crl):

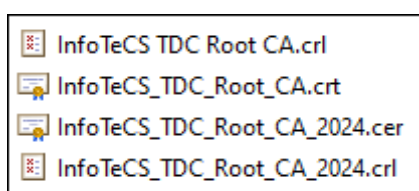


Рисунок 65. Корневые сертификаты и списки отозванных сертификатов

Внимание! Если ранее вы уже устанавливали корневой сертификат с именем InfoTeCS TDC Root CA, удостоверьтесь, что он отсутствует в хранилище «Промежуточные центры сертификации». Для этого:



- 1 Запустите диспетчер сертификатов с помощью команды `certmgr.msc` в Windows.
- 2 Откройте папку «Промежуточные центры сертификации» («Intermediate Certification Authorities»), а в ней — папку «Сертификаты» («Certificates»).
- 3 Если найдете сертификат с указанным именем, удалите его.

- 4 Нажмите **Добавить**:

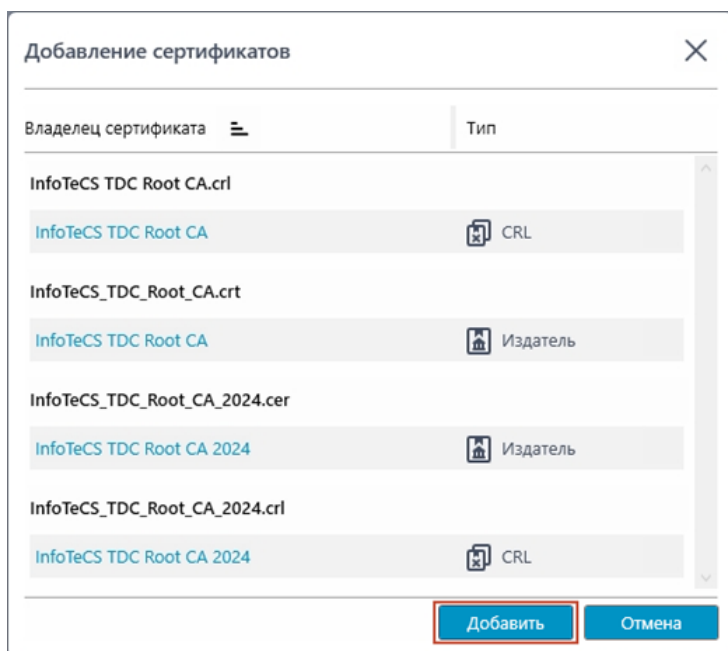


Рисунок 66. Подтверждение импорта корневых сертификатов и списков отозванных сертификатов

- 5 В окне **Предупреждение системы безопасности** подтвердите установку корневого сертификата.
- 6 В окне **Добавление сертификатов** нажмите кнопку **Заккрыть**.

Установка корневого сертификата и списка аннулированных сертификатов завершена.

Проверка электронной подписи

- 1 В главном окне программы ViPNet PKI Client выполните одно из действий:
 - Нажмите **Выбрать файлы** и выберите файл *.lzh и файл *.sig.
 - Перетащите файл *.lzh и файл *.sig в главное окно программы.

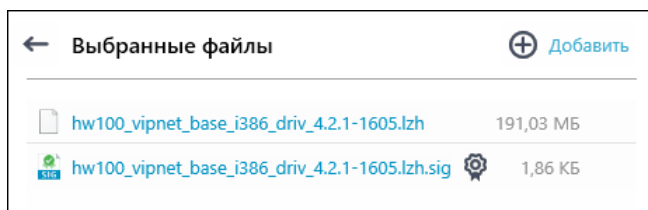


Рисунок 67. Выбор файла обновления и файла электронной подписи

- 2 Щелкните значок  напротив имени файла *.sig:

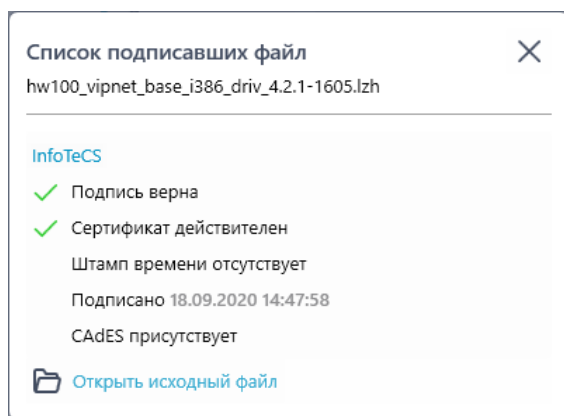


Рисунок 68. Проверка электронной подписи файла обновления

Проверка завершена.

Проведение обновления ПО

Удаленное обновление ПО



Внимание! Перед началом удаленного обновления обязательно ознакомьтесь с рекомендациями раздела [Особенности обновления ПО](#).

Перед обновлением сделайте [экспорт справочников, ключей и настроек ViPNet Coordinator HW в файл *.vbe](#). Это позволит вам восстановить их на ViPNet Coordinator HW, если обновление ПО пройдет некорректно (см. [Импорт справочников, ключей и настроек](#)).

Удаленное обновление ПО ViPNet Coordinator HW производится с помощью ViPNet ЦУС или Prime. Администратор сети ViPNet получает у представителей ИнфоТеКС файл обновления *.lzh, проверяет его целостность и из ViPNet ЦУС или Prime отправляет на ViPNet Coordinator HW.

Для разных исполнений ViPNet Coordinator HW передаются следующие файлы обновления ПО:

- для ViPNet Coordinator HW10 —
`hw10_vipnet_base_arm64_driv_<Major>.<Minor>.<Subminor>-<Build>.lzh;`
- для ViPNet Coordinator HW50 —
`hw50_vipnet_base_x86_64_driv_<Major>.<Minor>.<Subminor>-<Build>.lzh;`
- для ViPNet Coordinator HW100 —
`hw100_vipnet_base_x86_64_driv_<Major>.<Minor>.<Subminor>-<Build>.lzh;`
- для ViPNet Coordinator HW1000 —
`hw1000_vipnet_base_x86_64_driv_<Major>.<Minor>.<Subminor>-<Build>.lzh;`
- для ViPNet Coordinator HW2000 —
`hw2000_vipnet_base_x86_64_driv_<Major>.<Minor>.<Subminor>-<Build>.lzh;`
- для ViPNet Coordinator HW5000 —
`hw5000_vipnet_base_x86_64_driv_<Major>.<Minor>.<Subminor>-<Build>.lzh;`
- для ViPNet Coordinator VA —
`va_vipnet_base_x86_64_driv_<Major>.<Minor>.<Subminor>-<Build>.lzh.`

Удаленное обновление ПО подробно описано в документах на ViPNet ЦУС или Prime. На ViPNet Coordinator HW обновление принимается и выполняется автоматически. После обновления выполняется перезагрузка.

Внимание!

Если на ViPNet Coordinator HW недостаточно свободного дискового пространства для выполнения обновления, обновление не выполнится, а на узел с ViPNet ЦУС или Prime будет отправлена транспортная квитанция «искажение».

Если проверка подлинности транспортного конверта, содержащего обновление ПО, завершилась неудачно, обновление не выполнится, а на узел с ViPNet ЦУС или Prime будет отправлена транспортная квитанция с символом «!» в начале имени файла.

Если вы управляете ViPNet Coordinator HW с помощью веб-интерфейса, после обновления ПО очистите кеш вашего браузера.

Локальное обновление ПО

Прежде чем начать локальное обновление:

- Ознакомьтесь с рекомендациями раздела [Особенности обновления ПО](#).
- У администратора сети получите файл обновления *.lzh, проверьте ЭП (см. [Проверка ЭП файла обновления](#)) и запишите его на USB-носитель.



Примечание. На один USB-носитель можно записать несколько файлов обновления для различных исполнений ViPNet Coordinator HW (разместив их в разных каталогах). При обновлении будет доступен только тот файл, который соответствует данному исполнению.

- Выполните [экспорт справочников, ключей и настроек ViPNet Coordinator HW в файл *.vbe](#). Это позволит вам восстановить их на ViPNet Coordinator HW, если обновление ПО пройдет некорректно (см. [Импорт справочников, ключей и настроек](#)).



Внимание! Обновление ПО должно выполняться при запущенной службе failoverd. Если служба failoverd остановлена, обновление ПО может завершиться неудачно.

Для локального обновления:

- 1 На ViPNet Coordinator HW выполните команду:

```
hostname# admin upgrade software usb
```
- 2 Когда появится предложение подключить USB-носитель, подключите его и подтвердите подключение.

Если USB-носитель не найден или на нем нет файлов обновления, появится соответствующее сообщение.
- 3 Для каждого файла на USB-носителе программа проверяет целостность и соответствие данному исполнению ViPNet Coordinator HW.

Затем выводит пронумерованный список отобранных файлов.

- 4 Введите номер нужного файла обновления и нажмите **Enter**.



Внимание! При использовании кластера горячего резервирования локальное обновление ПО должно быть произведено на обоих узлах кластера (см. [Обновление ПО на кластере горячего резервирования](#)).

- 5 Дождитесь сообщения о результате обновления.
- 6 Перезагрузите ViPNet Coordinator HW, чтобы обновление вступило в силу.

Обновление ПО на кластере горячего резервирования

Обновление ПО ViPNet Coordinator HW на кластере горячего резервирования может выполняться удаленно или локально. Удаленное обновление выполняется автоматически и одновременно на обоих узлах кластера. При этом на пассивном узле обновление выполняется с задержкой, чтобы после перезагрузки он остался в пассивном режиме. Таким образом, время недоступности кластера определяется временем, необходимым на обновление ПО и перезагрузку активного узла и зависит от используемой аппаратной платформы и топологии сети ViPNet.



Внимание! При удаленном обновлении не разрывайте канал синхронизации, иначе обновление пройдет некорректно.

Для локального обновления ПО на кластере:

- 1 Убедитесь, что запущена служба failover:

```
hostname# failover show info
```

Если служба остановлена, обновление может завершиться неудачно.

- 2 Выключите интерфейс синхронизации на обоих узлах кластера:

```
hostname# inet ifconfig <интерфейс> down
```

Затем отсоедините Ethernet-кабель от ViPNet Coordinator HW.

- 3 Обновите ПО на пассивном ViPNet Coordinator HW (так же, как при работе в одиночном режиме (см. [Локальное обновление ПО](#))):

```
hostname# admin upgrade software usb
```

- 4 После обновления перезагрузите пассивный ViPNet Coordinator HW. Для этого в ответ на предложение перезагрузки нажмите **Enter**.

Убедитесь в стабильной работе ViPNet Coordinator HW:

- Проверьте текущее состояние служб ViPNet:

```
hostname# failover show info
```

Все службы должны быть запущены, система защиты от сбоев должна работать в режиме кластера.

- В течение некоторого времени (около 15 минут) наблюдайте за работой пассивного ViPNet Coordinator HW и убедитесь, что он не перезагружается.

5 Переключите активный ViPNet Coordinator HW (с необновленным ПО) в пассивный режим:

```
hostname# failover stop
```

```
hostname# failover start passive
```

В результате пассивный ViPNet Coordinator HW (с обновленным ПО) перейдет в активный режим, а ViPNet Coordinator HW со старой версией ПО окажется в пассивном режиме.

6 Обновите ПО на пассивном ViPNet Coordinator HW:

```
hostname# admin upgrade software usb
```

7 После обновления перезагрузите пассивный ViPNet Coordinator HW и убедитесь в его стабильной работе (как на шаге 3).

8 Соедините оба ViPNet Coordinator HW Ethernet-кабелем и включите интерфейс синхронизации:

```
hostname# inet ifconfig <интерфейс> up
```

9 Убедитесь, что синхронизация работает. Для этого измените какую-либо настройку в файле конфигурации на активном ViPNet Coordinator HW и через некоторое время проверьте наличие этих же изменений на пассивном ViPNet Coordinator HW. Чтобы изменения попали на пассивный ViPNet Coordinator HW, необходимо включить резервирование группы файлов конфигурации в файле `failover.ini` в секции `[sendconfig]`.

Начиная с версии 4.3.0, ViPNet Coordinator HW поддерживает работу DHCP-сервера в кластере горячего резервирования. Если вы обновили ViPNet Coordinator HW до версии 4.5.8 с предыдущей версии, в которой использовали функции DHCP-сервера и DHCP-relay, то для настройки работы этих функций в кластере:

1 На обоих узлах кластера остановите службу системы защиты от сбоев:

```
hostname# failover stop
```

2 Для слушающего интерфейса DHCP-сервера (DHCP-relay) назначьте статический IP-адрес:

```
hostname# inet ifconfig <интерфейс> address <IP-адрес> netmask <маска подсети>
```

Заданный IP-адрес должен быть указан в параметре `activeip` секции `[channel]` файла `failover.ini`.

3 На обоих узлах кластера запустите службу системы защиты от сбоев:

```
hostname# failover start
```

Проверка корректности обновления

После проведения обновления ПО:

- Проверьте номер версии обновленного ViPNet Coordinator HW:
`hostname# version`
- Проверьте содержимое конфигурационного файла `iplir.conf`.
- Проверьте значения параметров `tunnel` в секциях `[id]`.

16

Мониторинг состояния ViPNet Coordinator HW и просмотр журналов

Мониторинг по SNMP	314
Просмотр статистики передачи данных	324
Работа с журналом регистрации IP-пакетов	325
Работа с журналом MFTP	336
Работа с системным журналом	338
Работа с журналом DNS-запросов	345

Мониторинг по SNMP

С помощью [SNMP-агента](#), входящего в состав ПО ViPNet Coordinator HW, вы можете удаленно получать информацию о работе ViPNet Coordinator HW, а также настроить автоматическое оповещение о наиболее важных событиях (например, потеря связи с пассивным узлом кластера горячего резервирования).

Информация, доступная по протоколу SNMP, представлена в виде набора SNMP-параметров. Каждый параметр имеет уникальный идентификатор (OID). Список доступных SNMP-параметров ViPNet Coordinator HW вы можете просмотреть в приложении [Базы управляющей информации \(MIB\) для протокола SNMP](#).



Примечание. Все SNMP-параметры доступны только для чтения.

Для получения информации о состоянии ViPNet Coordinator HW вы можете использовать любую программу, предназначенную для сбора информации по протоколу SNMP (SNMP-менеджер), например:

- ViPNet StateWatcher. Данный программный комплекс собирает информацию о состоянии установленных на узлах компонентов ПО ViPNet, о текущей загрузке аппаратного обеспечения узлов, о состоянии сетевых интерфейсов и о статусе связей между узлами.
- ViPNet NVS. Система мониторинга для сбора информации о состоянии и составе устройств в сети ViPNet. ViPNet NVS — функциональный модуль системы управления ViPNet Prime.
- Утилита snmpwalk (пакет net-snmp).
- Система мониторинга Zabbix.

Сетевой узел с установленным SNMP-менеджером называется сервером мониторинга.



Примечание. В качестве сервера мониторинга мы рекомендуем использовать защищенный или туннелируемый узел. Если будет использоваться открытый узел, необходимо добавить сетевые фильтры, разрешающие SNMP-трафик:

```
hostname# firewall local add src <IP-адрес сервера мониторинга> dst @local  
udp dport 161 pass  
  
hostname# firewall local add src @local dst <IP-адрес сервера мониторинга>  
udp dport 162 pass
```

Протокол SNMP поддерживает следующие способы получения информации:

- Передача параметров по запросу. При этом сервер мониторинга посылает SNMP-запрос на ViPNet Coordinator HW и получает в ответ значения необходимых параметров.
- Передача оповещений. SNMP-агент ViPNet Coordinator HW автоматически отправляет SNMP-оповещения на сервер мониторинга при наступлении определенного события.



Примечание. Чтобы SNMP-запросы обрабатывались корректно, необходимо отправлять их на IP-адрес ViPNet Coordinator HW, который доступен с сервера мониторинга по кратчайшему маршруту.

Если вы импортируете на сервер мониторинга MIB-файлы, поддерживаемые ViPNet Coordinator HW, вы сможете:

- использовать в SNMP-запросах текстовые обозначения параметров вместо числовых (например, `infVipNetHostName` вместо `.1.3.6.1.4.1.10812.1.1.7`);
- получать текстовые обозначения некоторых числовых параметров в ответах на SNMP-запросы (например, при запросе состояния интерфейса, вместо значения 1 или 2, вы получите значение `up` или `down`);
- получать оповещения от ViPNet Coordinator HW в виде текстовых обозначений параметров вместо числовых (например, `infvpnWebStart` вместо `.1.3.6.1.4.1.10812.1.0.10`).

SNMP-агент ViPNet Coordinator HW поддерживает одновременную работу по протоколам SNMPv1, SNMPv2c и SNMPv3.

Для ограничения доступа к SNMP-параметрам используются следующие подходы:

- В SNMPv1 и SNMPv2c используется параметр `community string`, который является идентификатором и паролем доступа к SNMP-параметрам. ViPNet Coordinator HW ответит на SNMP-запрос, если `community string` в запросе совпадает с любым из `community string`, заданных на ViPNet Coordinator HW.

Для получения SNMP-оповещений сервером мониторинга в настройках его SNMP-менеджера необходимо задать параметр `community-trap string`. Чтобы сервер мониторинга смог принять адресованные ему оповещения, `community-trap string` должен быть указан и в настройках отправки оповещений на ViPNet Coordinator HW.

- В SNMPv3 используются учетные записи пользователей SNMP. ViPNet Coordinator HW ответит на SNMP-запрос, если пользователю разрешен доступ на чтение параметров и данные пользователя в запросе совпадают с данными любого из пользователей, заданных на ViPNet Coordinator HW:
 - имя пользователя;
 - пароль пользователя и алгоритм хэширования пароля;
 - способ аутентификации;
 - ключ и алгоритм шифрования (если заданы).

Для получения SNMP-оповещений сервером мониторинга в настройках его SNMP-менеджера необходимо указать данные одного из пользователей SNMP, заданных на ViPNet Coordinator HW. При использовании оповещений типа TRAP, в настройках SNMP-менеджера дополнительно к учетным данным пользователя необходимо указать параметр `engineID` (OID `.1.3.6.1.6.3.10.2.1.1.0`).

Настройка SNMP-агента

1 Для протоколов SNMPv1 и SNMPv2c:

1.1 Добавьте community string для чтения SNMP-параметров:

```
hostname# inet snmp community add
```

Длина community string — от 6 до 18 символов. Разрешенные символы — прописные и строчные буквы латинского алфавита, цифры и специальные символы: . * / - : _ ? = @ , & ().

Максимальное количество community string — 16.

1.2 Задайте адрес сервера мониторинга для отправки SNMP-оповещений:

```
hostname# inet snmp trapsink add <адрес> [port <номер>] [{v1 | inform}]
```

- <адрес> — IP-адрес или доменное имя сервера мониторинга,
- <номер> — номер UDP-порта,
- v1 — отправлять оповещения типа TRAP по протоколу SNMPv1,
- inform — отправлять оповещения типа INFORM по протоколу SNMPv2c.

Если порт и тип оповещений не указан, устанавливаются следующие значения:

- Тип отправляемых оповещений — TRAP.
- Протокол отправки оповещений — SNMPv2c.
- Порт, на который отправляются оповещения — UDP 162.

При выполнении команды необходимо ввести community-trap string, которое задано на сервере мониторинга для приема SNMP-оповещений.

Вы можете использовать до 16 сетевых узлов в качестве серверов мониторинга.

Внимание! Если задан порт, отличный от UDP 162, необходимо добавить сетевой фильтр, разрешающий передачу SNMP-оповещений на данный порт:



- для защищенного узла: hostname# firewall vpn add src @local dst <идентификатор защищенного узла> udp dport <номер порта> pass
 - для туннелируемого узла: hostname# firewall tunnel add src @local dst <IP-адрес> udp dport <номер порта> pass
 - для открытого узла: hostname# firewall local add src @local dst <IP-адрес> udp dport <номер порта> pass
-

1.3 Разрешите чтение SNMP-параметров ViPNet Coordinator HW:

```
hostname# inet snmp v2 ro on
```

1.4 Разрешите отправку SNMP-оповещений:

```
hostname# inet snmp v2 traps on
```

2 Для протокола SNMPv3:

2.1 Добавьте нового пользователя для доступа к SNMP-параметрам ViPNet Coordinator HW:

```
hostname# inet snmp user add <имя пользователя> [{md5 | sha}]
```

При добавлении пользователя требуется задать пароль и алгоритм хэширования пароля MD5 (по умолчанию) или SHA.

Длина имени пользователя — от 1 до 32 символов. Разрешенные символы — прописные и строчные буквы латинского алфавита и цифры.

Длина пароля — от 8 до 32 символов. Разрешенные символы — прописные и строчные буквы латинского алфавита, цифры и специальные символы: ! @ # \$ % ^ & * () - _ + = ; : ' " , . < > / ? \ | ` ~ [] { }.



Внимание! Не используйте повторяющиеся последовательности символов в пароле. Иначе его хэш может совпасть с хэшем другого пароля. Например, пароли AbcdAbcd и AbcdAbcdAbcd будут иметь одинаковые хэши.



Примечание. По умолчанию пользователю разрешено чтение SNMP-параметров. Вы можете запретить чтение SNMP-параметров:

```
hostname# inet snmp user set <имя пользователя> read off
```

2.2 Для шифрования SNMP-трафика между сервером мониторинга и ViPNet Coordinator HW (по алгоритму AES-128) создайте ключ шифрования для пользователя:

```
hostname# inet snmp user set <имя пользователя> key
```

Длина ключа шифрования — от 8 до 32 символов. Разрешенные символы — прописные и строчные буквы латинского алфавита, цифры и специальные символы: ! @ # \$ % ^ & * () - _ + = ; : ' " , . < > / ? \ | ` ~ [] { }.



Внимание! Не используйте повторяющиеся последовательности символов в ключе шифрования. Иначе его хэш может совпасть с хэшем другого ключа. Например, ключи AbcdAbcd и AbcdAbcdAbcd будут иметь одинаковые хэши.



Примечание. Если задан ключ шифрования, его необходимо указывать в SNMP-запросах наряду с другими данными пользователя SNMP.

2.3 Разрешите отправку SNMP-оповещений для пользователя:

```
hostname# inet snmp user set <имя пользователя> trapsess on
```

2.4 Задайте адрес сервера мониторинга для отправки SNMP-оповещений:

```
hostname# inet snmp user set <имя пользователя> trapsess add <адрес> [port <номер>] [inform]
```

- <адрес> — IP-адрес или доменное имя сервера мониторинга,
- <номер> — номер UDP-порта,
- inform — отправлять оповещения типа INFORM.

Если порт и тип оповещений не указан, устанавливаются следующие значения:

- Тип отправляемых оповещений — TRAP.
- Порт, на который отправляются сообщения — UDP 162.

Внимание! Если задан порт, отличный от UDP 162, добавьте сетевой фильтр, разрешающий передачу SNMP-оповещений на данный порт:



- для защищенного узла: `hostname# firewall vpn add src @local dst <идентификатор защищенного узла> udp dport <номер порта> pass`
 - для туннелируемого узла: `hostname# firewall tunnel add src @local dst <IP-адрес> udp dport <номер порта> pass`
 - для открытого узла: `hostname# firewall local add src @local dst <IP-адрес> udp dport <номер порта> pass`
-

2.5 Разрешите чтение SNMP-параметров по протоколу SNMPv3:

```
hostname# inet snmp v3 ro on
```

2.6 Разрешите отправку SNMP-оповещений по протоколу SNMPv3:

```
hostname# inet snmp v3 traps on
```

3 По умолчанию SNMP-агент принимает запросы на UDP-порт 161. При необходимости измените номер порта:

```
hostname# inet snmp port <порт>
```



Внимание! При изменении номера порта добавьте соответствующий сетевой фильтр.

4 Включите запуск SNMP-агента при загрузке ViPNet Coordinator HW:

```
hostname# inet snmp autostart on
```

5 По умолчанию SNMP-агент регистрирует в системном журнале события с уровнем важности error. При необходимости измените уровень важности:

```
hostname# inet snmp logging <уровень важности>
```

Список возможных значений параметра <уровень важности>:

- o off — регистрация событий выключена.
- o critical — регистрируются критические ошибки, после которых SNMP-агент не может продолжить работу.
- o error — регистрируются ошибки, после которых SNMP-агент может продолжать работу.
- o info — регистрируется полная информация о работе SNMP-агента.
- o debug — регистрируется служебная информация, используемая при отладке.

6 При необходимости задайте информацию о вашем ViPNet Coordinator HW, которая будет доступна по SNMP при чтении SNMP-параметров:

- Чтобы задать параметр **Контактное лицо** (sysContact OID .1.3.6.1.2.1.1.4.0):
hostname# inet snmp system contact
- Чтобы задать параметр **Имя устройства** (sysName OID .1.3.6.1.2.1.1.5.0):
hostname# inet snmp system name
- Чтобы задать параметр **Местоположение** (sysLocation OID .1.3.6.1.2.1.1.6.0):
hostname# inet snmp system location



Примечание. Параметры **Имя устройства** и **Местоположение** не синхронизируются между узлами кластера.

7 Включите SNMP-агент:

```
hostname# inet snmp start
```

8 Проверьте выполненную настройку SNMP-агента:

```
hostname# inet snmp show
```

Полный список команд по настройке SNMP-агента см. в документе «Справочник команд и конфигурационных файлов».

Мониторинг кластера

Вы можете следить за состоянием обоих узлов кластера по протоколу SNMP. Прием SNMP-запросов и отправку SNMP-оповещения выполняет активный узел. Для мониторинга пассивного узла кластера активный узел перенаправляет полученные SNMP-запросы на пассивный через интерфейс синхронизации и возвращает ответы отправителю.

Поскольку узлы кластера могут переключаться, SNMP-запрос должен указывать, какой из узлов должен на него ответить. Для этого используются следующие параметры SNMP-агента:

- При использовании SNMPv1 и SNMPv2c для каждого ViPNet Coordinator HW кластера создается community string для режима кластера. SNMP-запрос с таким community string будет адресован конкретному ViPNet Coordinator HW независимо от его текущего режима работы в кластере (активный или пассивный). При переключении системы защиты от сбоев в одиночный режим, community string для режима кластера автоматически удаляется.
- При использовании SNMPv3 каждому узлу кластера назначается свой контекст (строка, идентифицирующая ViPNet Coordinator HW в кластере). Контексты создаются автоматически на активном узле в формате Node<порядковый номер>, где <порядковый номер> — число от 1, которое выбирается по возрастанию в соответствии с IP-адресом интерфейса синхронизации.

Для мониторинга кластера по протоколу SNMP:

- 1 Настройте SNMP-агент на активном узле кластера (см. [Настройка SNMP-агента](#)).

Настройки будут переданы на пассивный узел при последующей синхронизации.



Примечание. Если на момент развертывания кластера SNMP-агент уже настроен на одном из ViPNet Coordinator HW, запускайте `failover` в режиме кластера сначала на ViPNet Coordinator HW с настроенным SNMP-агентом. Тогда другой узел кластера получит актуальную конфигурацию SNMP-агента от активного узла. Исходная конфигурация агента на пассивном узле при этом удаляется.

2 Для SNMPv1 и SNMPv2c выполните действия на активном узле кластера:

2.1 Задайте community string для мониторинга узла кластера:

```
hostname# inet snmp cluster node <IP-адрес> community
```

где <IP-адрес> — IP-адрес интерфейса синхронизации узла кластера.

В ответ на сообщение `Type community:` введите community string, отличное от других community string. Допустимая длина — от 6 до 18 символов: прописные и строчные буквы латинского алфавита, цифры и специальные символы: (. * / - : _ ? = @ , &).



Примечание. Просмотреть созданные для режима кластера community string вы можете:

```
hostname# inet snmp cluster show community
```

2.2 Разрешите чтение SNMP-параметров узлов кластера по протоколам SNMPv1 и SNMPv2c:

```
hostname# inet snmp cluster v2 on
```

Примечание. Если выводится ошибка: `SNMPv1 and SNMPv2c support is OFF. Please, turn it on first`, выполните действия:



- разрешите чтение SNMP-параметров на активном узле кластера: `hostname# inet snmp v2 ro on`
- повторно разрешите чтение SNMP-параметров: `hostname# inet snmp cluster v2 on`

3 Для SNMPv3 на активном узле кластера:

3.1 Просмотрите контексты, присвоенные узлам кластера, чтобы использовать их в SNMP-запросах:

```
hostname# inet snmp cluster show
```

3.2 Чтобы изменить контекст узла:

```
hostname# inet snmp cluster node <IP-адрес> context <контекст>
```

где <IP-адрес> – IP-адрес интерфейса синхронизации узла кластера.

Допустимая длина контекста — от 1 до 32 символов: прописные и строчные буквы латинского алфавита и цифры.

4 На пассивном узле запустите SNMP-агент:

```
hostname# inet snmp start
```

- 5 В зависимости от версии протокола SNMP используйте созданные community string или контексты в SNMP-запросах для получения информации о состоянии конкретного узла кластера (см. руководство по настройке используемого SNMP-менеджера):
- При использовании протоколов SNMPv1 и SNMPv2c отправляйте SNMP-запрос с community string соответствующего узла кластера.
 - При использовании протокола SNMPv3 отправляйте SNMP-запрос с учетными данными пользователя, созданного с помощью команды `inet snmp user add`, и контекстом соответствующего узла. Если запрос не содержит контекст, ответ будет получен от активного узла кластера.

Пример настройки системы мониторинга

Рассмотрим пример: необходимо узнать кто из ViPNet-клиентов, работающих в определенный промежуток времени подключался с известного IP-адреса к сети через ViPNet Coordinator HW.

По протоколу SNMP можно получить информацию о подключенных ViPNet-клиентах, для которых координатор является сервером IP-адресов (подробнее см. [VIPNET-MIB](#)):

- идентификатор узла;
- имя узла;
- IP-адрес точки доступа к узлу;
- порт точки доступа к узлу.

Используя эти данные настроим систему мониторинга так, чтобы в ней можно было посмотреть историю изменений IP-адреса ViPNet-клиента:



Примечание. Ниже представлен сценарий с особенностями настройки для системы мониторинга Zabbix 6.4. Подробные настройки системы читайте в документах на систему мониторинга.

Перед настройкой системы мониторинга настройте SNMP-агент ViPNet Coordinator HW (см. [Настройка SNMP-агента](#)).

- 1 Создайте шаблон для узлов мониторинга с именем ViPNet.
- 2 Создайте правило обнаружения. Система мониторинга будет автоматически создавать данные о ViPNet-клиентах в виде элементов данных узла мониторинга (координатора).

Заполните обязательные поля следующим образом:

- **Name:** vipnet.client.ip
- **Type:** SNMP agent
- **Key:** vipnet.client.ip

- **SNMP OID:** discovery[#{#CLIENTID},1.3.6.1.4.1.10812.1.4.1.1.1,#{#CLIENTNAME},1.3.6.1.4.1.10812.1.4.1.1.2]
- **Update interval:** время обнаружения новых клиентов на координаторе (по умолчанию 1m)
- **Keep lost resources period:** время, по истечению которого будут удалены данные пользователей, которые больше не зарегистрированы на координаторе.



Совет. В системе мониторинга идентификатор узла сети ViPNet будет отображаться через пробел: 00 5D 20 21 Client-01. Вы можете настроить отображение без пробелов (005D2021 Client-01):

- 1 Перейдите на вкладку **Preprocessing**.
- 2 В поле **Name** выберите **JavaScript**.
- 3 В поле **Parameters** введите: `return (value.replace(/(..)\s(..)\s(..)\s(..)/g, '$1$2$3$4'))`.

3 Создайте прототип элемента данных. Заполните обязательные поля следующим образом:

- **Name:** {#CLIENTID} {#CLIENTNAME}
- **Type:** SNMP agent
- **Key:** vipnet.client.ip[#{#CLIENTID}]
- **Type of information:** Text
- **SNMP OID:** 1.3.6.1.4.1.10812.1.4.1.1.3.{#SNMPINDEX}
- **Update interval:** время опроса координатора (по умолчанию 1m)
- **History storage period:** Storage period. Также укажите время хранения данных о ViPNet-клиентах (по умолчанию 90 дней)

4 Создайте прототип триггера для оповещений. Заполните обязательные поля следующим образом:

- **Name:** Client IP address changed
- **Event name:** Client {#CLIENTID} : IP address changed to {ITEM.LASTVALUE1}
- **Severity:** важность триггера
- **Expression:** `last(/ViPNet/vipnet.client.ip[#{#CLIENTID}]), #1)<>last(/ViPNet/vipnet.client.ip[#{#CLIENTID}], #2) and length(last(/ViPNet/vipnet.client.ip[#{#CLIENTID}]))>0`
- **OK event generation:** None
- Установите флажок **Allow manual close**

5 Добавьте координатор в систему мониторинга, используя созданный шаблон. **SNMP Community** должно быть таким же, какое вы устанавливали на координаторе.

6 Создайте экран мониторинга.

7 Создайте виджет, в котором будет отображаться информация о ViPNet-клиентах и их IP-адресах подключения.

- 8 Создайте виджет, в котором будут отображаться события со сменой IP-адресов ViPNet-клиентов. Для просмотра истории IP-адресов подключения ViPNet-клиента нажмите на IP-адрес клиента в созданном ранее виджете и выберите **Latest values**. Отобразится история изменения IP-адреса ViPNet-клиента с возможностью фильтрации по временному диапазону.

Также вы можете настроить систему мониторинга так, чтобы она отображала историю подключения или отключения ViPNet-клиентов. Для этого в предыдущем сценарии на 4 шаге создайте прототипы триггеров:

- **Name:** {#CLIENTID} {#CLIENTNAME} connected
- **Event name:** {#CLIENTID} {#CLIENTNAME} connected
- **Severity:** важность триггера
- **Expression:** `nodata(/ViPNet/vipnet.client.ip[{#CLIENTID}],1m)=0`
- **OK event generation:** None
- Установите флажок **Allow manual close**

- **Name:** {#CLIENTID} {#CLIENTNAME} disconnected
- **Event name:** {#CLIENTID} {#CLIENTNAME} disconnected
- **Severity:** Важность триггера
- **Expression:** `nodata(/ViPNet/vipnet.client.ip[{#CLIENTID}],1m)=1`
- **OK event generation:** None
- Установите флажок **Allow manual close**

История подключения или отключения ViPNet-клиентов будет доступна в разделе Zabbix: **Monitoring -> Problems**. Для удобства просмотра вы можете отфильтровать записи по имени координатора или ViPNet-клиента.

Просмотр статистики передачи данных

Статистика передачи данных ведется отдельно по всем интерфейсам с момента включения ViPNet Coordinator HW. Статистика учитывает текущую скорость интерфейсов и объем трафика. После запуска команды статистика отображается в режиме реального времени с заданным интервалом обновления (по умолчанию — 3 секунды). Чтобы завершить просмотр статистики, нажмите **Ctrl+C**. В кластере просмотр статистики доступен на обоих узлах. После перезагрузки ViPNet Coordinator HW или удаления интерфейса данные статистики сбрасываются. Для ViPNet Coordinator VA статистика может быть сброшена после перезапуска интерфейса.

Чтобы просмотреть статистику передачи данных на всех физических и агрегированных интерфейсах:

```
hostname> inet show traffic
```

Please wait for 3 sec. Gathering traffic....

Interface	RX speed	TX speed	RX Total	TX Total
-----	-----	-----	-----	-----
eth0	10,24 Kbit/s	12,37 Kbit/s	456,12 KB	517,37 KB
eth1	999,99 bit/s	634,12 bit/s	12,31 KB	6,27 KB
bond0	1,00 Gbit/s	1000,00 Mbit/s	1,10 TB	1022,12 GB
eth2	512,00 Mbit/s	500,00 Mbit/s	563,20 GB	511,06 GB
eth3	512,00 Mbit/s	500,00 Mbit/s	563,20 GB	511,06 GB

Mon Feb 2 09:55:16 2024

Чтобы просмотреть статистику передачи данных виртуального интерфейса eth2.10 с интервалом обновления 10 секунд:

```
hostname> inet show traffic interface eth2.10 interval 10
```

Please wait for 10 sec. Gathering traffic....

Interface	RX speed	TX speed	RX Total	TX Total
-----	-----	-----	-----	-----
eth2.10	500,00 Mbit/s	530,00 Mbit/s	565,25 GB	515,07 GB

Mon Feb 2 09:59:19 2024

Работа с журналом регистрации IP-пакетов

Служба `iplircfg` ведет журнал регистрации IP-пакетов, в который заносится информация обо всех IP-пакетах (зашифрованных и открытых), проходящих через сетевые интерфейсы узла.

В этот журнал вносится информация не о соединениях, а об IP-пакетах, которые проходят через сетевые интерфейсы, поэтому:

- каждый пропущенный и заблокированный локальный IP-пакет отображается в журнале один раз — на том интерфейсе, через который он прошел, или на том интерфейсе, на котором он был заблокирован;
- каждый пропущенный транзитный IP-пакет отображается в журнале дважды: первый раз — на интерфейсе, через который он пришел, второй раз — на интерфейсе, через который он ушел.

Примечание. По умолчанию регистрируются только записи о заблокированных IP-пакетах и изменении адресов сетевых узлов. Чтобы включить регистрацию записей обо всех пакетах, проходящих через сетевой интерфейс:



- 1 `hostname# iplir stop`
 - 2 `hostname# iplir config <интерфейс или группа интерфейсов>`
 - 3 в секции `[db]` для параметра `registerall` установите значение `on`.
 - 4 `hostname# iplir start`
-

Просмотр журнала регистрации IP-пакетов

Чтобы просмотреть записи из журнала регистрации IP-пакетов:

```
hostname# iplir view
```

В результате откроется окно **Set search parameters** для задания параметров поиска записей в журнале.

Примечание. При работе ViPNet Coordinator HW в режиме кластера горячего резервирования журнал регистрации IP-пакетов синхронизируется между узлами кластера (по умолчанию). При этом просмотреть журнал вы можете только на активном узле кластера.

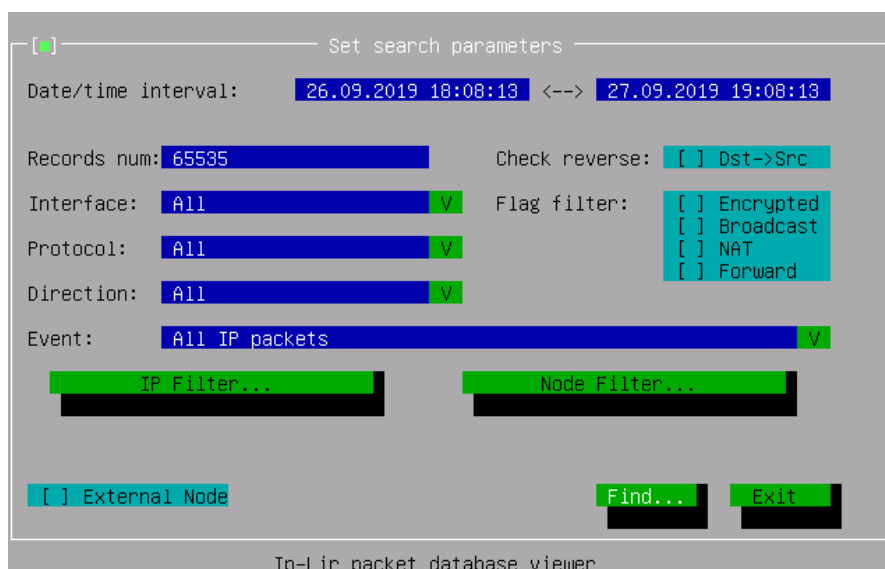


Рисунок 69. Задание параметров поиска записей в журнале регистрации IP-пакетов

По умолчанию предполагается просмотр записей обо всех IP-пакетах, прошедших через сетевые интерфейсы собственного узла за последние сутки (плюс один час). Чтобы просмотреть такие записи, нажмите **Find**.

Чтобы выбрать записи из журнала регистрации IP-пакетов для просмотра:

1 Укажите параметры пакетов:

- **Date/time interval** — период времени, в который были зарегистрированы пакеты в формате: `дд.мм.гггг чч:мм:сс <--> дд.мм.гггг чч:мм:сс`

По умолчанию — текущие сутки.



Внимание! Если на ViPNet Coordinator HW было изменено время, учитывайте временной сдвиг при указании периода времени, в который были зарегистрированы пакеты.

Если на ViPNet Coordinator HW время перевели назад нарушается монотонность времени событий. Журнал регистрации может перестать отображать записи. При необходимости проанализировать события в данный период времени (в котором была нарушена монотонность) экспортируйте журнал регистрации IP-пакетов и проанализируйте его сторонними средствами (см. [Экспорт журнала регистрации IP-пакетов на USB-носитель](#)).

- **Records num** — количество записей о зарегистрированных пакетах, которое будет отображено (по умолчанию — 65535).
- **Interface** — сетевой интерфейс, на котором были обработаны пакеты. По умолчанию — все доступные интерфейсы (**All**).
- **Protocol** — IP-протокол, по которому были переданы пакеты. Возможные значения:
 - **All** (по умолчанию) — любой протокол;
 - **ICMP** — только протокол ICMP;
 - **TCP** — только протокол TCP;
 - **UDP** — только протокол UDP.

- **Except TCP, UDP, ICMP** — любой протокол, кроме TCP, UDP и ICMP.
- **Direction** — направление пакетов:
 - **All** (по умолчанию) — входящие и исходящие пакеты;
 - **Incoming** — входящие пакеты;
 - **Outgoing** — исходящие пакеты.
- **Event** — тип события, которому соответствует IP-пакет (см. [Типы событий в журнале регистрации IP-пакетов](#)). По умолчанию — события всех типов (**All IP packets**).
- **Check reverse** — включение в журнал ответных пакетов от получателя отправителю (**Dst -> Src**). Устанавливать этот флажок имеет смысл только, если в качестве отправителя и/или получателя пакетов указаны конкретные IP-адреса (**IP Filter**) или узлы (**Node Filter**).
- **Flag filter** — вид пакетов:
 - **Encrypted** — зашифрованные пакеты;
 - **Broadcast** — широковещательные пакеты;
 - **NAT** — транслированные пакеты;
 - **Forward** — транзитные пакеты.

Вы можете выбрать один или несколько видов пакетов, установив соответствующие флажки.

2 Чтобы задать IP-адреса или порты отправителя или получателя пакетов, нажмите **IP Filter** и в открывшемся окне:

- если в списке **Protocol** вы выбрали все протоколы или все протоколы, кроме TCP, UDP и ICMP, укажите:
 - **Source IP address** — IP-адрес (**Value**) или диапазон IP-адресов (**Range**) отправителей пакетов. По умолчанию — любой адрес (**All**);
 - **Destination IP address** — IP-адрес (**Value**) или диапазон IP-адресов (**Range**) получателей пакетов. По умолчанию — любой адрес (**All**);
- если в списке **Protocol** вы выбрали протокол TCP или UDP, помимо IP-адресов отправителя и получателя, укажите:
 - **Source port** — порт (**Value**) или диапазон портов (**Range**) отправителей. По умолчанию — все порты (**All**);
 - **Destination port** — порт (**Value**) или диапазон портов (**Range**) получателей. По умолчанию — все порты (**All**);



Примечание. При задании в поле **Value** числового значения использование числа 0 не допускается.

- если в списке **Protocol** вы выбрали протокол ICMP, помимо IP-адресов отправителя и получателя, укажите:

- ICMP type — тип (Value) или диапазон типов (Range) ICMP-пакетов.
- ICMP code — код (Value) или диапазон кодов (Range) ICMP-пакетов.

3 Чтобы выбрать узел отправителя или получателя, нажмите **Node Filter** и в открывшемся окне:

3.1 Для выбора узла отправителя нажмите **Select source Node**, для выбора узла получателя — **Select destination Node**.

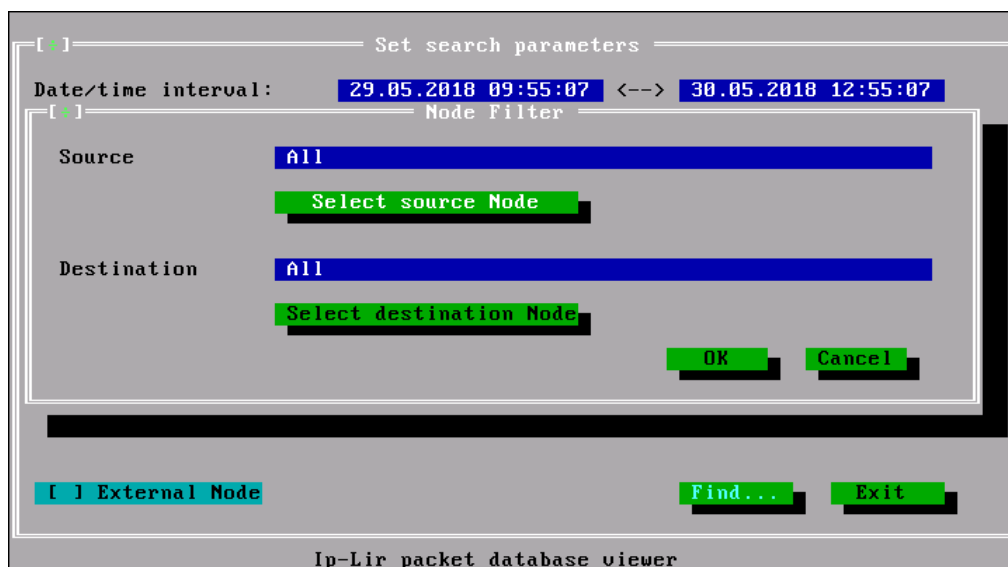


Рисунок 70. Выбор узла отправителя и получателя

В результате откроется таблица со списком защищенных узлов, с которыми ViPNet Coordinator HW имеет связь. В левом столбце таблицы указан шестнадцатеричный идентификатор узла (ID), в правом — имя узла (Name).

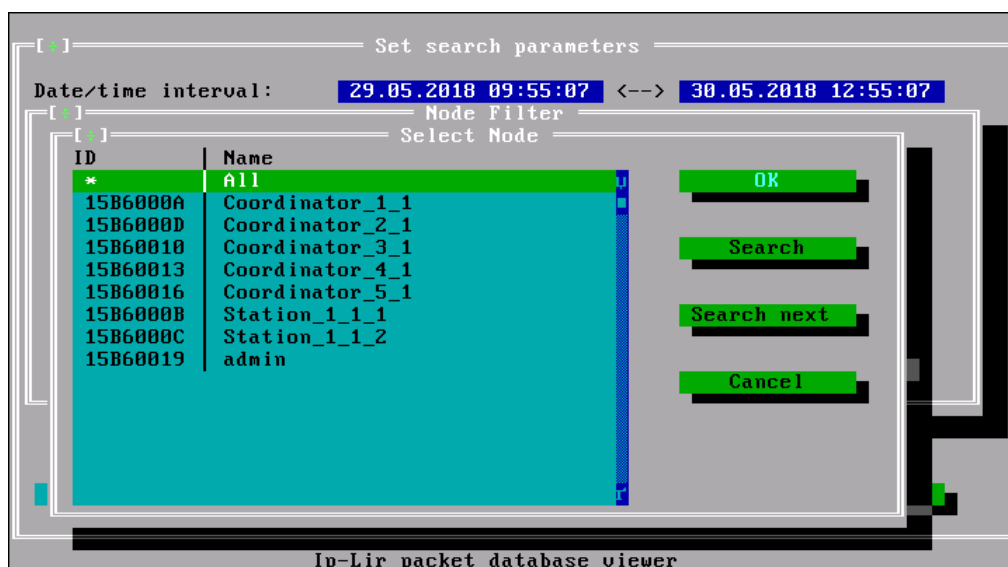


Рисунок 71. Выбор узла отправителя и получателя

3.2 Выберите нужный узел и нажмите **OK**.



Совет. Чтобы быстро найти нужный узел в списке, нажмите **Search**. В открывшемся окне введите часть идентификатора или имени узла и нажмите **OK**. В результате курсор будет переведен на строку первого узла, идентификатор или имя которого удовлетворяют условиям поиска. Для перехода к следующему такому узлу нажмите **Search next**.

4 Чтобы просмотреть журнал регистрации IP-пакетов удаленного узла:



Внимание! Для просмотра журнала регистрации IP-пакетов удаленного узла необходимо, чтобы этот узел был доступен.

4.1 Выберите **External Node** и введите пароль администратора узла ViPNet.

4.2 С помощью кнопки **Select** откройте окно со списком защищенных узлов ViPNet и выберите нужный узел.

5 Нажмите **Find**.

В результате откроется окно **View results**, содержащее таблицу со списком записей об IP-пакетах, которые удовлетворяют выбранным параметрам. Записи упорядочены по времени регистрации. Таблица включает следующие столбцы:

- **Date/time** — дата и время регистрации пакета;
- **Dev** — имя сетевого интерфейса, на котором был обработан пакет;
- **Flags** — направление и флаги пакетов:
 - **<** — исходящий пакет;
 - **>** — входящий пакет;
 - **C** — шифрованный пакет;
 - **B** — широковещательный пакет;
 - **D** — заблокированный пакет;
 - **T** — транзитный пакет;
 - **R** — пакет, который будет обработан правилами NAT открытой сети;
 - **N** — пакет, который был обработан правилами NAT открытой сети;
- **Prot** — протокол, по которому был передан пакет;
- **Source IP** — IP-адрес отправителя пакета;
- **Port** — локальный порт для TCP- и UDP-пакетов или тип для ICMP-пакетов;
- **Destination IP** — IP-адрес получателя пакета;
- **Port** — порт удаленного узла для TCP- и UDP-пакетов или код для ICMP-пакетов.



Примечание. Единицы измерения суммарного размера IP-пакетов: **В** — байты, **KB** — килобайты, **MB** — мегабайты.

Символ «*» в поле **Total In** или **Total Out** пакета означает, что этот пакет не учтен в общем объеме входящего или исходящего трафика. Символ «N/A» — означает отсутствие информации о размере пакетов, соответствующих выбранному типу события.

- **Source Node** — номер сетевого узла ViPNet (отправителя пакетов);
- **Destin Node** — номер сетевого узла ViPNet (получателя пакетов).
- Просмотреть подробную информацию о записи в журнале регистрации IP-пакетов, выбрав его в списке и нажав **Enter**. Подробная информация о записи журнала откроется в отдельном окне **Record details**.

```
[ ] Record details
Events: 30 - Local IP packet blocked by Public Network filter

Interval Begin: 04.03.2019 16:49:31
                End: 04.03.2019 16:49:33

Interface: eth0      Ethernet protocol: 800h
Size: 234 B          Count: 3

Drop: YES            Encrypted NO
Direction: Incoming NAT: NO
Broadcast: NO         Forward: NO

IP protocol: 17 - UDP (User Datagram)
Source IP: 10.0.17.19 Port: 137
Destination IP: 10.0.17.255 Port: 137

Key number: 00000000
Source Node 00000000
Destination Node 00000000

Esc or Enter - return to view results
```

Рисунок 73. Просмотр подробной информации об IP-пакете

Для каждой записи журнала отображается следующая информация:

- **Events** — номер и тип событий, объединенных в одну запись журнала;
- **Interval Begin, End** — начало и конец интервала времени, в течение которого несколько событий были объединены в одну запись журнала;
- **Interface** — имя сетевого интерфейса, на котором были зарегистрированы события;
- **Ethernet protocol** — шестнадцатеричный идентификатор протокола Ethernet;
- **Size** — размер пакетов, агрегированных в выбранной записи;
- **Count** — число пакетов, агрегированных в выбранной записи;
- **Drop** — флаг блокировки пакета;
- **Encrypted** — флаг зашифрованного пакета;
- **Direction** — направление пакета (входящий, исходящий);
- **NAT** — флаг транслированного пакета;
- **Broadcast** — флаг широковещательного пакета;

- **Forward** — флаг транзитного пакета;
- **IP protocol** — тип транспортного протокола (TCP/UDP);
- **Source IP, Port** — IP-адрес и порт отправителя пакетов;
- **Destination IP, Port** — IP-адрес и порт получателя пакетов;
- **Key number** — номер ключа, на котором зашифрован пакет;
- **Source Node** — номер сетевого узла ViPNet — отправителя пакетов;
- **Destination Node** — номер сетевого узла ViPNet — получателя пакетов.
- Для завершения просмотра журнала регистрации IP-пакетов или изменения параметров поиска пакетов в журнале нажмите **Esc**.

Экспорт журнала регистрации IP-пакетов на USB-носитель

Чтобы экспортировать список записей из журнала регистрации IP-пакетов на USB-носитель:



Внимание! Если вы используете удаленное подключение по SSH, то для корректного экспорта установите в настройках SSH-клиента тип терминала VT100+.

- 1 Выполните команду:

```
hostname# iplir view
```
- 2 Задайте параметры для выбора из журнала нужных записей и нажмите **Find**.
- 3 В окне **View results** нажмите клавишу **F2**.
- 4 Введите имя файла, в который вы хотите сохранить журнал, или выберите из списка существующий файл. Затем нажмите **Enter**.
- 5 Завершите просмотр журнала: нажмите **Esc**, затем **Exit**.
- 6 Выполните команду экспорта журнала на USB-носитель:

```
hostname# admin export packetdb usb <имя файла>
```

В качестве имени файла укажите имя, которое вы задали при сохранении журнала (на шаге 4).
- 7 Подключите USB-носитель к ViPNet Coordinator HW и нажмите **Enter**.
- 8 В списке подключенных к ViPNet Coordinator HW USB-носителей и имеющихся на них разделов найдите нужный, введите его номер, нажмите **Enter** и дождитесь завершения экспорта.

В результате журнал регистрации IP-пакетов будет экспортирован в корневой каталог выбранного USB-носителя в виде архива *.tar.gz, содержащего текстовый файл. Данные в этом текстовом файле разделены символом «|».

Чтобы преобразовать журнал в удобный для чтения вид, распакуйте архив, откройте его с помощью программы Microsoft Excel и выполните импорт данных, указав в качестве разделителя символ «|». В результате данные отобразятся в таблице, содержащей следующие столбцы:

- **Time begin** — дата и время начала регистрации пакета;
- **Time end** — дата и время окончания регистрации пакета;
- **Dev** — интерфейс, на котором был обработан пакет;
- **Eth** — шестнадцатеричный идентификатор протокола Ethernet;
- **Flags** — направление и флаги пакета;
- **Prot** — протокол, по которому был передан пакет;
- **Source IP** — IP-адрес отправителя пакета;
- **Port** — локальный порт TCP- и UDP-пакетов или тип для ICMP-пакетов;
- **Destination IP** — IP-адрес получателя пакета;
- **Port** — порт удаленного компьютера для TCP- и UDP-пакетов или код для ICMP-пакетов;
- **Size** — размер пакета (или суммарный размер пакетов, если количество пакетов больше единицы);
- **Cnt** — количество пакетов, относящихся к данному событию;
- **SourceNode** — узел отправителя;
- **DestinNode** — узел получателя;
- **Event** — код и название события, присвоенного пакету.

Экспорт журнала регистрации IP-пакетов по сети в формате CEF

Если в вашей компании используется система управления информационной безопасностью, вы можете настроить экспорт записей журнала регистрации IP-пакетов по сети в формате [CEF](#):

1 Остановите службу `iplircfg`:

```
hostname# iplir stop
```

2 Откройте файл конфигурации `iplir.conf`:

```
hostname# iplir config
```

3 В секции `[misc]` укажите параметры экспорта журнала регистрации IP-пакетов:

- o `cef_enabled` — разрешение экспорта записей журнала по сети:
 - `yes` — запустить экспорт;
 - `no` — остановить экспорт.
- o `cef_ip` — IP-адрес сетевого узла, на который будут отправляться сообщения CEF.



Внимание! В качестве сетевого узла нельзя указывать локальный узел `localhost`, собственные IP-адреса и дополнительные IP-адреса (алиасы) интерфейсов.

- o `cef_port` — порт UDP для передачи сообщений CEF. Значение по умолчанию — 514.

4 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).

5 Откройте файл конфигурации интерфейса `iplir.conf-eth<номер>`, записи журнала регистрации IP-пакетов которого вы хотите экспортировать:

```
hostname# iplir config eth<номер>
```

6 Укажите параметры формирования сообщений CEF в секции `[cef]`:

- o `event` — определяет формирование сообщений CEF при регистрации IP-пакетов, проходящих через интерфейс:
 - `all` — для всех IP-пакетов.
 - `blocked` — для заблокированных IP-пакетов (значение по умолчанию).
- o `exclude` — определяет формирование сообщений CEF, заданных в `event`, за исключением типов событий в журнале регистрации IP-пакетов (см. [Типы событий в журнале регистрации IP-пакетов](#)). Допускается перечисление номеров типов событий в любом порядке через запятую.

Примечание. При необходимости вы можете изменить значение параметра `timediff` секции `[db]` в файле конфигурации интерфейса `iplir.conf-ethX`. Параметр `timediff` задает:



- интервал времени, в течение которого одинаковые события объединяются в одну запись журнала регистрации IP-пакетов;
- периодичность отправки записей журнала регистрации IP-пакетов в формате CEF по сети.

Подробнее см. в документе «Справочник команд и конфигурационных файлов», раздел «Файл `iplir.conf`-<интерфейс или группа интерфейсов>».

7 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).

8 Запустите службу `iplircfg`:

```
hostname# iplir start
```

9 Если сетевой узел, на который будут отправляться сообщения CEF, расположен в открытой сети, создайте разрешающее исходящее правило:

```
hostname# firewall local add src @local dst <IP-адрес> service @Syslog pass
```



Примечание. Для сетевого узла, расположенного в сети ViPNet, разрешающее правило для исходящего трафика протокола `syslog` создается автоматически.

Пример записи журнала регистрации IP-пакетов для экспорта по сети в формате CEF о событии блокировки открытого IP-пакета межсетевым экраном:

```
Sep 19 08:26:10 hw CEF:0|infotecs|hw|4.5|20|unencrypted IP packet  
blocked|10|deviceExternalId=27e8000a start=1539756183000 end=1539756184000 src=10.0.0.1  
dst=2.1.2.2 spt=1232 dpt=80 proto=tcp cnt=5 in=20 deviceInboundInterface=eth0 cnl=800h  
cnLabel=Eth protocol hex cs1=NAT yes cs2=Drop yes cs3=Broadcast no cs4=Forward no  
cs5=Encrypted no cs1Label=Flags deviceDirection=inbound devProd=VA VMWARE  
devVer=4.5.7-1047 dvchost=va1000-5804000a customerURI=http://va1000-5804000a:8080
```

Описание структуры записи см. в приложении [Структура записи журнала регистрации IP-пакетов для экспорта по сети в формате CEF](#).

Работа с журналом MFTP

Транспортный сервер (MFTP) записывает информацию о полностью принятых, отправленных, удаленных и поврежденных транспортных конвертах в специальный журнал.

Для журнала предусмотрена автоматическая ротация. При превышении максимального размера (по умолчанию — 1 Мбайт) наиболее ранние записи перезаписываются новыми. Это позволяет экономить место на ViPNet Coordinator HW.

Для просмотра журнала в командном интерпретаторе транспортный сервер выгружает информацию об обработанных конвертах из журнала в текстовый файл (`mftpenv.log.<дата>`). По истечении периода времени (по умолчанию — 1 день) создается новый файл, в имени которого содержится текущая дата.

Просмотр журнала

Чтобы посмотреть записи журнала MFTP:

```
hostname# mftp view
```

При просмотре выводится следующая информация:

- имя конверта;
- размер конверта;
- имя узла-отправителя;
- имя узла-получателя;
- название события;
- имя прикладной задачи (роли) или тип узла-отправителя;
- описание конверта;
- дата и время события.

Настройка параметров журнала

1 Завершите работу службы `mftpd`:

```
hostname# mftp stop
```

2 Откройте файл `mftp.conf` для редактирования:

```
hostname# mftp config
```

3 В секции `[journal]` при необходимости измените значения:

- `max_size` — максимальный размер журнала конвертов в мегабайтах (по умолчанию — 1).
В случае уменьшения размера журнала наиболее ранние записи удаляются из него.



Внимание! Не рекомендуется задавать размер журнала более чем 100 мегабайт, так как при превышении этого значения могут возникнуть проблемы с просмотром журнала конвертов в веб-интерфейсе.

- o `dump_interval` — период выгрузки информации из журнала в текстовый файл в днях (по умолчанию — 1).

4 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).

5 Запустите службу `mftpd`:

```
hostname# mftp start
```

Работа с системным журналом

В ViPNet Coordinator HW службы `iplircfg`, `mftpd`, `failoverd` записывают информацию о событиях, происходящих в процессе их работы, в системный журнал. Этот журнал позволяет контролировать работоспособность ViPNet Coordinator HW, своевременно выявлять и устранять причины сбоев в его работе.

Информация в системный журнал записывается системными средствами по протоколу `syslog`. Этот журнал может храниться в системном журнале `syslog` на самом сервере ViPNet Coordinator HW (локальное протоколирование) или на удаленном сетевом узле (удаленное протоколирование).

Внимание! Во всех исполнениях ViPNet Coordinator HW по умолчанию настроено локальное протоколирование. При этом по умолчанию:



- в исполнениях с одним дисковым накопителем в журнал записывается информация первого уровня детализации;
- во всех остальных исполнениях в журнал записывается информация третьего уровня детализации.

Подробнее об уровнях детализации см. в документе «Справочник команд и конфигурационных файлов», в описании секций `[debug]` конфигурационных файлов служб `iplircfg`, `mftpd`, `failoverd`.

В случае локального хранения журнала предусмотрена автоматическая ротация файла журнала с целью экономии места на ViPNet Coordinator HW. Ротация выполняется, когда размер файла журнала достигает следующих размеров:

- 50 Мбайт — для всех исполнений с одним дисковым накопителем;
- 1 Гбайт — для остальных исполнений.

В процессе ротации заполненный файл журнала переименовывается, при этом другой файл журнала, который был переименован раньше остальных, удаляется. Это делается для того, чтобы на ViPNet Coordinator HW всегда было не более:

- 3 заполненных файлов журнала и 1 текущего файла, в который ведется запись событий, — для всех исполнений с одним дисковым накопителем;
- 10 заполненных файлов и 1 текущего файла — для остальных исполнений.

При локальном хранении журнала вы можете выполнить с ним следующие действия:

- Просмотреть содержимое журнала (см. [Просмотр системного журнала](#)).
- Экспортировать файлы журнала на другой компьютер или на USB-носитель.
- Экспортировать файлы журнала на другой компьютер или на USB-носитель с их последующим удалением. Для этого выполните команду:

```
hostname# admin export-and-clear logs {usb | tftp}
```



Внимание! Удаление файлов журнала требуется при завершении эксплуатации ViPNet Coordinator HW. В других случаях удалять файлы журнала настоятельно не рекомендуется.

Вы также можете настроить ряд параметров ведения журнала. Подробнее см. раздел [Настройка параметров ведения системного журнала](#).

Перечень протоколируемых событий, связанных с аутентификацией и настройкой оборудования, приведен в приложении.

Просмотр системного журнала

Чтобы просмотреть записи системного журнала, выполните команду:

```
hostname# machine show logs [reversed] [{since <время> | filtered {<служба> | string <строка>}}],
```

указав следующие параметры:

- `reversed` — вывод списка записей в обратном хронологическом порядке.
- `<время>` — вывод списка записей, начиная с указанного момента времени. Момент времени задается в формате `YYYY-MM-DD hh:mm:ss`.
- `<служба>` — вывод записей только для указанной службы в составе ПО ViPNet Coordinator HW.
- `<строка>` — вывод записей журнала по части строки. При поиске по части строки можно использовать символы `A-Z`, `a-z`, `0-9`, а также следующие символы:

`!# $ % & ( ) * + , - . / : ; < = > @ [ ] _ { | } ~`

Также можно использовать пробел, в этом случае необходимо взять часть строки в двойные кавычки (`"`).



Внимание! В одной команде `machine show logs` можно одновременно использовать только одну из лексем: `since` или `filtered`. При использовании лексемы `filtered` можно указать только один из ее параметров: `<служба>` или `<строка>`.

Например, чтобы найти все записи журнала, начиная с 14:50 22 сентября 2024 года, и отобразить их в обратном порядке, выполните команду:

```
hostname# machine show logs reversed since 2024-09-22 14:50:00
```

Настройка параметров ведения системного журнала

Особенности удаленного протоколирования

- Если для удаленного протоколирования будет использоваться открытый узел, на ViPNet Coordinator HW вручную задайте фильтр открытой сети, разрешающий исходящий трафик по протоколу UDP на порт 514 этого открытого узла.
- При переключении интерфейса платформы, через который передается информация о работе ViPNet Coordinator HW, в режим получения настроек от DHCP-сервера, возможна временная потеря связи с удаленным сервером, и, как следствие потеря информации.

Чтобы избежать этого, перед настройкой интерфейса включите локальное протоколирование:

```
hostname# machine set loghost local
```

После применения настроек снова включите удаленное протоколирование.

- Не рекомендуем использовать удаленное протоколирование на ViPNet Coordinator HW в режиме кластера горячего резервирования, так как на удаленный сетевой узел не будут передаваться журналы с пассивного узла (то есть часть информации о работе ViPNet Coordinator HW будет утеряна).

Если все же необходимо настроить удаленное протоколирование при работе в режиме кластера горячего резервирования, задайте параметры протоколирования на обоих узлах кластера, так как эти параметры не передаются с активного узла на пассивный через интерфейс синхронизации.

Настройка параметров ведения системного журнала

- Чтобы выбрать место хранения системного журнала, выполните:

```
hostname# machine set loghost {<IP-адрес> | <доменное имя> | local},
```

указав IP-адрес или доменное имя удаленного сетевого узла, на котором будет храниться журнал, или параметр `local` для локального протоколирования.

- Чтобы отключить ведение системного журнала, выполните:

```
hostname# machine set loghost null
```



Внимание! Отключать ведение журнала не рекомендуется.

- Чтобы изменить уровень важности информации, выводимой в журнал, выполните настройку параметра `debuglevel` в секции `[debug]` конфигурационных файлов `iplir.conf`, `failover.ini`, `mftp.conf`. Описание параметров секции `[debug]` в данных файлах см. в документе «Справочник команд и конфигурационных файлов».



Совет. В большинстве случаев изменять уровень важности, выбранный по умолчанию, не рекомендуется. Особенно не рекомендуется повышать уровень важности в исполнениях с одним дисковым накопителем (ViPNet Coordinator HW50). При повышении уровня необходимо учитывать, что размер файла журнала будет быстро расти, вследствие чего, будет чаще производиться его ротация при достижении максимального размера (см. [Работа с системным журналом](#)).

- Чтобы сообщения о работе службы выводились на консоль командного интерпретатора, выполните команду:

```
hostname# debug on <источник> <уровень важности>
```

Параметры `источник` и `уровень важности` должны быть заданы в секции `[debug]` конфигурационных файлов `iplir.conf`, `failover.ini`, `mftp.conf`. Описание параметров секции `[debug]` в данных файлах см. в документе «Справочник команд и конфигурационных файлов».

- Чтобы выключить вывод сообщений на консоль командного интерпретатора, выполните команду `debug off`. С помощью этой команды вы можете выключить как вывод всех сообщений, так и вывод только тех сообщений, которые соответствуют конкретным значениям параметров `facility` и `level`.

Например, вы можете настроить протоколирование так, чтобы следить за работой разных служб на разных терминалах. Для этого выполните следующее:

- В конфигурационном файле одной из служб в секции `[debug]` задайте нужные параметры, например:

```
[debug]
debuglogfile = syslog:daemon.debug
debuglevel = 3
```

И на одном из терминалов выполните команду:

```
hostname# debug on daemon debug
```

- В конфигурационном файле другой службы в секции `[debug]` задайте параметры протоколирования, указав другое значение `источника`, например:

```
[debug]
debuglogfile = syslog:local0.debug
debuglevel = 3
```

И на другом терминале выполните команду:

```
hostname# debug on local0 debug
```

В результате на первый терминал будут поступать сообщения только от первой службы, на второй терминал — только от второй службы. Чтобы на первом терминале можно было следить за работой обеих служб, на нем в дополнение к первой команде выполните команду:

```
hostname# debug on local0 debug
```



Примечание. Выводимые на консоль командного интерпретатора сообщения о работе служб могут смешиваться с сообщениями самого интерпретатора и командами, набираемыми на консоли. Для просмотра набранной к данному моменту части команды и продолжения ее ввода используйте комбинацию клавиш **Ctrl+L**.

Экспорт диагностической информации на компьютер

К диагностической информации относятся:

- Журналы — системный журнал, журнал MFTP.
- Конфигурация — журнал IP-пакетов по каждому интерфейсу и информация о конфигурации ViPNet Coordinator HW, например, таблицы маршрутизации, правила фильтрации, правила трансляции и информация о службах и интерфейсах.

Вы можете отдельно скачать журналы и конфигурацию.



Внимание! Во время экспорта журналов на другой компьютер завершается работа всех служб и выгружаются все драйверы ViPNet, то есть ViPNet Coordinator HW будет незащищенным.

При экспорте диагностической информации с ViPNet Coordinator HW на другой компьютер используется стандартная служба TFTP. В Windows 10 эта служба по умолчанию выключена и ее необходимо включить вручную:

- 1 Выберите **Пуск (Start) > Панель управления (Control Panel) > Программы > Программы и компоненты (Programs and Features)**.
- 2 Зайдите в меню **Включение или отключение компонентов Windows (Turn Windows features on or off)** и установите флажки рядом с названием служб **Клиент TFTP (TFTP Client)** и **Простые службы TCP/IP (Simple TCP/IP services)**.
- 3 Перезагрузите компьютер.

На время выполнения экспорта автоматически изменяются настройки сетевых интерфейсов ViPNet Coordinator HW: на интерфейсе `eth1` устанавливается IP-адрес `169.254.241.1`, все остальные интерфейсы выключаются. После успешного завершения экспорта настройки интерфейсов ViPNet Coordinator HW автоматически восстанавливаются.



Внимание! Чтобы не потерять удаленный доступ к ViPNet Coordinator HW запрещается выполнять экспорт диагностической информации на другой компьютер в удаленной SSH-сессии.

Чтобы экспортировать диагностическую информацию на компьютер:

- 1 Подключите ноутбук к порту Ethernet1 ViPNet Coordinator HW с помощью кросс-кабеля Ethernet.
- 2 Установите вручную на сетевом интерфейсе ноутбука IP-адрес 169.254.241.5.
- 3 Подключите к ViPNet Coordinator HW обычную консоль (монитор и клавиатуру) или COM-консоль.
- 4 Завершите работу служб `iplircfg`, `failoverd` и `mftpd`:

```
hostname# vpn stop
```

- 5 Вы можете экспортировать журналы и конфигурацию:

- Чтобы экспортировать журналы:

```
hostname# admin export logs tftp nodiag
```

- Чтобы экспортировать журналы и конфигурацию:

```
hostname# admin export logs tftp
```

- Чтобы экспортировать конфигурацию:

```
hostname# admin export diagnostics tftp
```

Примечание. При выполнении команд файлы журналов архивируются в файл экспорта. Файлы конфигурации помещаются в отдельный архив.

Имя файла экспорта формируется по шаблону:



- Архив системного журнала: `logs-<название ViPNet Coordinator HW>-%Y-%m-%d-%H-%M-%S.tar.gz`.
- Архив с файлами конфигурации: `diagnostics-<название ViPNet Coordinator HW>-%Y-%m-%d-%H-%M-%S.tar.gz`.

По умолчанию `<название ViPNet Coordinator HW>` включает `<платформа ViPNet Coordinator HW>-<идентификатор узла>`.

Чтобы прервать экспорт во время архивирования, нажмите **Ctrl+C**.

После завершения архивирования появится сообщение, содержащее предложение скачать полученный файл.

- 6 Перенесите файл экспорта на ноутбук по TFTP. Для этого выполните на ноутбуке команду:

```
tftp -i 169.254.241.1 get <файл экспорта>
```

- 7 Нажмите **Enter**.

- 8 Запустите службы `iplircfg`, `failoverd` и `mftpd`:

```
hostname# vpn start
```

В результате архив с диагностической информацией будет скопирован на компьютер.

Экспорт диагностической информации на USB-носитель

1 Вы можете экспортировать журналы и конфигурацию:

- Чтобы экспортировать журналы:

```
hostname# admin export logs usb nodiag
```

- Чтобы экспортировать журналы и конфигурацию:

```
hostname# admin export logs usb
```

- Чтобы экспортировать конфигурацию:

```
hostname# admin export diagnostics usb
```

Примечание. При выполнении команд файлы журналов архивируются в файл экспорта. Файлы конфигурации помещаются в отдельный архив.

Имя файла экспорта формируется по шаблону:



- Архив с файлами журналов: logs-<название ViPNet Coordinator HW>-%Y-%m-%d-%H-%M-%S.tar.gz.
- Архив с файлами конфигурации: diagnostics-<название ViPNet Coordinator HW>-%Y-%m-%d-%H-%M-%S.tar.gz.

По умолчанию <название ViPNet Coordinator HW> включает <платформа ViPNet Coordinator HW>-<идентификатор узла>.

Чтобы прервать экспорт во время архивирования, нажмите **Ctrl+C**.

2 Подключите USB-носитель к ViPNet Coordinator HW и нажмите **Enter**.

3 По запросу командного интерпретатора введите номер USB-носителя и нажмите **Enter**.

Архив с системным журналом будет скопирован на USB-носитель.

Работа с журналом DNS-запросов

ViPNet Coordinator HW может выполнять функцию DNS-сервера в сети. Чтобы вести дополнительный контроль за тем, к каким ресурсам обращаются узлы сети и выявлять проблемы в настройках сети, в ViPNet Coordinator HW записывается информация о DNS-запросах:

- дата и время запроса;
- IP-адрес и порт, с которого поступил запрос;
- запрошенное доменное имя;
- тип запрошенной DNS-записи.

Настройте журнал DNS-запросов:

1 Проверьте настройки журнала DNS-запросов:

```
hostname# inet show dns
```

2 Включите ведение журнала, если оно выключено:

```
hostname# inet dns querylog on
```

3 Настройте размер журнала DNS-запросов:

При локальном хранении журнала предусмотрена автоматическая ротация. Это позволяет экономить место на ViPNet Coordinator HW. Ротация выполняется, когда журнал DNS-запросов достигает максимального размера. По умолчанию:

- 10 Мбайт — для всех исполнений с одним дисковым накопителем (HW10 и HW50);
- 50 Мбайт — для остальных исполнений.

Вы можете изменить размер журнала DNS-запросов:

```
hostname# inet dns querylog size <размер>
```

Допустимые значения параметра <размер>:

- 1 — 50 Мбайт: для всех исполнений с одним дисковым накопителем (HW10 и HW50);
- 1 — 200 Мбайт: для остальных исполнений.

В процессе ротации заполненный файл журнала переименовывается, при этом другой файл, который был переименован раньше, удаляется. Таким образом на ViPNet Coordinator HW всегда находится 1 заполненный и 1 текущий файл, в который ведется запись DNS-запросов.



Внимание! Настройки журнала применяются сразу после их изменения. Поэтому, если вы уменьшили размер журнала, а его текущий размер достиг или превышает новое значение, он автоматически будет ротирован. Предыдущий файл будет удален.

При локальном хранении журнала вы можете:

- Просмотреть содержимое журнала:

```
hostname# inet dns querylog show
```

Также вы можете посмотреть записи журнала, содержащие определенный текст:

```
hostname# inet dns querylog show filtered <строка>
```

- Экспортировать файлы журнала и конфигурационную информацию на другой компьютер или на USB-носитель. Файлы журнала экспортируются вместе с системным журналом в архив `logs.tar.gz`, конфигурационная информация сохраняется в архив `diagnostics.tar.gz`:

- Чтобы экспортировать файлы журнала:

```
hostname# admin export logs {usb|tftp} nodiag
```

- Чтобы экспортировать файлы журнала и конфигурационную информацию:

```
hostname# admin export logs {usb|tftp}
```

- Чтобы экспортировать конфигурационную информацию:

```
hostname# admin export diagnostics {usb|tftp}
```

Вы можете также можете экспортировать файлы журнала и конфигурационную информацию через веб-интерфейс вместе с системным журналом. Подробнее см. документ «Настройка с помощью веб-интерфейса» раздел «Просмотр системного журнала».

- Экспортировать файлы журнала на другой компьютер или на USB-носитель с их последующим удалением:

```
hostname# admin export-and-clear logs {usb | tftp}
```

- Удалить журнал:

```
hostname# machine logs clear dns
```



Сетевые фильтры по умолчанию

Сетевые фильтры, в том числе фильтры системы защиты от сбоев, создаваемые в ViPNet Coordinator HW по умолчанию, перечислены в таблицах ниже.

Таблица 12. Фильтры защищенной сети (vpn)

Название	Источник	Назначение	Протокол	Действие
В режиме кластера (нередактируемый)				
ViPNet Service	@any	@local	udp: from 2060 to 2060	pass
ViPNet Service	@local	@any	udp: from 2060 to 2060	pass
Сервисный фильтр (нередактируемый)				
Block not original udp port	@local	@any	udp: from 0-2045 to 2046 udp: from 2047-65535 to 2046	drop
Настраиваемые фильтры				
ICMP redirect	@any	@any	ICMP:5	drop
Allow ViPNet base services	@any	@any	udp: from 2048 to 2048, from 2050 to 2050	pass
Allow ViPNet base services	@any	@any	udp: to 2046	pass
Allow DHCP Service	@any	@any	udp: from 67 to 68	pass
Allow DHCP Service	@any	@any	udp: from 68 to 67	pass

Название	Источник	Назначение	Протокол	Действие
Allow DHCP-Relay service	@any	@any	udp: from 67 to 67	pass
Allow ViPNet DBViewer	@any	@any	tcp: to 2047	pass
Allow ViPNet StateWatcher (MFTP info)	@any	@any	tcp: to 5100, 10092	pass
Allow ViPNet MFTP	@any	@any	tcp: to 5000-5003	pass
Allow ViPNet WebGUI	@any	@local	tcp: to 8080	pass
Allow ICMP Ping	@any	@any	icmp: 8	pass
Allow SSH	@any	@any	tcp: to 22	pass
Allow DNS	@any	@any	udp: to 53, tcp: to 53	pass
Allow NTP	@any	@any	udp: to 123	pass
Allow UPS service (нет в исполнении ViPNet Coordinator VA и HW10)	@any	@any	tcp: to 3493	pass
Allow syslog outgoing	@local	@any	udp: to 514	pass
Allow SNMP	@any	@local	udp: to 161	pass
Allow SNMP traps	@local	@any	udp: to 162	pass
Блокирующий фильтр (нередактируемый)				
Примечание. Данный фильтр имеет самый низкий приоритет и действует для трафика, не попадающего под действие всех остальных фильтров.				
Default VPN rule	@any	@any	@any	drop

Таблица 13. Локальные фильтры открытой сети (*local*)

Название	Источник	Назначение	Протокол	Действие
В режиме кластера (нередактируемые)				
Failover test IP	@local	<список значений параметров testip из failover.ini>	icmp	pass
Failover Channel	@local	interface <имя интерфейса синхронизации>	@any	pass
Failover Channel	<IP-адрес интерфейса синхронизации второго узла кластера>	interface <имя интерфейса синхронизации>	@any	pass

Название	Источник	Назначение	Протокол	Действие
В пассивном режиме кластера (нередатируемые)				
Failover Channel	@local	iface <значение параметра device из failover.ini>	@any	pass
Failover Channel	<значение параметра activeip второго узла из failover.ini>	iface <значение параметра device из failover.ini>	@any	pass
ARP requests to active	@local	<список значений параметров activeip из failover.ini>	udp: to <значение connect_port из failover.ini>	pass
Block all local traffic	@any	@any	@any	drop
Сервисные фильтры (нередатируемые)				
ViPNet Service Common In	@any	@local	tcp/udp: to 2046, 2047, 10096, 5100, 10092	drop
ViPNet Service Common Out	@local	@any	tcp/udp: from 2046	drop
Настраиваемые фильтры				
ICMP redirect	@any	@any	ICMP:5	drop
Allow DHCP Service	@any	@any	udp: from 67 to 68	pass
Allow DHCP Service	@any	@any	udp: from 68 to 67	pass
Allow DHCP-Relay service	@any	@any	udp: from 67 to 67	pass
Block ICMP timestamp response	@local	@any	icmp:14	drop
Allow ICMP type 3 code 4	@local	@any	ICMP type 3 code 4	pass
Allow ICMP Ping	@local	@any	icmp: 8	pass
Allow DNS	@local	@any	tcp/udp: to 53	pass
Allow NTP	@local	@any	udp: to 123	pass
Блокирующий фильтр (нередатируемый)				
Примечание. Данный фильтр имеет самый низкий приоритет и действует для трафика, не попадающего под действие всех остальных фильтров.				

Название	Источник	Назначение	Протокол	Действие
Default local rule	@any	@any	@any	drop

Таблица 14. Транзитные фильтры открытой сети (*forward*)

Название	Источник	Назначение	Протокол	Действие
Блокирующий фильтр (нередатируемый)				
Примечание. Данный фильтр имеет самый низкий приоритет и действует для трафика, не попадающего под действие всех остальных фильтров				
Default transit rule.	@any	@any	@any	drop

Таблица 15. Фильтры туннелируемых узлов (*tunnel*)

Название	Источник	Назначение	Протокол	Действие
Общие фильтры				
To all tunnel nodes	@any	@tunneledip	@any	pass
From all tunnel nodes	@tunneledip	@any	@any	pass
Блокирующий фильтр (нередатируемый)				
Примечание. Данный фильтр имеет самый низкий приоритет и действует для трафика, не попадающего под действие всех остальных фильтров				
Default tunnel rule	@any	@any	@any	drop

В

Пользовательские группы протоколов по умолчанию

Таблица 16. Пользовательские группы протоколов по умолчанию

Имя группы протоколов	Состав группы протоколов
DHCP	UDP:from 67-68 to 67-68
CITRIX	TCP:to 1494
DNS	UDP:to 53
FTP	TCP:to 21
GRE	IP:47
H323	TCP:to 1720
HTTP	TCP:to 80, TCP:to 8080
HTTP-Proxy	TCP:to 3128
HTTPS	TCP:to 443
IGMP	IP:2
IKE	UDP:to 500
IMAP	TCP:to 143
IPSecESP	IP:50
Kerberos	TCP:to 88, TCP:to 749, UDP:to 88, UDP:to 749
L2TP	UDP:to 1701
LDAP	TCP:to 389, UDP:to 389

Имя группы протоколов	Состав группы протоколов
LotusNotes	TCP:to 1352
MS-SQL	TCP:to 1433-1434, UDP:to 1433-1434
MySQL	TCP:to 3306
NetBIOS-DGM	UDP:from 138 to 138
NetBIOS-NC	UDP:from 137 to 137
NetMeeting	TCP:to 1503
NTP	UDP:to 123
PING	ICMP:8
POP3	TCP:to 110
Postgres	TCP:to 5432
PPTP	TCP:to 1723
RADIUS	UDP:to 1812-1813
RDP	TCP:to 3389
RTSP	TCP:to 554
SCCP	TCP:to 2000
SIP	TCP:to 5060, UDP:to 5060
SMTP	TCP:to 25
SNMP	UDP:to 161
SNMP-Traps	UDP:to 162
SSH	TCP:to 22
Syslog	UDP:to 514
Telnet	TCP:to 23
TFTP	UDP:to 69
UPnP	TCP:to 1900, TCP:to 2869, UDP:to 1900, UDP:to 2869
MFTP	TCP:to 5000-5003
SMCMonitoring	TCP:to 2047, TCP:to 5100, TCP:to 10092
VPNBase	UDP:to 2046, UDP:from 2048 to 2048, UDP:from 2050 to 2050
Cluster	UDP:from 2060 to 2060
ClusterMonitoring	UDP:from 2060 to 2065, UDP:from 2065 to 2060
VNC	TCP:to 5900
OSPF	IP:89

С

Типы событий в журнале регистрации IP-пакетов

Типы событий, регистрируемых в журнале IP-пакетов ViPNet Coordinator HW, можно поделить на следующие группы и подгруппы:

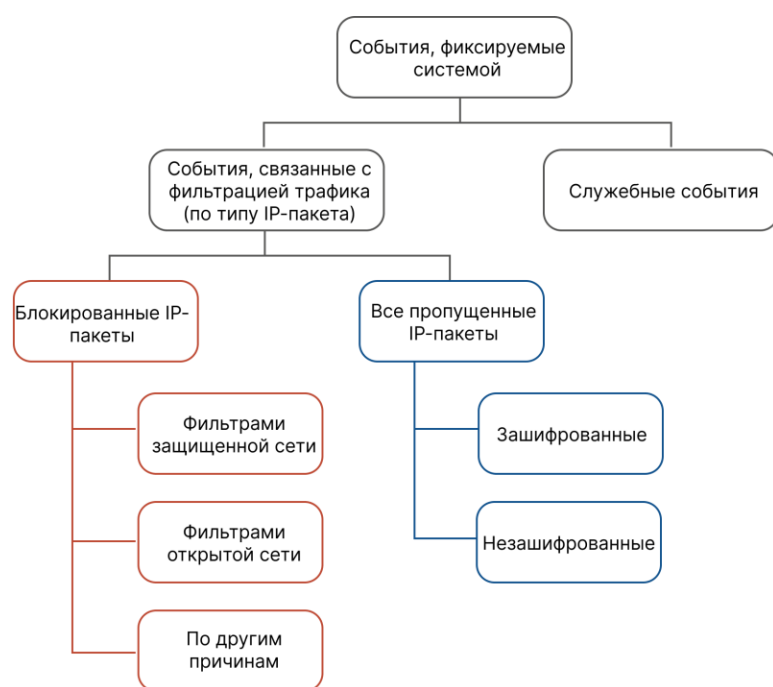


Рисунок 74. Классификация событий в журнале IP-пакетов

Описание типов событий каждой подгруппы приведено в таблицах.

Таблица 17. События подгруппы **Все IP-пакеты/Блокированные IP-пакеты/IP-пакеты, блокированные фильтрами защищенной сети**

Номер события	Описание события
1	Не найден ключ для передачи пакета сетевому узлу с идентификатором, указанным в пакете
2	Неверное значение имитозащитной вставки пакета. Защищенные данные или открытая информация в пакете были изменены при передаче
3	Входящий зашифрованный или предназначенный для шифрования исходящий открытый пакет заблокирован фильтром защищенной сети
4	Слишком большой тайм-аут пакета, то есть время его отправки отличается от времени его получения на величину, большую указанной в соответствующей настройке
7	Метод шифрования, код которого указан во входящем пакете, не поддерживается
8	IP-пакет поврежден
9	Неизвестный идентификатор сетевого узла
13	Превышено максимальное время пребывания пакета в сети
14	Неверный адрес получателя пакета
15	Превышено допустимое количество одновременно обрабатываемых фрагментированных пакетов
16	Исчерпана лицензия на количество туннелируемых узлов
17	Неверный IP-адрес получателя
18	Неизвестный IP-адрес получателя или узла, указанного в параметре <code>proxyid</code>
19	Подмена узла отправителя
26	Узел с данного координатора недоступен
70	Транзитный IP-пакет заблокирован

Таблица 18. События подгруппы **Все IP-пакеты/Блокированные IP-пакеты/IP-пакеты, блокированные фильтрами открытой сети**

Номер события	Описание события
22	От защищенного узла получен открытый пакет
23	От защищенного узла получен открытый широкоэвещательный пакет
24	На порт, используемый одной из служб ViPNet, получен открытый пакет
30	Локальный пакет заблокирован фильтром открытой сети
31	Транзитный пакет заблокирован фильтром открытой сети

Номер события	Описание события
32	Широковещательный пакет заблокирован фильтром открытой сети
33	Пакет заблокирован фильтром антиспуфинга
37	Пакет заблокирован фильтром туннелируемых ресурсов
39	Пакет заблокирован фильтром по умолчанию при загрузке компьютера

Таблица 19. События подгруппы **Все IP-пакеты/Блокированные IP-пакеты/IP-пакеты, блокированные по другим причинам**

Номер события	Описание события
80	Размер заголовка IP-пакета меньше допустимого
81	Недопустимая версия протокола IP (поддерживается только IPv4)
82	Длина заголовка IP-пакета меньше допустимой
83	Длина IP-пакета меньше указанной в IP-заголовке
84	Значение контрольной суммы в заголовке IP-пакета отличается от ее значения в IP-пакете
85	Размер TCP-заголовка меньше допустимого
86	Размер UDP-заголовка меньше допустимого
87	Обработаны не все фрагменты IP-пакета
88	Широковещательный адрес отправителя в IP-пакете
89	Перекрытие фрагментов IP-пакета. Наиболее устаревший из перекрываемых фрагментов IP-пакета отброшен
90	IP-пакет не был обработан, так как для его обработки недостаточно вычислительных ресурсов
91	IP-пакет получен во время инициализации драйвера ViPNet
92	Размер IP-пакета превышает 48 Кбайт
93	По истечении допустимого времени получены не все фрагменты IP-пакета
95	Получены два IP-пакета от узлов с разными IP-адресами и одинаковыми идентификаторами
97	Заблокирован пакет с установленным флагом DF, передача которого без фрагментации невозможна
99	Заблокирован фрагментированный IP-пакет
101	Транзитный IP-пакет не может быть маршрутизирован

Номер события	Описание события
103	Количество установленных соединений превышает допустимое значение, заданное в соответствующих настройках
104	Соединение заблокировано, так как параметры исходящих пакетов (socketpair) для этого соединения совпадают с такими параметрами для ранее установленного соединения
105	Не удалось выделить динамический порт для правила трансляции адресов (в пуле нет свободных портов)
111	Не найден ключ обмена
112	Неверное значение имитозащитной вставки в незашифрованном пакете версии 4.2
113	Неизвестный идентификатор сетевого узла отправителя
116	Не найден сетевой адаптер
117	Не удалось определить MAC-адрес получателя пакета по его IP-адресу
118	Ошибка при шифровании исходящего IP-пакета для защищенного узла
119	Получен зашифрованный IP-пакет неизвестного формата, который не может быть расшифрован
120	Ошибка при отправке IP-пакета защищенному узлу из-за проблем с доступом к этому узлу
121	Ошибка в работе кластера горячего резервирования
122	Получен IP-пакет неизвестного протокола канального уровня
129	Получен широковещательный пакет с IP-адресом назначения, не принадлежащим подсети интерфейса
130	Отсутствуют свободные динамические порты, необходимые для создания соединения
131	Ошибка обработки прикладных протоколов: не удалось построить маршрут
132	Ошибка обработки прикладных протоколов: отсутствуют свободные динамические порты UDP
133	Ошибка обработки прикладных протоколов: порты источника и назначения IP-пакета принадлежат разным прикладным сервисам
134	Ошибка обработки прикладных протоколов: ошибка в таблице соответствия между прикладными сервисами и сетевыми протоколами, портами
135	Ошибка обработки модуля ALG
136	Недостаточно памяти для выполнения трансляции адреса (NAT)
139	Отсутствуют лицензии для туннелирования на узле назначения

Таблица 20. События группы **Все IP-пакеты/Все пропущенные IP-пакеты/Пропущенные зашифрованные IP-пакеты**

Номер события	Описание события
40	Пропущен зашифрованный IP-пакет
41	Пропущен зашифрованный широковежательный IP-пакет
44	Маршрутизация зашифрованного транзитного IP-пакета с подменой адреса получателя
45	Пакет пропущен на туннелирующем координаторе, так как он поступил от туннелируемого узла или предназначен для такого узла

Таблица 21. События группы **Все IP-пакеты/Все пропущенные IP-пакеты/Пропущенные незашифрованные IP-пакеты**

Номер события	Описание события
60	Пропущен незашифрованный IP-пакет
61	Пропущен незашифрованный широковежательный IP-пакет
62	Пропущен незашифрованный транзитный IP-пакет
63	IP-пакет пропущен фильтром для туннелируемых ресурсов
64	IP-пакет пропущен при запуске операционной системы
65	Пропущен зашифрованный IP-пакет неизвестной ViPNet сети

Таблица 22. События группы **Все IP-пакеты/Служебные события**

Номер события	Описание события
42	Изменился IP-адрес сетевого узла
46	Изменились параметры доступа к сетевому узлу
47	Истек интервал отправки IP-пакетов, передаваемых сетевым узлом своему серверу соединений. Событие может возникать только для тех узлов, которые работают через межсетевой экран с динамической трансляцией адресов
48	Узел доступен по широковежательному адресу
49	Изменился IP-адрес собственного сетевого узла
110	Новый IP-адрес сетевого узла зарегистрирован на DNS-сервере

D

Структура записи журнала регистрации IP-пакетов для экспорта по сети в формате CEF

Запись журнала регистрации IP-пакетов для экспорта по сети в формате CEF состоит из полей заголовка CEF и полей записи журнала регистрации IP пакетов ViPNet Coordinator HW. Ниже приведено описание структуры записи.

```
<Prival> Dec 2 13:55:47 hw CEF:Version|Device Vendor|Device Product|Device  
Version|Signature ID|Name|Severity|Extension
```

где:

- `Prival` — идентификатор важности события. Всегда равен 38;
- `Version` — номер версии CEF-нотации (значение равно нулю);
- `Device Vendor`, `Device Product`, `Device Version` — идентификатор устройства (производитель, продукт, версия);
- `Signature ID` — номер события (см. [Типы событий в журнале регистрации IP-пакетов](#)) в журнале регистрации IP-пакетов;
- `Name` — описание события (см. [Типы событий в журнале регистрации IP-пакетов](#)) в журнале регистрации IP-пакетов;
- `Severity` — важность события;

- 5 — если IP-пакет пропущен;
- 10 — если IP-пакет блокирован;
- `Extension` — набор пар ключ-значение из словаря CEF, разделенных пробелом.

Таблица 23. Описание ключей словаря CEF поля `Extension`

Ключ словаря CEF	Описание ключа	Название поля в журнале
<code>deviceExternalId</code>	Идентификатор сенсора	
<code>dvc</code>	Адрес интерфейса, на котором зарегистрировано событие	
<code>dvchost</code>	Имя сетевого узла	Hostname
<code>customerURI</code>	Ссылка на начальную страницу управления ViPNet Coordinator HW	
<code>start</code>	Начало интервала времени UTC, в течение которого несколько событий были объединены в одну запись журнала IP-пакетов	Interval Begin
<code>end</code>	Конец интервала времени UTC, в течение которого несколько событий были объединены в одну запись журнала IP-пакетов	Interval End
<code>src</code>	IP-адрес отправителя пакетов	Source IP
<code>spt</code>	Порт отправителя пакетов	Source port
<code>dst</code>	IP-адрес получателя пакетов	Destination IP
<code>dpt</code>	Порт получателя пакетов	Destination port
<code>proto</code>	Тип транспортного протокола (TCP или UDP)	IP protocol
<code>cnt</code>	Число пакетов, соответствующих выбранному типу события	Count
<code>in</code>	Размер входящих пакетов, соответствующих выбранному типу события	size in
<code>out</code>	Размер исходящих пакетов, соответствующих выбранному типу события	size out
<code>deviceInboundInterface</code> <code>deviceOutboundInterface</code>	Имя сетевого интерфейса, на котором зарегистрировано событие	Interface
<code>cn1</code>	Шестнадцатеричный идентификатор протокола Ethernet	Eth. protocol
<code>cn1Label=Eth protocol hex</code>	Имя параметра <code>cn1</code>	

Ключ словаря CEF	Описание ключа	Название поля в журнале
deviceDirection	Направление пакета (входящий, исходящий)	Direction
• cs1=NAT {yes no} • cs2=Drop {yes no} • cs3=Broadcast {yes no} • cs4=Forward {yes no} • cs5=Encrypted {yes no}	• NAT — флаг транслированного пакета; • Drop — флаг блокировки пакета; • Broadcast — флаг широковещательного пакета; • Forward — флаг транзитного пакета; • Encrypted — флаг зашифрованного пакета	
• cs1Label= FlagNAT • cs2Label=FlagDrop • cs3Label=FlagBroadcast • cs4Label=FlagForward • cs5Label=FlagEncrypted	Имена параметров, передаваемых в полях cs1-cs5	
cs1Label=Flags		
shost	Номер и имя сетевого узла ViPNet (отправителя)	Source node
dhost	Номер и имя сетевого узла ViPNet (получателя)	Destination node
devProd	Идентификатор устройства (продукт)	Device Product
devVer	Идентификатор устройства (версия)	Device Version



Записи системного журнала, связанные с аутентификацией и настройкой оборудования

Ниже приведены протоколируемые события ViPNet Coordinator HW, связанные с аутентификацией и настройкой оборудования, и соответствующие им записи в системном журнале.

Таблица 24. Протоколируемые события операционной системы Linux

Событие	Пример записей в журнале
Запуск операционной системы	<pre>vmunix: [0.000000] Linux version 4.4.163 (vipnet@build-linux) (gcc version 6.3.0 20170516 (Debian 6.3.0-18+deb9u1)) #1 SMP Tue Mar 5 15:14:33 UTC 2019</pre>
Завершение работы операционной системы	<pre>rvpn_shell[2690]: [03-11 12:28:09]< 2690>(24df0740) syslog <HALT> Command 'machine halt'. rvpn_shell[2690]: [03-11 12:28:09]< 2690>(24df0740) syslog The session has been forced to close. rvpn_shell[2690]: [03-11 12:28:09]< 2690>(24df0740) syslog Your administrator session has been forced to close, as another administrator session of the command line interface has been started on the terminal tty1 rvpn_shell[2690]: [03-11 12:28:09]< 2690>(24df0740) syslog with the connection address local console</pre>

Событие	Пример записей в журнале
	<pre> rvpn_shell[2690]: [03-11 12:28:09]< 2690>(24df0740) ClUnilogFree rvpn_shell[2690]: === ended log session at Mon Mar 11 12:28:09 2019 dhcp-relay[3226]: DHCP relays are already STOPPED dhcp-server[3240]: DHCP server is already STOPPED sshd[1701]: Received signal 15; terminating. watchquagga[1289]: Terminating on signal zdhcpd[1284]: Terminating on signal zebra[1280]: client 10 disconnected. 0 dhcp routes removed from the rib zebra[1280]: Terminating on signal zebra[1280]: IRDP: Received shutdown notification. enterpin: Using file based authentication by ViPNet password acpid: exiting syslogd: exiting on signal 15 </pre>
Сброс ViPNet Coordinator HW к заводским настройкам	<pre> factory_reset.sh[554]: Deletion of keys and host links has been started factory_reset.sh[554]: Keys and host links will be deleted in 29 seconds. To cancel, press Ctrl+C ... factory_reset.sh[554]: Keys and host links will be deleted in 29 seconds. To cancel, press Ctrl+C factory_reset.sh[554]: Device has been reset to factory defaults. Press Enter to reboot </pre>

Таблица 25. События, регистрируемые при работе пользователя или администратора ViPNet Coordinator HW

Событие	Пример записей в журнале
Завершение работы операционной системы по команде	<pre> rvpn_shell[1380]: <HALT> Command 'machine halt'. </pre>
Перезагрузка операционной системы по команде	<pre> rvpn_shell[3926]: [03-11 12:18:35]< 3926>(85d0e740) syslog <REBOOT> Command: machine reboot </pre>
Выход из командного интерпретатора, работающего в режиме пользователя	<pre> login[2675]: pam_unix(login:session): session closed for user user rvpn_shell[3214]: [03-11 17:00:58]< 3214>(48295740) syslog <EXIT> Command 'exit' </pre>
Успешный удаленный запуск командного интерпретатора по протоколу SSH	<pre> login[2705]: pam_unix(login:session): session opened for user user by LOGIN(uid=0) Accepted password for user from 192.168.120.2 port 49449 ssh2 </pre>

Событие	Пример записей в журнале
Не удалось запустить командный интерпретатор при удаленном подключении по протоколу SSH	sshd[20448]: Failed password for user from 13.0.1.94 port 55650 ssh2
Завершение удаленного подключения по протоколу SSH	sshd[20448]: pam_unix(sshd:session): session closed for user user
Успешный переход в режим администратора	rvpn_shell[1334]: <ENABLE> Command: enable - Administrator session started.
Не удалось перейти в режим администратора	rvpn_shell[2748]: [03-11 12:33:32]< 2748>(2e5b1740) syslog <ENABLE> Command 'enable'. Failed to log on as an administrator.
Переход из режима администратора в режим пользователя	rvpn_shell[2748]: [03-11 12:33:23]< 2748>(2e5b1740) syslog <EXIT> Command 'exit'
Вход в системную командную оболочку	rvpn_shell[2748]: [03-11 18:03:53]< 2748>(2e5b1740) syslog <I_SHELL> You have exited to the Linux system shell. rvpn_shell[2748]: [03-11 18:03:53]< 2748>(2e5b1740) syslog The session has been forced to close. rvpn_shell[2748]: [03-11 18:03:53]< 2748>(2e5b1740) syslog Your administrator session has been forced to close, as another administrator session of the command line interface has been started on the terminal tty1 rvpn_shell[2748]: [03-11 18:03:53]< 2748>(2e5b1740) syslog with the connection address local console
Выход из системной командной оболочки	Запись о событии не заносится в журнал.
Принудительное завершение сессии командного интерпретатора	rvpn_shell: <I_CFG> Command: iplir config - starting
Успешное изменение файла iplir.conf	rvpn_shell: <I_CFG> Command: iplir.conf has been edited successfully
Не удалось изменить содержимое файла iplir.conf	rvpn_shell: <I_CFG> Command: iplir config - starting rvpn_shell: <I_CFG> Command: iplir.config: incorrect configuration или rvpn_shell: <I_CFG> Command: iplir config - starting rvpn_shell: <I_CFG> Command: iplir config Cannot edit config while iplir is running
Успешная смена пароля пользователя	rvpn_shell[1949]: <PSW> Command: 'admin password'. The new password has been set successfully.

Событие	Пример записей в журнале
Не удалось сменить пароль пользователя	rvpn_shell[2748]: [03-11 15:53:57]< 2748>(2e5b1740) syslog <PSW> Command 'admin password' could not be executed. The old password is incorrect.
Удаление справочников и ключей	rvpn_shell[1664]: <RMKEYS> Command 'admin keys remove'
Успешная установка справочников и ключей	root: Keys will be installed from usb root: Partition /dev/sdc1 was successfully mounted on /usb /sbin/unmerge[2552]: [03-11 15:14:17] Keys successfully installed from DST root: Unmerging /tmp/vipnet/abn_0004.dst to /tmp/vipnet/dst
Добавление ключей РНПК	rvpn_shell[2748]: [03-11 16:13:41]< 2748>(2e5b1740) syslog <ADM_IMP> Command 'add spare keys'..
Изменение способа аутентификации пользователя на «Устройство»	rvpn_shell[1664]: command : admin authentication-type-token rvpn_shell[1664]: The two-factor authentication has been enabled successfully.
Локальное обновление ПО	rvpn_shell[2748]: command: admin upgrade software usb
Удаленное обновление ПО	mftpd[7771]: [03-12 10:13:46]< 7771>(4565a740) StartVUpgrade Unpacking stuff with '//usr/bin/lha -xqfw=/mnt/main/tmp//vnsftWumFZO /opt/vipnet/ccc/driv.lzh'. mftpd[7771]: [03-12 10:13:51]< 7771>(4565a740) syslog run command: /sbin/driv_upgrade.sh. Arguments: vupgrade: Check /mnt/main/opt/vipnet/user/mftpd-config.crg success. vupgrade: Check /mnt/main/etc/failoverd-config.crg success.
Задание IP-адреса для сетевого интерфейса	rvpn_shell[2748]: [03-11 16:23:15]< 2748>(2e5b1740) syslog <IFCONF> Command inet ifconfig eth2 address 10.0.0.1 netmask 255.255.255.0 Old ifconfig value: ether 00:0c:29:95:92:ba txqueuelen 1000 (Ethernet)
Сброс настроек сетевого интерфейса	rvpn_shell[2748]: command: inet ifconfig eth2 reset
Включение сетевого интерфейса	rvpn_shell[2748]: [03-11 16:23:39]< 2748>(2e5b1740) syslog <IFCONF> The command 'inet ifconfig eth2 up' executed.
Выключение сетевого интерфейса	rvpn_shell[2748]: [03-11 16:23:42]< 2748>(2e5b1740) syslog <IFCONF> The command 'inet ifconfig eth2 down' executed.

Событие	Пример записей в журнале
Установка параметров скорости сетевого интерфейса	rvpn_shell[1949]: <IFCONF_SPD> Command: inet ifconfig eth2 speed 10 duplex full autoneg on
Добавление статического маршрута	rvpn_shell[6176]: [08-02 14:22:53.443245]< 6176> (00007f26392cf7c0) syslog <RT_ADD> Command: inet route add 123.45.67.89 netmask 255.255.255.0 gateway 123.45.67.1 distance 255 weight 255 table name routes table number 1024. Static route has been added successfully
Удаление статического маршрута	rvpn_shell[6167]: The following route was deleted: Table 1024 (ROUTES), Destination Netmask Next hop Distance Weight, ----- -----, 123.45.67.0 255.255.255.0 123.45.67.1 255 255
Создание виртуального интерфейса VLAN	rvpn_shell[2748]: [03-11 16:36:34]< 2748>(2e5b1740) syslog <I_VLAN_ADD> Command: 'inet ifconfig eth2 vlan add 2'. The new vlan interface eth2.2 has been created.
Удаление виртуального интерфейса VLAN	rvpn_shell[2748]: [03-11 16:37:05]< 2748>(2e5b1740) syslog <I_VLAN_DELETE> Command: 'inet ifconfig eth2 vlan delete 2'. The vlan interface eth2.2 has been deleted.
Сохранение копии конфигурации VPN	[03-11 16:38:20]< 2748>(2e5b1740) syslog <CFG_LD> Command: admin config save test_conf.
Загрузка копии конфигурации VPN	rvpn_shell[1949]: <CFG_LD> Command: admin config load test_conf.
Экспорт журнала регистрации IP-пакетов на USB-носитель	rvpn_shell[2748]: [03-11 16:40:04]< 2748>(2e5b1740) syslog <ADM_EXP> Command: 'admin export packetdb usb'.
Экспорт файлов системного журнала	rvpn_shell[2748]: [03-11 16:49:26]< 2748>(2e5b1740) syslog <ADM_EXP> Command: admin export logs usb.
Экспорт конфигурационной информации	rvpn_shell[2748]: [03-11 16:49:26]< 2748>(2e5b1740) syslog <ADM_EXP> Command: admin export diagnostics usb.
Экспорт справочников, ключей и настроек	rvpn_shell[1949]: <ADM_EXP> Command: 'admin export keys binary-encrypted'

Событие	Пример записей в журнале
Изменение сетевых фильтров и правил трансляции адресов	Отображение изменения: <pre>iplircfg[12145]:[03-11 17:48:05]<12145>(a0ff9700) syslog User rules dataname has been changed. iplircfg[12145]:[03-11 17:48:05]<12145>(a0ff9700) syslog <Rules> iplircfg[12145]:[03-11 17:48:05]<12145>(a0ff9700) syslog + <Rule IsActive="true" xsi:type="vn:LocalRule"> iplircfg[12145]:[03-11 17:48:05]<12145>(a0ff9700) syslog + <RuleId>100001</RuleId> iplircfg[12145]:[03-11 17:48:05]<12145>(a0ff9700) syslog </Rule></pre>
Включение вывода сообщений о событиях	rvpn_shell[1949]: <DBG_ON> Command 'debug on'
Выключение вывода сообщений о событиях	rvpn_shell[1949]: <DBG_ON> Command 'debug off'

Таблица 26. События, связанные с запуском или остановкой служб ViPNet Coordinator HW

Событие	Пример записей в журнале
Загрузка драйверов и запуск служб ViPNet Coordinator HW	<pre>rvpn_shell[2748]: [03-11 16:44:37]< 2748>(2e5b1740) syslog <ADM_VSTART> Command: 'vpn start'.</pre>
Выгрузка драйверов и завершение работы служб ViPNet Coordinator HW	<pre>rvpn_shell[2748]: [03-11 16:44:45]< 2748>(2e5b1740) syslog <ADM_VSTOP> Command 'vpn stop'.</pre>
Запуск службы failoverd, отвечающую за работу системы защиты от сбоев	<pre>rvpn_shell[2748]: [03-11 16:45:28]< 2748>(2e5b1740) syslog <F_STR> Command: failover start</pre>
Завершение работы службы failoverd, отвечающую за работу системы защиты от сбоев	<pre>rvpn_shell[2748]: [03-11 16:37:31]< 2748>(2e5b1740) syslog <F_STP> Command: failover stop</pre>
Запуск управляющей службы iplircfg	<pre>rvpn_shell[2748]: <I_STR> Command: iplir start : command: iplir start rvpn_shell: <I_CFG> Command: iplir config - starting</pre>
Завершение работы управляющей службы iplircfg	<pre>rvpn_shell[2748]: <I_STP> Command: iplir stop : command: iplir stop</pre>
Запуск транспортного сервера MFTP	<pre>rvpn_shell[2748]: [03-11 16:38:04]< 2748>(2e5b1740) syslog <M_STR> Command: mftp start mftp[3989]: [03-11 15:16:41]< 3989>(62cad740) syslog Init MFTP manager ... OK.</pre>

Событие	Пример записей в журнале
Завершение работы транспортного сервера MFTP	rvpn_shell[2748]: [03-11 16:37:47]< 2748>(2e5b1740) syslog <M_STP> Command: mftp stop mftp[2267]: [03-11 16:37:46]< 2267>(d7e13740) syslog Exiting MFTP manager.
Перезапуск веб-сервера ViPNet Coordinator HW	rvpn_shell[2748]: [03-11 16:48:01]< 2748>(2e5b1740) syslog <WUI_RST> Command: webui restart

Таблица 27. События, связанные с работой системы защиты от сбоев

Событие	Пример записей в журнале
Смена режима работы узла кластера из пассивного в активный	failoverd[5261]: [03-12 17:02:08]< 5261>(155c9740) syslog switching to active mode
Перезагрузка узла кластера из-за сбоя на сетевом интерфейсе	failoverd[2341]: [03-12 17:11:54]< 2341>(2509b740) active_failed failure detected on interface eth2 failoverd[2341]: [03-12 17:11:54]< 2341>(2509b740) syslog one or more subsystems failed, rebooting failoverd[2341]: [03-12 17:11:54]< 2341>(2509b740) syslog Rebooted due to network error on eth2 at 03-12 17:11:54
Перезагрузка узла кластера из-за конфликта режимов	failoverd[2341]: [03-12 17:11:54]< 2341>(2509b740) syslog Rebooted due to network error on eth2 at 03-12 17:11:54
Перезагрузка узла кластера из-за обнаружения службой failoverd ошибки другого контролируемого сервиса	failoverd[1411]: [02-01 21:12:54] CheckApp: [mentor.cpp]: Reached maximum count of application restarts [iplircfg] pid 0 failoverd[1411]: Rebooted due to watcher detected error at 03-12 17:11:54

Таблица 28. События, связанные с обменом служебной информацией между узлами ViPNet

Событие	Пример записей в журнале
Получение сообщения о доступности узла ViPNet (сообщения рассылаются при включении узла, а затем периодически)	iplircfg[8281]: [03-12 10:03:35]< 8281>(23aae740) ProcessMonChannel got message: type 1 from ID 0x31DB000A (msg name '@4\$2)F3S.01B')
Получение сообщения о доступности узла ViPNet (сообщения рассылаются при первом обращении к серверу IP-адресов сети ViPNet)	iplircfg[2763]: [03-12 10:29:44]< 2763>(534c4740) ProcessMonChannel got message: type 3 from ID 0x31DB000A (msg name '@!VQ`7ZS.01B')

Событие	Пример записей в журнале
Первое обращение к серверу соединений, если сервер не является также сервером IP-адресов сети ViPNet	iplircfg[13787]: [11-16 16:40:39] ProcessMonChannel: [src/ipserver.cpp]: got message: type 23 from ID 1C2000A (msg name '773PYNG3.2HH')
Отключение узла ViPNet от сервера IP-адресов сети ViPNet	iplircfg[13787]: [11-16 16:40:44] ProcessMonChannel: [src/ipserver.cpp]: got message: type 2 from ID 1C2000A (msg name '85FTKIK3.IAY')
Проверка соединения с узлом ViPNet (запрос)	iplircfg[8281]: [03-12 10:03:34]< 8281>(23aae740) ProcessMonChannel got message: type 5 from ID 0x31DB000C (msg name '@')2)F3S.01B')
Подтверждение получения сообщения (кроме уведомлений об изменении адреса доступа удаленного узла ViPNet)	iplircfg[8281]: [03-12 10:03:34]< 8281>(23aae740) ProcessMonChannel got message: type 7 from ID 0x31DB000A (msg name '0RUX6QK1.PJD')
Уведомление об изменении адреса доступа удаленного узла ViPNet	iplircfg[2763]: [03-12 10:29:43]< 2763>(534c4740) ProcessDrvChannelResolve got message: type 50 from ID 0x31DB000A (msg name '@2VQ`7ZS.01B')
Подтверждение получения уведомления об изменении адреса доступа удаленного узла ViPNet	iplircfg[2763]: [03-12 10:16:09]< 2763>(534c4740) ProcessDrvChannelResolve got message: type 70 from ID 0x31DB000A (msg name 'D09HKJAU.UYN')
Изменение параметров адреса доступа удаленного узла ViPNet	iplircfg[10937]: [12-07 15:33:50] CConfig.SubstNatSettings: subst proxy 01C3000A natsettings for changed ID 01C30B4D: firewallIp 79.133.171.237, port 55777 iplircfg[10937]: [12-07 15:33:46] CConfig::SubstNatSettingsByProxy: subst changed proxy 01D60032 natsettings for ID 01D6004D: firewallIp 10.54.100.10, port 55777 iplircfg[10937]: [12-07 15:35:26] CConfig.SetIpsForId: load natsettings for 5C1A37: firewallip = 88.151.200.50, forwardip = 88.151.200.50, port = 55777, timeout = 0, virtualip = 0, proxyid = 5C05CF, DRV_ALWAYS_USE_SERVER = yes DRV_GREEN_NODE = yes DRV_STOP_FIREWALL_IP=no

Таблица 29. События, связанные с ошибками датчика случайных чисел (ДСЧ) и шифратора

Событие	Пример записей в журнале
Проверка при запуске и регламентная проверка ДСЧ в службах iplircfg и mftpd	iplircfg[2736]: NewRngInitializer failure: NewRandInit has failed: 100 iplircfg[2736]: Failed to initialize CSP iplircfg[2736]: Please, restart application manually.

Событие	Пример записей в журнале
Проверка при запуске и регламентная проверка шифраторов в службах iplircfg и mftpd	iplircfg[2370]: InitCSP has failed iplircfg[2370]: Failed to initialize CSP iplircfg[2370]: Please, restart application manually.
Периодическая проверка ДСЧ в службах	mftpd[1950]: [04-25 17:15:12] CPGenRandom: CPGenRandom unsuccessful, length 8, ErrorCode -2146893795 mftpd[1950]: [04-25 17:15:12] EncryptData: FD 15: Can't encrypt data for alien ID 28750011: 'CCryptoApi::Encrypt()' mftpd[1950]: [04-25 17:15:12] FormINF8Data: FD 15: Can't encrypt data mftpd[1950]: [04-25 17:15:12] Init: FD 15: Can't form INF8 data mftpd[1950]: [04-25 17:15:12] SendERRCommand: FD 15 -> "ERR" mftpd[1950]: [04-25 17:15:12] ProcessMftpConnection: Can't init MFTP connection. Sock 15. mftpd[1950]: [04-25 17:15:12] CMftpConnection.Close: Close connection. Sock 15.
Периодическая проверка шифраторов в службах	mftpd[1950]: [04-25 17:15:12] EncryptData: FD 15: Can't encrypt data for alien ID 28750011: 'CCryptoApi::Encrypt()' mftpd[1950]: [04-25 17:15:12] FormINF8Data: FD 15: Can't encrypt data mftpd[1950]: [04-25 17:15:12] Init: FD 15: Can't form INF8 data mftpd[1950]: [04-25 17:15:12] SendERRCommand: FD 15 -> "ERR" mftpd[1950]: [04-25 17:15:12] ProcessMftpConnection: Can't init MFTP connection. Sock 15. mftpd[1950]: [04-25 17:15:12] CMftpConnection.Close: Close connection. Sock 15.
Проверка при запуске и регламентная проверка ДСЧ в криптодрайвере	vmunix: [34.573918] itcscrt: Run CSP test. vmunix: [34.573920] itcscrt: Driver selftest: TestLinealRecurrentGenerator failed
Проверка при запуске и регламентная проверка шифраторов в криптодрайвере	vmunix: [34.206950] itcscrt: Run CSP test. vmunix: [34.206957] itcscrt: Driver selftest: TestLowStreamImit failed vmunix: [34.206957] itcscrt: Driver selftest: TestLowCloseStreamImit failed vmunix: [34.206958] itcscrt: Driver selftest: TestLowStreamEnCryptCFB failed vmunix: [34.206959] itcscrt: Driver selftest: TestLowStreamDeCryptCFB failed

Событие	Пример записей в журнале
Периодическая проверка ДСЧ в криптодрайвере	<pre>vmunix: [652.427598] itcsrpt: Driver selftest: TestLowStreamImit failed vmunix: [652.427600] itcsrpt: Driver selftest: TestLowCloseStreamImit failed vmunix: [652.427601] itcsrpt: Driver selftest: TestLowStreamEnCryptCFB failed vmunix: [652.427602] itcsrpt: Driver selftest: TestLowStreamDeCryptCFB failed vmunix: [652.427606] itcsrpt: Driver selftest: There are tests with failures.</pre>



Примечание. Записи об успешной работе ДСЧ или шифратора не протоколируются.

Таблица 30. События, связанные с антивирусной проверкой и контролем содержимого трафика прокси-сервера

Событие	Пример записей в журнале
Сработало запрещающее правило контроля содержимого трафика по HTTP-методу	<pre>0 192.168.56.121 TCP_DENIED/403 4293 GET http://www.address.com/ - NONE - text/html</pre>
Сработало запрещающее правило контроля содержимого трафика по MIME-типу	<pre>657 192.168.56.121 TCP_DENIED_REPLY/403 3974 GET http://www.address.com/file.zip - FIRST_UP_PARENT/10.0.25.3 text/html</pre>
Антивирус заблокировал скачивание зараженного файла	<pre>Sep 19 20:49:01 Coord-164a0013 icap[6069]: RESPMOD 127.0.0.1 - 192.168.15.233 http://www.rexswain.com/eicar.zip INFECTED EICAR-Test-File</pre>



Базы управляющей информации (MIB) для протокола SNMP

MIB-II System

MIB-II System (.1.3.6.1.2.1.1) включает в себя объекты, предназначенные для получения информации о системных параметрах сетевого узла (например, имя узла, ОС, количество и параметры сетевых интерфейсов, статистика по SNMP-сообщениям). Подробнее см. в [RFC 1213](#).



Примечание. Количество и индексы сетевых интерфейсов ViPNet Coordinator HW могут различаться для разных групп OID в описывающих их таблицах. При настройке мониторинга по протоколу SNMP рекомендуется вместо индексов использовать имена интерфейсов.

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 31. SNMP-параметры. MIB-II System

OID	Описание
.1.3.6.1.2.1.1.0	Информация о сетевом узле (имя, назначенное сетевому узлу, исполнение и версия ViPNet Coordinator HW).
.1.3.6.1.2.1.1.2.0	OID SNMP-агента.
.1.3.6.1.2.1.1.3.0	Время работы SNMP-агента с последнего запуска (sysUpTime).

OID	Описание
.1.3.6.1.2.1.1.4.0	Контактная информация администратора. Задается командой <code>inet snmp system contact</code> . Если информация не задана, параметр отсутствует.
.1.3.6.1.2.1.1.5.0	Имя, назначенное сетевому узлу. Задается командой <code>inet snmp system name</code> . Если имя не задано, параметр равен <code>hostname</code> .
.1.3.6.1.2.1.1.6.0	Место расположения сетевого узла. Задается командой <code>inet snmp system location</code> . Если не задано, параметр отсутствует.
.1.3.6.1.2.1.1.7.0	Число, означающее наличие сетевых служб на определенных уровнях модели OSI. Для каждого уровня, для которого на узле есть службы, число 2 возводится в степень (L-1) и добавляется к общей сумме. Пример для координатора: $2^{(4-1)} + 2^{(7-1)} = 72$ – узел работает на транспортном и прикладном уровне.
.1.3.6.1.2.1.1.8.0	Значение <code>sysUpTime</code> с момента последнего изменения состояния или значения столбца .1.3.6.1.2.1.1.9.1.2.A из таблицы .1.3.6.1.2.1.1.9.1.

Таблица поддерживаемых MIB (.1.3.6.1.2.1.1.9.1):

A — индекс таблицы

.1.3.6.1.2.1.1.9.1.1.A	Индекс таблицы.
.1.3.6.1.2.1.1.9.1.2.A	OID для MIB, с которыми работает SNMP-агент.
.1.3.6.1.2.1.1.9.1.3.A	Текстовое описание MIB.
.1.3.6.1.2.1.1.9.1.4.A	Время с момента подключения MIB.

ARP-таблица (.1.3.6.1.2.1.3.1.1):

B — индекс интерфейса, присвоенный сетевым узлом

E — IP-адрес сетевого узла

.1.3.6.1.2.1.3.1.1.1.B.1.E	Индекс интерфейса, для которого видна связь IP-MAC.
.1.3.6.1.2.1.3.1.1.2.B.1.E	MAC-адрес.
.1.3.6.1.2.1.3.1.1.3.B.1.E	IP-адрес, соответствующий MAC-адресу.

Таблица описания маршрутов (.1.3.6.1.2.1.4.21.1):

E — IP-адрес сетевого узла или сети

.1.3.6.1.2.1.4.21.1.1.E	IP-адрес назначения маршрутов (маршрут по умолчанию 0.0.0.0).
.1.3.6.1.2.1.4.21.1.2.E	Индекс интерфейса (.1.3.6.1.2.1.2.2.1.1), через который будет производится маршрутизация.
.1.3.6.1.2.1.4.21.1.3.E	Основная метрика протокола маршрутизации (если не используется, передается значение 1).
.1.3.6.1.2.1.4.21.1.4.E	Дополнительная метрика протокола маршрутизации (если не используется, передается значение 1).

OID	Описание
.1.3.6.1.2.1.4.21.1.5.E	Дополнительная метрика протокола маршрутизации (если не используется, передается значение 1).
.1.3.6.1.2.1.4.21.1.6.E	Дополнительная метрика протокола маршрутизации (если не используется, передается значение 1).
.1.3.6.1.2.1.4.21.1.7.E	IP-адрес следующего маршрутизатора по маршруту (Next Hop).
.1.3.6.1.2.1.4.21.1.8.E	Тип маршрута: • 1 (other) — другое, не может быть причислено к другим категориям, • 2 (invalid) — некорректный, • 3 (direct) — маршрут к подключенной сети, • 4 (indirect) — маршрут к удаленной сети.
.1.3.6.1.2.1.4.21.1.9.E	Механизм маршрутизации, через который был получен маршрут: • 1 (other) — другое, не может быть отнесен к остальным группам, • 2 (local) — статический маршрут, заданный в конфигурации, • 3 (netmgmt) — протокол SNMP, • 4 (icmp) — протокол ICMP (Redirect), • 5 (egp) — протокол EGP, • 6 (ggp) — протокол GGP, • 7 (hello) — протокол Hello, • 8 (rip) — протокол RIP, • 9 (is-is) — протокол IS-IS, • 10 (es-is) — протокол ES-IS, • 11 (ciscoIgrp) — протокол cisco IGRP, • 12 (bbnSpflgp) — протокол BBNSPFIGP, • 13 (ospf) — протокол OSPF, • 14 (bgp) — протокол BGP.
.1.3.6.1.2.1.4.21.1.10.E	Время жизни маршрута с последнего обновления (в секундах).
.1.3.6.1.2.1.4.21.1.11.E	Маска маршрута.
.1.3.6.1.2.1.4.21.1.12.E	Дополнительная метрика протокола маршрутизации.
.1.3.6.1.2.1.4.21.1.13.E	Информация о протоколе маршрутизации, с помощью которого был получен маршрут. Если эта информация не представлена, параметр равен 0.0.
.1.3.6.1.2.1.11.1.0	Количество полученных сообщений SNMP.
.1.3.6.1.2.1.11.2.0	Количество отправленных сообщений SNMP.
.1.3.6.1.2.1.11.3.0	Количество полученных сообщений SNMP неподдерживаемой версии.

OID	Описание
.1.3.6.1.2.1.11.4.0	Количество полученных сообщений SNMP, адресованных неизвестному сообществу.
.1.3.6.1.2.1.11.5.0	Количество полученных сообщений SNMP, содержащих операцию, не разрешенную на сетевом узле для сообщества.
.1.3.6.1.2.1.11.6.0	Количество ошибок ASN.1 или BER при разборе SNMP-сообщений.
.1.3.6.1.2.1.11.8.0	Количество входящих SNMP PDU, для которых значение поля error-status – «tooBig».
.1.3.6.1.2.1.11.9.0	Количество входящих SNMP PDU, для которых значение поля error-status – «noSuchName».
1.3.6.1.2.1.11.10.0	Количество входящих SNMP PDU, для которых значение поля error-status – «badValue».
.1.3.6.1.2.1.11.11.0	Количество входящих SNMP PDU, для которых значение поля error-status – «readOnly».
.1.3.6.1.2.1.11.12.0	Количество входящих SNMP PDU, для которых значение поля error-status – «genErr».
.1.3.6.1.2.1.11.13.0	Количество успешно полученных MIB объектов с использованием Get-Request и Get-Next.
.1.3.6.1.2.1.11.14.0	Количество успешно записанных MIB объектов с использованием Set-Request.
.1.3.6.1.2.1.11.15.0	Количество успешно обработанных SNMP Get-Request PDUs.
.1.3.6.1.2.1.11.16.0	Количество успешно обработанных SNMP Get-Next PDUs.
.1.3.6.1.2.1.11.17.0	Количество успешно обработанных SNMP Set-Request PDUs.
.1.3.6.1.2.1.11.18.0	Количество успешно обработанных SNMP Get-Response PDUs.
.1.3.6.1.2.1.11.19.0	Количество успешно обработанных SNMP Trap PDUs.
.1.3.6.1.2.1.11.20.0	Количество исходящих SNMP PDU, для которых значение поля error-status – «tooBig».
.1.3.6.1.2.1.11.21.0	Количество исходящих SNMP PDU, для которых значение поля error-status – «noSuchName».
.1.3.6.1.2.1.11.22.0	Количество исходящих SNMP PDU, для которых значение поля error-status – «badValue».
.1.3.6.1.2.1.11.24.0	Количество исходящих SNMP PDU, для которых значение поля error-status – «genErr».
.1.3.6.1.2.1.11.25.0	Количество успешно сгенерированных исходящих SNMP Get-Request PDUs.
.1.3.6.1.2.1.11.26.0	Количество успешно сгенерированных исходящих SNMP Get-Next PDUs.

OID	Описание
.1.3.6.1.2.1.11.27.0	Количество успешно сгенерированных исходящих SNMP Set-Request PDUs.
.1.3.6.1.2.1.11.28.0	Количество успешно сгенерированных исходящих SNMP Get-Response PDUs.
.1.3.6.1.2.1.11.29.0	Количество успешно сгенерированных исходящих SNMP Trap PDUs.
.1.3.6.1.2.1.11.30.0	Возможность отправки authenticationFailure traps: • 1 — разрешено, • 2 — запрещено.
.1.3.6.1.2.1.11.31.0	Количество параметров отброшенных SNMP без уведомлений в связи с несоответствием параметров агента и сервера.
.1.3.6.1.2.1.11.32.0	Количество параметров отброшенных SNMP без уведомлений в связи с проблемами проху.

IF-MIB

IF-MIB (.1.3.6.1.2.1.2) включает в себя объекты, предназначенные для получения информации о параметрах сетевых интерфейсов сетевого узла (например, тип сетевого интерфейса, физический адрес, пропускная способность). Подробнее см. в [RFC 2863](#).



Примечание. Количество и индексы сетевых интерфейсов ViPNet Coordinator HW могут различаться для разных групп OID в описывающих их таблицах. При настройке мониторинга по протоколу SNMP вместо индексов используйте названия (логические имена) сетевых интерфейсов.

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 32. SNMP-параметры. IF-MIB

OID	Описание
.1.3.6.1.2.1.2.1.0	Количество сетевых интерфейсов (физических, виртуальных, включая loopback).
Таблица объектов, описывающих сетевые интерфейсы (.1.3.6.1.2.1.2.1):	
A — индекс интерфейса	
.1.3.6.1.2.1.2.2.1.1.A	Индекс сетевого интерфейса (A).
.1.3.6.1.2.1.2.2.1.2.A	Описание интерфейса (производитель, название и версия оборудования или ПО).

OID	Описание
.1.3.6.1.2.1.2.2.1.3.A	Тип сетевого интерфейса: • 6 — ethernet, • 24 — localhost. Остальные значения можно посмотреть здесь.
.1.3.6.1.2.1.2.2.1.4.A	MTU (в байтах).
.1.3.6.1.2.1.2.2.1.5.A	Максимальная пропускная способность (бит/сек).
.1.3.6.1.2.1.2.2.1.6.A	MAC-адрес.
.1.3.6.1.2.1.2.2.1.7.A	Назначенное состояние интерфейса: • 1 (up) — включен, • 2 (down) — выключен, • 3 (testing) — состояние тестирования, указывает, что никакие рабочие пакеты не могут быть переданы.
.1.3.6.1.2.1.2.2.1.8.A	Текущее рабочее состояние интерфейса: • 1 (up) — включен, • 2 (down) — выключен, • 3 (testing) — состояние тестирования. Никакие рабочие пакеты не могут быть переданы, • 4 (unknown) — неизвестно, • 5 (dormant) — интерфейс ожидает внешних действий (таких как последовательная линия, ожидающая входящее соединение), • 6 (notPresent) — в интерфейсе отсутствуют компоненты (обычно аппаратные), • 7 (lowerLayerDown) — не работает из-за состояния интерфейса нижнего уровня.
.1.3.6.1.2.1.2.2.1.9.A	Время работы SNMP-агента с начала его запуска (sysUpTime) до момента перехода интерфейса в его текущее состояние.
.1.3.6.1.2.1.2.2.1.10.A	Количество полученных байт (октетов).
.1.3.6.1.2.1.2.2.1.11.A	Количество полученных одноадресных (unicast) пакетов.
.1.3.6.1.2.1.2.2.1.12.A	Количество полученных широковещательных (broadcast) и многоадресных (multicast) пакетов.
.1.3.6.1.2.1.2.2.1.13.A	Количество отклоненных пакетов.
.1.3.6.1.2.1.2.2.1.14.A	Количество пакетов, отклоненных из-за ошибок.
.1.3.6.1.2.1.2.2.1.15.A	Количество пакетов, которые были отклонены из-за неизвестного или не поддерживаемого протокола.
.1.3.6.1.2.1.2.2.1.16.A	Количество отправленных байт (октетов).
.1.3.6.1.2.1.2.2.1.17.A	Количество отправленных одноадресных (unicast) пакетов.

OID	Описание
.1.3.6.1.2.1.2.2.1.18.A	Количество отправленных широковещательных (broadcast) и многоадресных (multicast) пакетов.
.1.3.6.1.2.1.2.2.1.19.A	Количество отклоненных отправленных пакетов.
.1.3.6.1.2.1.2.2.1.20.A	Количество отправленных пакетов, отклоненных из-за ошибок.
.1.3.6.1.2.1.2.2.1.21.A	Количество исходящих пакетов в очереди на отправку.

Таблица статистики дополнительных параметров сетевых интерфейсов (.1.3.6.1.2.1.31.1.1.1):

A — индекс интерфейса из таблицы 1.3.6.1.2.1.2.2.1

.1.3.6.1.2.1.31.1.1.1.1.A	Имя интерфейса.
.1.3.6.1.2.1.31.1.1.1.2.A	Количество входящих многоадресных (multicast) пакетов.
.1.3.6.1.2.1.31.1.1.1.3.A	Количество входящих широковещательных (broadcast) пакетов.
.1.3.6.1.2.1.31.1.1.1.4.A	Количество исходящих многоадресных (multicast) пакетов.
.1.3.6.1.2.1.31.1.1.1.5.A	Количество исходящих широковещательных (broadcast) пакетов.
.1.3.6.1.2.1.31.1.1.1.6.A	Количество байт (октетов), принятых интерфейсом (64-разрядный счетчик).
.1.3.6.1.2.1.31.1.1.1.7.A	Количество входящих одноадресных (unicast) пакетов (64-разрядный счетчик).
.1.3.6.1.2.1.31.1.1.1.8.A	Количество входящих многоадресных (multicast) пакетов (64-разрядный счетчик).
.1.3.6.1.2.1.31.1.1.1.9.A	Количество входящих широковещательных (broadcast) пакетов (64-разрядный счетчик).
.1.3.6.1.2.1.31.1.1.1.10.A	Количество байт (октетов), отправленных интерфейсом (64-разрядный счетчик).
.1.3.6.1.2.1.31.1.1.1.11.A	Количество исходящих одноадресных (unicast) пакетов (64-разрядный счетчик).
.1.3.6.1.2.1.31.1.1.1.12.A	Количество исходящих многоадресных (multicast) пакетов (64-разрядный счетчик).
.1.3.6.1.2.1.31.1.1.1.13.A	Количество исходящих широковещательных (broadcast) пакетов (64-разрядный счетчик).
.1.3.6.1.2.1.31.1.1.1.14.A	Необходимость отправки trap на события linkUp/linkDown: • 1 — отправлять trap, • 2 — не отправлять trap.
.1.3.6.1.2.1.31.1.1.1.15.A	Максимальная скорость интерфейса (Мегабит/сек).
.1.3.6.1.2.1.31.1.1.1.16.A	Режим работы интерфейса: • 1 (true) — принимает все пакеты, • 2 (false) — принимает только пакеты, адресованные сетевому узлу.

OID	Описание
.1.3.6.1.2.1.31.1.1.17.A	Наличие физического коннектора у интерфейса: 1 (true) — есть коннектор (для ethernet), 2 (false) — нет коннектора (для localhost).
.1.3.6.1.2.1.31.1.1.18.A	Имя интерфейса, заданное администратором. Для координатора всегда равно 0.
.1.3.6.1.2.1.31.1.1.19.A	Время работы SNMP-агента с последнего запуска (sysUpTime) до появления разрыва в подсчете счетчиков на интерфейсе.
.1.3.6.1.2.1.31.1.5.0	Время работы SNMP-агента с последнего запуска (sysUpTime) до последнего добавления или удаления записи в таблице статистики дополнительных параметров сетевых интерфейсов.

IP-MIB

IP-MIB (.1.3.6.1.2.1.4) включает в себя объекты, предназначенные для получения информации о работе сетевого узла по протоколу IP (например, параметры сетевых интерфейсов, статистика для IP-трафика, таблица IP-адресов). Подробнее см. в [RFC 4293](#).

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 33. SNMP-параметры. IP-MIB

OID	Описание
.1.3.6.1.2.1.4.1.0	Указывает, является ли сетевой узел маршрутизатором: 1 — является (координатор всегда работает в данном режиме), 2 — не является.
.1.3.6.1.2.1.4.2.0	TTL по умолчанию для IP-пакетов.
.1.3.6.1.2.1.4.3.0	Общее количество IP-пакетов, принятых сетевым узлом, включая пакеты с ошибками.
.1.3.6.1.2.1.4.4.0	Количество входящих IP-пакетов, отклоненных сетевым узлом по причине ошибок в заголовке пакета.
.1.3.6.1.2.1.4.5.0	Количество входящих IP-пакетов, отклоненных сетевым узлом по причине ошибок в адресе.
.1.3.6.1.2.1.4.6.0	Количество IP-пакетов, переданных на следующий шлюз.
.1.3.6.1.2.1.4.7.0	Количество IP-пакетов, отклоненных по причине неизвестного или неподдерживаемого протокола.
.1.3.6.1.2.1.4.8.0	Количество отклоненных пакетов по причине недостатка буфера.
.1.3.6.1.2.1.4.9.0	Количество успешно доставленных IP-пакетов.

OID	Описание
.1.3.6.1.2.1.4.10.0	Количество IP-пакетов, отправленных сетевым узлом (не включая .1.3.6.1.2.1.4.6.0).
.1.3.6.1.2.1.4.11.0	Количество IP-пакетов, отклонённых по причине недостатка буфера.
.1.3.6.1.2.1.4.12.0	Количество IP-пакетов, отклоненных к дальнейшей передаче по причине отсутствия маршрута.
.1.3.6.1.2.1.4.13.0	Максимальное количество секунд ожидания сборки фрагментов.
.1.3.6.1.2.1.4.14.0	Количество принятых IP-фрагментов.
.1.3.6.1.2.1.4.15.0	Количество успешных сборок IP-пакетов из фрагментов.
.1.3.6.1.2.1.4.16.0	Количество ошибок сборки IP-пакетов из фрагментов.
.1.3.6.1.2.1.4.17.0	Количество успешно фрагментированных исходящих IP-пакетов.
.1.3.6.1.2.1.4.18.0	Количество ошибок при фрагментации исходящих IP-пакетов.
.1.3.6.1.2.1.4.19.0	Количество фрагментов IP-пакетов, сгенерированных данным сетевым узлом.

Таблица IP-адресов сетевого узла для протокола IPv4 (.1.3.6.1.2.1.4.20.1.):

E — IP-адрес

.1.3.6.1.2.1.4.20.1.1.E	IP-адрес.
.1.3.6.1.2.1.4.20.1.2.E	Указатель на интерфейс, которому присвоен этот IP-адрес (A - из таблицы .1.3.6.1.2.1.2.2.1).
.1.3.6.1.2.1.4.20.1.3.E	Маска сети.
.1.3.6.1.2.1.4.20.1.4.E	Значение наименее значимого бита в широковещательном (broadcast) IP-адресе.

Таблица разрешения IP-адресов в физические адреса (ARP-таблица) (.1.3.6.1.2.1.4.22.1.):

A — индекс интерфейса из таблицы 1.3.6.1.2.1.2.2.1.1 IF-MIB

E — IP-адрес

.1.3.6.1.2.1.4.22.1.1.A.E	Индекс интерфейса, для которого видна связка IP-МАС.
.1.3.6.1.2.1.4.22.1.2.A.E	МАС-адрес.
.1.3.6.1.2.1.4.22.1.3.A.E	IP-адрес.
.1.3.6.1.2.1.4.22.1.4.A.E	Тип маппинга: • 1 — другое, • 2 — некорректный, • 3 — динамический, • 4 — статический.
.1.3.6.1.2.1.4.23.0	Количество отклоненных маршрутов.

OID	Описание
.1.3.6.1.2.1.4.31.1.1.	Таблица, содержащая статистику (счетчики) IP-трафика различных версий на сетевом узле для всех интерфейсов.
.1.3.6.1.2.1.4.31.2.0	Значение sysUpTime в момент добавления/удаления столбцов в таблицу .1.3.6.1.2.1.4.31.1.1.
Таблица префиксов IP-адресов (.1.3.6.1.2.1.4.32.1.):	
A — индекс интерфейса из таблицы 1.3.6.1.2.1.2.2.1.1 IF-MIB	
B — тип адреса:	
• значение 0 — неизвестно, • значение 1 — ipv4, • значение 2 — ipv6, • значение 3 — ipv4z, • значение 4 — ipv6z, • значение 16 — dns.	
C — префикс (адрес сети)	
D — длина префикса	
.1.3.6.1.2.1.4.32.1.5.A.B.C.D	Источник префикса: • 1 — другое, • 2 — вручную, • 3 — общеизвестный префикс, • 4 — dhcp, • 3 — routeradv (значение получено от роутера).
.1.3.6.1.2.1.4.32.1.6.A.B.C.D	Указывает, может ли префикс использоваться как маршрут: • 1 — может, • 2 — не может.
.1.3.6.1.2.1.4.32.1.7.A.B.C.D	Указывает, может ли префикс использоваться для автоматической конфигурации адреса: • 1 — может, • 2 — не может.
.1.3.6.1.2.1.4.32.1.8.A.B.C.D	Время (в секундах), в течение которого префикс будет оставаться предпочтительным. Значение 4294967295 означает бесконечность.
.1.3.6.1.2.1.4.32.1.9.A.B.C.D	Время (в секундах), в течение которого префикс будет существовать. Значение 4294967295 означает бесконечность.

OID	Описание
Таблица адресной информации по интерфейсам (.1.3.6.1.2.1.4.34.1):	
B — тип адреса:	
• значение 0 — неизвестно, • значение 1 — ipv4, • значение 2 — ipv6, • значение 3 — ipv4z, • значение 4 — ipv6z, • значение 16 — dns.	
E — IP-адрес	
.1.3.6.1.2.1.4.34.1.3.B.E	Индекс сетевого интерфейса.
.1.3.6.1.2.1.4.34.1.4.B.E	Тип адреса: • 1 (unicast) — одноадресный, • 2 (anycast) — к ближайш ему узлу, • 3 (broadcast) — ш ироковещательный.
.1.3.6.1.2.1.4.34.1.5.B.E	Ссылка на строку таблицы префиксов.
.1.3.6.1.2.1.4.34.1.6.B.E	Происхождение адреса: • 1 (other) — другое, • 2 (manual) — назначен вручную, • 4 (dhcp) — назначен по dhcp, • 5 (linklayer) — назначен механизмом автоконфигурации протокола IPv6 (SLAAC), • 6 (random) — назначен случайным образом.
.1.3.6.1.2.1.4.34.1.7.B.E	Статус адреса: • 1 (preferred) — предпочтительный, • 2 (deprecated) — устарел, • 3 (invalid) — недействителен, • 4 (inaccessible) — недоступен, • 5 (unknown) — неизвестен, • 6 (tentative) — предварительный, • 7 (duplicate) — дублированный, • 8 (optimistic) — оптимистичный.
.1.3.6.1.2.1.4.34.1.8.B.E	Время работы SNMP-агента после запуска (sysUpTime) до момента создания записи.
.1.3.6.1.2.1.4.34.1.9.B.E	Время работы SNMP-агента после запуска (sysUpTime) до момента изменения записи.

OID	Описание
.1.3.6.1.2.1.4.34.1.10.B.E	Статус строки: • 1 (active) — строка доступна для использования, • 2 (notInService) — строка недоступна для использования, • 3 (notReady) — отсутствует информация для использования, • 4 (createAndGo) — создание строки и перевод ее в статус active, • 5 (createAndWait) — создание строки и перевод ее в статус notInService, • 6 (destroy) — удаление.
.1.3.6.1.2.1.4.34.1.11.B.E	Механизм хранения строки: • 1 (other) — другое, • 2 (volatile) — изменяемая, • 3 (nonvolatile) — энергонезависимая, • 4 (permanent) — постоянная, • 5 (readOnly) — только для чтения.
Таблица трансляции MAC IP (.1.3.6.1.2.1.4.35.1): A — индекс интерфейса из таблицы .1.3.6.1.2.1.2.1.1 MIB-II B — тип адреса: • 0 — неизвестно, • 1 — ipv4, • 2 — ipv6, • 3 — ipv4z, • 4 — ipv6z, • 16 — dns. E — IP-адрес	
.1.3.6.1.2.1.4.35.1.4.A.B.E	Транслируемый MAC-адрес.
.1.3.6.1.2.1.4.35.1.5.A.B.E	Время работы SNMP-агента с момента последнего запуска (sysUpTime) до момента изменения MAC-адреса.
.1.3.6.1.2.1.4.35.1.6.A.B.E	Тип маппинга: • 1 (other) — другое, • 2 (invalid) — недопустимое значение, • 3 (dynamic) — динамическое, • 4 (static) — статическое, • 5 (local) — предоставляется для собственного адреса интерфейса объекта.

OID	Описание
.1.3.6.1.2.1.4.35.1.7.A.B.E	Статус маппинга: • 1(reachable) — подтвержден, • 2 (stale) — не подтвержден, • 3 (delay) — ожидание, • 4 (proberprobe) — активное, • 5 (invalid) — недействителен, • 6 (unknown) — не определен, • 7 (incomplete) — выполняется разрешение адреса.
.1.3.6.1.2.1.4.35.1.8.A.B.E	Статус строки: • 1 (active) — строка доступна для использования, • 2 (notInService) — строка недоступна для использования, • 3 (notReady) — отсутствует информация для использования, • 4 (createAndGo) — создание строки и перевод ее в статус active, • 5 (createAndWait) — создание строки и перевод ее в статус notInService, • 6 (destroy) — удаление.
.1.3.6.1.2.1.5.1.0	Общее число ICMP-запросов, полученных сетевым узлом.
.1.3.6.1.2.1.5.2.0	Число ICMP-запросов с ошибкой, полученных сетевым узлом.
.1.3.6.1.2.1.5.3.0	Количество сообщений о недоступности заданного сетевого узла, полученных данным сетевым узлом.
.1.3.6.1.2.1.5.4.0	Количество сообщений об истечении времени ожидания ответа, полученных сетевым узлом.
.1.3.6.1.2.1.5.5.0	Количество полученных сообщений ICMP Parameter Problem.
.1.3.6.1.2.1.5.6.0	Количество полученных сообщений ICMP Source Quench.
.1.3.6.1.2.1.5.7.0	Количество полученных сообщений ICMP Redirect messages.
.1.3.6.1.2.1.5.8.0	Количество полученных сообщений ICMP Echo (request).
.1.3.6.1.2.1.5.9.0	Количество полученных сообщений ICMP Echo Reply.
.1.3.6.1.2.1.5.10.0	Количество полученных сообщений ICMP Timestamp (request).
.1.3.6.1.2.1.5.11.0	Количество полученных сообщений ICMP Timestamp Reply.
.1.3.6.1.2.1.5.12.0	Количество полученных сообщений ICMP Address Mask Request.
.1.3.6.1.2.1.5.13.0	Количество полученных сообщений ICMP Address Mask Reply.
.1.3.6.1.2.1.5.14.0	Количество ICMP сообщений, отправленных сетевым узлом.
.1.3.6.1.2.1.5.15.0	Количество ICMP сообщений, которые сетевой узел не смог отправить из-за проблем с ICMP.
.1.3.6.1.2.1.5.16.0	Количество отправленных сообщений ICMP Destination Unreachable.

OID	Описание
.1.3.6.1.2.1.5.17.0	Количество отправленных сообщений ICMP Time Exceeded.
.1.3.6.1.2.1.5.18.0	Количество отправленных сообщений ICMP Parameter Problem.
.1.3.6.1.2.1.5.19.0	Количество отправленных сообщений ICMP Source Quench.
.1.3.6.1.2.1.5.20.0	Количество отправленных сообщений ICMP Redirect.
.1.3.6.1.2.1.5.21.0	Количество отправленных сообщений ICMP Echo (request).
.1.3.6.1.2.1.5.22.0	Количество отправленных сообщений ICMP Echo Reply.
.1.3.6.1.2.1.5.23.0	Количество отправленных сообщений ICMP Timestamp (request).
.1.3.6.1.2.1.5.24.0	Количество отправленных сообщений ICMP Timestamp Reply.
.1.3.6.1.2.1.5.25.0	Количество отправленных сообщений ICMP Address Mask Request.
.1.3.6.1.2.1.5.26.0	Количество отправленных сообщений ICMP Address Mask Reply.
.1.3.6.1.2.1.5.29.1	Таблица счетчиков ICMP сообщений с указанием версии протокола IP.
.1.3.6.1.2.1.5.30.1	Таблица счетчиков входящих-исходящих ICMP пакетов с указанием версии протокола IP.

IP-FORWARD-MIB

IP-FORWARD-MIB (.1.3.6.1.2.1.4.24) включает в себя объекты, предназначенные для получения информации о параметрах маршрутизации сетевого узла. Подробнее см. в [RFC 4292](#).

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 34. SNMP-параметры. IP-FORWARD-MIB

OID	Описание
Таблица маршрутизации сетевого узла (.1.3.6.1.2.1.4.24.4.1):	
.1.3.6.1.2.1.4.24.4.1.1.A.B.C.D	Адрес назначения маршрута (A).
.1.3.6.1.2.1.4.24.4.1.2.A.B.C.D	Маска адреса назначения маршрута (B).
.1.3.6.1.2.1.4.24.4.1.3.A.B.C.D	Параметр, обозначающий значение поля IP TOS Field (C).
.1.3.6.1.2.1.4.24.4.1.4.A.B.C.D	Адрес следующего маршрутизатора в маршруте (D).
.1.3.6.1.2.1.4.24.4.1.5.A.B.C.D	Индекс интерфейса, для которого задан маршрут из таблицы .1.3.6.1.2.1.2.2.1 IF-MIB.

OID	Описание
.1.3.6.1.2.1.4.24.4.1.6.A.B.C.D	Тип маршрута: • 1 (other) — другое, не может быть отнесен к остальным группам, • 2 (reject) — недостижимый, • 3 (local) — маршрут, в котором адрес следующего маршрутизатора является адресом назначения, • 4 (remote) — маршрут, в котором адрес следующего маршрутизатора не является адресом назначения.
.1.3.6.1.2.1.4.24.4.1.7.A.B.C.D	Механизм маршрутизации, с помощью которого был получен маршрут: • 1 (other) — другое, не может быть отнесен к остальным группам, • 2 (local) — статический маршрут, заданный в конфигурации, • 3 (netmgmt) — протокол SNMP, • 4 (icmp) — протокол ICMP (Redirect), • 5 (egp) — протокол EGP, • 6 (ggp) — протокол GGP, • 7 (hello) — протокол Hello, • 8 (rip) — протокол RIP, • 9 (is-is) — протокол IS-IS, • 10 (es-is) — протокол ES-IS, • 11 (ciscoIgrp) — протокол cisco IGRP, • 12 (bbnSpfigp) — протокол BBNSPFIGP, • 13 (ospf) — протокол OSPF, • 14 (bgp) — протокол BGP, • 15 (idpr) — протокол IDRP, • 16 (ciscoEigrp) — протокол cisco EIGRP.
.1.3.6.1.2.1.4.24.4.1.8.A.B.C.D	Время жизни маршрута с последнего обновления (в секундах).
.1.3.6.1.2.1.4.24.4.1.9.A.B.C.D	Информация о протоколе маршрутизации, с помощью которого был получен маршрут. Если эта информация не представлена, параметр равен 0.0.
.1.3.6.1.2.1.4.24.4.1.10.A.B.C.D	Номер AS, к которой относится следующий маршрутизатор в маршруте. Если неизвестен или если для маршрутизации не используется, параметр равен 0.
.1.3.6.1.2.1.4.24.4.1.11.A.B.C.D	Основная метрика маршрута. Если метрика не используется, параметр равен -1.
.1.3.6.1.2.1.4.24.4.1.12.A.B.C.D	Дополнительная метрика маршрута. Если метрика не используется, параметр равен -1.
.1.3.6.1.2.1.4.24.4.1.13.A.B.C.D	Дополнительная метрика маршрута. Если метрика не используется, параметр равен -1.

OID	Описание
.1.3.6.1.2.1.4.24.4.1.14.A.B.C.D	Дополнительная метрика маршрута. Если метрика не используется, параметр равен -1.
.1.3.6.1.2.1.4.24.4.1.15.A.B.C.D	Дополнительная метрика маршрута. Если метрика не используется, параметр равен -1.
.1.3.6.1.2.1.4.24.4.1.16.A.B.C.D	Статус строки таблицы: • 1 (active) — строка доступна для использования, • 2 (notInService) — строка недоступна для использования, • 3 (notReady) — отсутствует информация для использования, • 4 (createAndGo) — создание строки и перевод ее в статус active, • 5 (createAndWait) — создание строки и перевод ее в статус notInService, • 6 (destroy) — удаление
1.3.6.1.2.1.4.24.6.0	Количество текущих корректных маршрутов из таблицы
.1.3.6.1.2.1.4.24.7.1	
Таблица маршрутизации сетевого узла с указанием версии протокола IP (.1.3.6.1.2.1.4.24.7.1):	
.1.3.6.1.2.1.4.24.7.1.7	Индекс интерфейса, для которого задан маршрут из таблицы .1.3.6.1.2.1.2.2 IF-MIB.
.1.3.6.1.2.1.4.24.7.1.8	Тип маршрута: • 1 (other) — другое, не может быть отнесен к остальным группам, • 2 (reject) — недостижимый, • 3 (local) — маршрут, в котором адрес следующего маршрутизатора является адресом назначения, • 4 (remote) — маршрут, в котором адрес следующего маршрутизатора не является адресом назначения, • 5 (blackhole) — маршрут, по которому пакеты теряются без сообщений об ошибках.

OID	Описание
.1.3.6.1.2.1.4.24.7.1.9	Механизм маршрутизации, с помощью которого был получен маршрут: • 1 (other) — другое, не может быть отнесен к остальным группам, • 2 (local) — статический маршрут, заданный в конфигурации, • 3 (netmgmt) — протокол SNMP, • 4 (icmp) — протокол ICMP (Redirect), • 5 (egp) — протокол EGP, • 6 (ggp) — протокол GGP, • 7 (hello) — протокол Hello, • 8 (rip) — протокол RIP, • 9 (is-is) — протокол IS-IS, • 10 (es-is) — протокол ES-IS, • 11 (ciscoIgrp) — протокол cisco IGRP, • 12 (bbnSpfIgp) — протокол BBNSPFIGP, • 13 (ospf) — протокол OSPF, • 14 (bgp) — протокол BGP, • 15 (idpr) — протокол IDRP, • 16 (ciscoEigrp) — протокол cisco EIGRP.
.1.3.6.1.2.1.4.24.7.1.10	Время жизни маршрута с последнего обновления (в секундах).
.1.3.6.1.2.1.4.24.7.1.11	Номер AS, к которой относится следующий маршрутизатор в маршруте. Если неизвестен или если для маршрутизации не используется, параметр равен 0.
.1.3.6.1.2.1.4.24.7.1.12	Основная метрика маршрута. Если метрика не используется, параметр равен -1.
.1.3.6.1.2.1.4.24.7.1.13	Дополнительная метрика маршрута. Если метрика не используется, параметр равен -1.
.1.3.6.1.2.1.4.24.7.1.14	Дополнительная метрика маршрута. Если метрика не используется, параметр равен -1.
.1.3.6.1.2.1.4.24.7.1.15	Дополнительная метрика маршрута. Если метрика не используется, параметр равен -1.
.1.3.6.1.2.1.4.24.7.1.16	Дополнительная метрика маршрута. Если метрика не используется, параметр равен -1.

OID	Описание
.1.3.6.1.2.1.4.24.7.1.17	Статус строки таблицы: 1 (active) — строка доступна для использования, 2 (notInService) — строка недоступна для использования, 3 (notReady) — отсутствует информация для использования, 4 (createAndGo) — создание строки и перевод ее в статус active, 5 (createAndWait) — создание строки и перевод ее в статус notInService, 6 (destroy) — удаление.

Ethernet-Like MIB

Ethernet-Like MIB (1.3.6.1.2.1.10) включает в себя объекты, предназначенные для получения статистической информации о работе интерфейсов сетевого узла. Подробнее см. в [RFC 3635](#).

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 35. SNMP-параметры. Ethernet-Like MIB

OID	Описание
Таблица статистики Ethernet-интерфейсов (.1.3.6.1.2.1.10.7.2.1):	
A — индекс интерфейса из таблицы 1.3.6.1.2.1.2.2.1.1 IF-MIB	
.1.3.6.1.2.1.10.7.2.1.1.A	Индекс интерфейса из таблицы 1.3.6.1.2.1.2.2.1.1 IF-MIB.
.1.3.6.1.2.1.10.7.2.1.2.A	Количество ошибок на интерфейсе.
.1.3.6.1.2.1.10.7.2.1.3.A	Количество FCS ошибок на интерфейсе.
.1.3.6.1.2.1.10.7.2.1.4.A	Количество фреймов, вовлеченных в SingleCollision на интерфейсе.
.1.3.6.1.2.1.10.7.2.1.5.A	Количество фреймов, вовлеченных в MultiplyCollision на интерфейсе.
.1.3.6.1.2.1.10.7.2.1.6.A	Количество ошибок SQE TEST ERROR на интерфейсе.
.1.3.6.1.2.1.10.7.2.1.7.A	Количество фреймов, для которых первая попытка отправки была отложена на интерфейсе.
.1.3.6.1.2.1.10.7.2.1.8.A	Количество обнаружения коллизий на интерфейсе, при котором коллизия обнаружена позднее одного slotTime.
.1.3.6.1.2.1.10.7.2.1.9.A	Количество фреймов, которые не удалось передать в связи с коллизиями.
.1.3.6.1.2.1.10.7.2.1.10.A	Количество фреймов, которые не удалось передать в связи с проблемами MAC уровня.
.1.3.6.1.2.1.10.7.2.1.11.A	Количество ошибок передачи в связи с потерей carrier sense condition на интерфейсе.

OID	Описание
.1.3.6.1.2.1.10.7.2.1.13.A	Количество фреймов, принятых на интерфейсе, размер которых превышал максимально допустимый.
.1.3.6.1.2.1.10.7.2.1.16.A	Количество фреймов, которые не удалось принять на интерфейсе в связи с ошибками MAC уровня.
.1.3.6.1.2.1.10.7.2.1.17.A	Идентификатор чипа, который реализует Ethernet-интерфейс.
.1.3.6.1.2.1.10.7.2.1.18.A	Количество ошибок.
.1.3.6.1.2.1.10.7.2.1.19.A	Текущий режим дуплекса на интерфейсе: • 1 — unknown, • 2 — halfDuplex, • 3 — fullDuplex. Для неподключенных интерфейсов параметр равен 1.
1.3.6.1.2.1.10.7.2.1.20.A	Возможность контроля скорости передачи на интерфейсе: • 1 — разрешено, • 2 — запрещено.
.1.3.6.1.2.1.10.7.2.1.21AA	Статус контроля скорости передачи на интерфейсе: • 1 — выключен, • 2 — включен, • 3 — неизвестно.

TCP-MIB

TCP-MIB (.1.3.6.1.2.1.6) включает в себя объекты, предназначенные для получения информации о работе сетевого узла по протоколу TCP (например, количество отправленных сегментов, активных соединений). Подробнее см. в [RFC 4022](#).

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 36. SNMP-параметры. TCP-MIB

OID	Описание
.1.3.6.1.2.1.6.1.0	Алгоритм определения тайм-аута для повторной передачи неопознанных октетов.
.1.3.6.1.2.1.6.2.0	Минимальное допустимое значение тайм-аута для повторной передачи (в миллисекундах).
.1.3.6.1.2.1.6.3.0	Максимальное допустимое значение тайм-аута для повторной передачи (в миллисекундах).
.1.3.6.1.2.1.6.4.0	Максимальное количество TCP-соединений. При динамическом определении, параметр равен -1.

OID	Описание
.1.3.6.1.2.1.6.5.0	Количество прямых переходов TCP-соединений из состояния CLOSED в состояние SYN-SENT.
.1.3.6.1.2.1.6.6.0	Количество прямых переходов TCP-соединений из состояния LISTEN в состояние SYN-RCVD.
.1.3.6.1.2.1.6.7.0	Количество прямых переходов TCP-соединений в состояние CLOSED из состояния SYN-SENT или SYN-RCVD, а также прямых переходов в состояние LISTEN из состояния SYN-RCVD.
.1.3.6.1.2.1.6.8.0	Количество переходов TCP-соединений в состояние CLOSED из состояния ESTABLISHED или CLOSE-WAIT.
.1.3.6.1.2.1.6.9.0	Количество TCP-соединений, находящихся в состоянии ESTABLISHED или CLOSE-WAIT.
.1.3.6.1.2.1.6.10.0	Количество полученных сегментов, в том числе с ошибками. Включает только количество полученных сегментов на соединениях, установленных в настоящий момент.
.1.3.6.1.2.1.6.11.0	Количество отправленных сегментов.
.1.3.6.1.2.1.6.12.0	Общее количество повторно отправленных сегментов.

Таблица TCP-соединений для протокола IPv4 (.1.3.6.1.2.1.6.13.1):

A - локальный IP-адрес

B - локальный порт

C - удаленный IP-адрес

D - удаленный порт

OID	Описание
.1.3.6.1.2.1.6.13.1.1.A.B.C.D	Состояние TCP-соединения: • 1 (closed) — закрыто, • 2 (listen) — ожидание запроса на соединение со стороны чужих портов и программ TCP, • 3 (synSent) — ожидание парного запроса на установление соединения, • 4 (synReceived) — ожидание подтверждения после того, как запрос соединения уже принят и отправлен, • 5 (established) — установлено, • 6 (finWait1) — ожидание запроса от программы TCP или подтверждения ранее отправленного запроса на закрытие соединения, • 7 (finWait2) — ожидание запроса на закрытие соединения со стороны программы TCP, • 8 (closeWait) — ожидание запроса на закрытие соединения со стороны своего клиента, • 9 (lastAck) — ожидание запроса на закрытие соединения, ранее отправленного программе TCP, • 10 (closing) — ожидание подтверждения со стороны программы TCP запроса о закрытии соединения, • 11 (timeWait) — ожидание, когда программа TCP получила подтверждение своего запроса на закрытие соединения, • 12 (deleteTCB) — удаление буфера управления передачей (TCB) для подключения TCP.
.1.3.6.1.2.1.6.14.0	Количество сегментов с ошибками.
.1.3.6.1.2.1.6.15.0	Количество отправленных сегментов с флагом RST.
Таблица TCP-соединений для протоколов IPv4 и IPv6 (.1.3.6.1.2.1.6.19.1): V1 — версия протокола IP для локального адреса, A — локальный IP-адрес, B — локальный порт, V2 — версия протокола IP для удаленного адреса, C — удаленный IP-адрес, D — удаленный порт	
.1.3.6.1.2.1.6.19.1.7.V1.4.A.B.V2.C.D	Состояние TCP-соединения: • 1 (closed) — закрыто, • 2 (listen) — ожидание запроса на соединение со стороны чужих портов и программ TCP, • 3 (synSent) — ожидание парного запроса на установление соединения, • 4 (synReceived) — ожидание подтверждения после

OID	Описание
	того, как запрос соединения уже принят и отправлен,
	• 5 (established) — установлено, • 6 (finWait1) — ожидание запроса от программы TCP или подтверждения ранее отправленного запроса на закрытие соединения, • 7 (finWait2) — ожидание запроса на закрытие соединения со стороны программы TCP, • 8 (closeWait) — ожидание запроса на закрытие соединения со стороны своего клиента, • 9 (lastAck) — ожидание запроса на закрытие соединения, ранее отправленного программе TCP, • 10 (closing) — ожидание подтверждения со стороны программы TCP запроса о закрытии соединения, • 11 (timeWait) — ожидание, когда программа TCP получила подтверждение своего запроса на закрытие соединения, • 12 (deleteTCB) — удаление буфера управления передачей (TCB) для подключения.
.1.3.6.1.2.1.6.19.1.8.V1.4.A.B.V2.C.D	Системный идентификатор процесса, связанного с этим соединением. Если такого процесса нет, параметр равен 0.
Таблица открытых TCP-портов (.1.3.6.1.2.1.6.20.1): V1 — версия протокола IP для локального адреса, A — локальный IP-адрес, B — локальный порт	
.1.3.6.1.2.1.6.20.1.4.V1.4.A.B	Системный идентификатор процесса, использующего порт. Если такого процесса нет, параметр равен 0.

UDP-MIB

UDP-MIB (.1.3.6.1.2.1.7) включает в себя объекты, предназначенные для получения информации о различных характеристиках работы сетевого узла по протоколу UDP (например, общее количество полученных и отправленных датаграмм). Подробнее см. в [RFC 4113](#).

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 37. SNMP-параметры. UDP-MIB

OID	Описание
.1.3.6.1.2.1.7.1.0	Количество входящих UDP-датаграмм.
.1.3.6.1.2.1.7.2.0	Количество входящих UDP-датаграмм, для которых нет открытого порта.

OID	Описание
.1.3.6.1.2.1.7.3.0	Количество входящих UDP-датаграмм, для которых нет открытого порта.
.1.3.6.1.2.1.7.4.0	Количество исходящих UDP-датаграмм.
Таблица открытых UDP-портов для протокола IPv4 (.1.3.6.1.2.1.7.5.1):	
A — IP-адрес локального порта,	
B — номер локального порта.	
.1.3.6.1.2.1.7.5.1.1.A.B	IP-адрес локального порта.
.1.3.6.1.2.1.7.5.1.2.A.B	Номер порта.
Таблица открытых UDP-соединений (.1.3.6.1.2.1.7.7.1):	
V1 — версия IP-протокола для локального адреса,	
A — IP-адрес локального порта,	
B — номер локального порта,	
V2 — версия IP-протокола для удаленного адреса,	
C — IP-адрес удаленного порта,	
D — номер удаленного порта.	
.1.3.6.1.2.1.7.7.1.8.V1.4.A.B.V2.C.D	Системный идентификатор процесса, использующего порт. Если такого процесса нет, параметр равен 0.

HOST-RESOURCES-MIB

HOST-RESOURCES-MIB (.1.3.6.1.2.1.25) включает в себя объекты, предназначенные для получения информации о системных параметрах и оборудовании сетевого узла. Подробнее см. [RFC 2790](#).

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 38. SNMP-параметры. HOST-RESOURCES-MIB

OID	Описание
.1.3.6.1.2.1.25.1.1.0	Время работы системы с последнего запуска.

OID	Описание
.1.3.6.1.2.1.25.1.2.0	Системное время на сетевом узле (значение передается в виде строки байт): • 1-2 байт — год, • 3 байт — месяц, • 4 байт — день, • 5 байт — часы, • 6 байт — минуты, • 7 байт — секунды, • 8 байт — десятые секунды, • 9 байт — «направление» от UTC (соответствующие коды символов ascii), • 10 байт — часы от UTC, • 11 байт — минуты от UTC. Если часовой пояс неизвестен, то последние 3 байта отсутствуют.
1.3.6.1.2.1.25.1.3.0	Индекс устройства из таблицы .1.3.6.1.2.1.25.3.2.1, с которого загружается сетевой узел.
.1.3.6.1.2.1.25.1.4.0	Параметры при загрузке сетевого узла (в том числе название аппаратной платформы — для координатора).
.1.3.6.1.2.1.25.1.5.0	Количество сессий пользователя.
.1.3.6.1.2.1.25.1.6.0	Количество процессов загруженных или выполняемых на сетевом узле.
.1.3.6.1.2.1.25.1.7.0	Максимальное поддерживаемое сетевым узлом количество процессов. Если количество не ограничено, параметр равен 0.
1.3.6.1.2.1.25.2.2.0	Объем RAM (в килобайтах).
Таблица локальных логических хранилищ сетевого узла (.1.3.6.1.2.1.25.2.3.1):	
A — индекс хранилища	
.1.3.6.1.2.1.25.2.3.1.1.A	Идентификатор, присвоенный сетевым узлом (A).
.1.3.6.1.2.1.25.2.3.1.2.A	OID, присвоенный данному типу хранилища: 1.3.6.1.2.1.25.2.1 + • 1 — устройство не может быть причислено к какой-либо другой категории, • 2 — RAM, • 3 — виртуальная память, временное хранилище, SWAP, • 4 — не извлекаемые НЖМД, • 5 — извлекаемые НЖМД.
.1.3.6.1.2.1.25.2.3.1.3.A	Описание хранилища.

OID	Описание
.1.3.6.1.2.1.25.2.3.1.4.A	Размер в байтах объектов (размеры секторов/блоков/буферов/пакетов), размещенных в данном хранилище.
.1.3.6.1.2.1.25.2.3.1.5.A	Общее количество секторов/блоков/буферов/пакетов в хранилище.
.1.3.6.1.2.1.25.2.3.1.6.A	Количество использованных секторов/блоков/буферов/пакетов в хранилище.
.1.3.6.1.2.1.25.2.3.1.7.A	Количество ошибок операций записи по причине недостатка места.

Таблица устройств сетевого узла (.1.3.6.1.2.1.25.3.2.1):

В — индекс устройства

.1.3.6.1.2.1.25.3.2.1.1.B	Идентификатор, присвоенный сетевым узлом (В).
.1.3.6.1.2.1.25.3.2.1.2.B	OID, присвоенный данному типу устройства: 1.3.6.1.2.1.25.3.1 + • 1 — применяется, если не может быть отнесено к другой группе, • 2 — применяется, если устройство не опознано, • 3 — CPU (ядра), • 4 — сетевой интерфейс, • 5 — принтер, • 6 — жесткий диск (HDD, SSD), • 10 — видеоустройство, • 11 — аудиоустройство, • 12 — сопроцессор, • 13 — клавиатура, • 14 — модем, • 15 — параллельный порт, • 16 — манипулятор типа мышь, • 17 — последовательный порт, • 18 — ленточное хранилище, • 19 — часы, • 20 — энергозависимая память, • 21 — энергонезависимая память.
.1.3.6.1.2.1.25.3.2.1.3.B	Описание устройства.
.1.3.6.1.2.1.25.3.2.1.4.B	ID продукта.

OID	Описание
.1.3.6.1.2.1.25.3.2.1.5.B	Текущий статус устройства: • 1 — статус неизвестен (unknown), • 2 — работает (running), • 3 — предупреждение (warning), • 4 — проверка (testing), • 5 — не работает (down).
.1.3.6.1.2.1.25.3.2.1.6.B	Счетчик количества ошибок на устройстве.
Таблица загрузки процессоров сетевого узла (.1.3.6.1.2.1.25.3.3.1):	
B — индекс устройства из таблицы .1.3.6.1.2.1.25.3.2.1	
.1.3.6.1.2.1.25.3.3.1.1.B	ID продукта.
.1.3.6.1.2.1.25.3.3.1.2.B	Средняя нагрузка на ядро за последнюю минуту.
Таблица сетевых устройств (.1.3.6.1.2.1.25.3.4.1):	
B — индекс устройства из таблицы .1.3.6.1.2.1.25.3.2.1	
.1.3.6.1.2.1.25.3.4.1.1.B	Индекс интерфейса из таблицы 1.3.6.1.2.1.2.2.1.1 IF-MIB.
Таблица дисков сетевого узла (.1.3.6.1.2.1.25.3.6.1):	
B — индекс устройства из таблицы .1.3.6.1.2.1.25.3.2.1	
.1.3.6.1.2.1.25.3.6.1.1.B	Доступ к диску: • 1 — чтение и запись, • 2 — чтение.
.1.3.6.1.2.1.25.3.6.1.2.B	Тип устройства: • 1 — применяется, если не может быть отнесено к другой группе, • 2 — неизвестен, • 3 — жесткий диск, • 4 — дискета, • 5 — оптический диск ROM, • 6 — оптический диск WORM, • 10 — оптический диск RW, • 11 — RAM диск.
.1.3.6.1.2.1.25.3.6.1.3.B	Возможность извлечения диска: • 1 — Разрешено, • 2 — Запрещено.
.1.3.6.1.2.1.25.3.6.1.4.B	Размер диска (в килобайтах). Если диск извлекаемый и он извлечен, параметр равен 0.
Таблица разделов сетевого узла (.1.3.6.1.2.1.25.3.7.1):	
B — индекс устройства из таблицы .1.3.6.1.2.1.25.3.2.1	
D — индекс устройства из таблицы .1.3.6.1.2.1.25.3.7.1	

OID	Описание
.1.3.6.1.2.1.25.3.7.1.1.B.D	Индекс устройства (D).
.1.3.6.1.2.1.25.3.7.1.2.B.D	Описание раздела.
.1.3.6.1.2.1.25.3.7.1.3.B.D	ID раздела в операционной системе — октетная строка.
.1.3.6.1.2.1.25.3.7.1.4.B.D	Размер раздела (в килобайтах).
.1.3.6.1.2.1.25.3.7.1.5.B.D	Файловая система, используемая в разделе. Если файловой системы нет, параметр равен 0.

Таблица файловых систем сетевого узла (.1.3.6.1.2.1.25.3.8.1):

E — индекс устройства из таблицы .1.3.6.1.2.1.25.3.7.1

.1.3.6.1.2.1.25.3.8.1.1.E	Индекс устройства (E).
.1.3.6.1.2.1.25.3.8.1.2.E	Корневая папка файловой системы (точка монтирования).
.1.3.6.1.2.1.25.3.8.1.3.E	Описание удаленного сервера, с которого смонтирована файловая система. Может включать корневую папку файловой системы (точка монтирования) на удаленном сервере. Для локальных файловых систем параметр содержит пустую строку.

OID	Описание
.1.3.6.1.2.1.25.3.8.1.4.E	OID, присвоенный данному типу файловой системы: 1.3.6.1.2.1.25.3.9 + • 1 — Применяется, если не может быть отнесена к другой группе, • 2 — Если тип файловой системы неизвестен, • 3 — BFFS, • 4 — System V, • 5 — DOS FAT, • 6 — OS/2 HPFS, • 7 — MHFS, • 8 — MFS, • 9 — NTFS, • 10 — VNode File System, • 11 — Journaled File System, • 12 — ISO 9660 File System for CD, • 13 — RockRidge File System for CD, • 14 — NFS File System, • 15 — Netware File System, • 16 — Andrew File System, • 17 — OSF DCE Distributed File System, • 18 — AppleShare File System, • 19 — RFS File System, • 20 — Data General DGCFS, • 21 — SVR4 Boot File System, • 22 — Windows FAT32, • 23 — Linux EXT2 File System.
.1.3.6.1.2.1.25.3.8.1.5.E	Права доступа к файловой системе: • 1 — запись, чтение, • 2 — чтение.
.1.3.6.1.2.1.25.3.8.1.6.E	Возможность запуска с файловой системы: • 1 — возможно, • 2 — невозможно.
.1.3.6.1.2.1.25.3.8.1.7.E	Указывает на связь файловой системы и хранилищ.
.1.3.6.1.2.1.25.3.8.1.8.E	Дата последнего полного резервного копирования файловой системы. Октетная строка. Если данные отсутствуют, параметр равен: 00 00 01 01 00 00 00 00.

OID	Описание
.1.3.6.1.2.1.25.3.8.1.9.E	Дата последнего частичного резервного копирования файловой системы. Октетная строка. Если данные отсутствуют, параметр равен: 00 00 01 01 00 00 00 00.
Таблица ПО, запущенного на сетевом узле (.1.3.6.1.2.1.25.4.2.1):	
F — индекс ПО	
.1.3.6.1.2.1.25.4.2.1.1.F	Индекс запущенного ПО (F).
.1.3.6.1.2.1.25.4.2.1.2.F	Наименование запущенного ПО.
.1.3.6.1.2.1.25.4.2.1.3.F	ID ПО производителя.
.1.3.6.1.2.1.25.4.2.1.4.F	Папка, из которой запущено ПО.
.1.3.6.1.2.1.25.4.2.1.5.F	Параметры, с которыми запущено ПО.
.1.3.6.1.2.1.25.4.2.1.6.F	Тип программного обеспечения: • 1 — неизвестно, • 2 — ОС, • 3 — драйвер устройства, • 4 — приложение.
.1.3.6.1.2.1.25.4.2.1.7.F	Текущее состояния ПО: • 1 — запущено (running), • 2 — работоспособно (runnable), • 3 — неработоспособно (notRunnable), • 4 — работает некорректно (invalid).
Таблица с потреблением каждого ПО ЦПУ и ОЗУ (.1.3.6.1.2.1.25.5.1.1):	
F — индекс ПО из таблицы .1.3.6.1.2.1.25.4.2.1	
.1.3.6.1.2.1.25.5.1.1.1.F	Количество процессорного времени, выделенного ПО, в сантисекундах (0,01 с).
.1.3.6.1.2.1.25.5.1.1.2.F	Количество занимаемой оперативной памяти (в килобайтах).

LM-SENSORS-MIB

LM-SENSORS-MIB (.1.3.6.1.4.1.2021) включает в себя объекты, предназначенные для получения информации от различных датчиков, которые установлены в оборудовании сетевого узла.

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 39. SNMP-параметры. LM-SENSORS-MIB

OID	Описание
Таблица температурных датчиков сетевого узла (.1.3.6.1.4.1.2021.13.16.2.1):	
A — индекс температурного датчика	
.1.3.6.1.4.1.2021.13.16.2.1.1.A	Индекс температурного датчика (A).
.1.3.6.1.4.1.2021.13.16.2.1.2.A	Имя температурного датчика.
.1.3.6.1.4.1.2021.13.16.2.1.3.A	Температура (в м°С).
Таблица датчиков скорости вращения вентиляторов (.1.3.6.1.4.1.2021.13.16.3.1):	
B — индекс датчика скорости вращения вентилятора	
.3.6.1.4.1.2021.13.16.3.1.1.B	Индекс датчика скорости вращения вентилятора (B).
.3.6.1.4.1.2021.13.16.3.1.2.B	Имя датчика скорости вращения вентилятора.
.3.6.1.4.1.2021.13.16.3.1.3.B	Скорость вращения вентилятора (в оборотах в минуту). Если вентилятор не подключен, параметр равен 0.

Таблица 40. Температурные датчики

Платформа	Имя датчика	Описание	Предельное значение
HW10 F1	gpu_thermal-virtual-0: temp1	Температура графического ядра	+115 °C (+115 000 м°С)
	soc_thermal-virtual-0: temp1	Температура ядра процессора	+115 °C (+115 000 м°С)
HW50 N1, N2, N3	Core 0	Температура ядра процессора	+110 °C (+110 000 м°С)
HW50 N4	Core 0	Температура ядер процессора	+90 °C (+90 000 м°С)
	Core 2		
HW50 A1	Core0	Температура ядер процессора	+110 °C (+110 000 м°С)
	Core1		
	Core2		
	Core3		
HW100 N1, N2, N3	Core 0	Температура ядер процессора	+105 °C (+105 000 м°С)
	Core 1		
HW100 Q1, Q2	Physical id 0	Температура процессора	+75 °C (+75 000 м°С)
	Core 6	Температура ядер процессора	+75 °C (+75 000 м°С)
	Core 12		
HW1000 Q4	Core 0	Температура ядер процессора	+80 °C (+80 000 м°С)
	Core 1		
	Physical id 0	Температура процессора	+80 °C (+80 000 м°С)

Платформа	Имя датчика	Описание	Предельное значение
HW1000 Q5, Q6	temp1	Температура модуля памяти	+105 °C (+105 000 м°C)
	Core 0	Температура ядер процессора	+80 °C (+80 000 м°C)
	Core 1		
	Physical id 0	Температура процессора	+80 °C (+80 000 м°C)
HW1000 Q7, Q8, Q9	jc42-i2c-*.temp1	Температура модуля памяти	+105 °C (+105 000 м°C)
	Core 0	Температура ядер процессора	+80 °C (+80 000 м°C)
	Core 1		
	Core 2		
	Core 3		
	Physical id 0	Температура процессора	+80 °C (+80 000 м°C)
HW1000 Q10	temp1	Температура модуля памяти	+119 °C (+119 000 м°C)
	Core 0	Температура ядер процессора	+73 °C (+73 000 м°C)
	Core 2		
	Core 4		
	Core 6		
	Core 8		
	Core 10		
	Core 12		
	Core 14		
	Physical id 0	Температура процессора	+73 °C (+73 000 м°C)
	SYSTIN	Температура материнской платы	+80 °C (+80 000 м°C)
	CPUTIN	Температура центрального процессора	+80 °C (+80 000 м°C)
HW2000 Q4	Core 0	Температура ядер процессора 1	+71 °C (+71 000 м°C)
	Core 1		
	Core 2		
	Core 3		
	Core 4		
	Core 5		
	Physical id 0	Температура процессора 1	+71 °C (+71 000 м°C)
	coretemp-isa-0001: Core 0	Температура ядер процессора 2	+71 °C (+71 000 м°C)

Платформа	Имя датчика	Описание	Предельное значение
HW2000 Q5	coretemp-isa-0001: Core 1		
	coretemp-isa-0001: Core 2		
	coretemp-isa-0001: Core 3		
	coretemp-isa-0001: Core 4		
	coretemp-isa-0001: Core 5		
	Physical id 1	Температура процессора 2	+71 °C (+71 000 м°C)
	Core 0	Температура ядер процессора	+80 °C (+80 000 м°C)
	Core 1		
	Core 2		
	Core 3		
HW5000 Q1	Core 4		
	Core 5		
	Physical id 0	Температура процессора	+80 °C (+80 000 м°C)
	Core 0	Температура ядер процессора 1	+75 °C (+75 000 м°C)
	Core 1		
	Core 2		
	Core 3		
	Core 4		
	Core 5		
	Physical id 0	Температура процессора 1	+75 °C (+75 000 м°C)
HW5000 Q2	coretemp-isa-0001: Core 0	Температура ядер процессора 2	+75 °C (+75 000 м°C)
	coretemp-isa-0001: Core 1		
	coretemp-isa-0001: Core 2		
	coretemp-isa-0001: Core 3		
	coretemp-isa-0001: Core 4		
	coretemp-isa-0001: Core 5		
	Physical id 1	Температура процессора 2	+75 °C (+75 000 м°C)
	Core 0	Температура ядер процессора	+78 °C (+78 000 м°C)
	Core 1		
	Core 2		
	Core 3		

Платформа	Имя датчика	Описание	Предельное значение
	Core 4		
	Core 5		
	Core 6		
	Core 7		
	Physical id 0	Температура процессора	+78 °C (+78 000 м°C)



Внимание! Рекомендуемые значения скорости вращения вентиляторов находятся в промежутке 5 000 - 9 000 оборотов в минуту. Если значение скорости выше 15 000 или ниже 1 000 оборотов в минуту обратитесь в службу поддержки ИнфоТеКС.

Имена датчиков скорости могут отличаться в зависимости от ревизии аппаратной платформы.

Таблица 41. Датчики скорости вращения вентиляторов

Платформа	Количество датчиков	Имя
HW1000 Q4, Q5, Q6	2	fan3 fan4
HW1000 Q7, Q8, Q9	3	FRNT_FAN1 FRNT_FAN2 FRNT_FAN3
HW1000 Q10	2	SYS_FAN1 SYS_FAN2
HW2000 Q4	6	fan3 fan6 fan9 nct7904-i2c-0-2d:fan1, nct7904-i2c-0-2d:fan2, nct7904-i2c-0-2d:fan4
HW2000 Q5	3	FRNT_FAN1 FRNT_FAN2 FRNT_FAN3

Платформа	Количество датчиков	Имя
HW5000 Q1	6	fan3
		fan6
		fan9
		nct7904-i2c-0-2d:fan1,
		nct7904-i2c-0-2d:fan2,
		nct7904-i2c-0-2d:fan4
HW5000 Q2	3	FRNT_FAN1
		FRNT_FAN2
		FRNT_FAN3

NOTIFICATION-LOG-MIB

NOTIFICATION-LOG-MIB (.1.3.6.1.2.1.92) включает в себя объекты, предназначенные для получения информации о журналах событий. Подробнее см. в [RFC 3014](#).

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 42. SNMP-параметры. NOTIFICATION-LOG-MIB

OID	Описание
.1.3.6.1.2.1.92.1.1.1.0	Максимальное количество записей, которое может хранить таблица журналов .1.3.6.1.2.1.92.1.3.1.1 (таблица не поддерживается в ViPNet Coordinator HW). Если ограничений нет, параметр равен 0.
.1.3.6.1.2.1.92.1.1.2.0	Время, в течение которого SNMP-агент должен хранить уведомления (в минутах). Если время не ограничено, параметр равен 0.
.1.3.6.1.2.1.92.1.2.1.0	Количество записей в таблице журналов .1.3.6.1.2.1.92.1.3.1.1 (таблица не поддерживается ViPNet Coordinator HW).
.1.3.6.1.2.1.92.1.2.2.0	Количество записей, отброшенных из-за необходимости освобождения места под новые, или по причине превышения значений параметров .1.3.6.1.2.1.92.1.1.1.0 или 1.3.6.1.2.1.92.1.1.3.1.3. Не включает в себя количество записей, удаленных в связи с истечением времени .1.3.6.1.2.1.92.1.1.2.0.

UCD-SNMP-MIB

UCD-SNMP-MIB (.1.3.6.1.4.1.2021) включает в себя объекты, предназначенные для получения информации о характеристиках и состоянии оборудования сетевого узла (например, состояние оперативной памяти и жесткого диска). Подробнее см. на [веб-ресурсе](#).

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 43. SNMP-параметры. UCD-SNMP-MIB

OID	Описание
.1.3.6.1.4.1.2021.4.1.0	Фиктивный индекс, всегда равен 0.
.1.3.6.1.4.1.2021.4.2.0	Фиктивное имя, всегда равно объему виртуальной памяти.
.1.3.6.1.4.1.2021.4.3.0	Объем виртуальной памяти (swar) на сетевом узле (в кибибайтах).
.1.3.6.1.4.1.2021.4.4.0	Объем доступной виртуальной памяти (в кибибайтах).
.1.3.6.1.4.1.2021.4.5.0	Объем физической памяти (RAM) на сетевом узле (в кибибайтах).
.1.3.6.1.4.1.2021.4.6.0	Объем доступной физической памяти (в кибибайтах).
.1.3.6.1.4.1.2021.4.7.0	Объем виртуальной памяти, выделенный для текстовых страниц (в кибибайтах).
.1.3.6.1.4.1.2021.4.9.0	Объем физической памяти (RAM), выделенный для текстовых страниц (в кибибайтах). Не выводится, если операционная система не выделяет текстовые страницы от других используемых в RAM.
.1.3.6.1.4.1.2021.4.11.0	Общий объем доступной памяти (RAM + swar) (в кибибайтах).
.1.3.6.1.4.1.2021.4.12.0	Минимальный объем свободной виртуальной памяти (в кибибайтах).
.1.3.6.1.4.1.2021.4.13.0	Общий объем памяти (в кибибайтах), выделенный под разделяемую память (shared memory).
.1.3.6.1.4.1.2021.4.14.0	Общий объем памяти (в кибибайтах), который выделен для использования в качестве буфера обмена.
.1.3.6.1.4.1.2021.4.15.0	Общий объем памяти (в кибибайтах), который выделен для использования в качестве кеш ированной памяти.
.1.3.6.1.4.1.2021.4.16.0	Объем виртуальной памяти, занятый текстовыми страницами (в кибибайтах).
.1.3.6.1.4.1.2021.4.17.0	Объем физической памяти (RAM), занятый текстовыми страницами (в кибибайтах). Не выводится если операционная система не выделяет текстовые страницы от других используемых в RAM.
.1.3.6.1.4.1.2021.4.18.0	Объем виртуальной памяти на сетевом узле (в кибибайтах). 64-битный аналог .1.3.6.1.4.1.2021.4.3.0.

OID	Описание
.1.3.6.1.4.1.2021.4.19.0	Объем доступной виртуальной памяти (в кибибайтах). 64-битный аналог .1.3.6.1.4.1.2021.4.4.0.
.1.3.6.1.4.1.2021.4.20.0	Объем физической памяти на сетевом узле (в кибибайтах). 64-битный аналог .1.3.6.1.4.1.2021.4.5.0.
.1.3.6.1.4.1.2021.4.21.0	Объем свободной физической памяти (в кибибайтах). 64-битный аналог .1.3.6.1.4.1.2021.4.6.0.
.1.3.6.1.4.1.2021.4.22.0	Общий объем доступной памяти (RAM + swap) (в кибибайтах). 64-битный аналог .1.3.6.1.4.1.2021.4.11.0.
.1.3.6.1.4.1.2021.4.23.0	Минимальный объем свободной виртуальной памяти (в кибибайтах). 64-битный аналог .1.3.6.1.4.1.2021.4.12.0.
.1.3.6.1.4.1.2021.4.24.0	Общий объем памяти (в кибибайтах), выделенный под разделяемую память (shared memory). 64-битный аналог .1.3.6.1.4.1.2021.4.13.0.
.1.3.6.1.4.1.2021.4.25.0	Общий объем памяти (в кибибайтах), который выделен для использования в качестве буфера обмена. 64-битный аналог .1.3.6.1.4.1.2021.4.14.0.
.1.3.6.1.4.1.2021.4.26.0	Общий объем памяти (в кибибайтах), который выделен для использования в качестве кеш ированной памяти. 64-битный аналог .1.3.6.1.4.1.2021.4.15.0.
.1.3.6.1.4.1.2021.4.27.0	Объем физической памяти в кибибайтах (RAM), который доступен для запуска приложений без вымещения данных в swap.
.1.3.6.1.4.1.2021.4.100.0	Признак, что объем виртуальной памяти меньше минимально установленного значения.
.1.3.6.1.4.1.2021.4.101.0	Сообщение о том, что доступный объем виртуальной памяти меньше минимально установленного значения.

Таблица сведений о разделах (.1.3.6.1.4.1.2021.9.1)

В — индекс диска, присвоенный сетевым узлом

.1.3.6.1.4.1.2021.9.1.1.B	Индекс раздела (B).
.1.3.6.1.4.1.2021.9.1.2.B	Точка монтирования раздела.
.1.3.6.1.4.1.2021.9.1.3.B	Устройство (или tmpfs), на котором находится раздел.
.1.3.6.1.4.1.2021.9.1.4.B	Минимальный объем свободного места на разделе (в килобайтах) до появления флага ошибки.
.1.3.6.1.4.1.2021.9.1.5.B	Минимальный объем свободного места на разделе (в процентах) от общего до появления флага ошибки.
.1.3.6.1.4.1.2021.9.1.6.B	Общий объём раздела (в килобайтах).
.1.3.6.1.4.1.2021.9.1.7.B	Свободное место в разделе (в килобайтах).
.1.3.6.1.4.1.2021.9.1.8.B	Занятое место в разделе (в килобайтах).
.1.3.6.1.4.1.2021.9.1.9.B	Занятое место в разделе (в процентах).

OID	Описание
.1.3.6.1.4.1.2021.9.1.10.B	Процент использования инодов в разделе.
.1.3.6.1.4.1.2021.9.1.100.B	Флаг ошибки, сигнализирующий о том, что свободного места в разделе осталось меньше заданного значения .1.3.6.1.4.1.2021.9.1.4.B или .1.3.6.1.4.1.2021.9.1.5.B: 0 — нет ошибки, 1 — есть ошибка.
.1.3.6.1.4.1.2021.9.1.101.B	Сообщение об ошибке. Если параметр .1.3.6.1.4.1.2021.9.1.10.B равен 0, сообщение пустое.
.1.3.6.1.4.1.2021.9.1.11.B	Общий объем раздела, представленный в формате 64 бит. Параметр равен 1-32 бит числа. Если раздел менее 2 Тбайт, параметр равен .1.3.6.1.4.1.2021.9.1.6.B.
.1.3.6.1.4.1.2021.9.1.12.B	Общий объем раздела, представленный в формате 64 бит. Параметр равен 33-64 бит числа. Если раздел менее 2 Тбайт, параметр равен 0.
.1.3.6.1.4.1.2021.9.1.13.B	Общий объем свободного места раздела, представленный в формате 64 бит. Параметр равен 1-32 бит числа. Если свободно менее 2 Тбайт, параметр равен .1.3.6.1.4.1.2021.9.1.7.B.
.1.3.6.1.4.1.2021.9.1.14.B	Общий объем свободного места раздела, представленный в формате 64 бит. Параметр равен 33-64 бит числа. Если свободно менее 2 Тбайт, параметр равен 0.
.1.3.6.1.4.1.2021.9.1.15.B	Общий объем занятого места раздела, представленный в формате 64 бит. Параметр равен 1-32 бит числа. Если занято менее 2 Тбайт, параметр равен .1.3.6.1.4.1.2021.9.1.8.B
.1.3.6.1.4.1.2021.9.1.16.B	Общий объем занятого места раздела, представленный в формате 64 бит. Параметр равен 33-64 бит числа. Если занято менее 2 Тбайт, параметр равен 0.

Таблица информации о средней загрузке системы за 1, 5, 15 минут (1.3.6.1.4.1.2021.10.1)

C — индекс, присвоенный сетевым узлом (постоянный):

- 1 — значение для 1 минуты,
- 2 — значение для 5 минут,
- 3 — значение для 15 минут

.1.3.6.1.4.1.2021.10.1.1.C	Индекс значения средней загрузки: • 1 — 1 минута, • 2 — 5 минут, • 3 — 15 минут
.1.3.6.1.4.1.2021.10.1.2.C	Имя загрузки (по умолчанию: Load-1, Load-5, Load-15).
.1.3.6.1.4.1.2021.10.1.3.C	Средняя загрузка системы в процентах (в виде строки).
.1.3.6.1.4.1.2021.10.1.4.C	Пороговое значение загрузки системы в процентах (в виде строки).
.1.3.6.1.4.1.2021.10.1.5.C	Загрузка системы в процентах (в виде целого числа).

OID	Описание
.1.3.6.1.4.1.2021.10.1.6.C	Загрузка системы в процентах (в виде числа с плавающей точкой).
.1.3.6.1.4.1.2021.10.1.100.C	Флаг, сигнализирующий о превышении средней загрузки порогового значения.
.1.3.6.1.4.1.2021.10.1.101.C	Сообщение о превышении порогового значения.
.1.3.6.1.4.1.2021.10.1.0	Фиктивный индекс, всегда равен 1.
.1.3.6.1.4.1.2021.10.2.0	Фиктивное имя, всегда равно systemStats.
.1.3.6.1.4.1.2021.11.3.0	Среднее количество информации (в килобайтах), записанное на диск за минуту.
.1.3.6.1.4.1.2021.11.4.0	Среднее количество информации (в килобайтах), прочитанное с диска за минуту.
.1.3.6.1.4.1.2021.11.5.0	Среднее количество информации (в блоках данных), записанных на диск за минуту.
.1.3.6.1.4.1.2021.11.6.0	Среднее количество информации (в блоках данных), прочитанное с диска за минуту.
.1.3.6.1.4.1.2021.11.7.0	Среднее количество системных прерываний за минуту.
.1.3.6.1.4.1.2021.11.8.0	Среднее количество переключений контекста за минуту.
.1.3.6.1.4.1.2021.11.9.0	Процент времени, занятого пользовательскими процессами (за минуту).
.1.3.6.1.4.1.2021.11.10.0	Процент времени, когда процессор занят системными процессами (за минуту).
.1.3.6.1.4.1.2021.11.11.0	Процент времени, когда процессор не занят процессами (за минуту).
.1.3.6.1.4.1.2021.11.50.0	Количество тиков процессорного времени, затраченного на исполнения user-level code.
.1.3.6.1.4.1.2021.11.51.0	Количество тиков процессорного времени, затраченного на исполнение reduced-priority code.
.1.3.6.1.4.1.2021.11.52.0	Количество тиков процессорного времени, затраченного на исполнение system-level code.
.1.3.6.1.4.1.2021.11.53.0	Количество тиков процессорного времени, проведенного в состоянии idle.
.1.3.6.1.4.1.2021.11.54.0	Количество тиков процессорного времени, затраченного на исполнение ожидания ввода-вывода.
.1.3.6.1.4.1.2021.11.55.0	Количество тиков процессорного времени, затраченного на исполнение kernel-level code.
.1.3.6.1.4.1.2021.11.56.0	Количество тиков процессорного времени, затраченного на исполнение аппаратных прерываний.
.1.3.6.1.4.1.2021.11.57.0	Количество блоков, отправленное блочному устройству (диску).

OID	Описание
.1.3.6.1.4.1.2021.11.58.0	Количество блоков, полученное блочным устройством (диском).
.1.3.6.1.4.1.2021.11.59.0	Количество обработанных аппаратных прерываний.
.1.3.6.1.4.1.2021.11.60.0	Количество выполненных переключений контекста.
.1.3.6.1.4.1.2021.11.61.0	Количество тиков процессорного времени, затраченного на исполнение программных прерываний.
.1.3.6.1.4.1.2021.11.62.0	Количество блоков, помещенных в виртуальную память.
.1.3.6.1.4.1.2021.11.63.0	Количество блоков, извлеченных из виртуальной памяти.
.1.3.6.1.4.1.2021.11.64.0	Количество тиков процессорного времени, затраченного кодом гипервизора (если узел работает в виртуальной среде).
.1.3.6.1.4.1.2021.11.65.0	Количество тиков процессорного времени, затраченного на виртуальные процессоры в гипервизоре.
.1.3.6.1.4.1.2021.11.66.0	Количество тиков процессорного времени, затраченного на виртуальные процессоры в гипервизоре с пониженным приоритетом.
.1.3.6.1.4.1.2021.11.67.0	Количество ядер процессоров.
.1.3.6.1.4.1.2021.100.1.0	Фиктивный индекс, всегда равен 0.
.1.3.6.1.4.1.2021.100.2.0	Ключевое слово CVS.
.1.3.6.1.4.1.2021.100.3.0	Строка времени из ключевого слова RCS.
.1.3.6.1.4.1.2021.100.4.0	Строка времени из ctime.
.1.3.6.1.4.1.2021.100.5.0	Идентификатор из ключевого слова RCS.
.1.3.6.1.4.1.2021.100.6.0	Опции конфигурации SNMP-агента.
.1.3.6.1.4.1.2021.101.1.0	Фиктивный индекс, всегда равен 0.
.1.3.6.1.4.1.2021.101.2.0	Фиктивная строка, всегда равная snmp.
.1.3.6.1.4.1.2021.101.100.0	Признак ошибки SNMP-агента.
.1.3.6.1.4.1.2021.101.101.0	Описание ошибки SNMP-агента.

UCD-DISKIO-MIB

UCD-DISKIO-MIB (.1.3.6.1.4.1.2021.13.15) включает в себя объекты, предназначенные для получения информации о состоянии памяти и дискового пространства сетевого узла. Подробнее см. на [веб-ресурсе](#).

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 44. SNMP-параметры. UCD-DISKIO-MIB

OID	Описание
Таблица значений текущих параметров оперативной памяти, виртуальной памяти, дискового пространства и их загрузки (.1.3.6.1.4.1.2021.13.15.1.1.)	
A — индекс таблицы	
.1.3.6.1.4.1.2021.13.15.1.1.1.A	Индекс таблицы (A).
.1.3.6.1.4.1.2021.13.15.1.1.2.A	Описание устройства, полученное от ОС. Для координатора принимает значения: ram0-ram15 loop0-loop7 sda, sda1, sda2, sdb1, sdb2, sdb3
.1.3.6.1.4.1.2021.13.15.1.1.3.A	Количество байт, считанных с устройства с момента запуска.
.1.3.6.1.4.1.2021.13.15.1.1.4.A	Количество байт, записанных на устройство с момента запуска.
.1.3.6.1.4.1.2021.13.15.1.1.5.A	Количество операций доступа к устройству на чтение.
.1.3.6.1.4.1.2021.13.15.1.1.6.A	Количество операций доступа к устройству на запись.
.1.3.6.1.4.1.2021.13.15.1.1.9.A	Средняя загрузка устройства за 1 минуту (в процентах).
.1.3.6.1.4.1.2021.13.15.1.1.10.A	Средняя загрузка устройства за 5 минут (в процентах).
.1.3.6.1.4.1.2021.13.15.1.1.11.A	Средняя загрузка устройства за 15 минут (в процентах).
.1.3.6.1.4.1.2021.13.15.1.1.12.A	Количество операций доступа к устройству на чтение. 64-разрядный параметр.
.1.3.6.1.4.1.2021.13.15.1.1.13.A	Количество байт, записанных на устройство с момента запуска. 64-разрядный параметр.
.1.3.6.1.4.1.2021.13.15.1.1.14.A	Время использования диска с момента запуска (в микросекундах).

NET-SNMP-MIB

NET-SNMP-MIB (.1.3.6.1.4.1.8072) включает в себя объекты, предназначенные для получения служебной информации о работе SNMP-агента.

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 45. SNMP-параметры. NET-SNMP-MIB

OID	Описание
Таблица, содержащая все OID, зарегистрированные MIB-модулями в SNMP-агенте (.1.3.6.1.4.1.8072.1.2.1.1):	

OID	Описание
A – имя MIB-модуля	
B – OID MIB-модуля	
C – приоритет MIB-модуля	
.1.3.6.1.4.1.8072.1.2.1.1.4.A.B.C	Имя модуля, который зарегистрировал OID.
.1.3.6.1.4.1.8072.1.2.1.1.5.A.B.C	Биты, означающие режим работы OID.
.1.3.6.1.4.1.8072.1.2.1.1.6.A.B.C	Таймаут для модуля.
Таблица зашифрованных расширений (.1.3.6.1.4.1.8072.1.3.2.2.1):	
D – идентификатор расширения	
.1.3.6.1.4.1.8072.1.3.2.2.1.2.D	Путь к бинарному файлу или скрипту для запуска.
.1.3.6.1.4.1.8072.1.3.2.2.1.3.D	Аргументы для команды.
.1.3.6.1.4.1.8072.1.3.2.2.1.4.D	Стандартные вводные для команды.
.1.3.6.1.4.1.8072.1.3.2.2.1.5.D	Продолжительность времени, в течение которого результат выполнения команды кешируется (секунд).
.1.3.6.1.4.1.8072.1.3.2.2.1.6.D	Механизм запуска команды: • 1 — fork-and-exec, • 2 — run via a sub-shell.
.1.3.6.1.4.1.8072.1.3.2.2.1.7.D	Механизм исполнения команды: • 1 — run-on-read, • 2 — run-on-set, • 3 — run-command.
.1.3.6.1.4.1.8072.1.3.2.2.1.20.D	Тип хранилища: • 1 (other) — другое, • 2 (volatile) — изменяемый, • 3 (nonvolatile) — неизменяемый, • 4 (permanent) — постоянный, • 5 (readOnly) — только для чтения.
.1.3.6.1.4.1.8072.1.3.2.2.1.21.D	Статус строки таблицы: • 1 (active) — строка доступна для использования, • 2 (notInService) — строка недоступна для использования, • 3 (notReady) — отсутствует информация для использования, • 4 (createAndGo) — создание строки и перевод ее в статус active, • 5 (createAndWait) — создание строки и перевод ее в статус notInService, • 6 (destroy) — удаление.

OID	Описание
Таблица результатов выполнения скриптов (.1.3.6.1.4.1.8072.1.3.2.3.1):	
D – идентификатор расширения	
.1.3.6.1.4.1.8072.1.3.2.3.1.1.D	Первая строка в результате запуска команды.
.1.3.6.1.4.1.8072.1.3.2.3.1.2.D	Полный результат запуска команды одной строки.
.1.3.6.1.4.1.8072.1.3.2.3.1.3.D	Количество строк в результате запуска команды.
.1.3.6.1.4.1.8072.1.3.2.3.1.4.D	Значение, возвращенное командой.
Таблица результатов выполнения команды построчно (.1.3.6.1.4.1.8072.1.3.2.4.1):	
D – идентификатор расширения	
E – номер строки (с единицы)	
.1.3.6.1.4.1.8072.1.3.2.4.1.D.E	Единичная строка вывода команды.
.1.3.6.1.4.1.8072.1.5.1.0	Время, на которое кеш ируется значение OID по умолчанию (в секундах), если оно не задано в параметре .1.3.6.1.4.1.8072.1.5.3.1.2.F.
.1.3.6.1.4.1.8072.1.5.2.0	Статус кеширования: • 1 — включено, • 2 — выключено.
Таблица кешированных OID (.1.3.6.1.4.1.8072.1.5.3.1):	
F – OID, значение которого кешировано	
.1.3.6.1.4.1.8072.1.5.3.1.2.F	Время, на которое кеш ируется значение OID (в секундах).
.1.3.6.1.4.1.8072.1.5.3.1.3.F	Статус записи: • 1 (enabled) — включена, • 2 (disabled) — выключена, • 3 (empty) — пустая, • 4 (cached) — кешированная, • 5 (expired) — истекла.
.1.3.6.1.4.1.8072.1.7.1.1.0	Статус режима debug SNMP-агента: • 1 — включен, • 2 — выключен.
.1.3.6.1.4.1.8072.1.7.1.2.0	Переключатель вывода для debug: • 1 — вывод всех значений, • 2 — фильтрация вывода. Для работы переключателя параметр .1.3.6.1.4.1.8072.1.7.1.1.0 должен быть равен 1.
.1.3.6.1.4.1.8072.1.7.1.3.0	Переключатель вывода сырого дампа пакетов: • 1 — включен, • 2 — выключен.

OID	Описание
Таблица результатов debug (.1.3.6.1.4.1.8072.1.7.1.4.1):	
G – префикс идентификатора	
.1.3.6.1.4.1.8072.1.7.1.4.4.G	Статус строки таблицы: • 1 (active) — строка доступна для использования, • 2 (notInService) — строка недоступна для использования, • 3 (notReady) — отсутствует информация для использования, • 4 (createAndGo) — создание строки и перевод ее в статус active, • 5 (createAndWait) — создание строки и перевод ее в статус notInService, • 6 (destroy) — удаление.
Таблица форм вывода сообщений SNMP-агента (.1.3.6.1.4.1.8072.1.7.2.1.1):	
H – уровень важности событий:	
0 – emergency	
1 – alert	
2 – critical	
3 – error	
4 – warning	
5 – notice	
6 – info	
7 – debug	
I – идентификатор журнала	
.1.3.6.1.4.1.8072.1.7.2.1.1.3.H.I	Форма вывода сообщений для данной строки: • 1 — stdout, • 2 — stderr, • 3 — file, • 4 — syslog, • 5 — callback.
.1.3.6.1.4.1.8072.1.7.2.1.1.4.H.I	Максимальное значение, для которого применима данная строка: • 0 — emergency, • 1 — alert, • 2 — critical, • 3 — error, • 4 — warning, • 5 — notice, • 7 — debug.

OID	Описание
.1.3.6.1.4.1.8072.1.7.2.1.1.5.H.I	Статус строки таблицы: • 1 (active) — строка доступна для использования, • 2 (notInService) — строка недоступна для использования, • 3 (notReady) — отсутствует информация для использования, • 4 (createAndGo) — создание строки и перевод ее в статус active, • 5 (createAndWait) — создание строки и перевод ее в статус notInService, • 6 (destroy) — удаление.
VACM таблица доступа сетевого узла (.1.3.6.1.4.1.8072.1.9.1.1.): J – имя группы: K – префикс контекста доступа L – модель безопасности доступа M – уровень безопасности доступа N – способ аутентификации	
.1.3.6.1.4.1.8072.1.9.1.1.2.J.K.L.M.N	Механизм выбора совпадений в VACM: • 1 — exact (полное совпадение), • 2 — prefix (совпадает префикс).
.1.3.6.1.4.1.8072.1.9.1.1.3.J.K.L.M.N	Значение секции ViewName.
.1.3.6.1.4.1.8072.1.9.1.1.4.J.K.L.M.N	Тип хранилища: • 1 (other) — другое, • 2 (volatile) — изменяемый, • 3 (nonvolatile) — неизменяемый, • 4 (permanent) — постоянный, • 5 (readOnly) — только для чтения.
.1.3.6.1.4.1.8072.1.9.1.1.5.J.K.L.M.N	Статус строки таблицы: • 1 (active) — строка доступна для использования, • 2 (notInService) — строка недоступна для использования, • 3 (notReady) — отсутствует информация для использования, • 4 (createAndGo) — создание строки и перевод ее в статус active, • 5 (createAndWait) — создание строки и перевод ее в статус notInService, • 6 (destroy) — удаление.

SNMP-MPD-MIB

SNMP-MPD-MIB (.1.3.6.1.6.3.11) включает в себя статистическую информацию об SNMP-пакетах.

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 46. SNMP-параметры. SNMP-MPD-MIB

OID	Описание
.1.3.6.1.6.3.11.2.1.1.0	Количество полученных SNMP-пакетов, заблокированных по причине неподдерживаемой или неизвестной модели безопасности.
.1.3.6.1.6.3.11.2.1.2.0	Количество полученных SNMP-пакетов, заблокированных по причине ошибок в SNMP-сообщении.
.1.3.6.1.6.3.11.2.1.3.0	Количество полученных SNMP-пакетов, заблокированных по причине несоответствия pduType.

SNMP-TARGET-MIB

SNMP-TARGET-MIB (.1.3.6.1.6.3.12) включает в себя статистическую информацию об SNMP-пакетах.

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 47. SNMP-параметры. SNMP-TARGET-MIB

OID	Описание
.1.3.6.1.6.3.12.1.1.0	Параметр для совместной работы нескольких SNMP-менеджеров, для одновременной работы с таблицами данного MIB.
.1.3.6.1.6.3.12.1.4.0	Количество полученных SNMP-пакетов, заблокированных по причине недоступного контекста.
.1.3.6.1.6.3.12.1.5.0	Количество полученных SNMP-пакетов, заблокированных по причине неизвестного контекста.

SNMP-USER-BASED-SM-MIB

SNMP-USER-BASED-SM-MIB (.1.3.6.1.6.3.15) включает в себя объекты, предназначенные для получения информации о параметрах и событиях безопасности SNMP-агента.

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 48. SNMP-параметры. SNMP-USER-BASED-SM-MIB

OID	Описание
.1.3.6.1.6.3.15.1.1.1.0	Количество заблокированных SNMP-пакетов по причине запроса неизвестного или недоступного уровня безопасности.
.1.3.6.1.6.3.15.1.1.2.0	Количество заблокированных SNMP-пакетов по причине появления вне определённого (доверенного) SNMP-агентом окна.
.1.3.6.1.6.3.15.1.1.3.0	Количество заблокированных SNMP-пакетов по причине использования неизвестного SNMP-агенту имени пользователя.
.1.3.6.1.6.3.15.1.1.4.0	Количество заблокированных SNMP-пакетов по причине обращения к неизвестному SNMP-агенту snmpEngineID.
.1.3.6.1.6.3.15.1.1.5.0	Количество заблокированных SNMP-пакетов по причине отсутствия ожидаемого значения digest value.
.1.3.6.1.6.3.15.1.1.6.0	Количество заблокированных SNMP-пакетов по причине невозможности их расшифровки.
.1.3.6.1.6.3.15.1.2.1.0	Параметр для совместной работы нескольких SNMP-менеджеров с таблицей .1.3.6.1.6.3.15.1.2.2.1.

Таблица параметров безопасности пользователя (.1.3.6.1.6.3.15.1.2.2.1):

A – EngineID пользователя

B – имя пользователя

.1.3.6.1.6.3.15.1.2.2.1.3.A.B	Имя пользователя в формате модели безопасности.
.1.3.6.1.6.3.15.1.2.2.1.4.A.B	Ссылка на другую строку таблицы, в которой создана копия пользователя из текущей строки.
.1.3.6.1.6.3.15.1.2.2.1.5.A.B	Тип протокола аутентификации пользователя.
.1.3.6.1.6.3.15.1.2.2.1.8.A.B	Тип протокола конфиденциальности пользователя.
.1.3.6.1.6.3.15.1.2.2.1.11.A.B	Параметр для определения факта смены ключей пользователя.
.1.3.6.1.6.3.15.1.2.2.1.12.A.B	Тип хранилища: • 1 (other) — другое, • 2 (volatile) — изменяемый, • 3 (nonvolatile) — неизменяемый, • 4 (permanent) — постоянный, • 5 (readOnly) — только для чтения.

OID	Описание
.1.3.6.1.6.3.15.1.2.2.1.13.A.B	Статус строки таблицы: • 1 (active) — строка доступна для использования, • 2 (notInService) — строка недоступна для использования, • 3 (notReady) — отсутствует информация для использования, • 4 (createAndGo) — создание строки и перевод ее в статус active, • 5 (createAndWait) — создание строки и перевод ее в статус notInService, • 6 (destroy) — удаление.

SNMP-VIEW-BASED-ACM-MIB

SNMP-VIEW-BASED-ACM-MIB (.1.3.6.1.6.3.16) включает в себя объекты, предназначенные для получения информации о правах доступа к SNMP-параметрам.

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 49. SNMP-параметры. SNMP-VIEW-BASED-ACM-MIB

OID	Описание
Таблица доступных контекстов сетевого узла (.1.3.6.1.6.3.16.1.1):	
A – строка контекста	
.1.3.6.1.6.3.16.1.1.1.A	Строка контекста.
Таблица групп доступа (.1.3.6.1.6.3.16.1.2.1):	
B – модель безопасности	
C – имя безопасности	
.1.3.6.1.6.3.16.1.2.1.3.B.C	Имя группы
.1.3.6.1.6.3.16.1.2.1.4.B.C	Тип хранилища: • 1 (other) — другое, • 2 (volatile) — изменяемый, • 3 (nonvolatile) — неизменяемый, • 4 (permanent) — постоянный, • 5 (readOnly) — только для чтения.

OID	Описание
.1.3.6.1.6.3.16.1.2.1.5.B.C	Статус строки таблицы: • 1 (active) — строка доступна для использования, • 2 (notInService) — строка недоступна для использования, • 3 (notReady) — отсутствует информация для использования, • 4 (createAndGo) — создание строки и перевод ее в статус active, • 5 (createAndWait) — создание строки и перевод ее в статус notInService, • 6 (destroy) — удаление.
Таблица прав доступа к группам (.1.3.6.1.6.3.16.1.4.1):	
D – имя группы	
E – контекст доступа	
F – модель безопасности	
G – уровень безопасности	
.1.3.6.1.6.3.16.1.4.1.4.D.E.F.G	Формат совпадения: • 1 (exact) — полное совпадение, • 2 (prefix) — совпадает префикс.
.1.3.6.1.6.3.16.1.4.1.5.D.E.F.G	Имя view на просмотр.
.1.3.6.1.6.3.16.1.4.1.6.D.E.F.G	Имя view на запись.
.1.3.6.1.6.3.16.1.4.1.7.D.E.F.G	Имя view для уведомлений.
.1.3.6.1.6.3.16.1.4.1.8.D.E.F.G	Тип хранилища: • 1 (other) — другое, • 2 (volatile) — изменяемый, • 3 (nonvolatile) — неизменяемый, • 4 (permanent) — постоянный, • 5 (readOnly) — только для чтения.
.1.3.6.1.6.3.16.1.4.1.9.D.E.F.G	Статус строки таблицы: • 1 (active) — строка доступна для использования, • 2 (notInService) — строка недоступна для использования, • 3 (notReady) — отсутствует информация для использования, • 4 (createAndGo) — создание строки и перевод ее в статус active, • 5 (createAndWait) — создание строки и перевод ее в статус notInService, • 6 (destroy) — удаление.
.1.3.6.1.6.3.16.1.5.1.0	Параметр для совместной работы нескольких SNMP-менеджеров с таблицами данного MIB.

OID	Описание
Таблица с параметрами view для семей поддеревьев (.1.3.6.1.6.3.16.1.5.2.1.):	
H – имя view	
I – поддерево	
.1.3.6.1.6.3.16.1.5.2.1.3.H.I	Маска, заданная для view.
.1.3.6.1.6.3.16.1.5.2.1.4.H.I	Тип view: • 1 — included, • 2 — excluded.
.1.3.6.1.6.3.16.1.5.2.1.5.H.I	Тип хранилища: • 1 (other) — другое, • 2 (volatile) — изменяемый, • 3 (nonvolatile) — неизменяемый, • 4 (permanent) — постоянный, • 5 (readOnly) — только для чтения.
.1.3.6.1.6.3.16.1.5.2.1.6.H.I	Статус строки таблицы: • 1 (active) — строка доступна для использования, • 2 (notInService) — строка недоступна для использования, • 3 (notReady) — отсутствует информация для использования, • 4 (createAndGo) — создание строки и перевод ее в статус active, • 5 (createAndWait) — создание строки и перевод ее в статус notInService, • 6 (destroy) — удаление.

SNMP-FRAMEWORK-MIB

SNMP-FRAMEWORK-MIB (.1.3.6.1.6.3.10) включает в себя объекты, предназначенные для получения служебной информации о работе SNMP-агента. Подробнее см. на [веб-ресурсе](#).

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 50. SNMP-параметры. SNMP-FRAMEWORK-MIB

OID	Описание
.1.3.6.1.6.3.10.2.1.1.0	Идентификатор SNMP-engine.
.1.3.6.1.6.3.10.2.1.2.0	Количество инициализаций SNMP-engine после создания идентификатора.
.1.3.6.1.6.3.10.2.1.3.0	Время, прошедшее с последнего изменения количества инициализаций SNMP-engine (в секундах).

OID	Описание
.1.3.6.1.6.3.10.2.1.4.0	Максимальная длина входящего и исходящего SNMP-сообщения (в байтах).

VIPNET-MIB

VIPNET-MIB (.1.3.6.1.4.1.10812) включает в себя объекты, предназначенные для получения информации о параметрах сетевых узлов ViPNet. Для получения этой информации используется ветвь: `.iso.org.dod.internet.private.enterprises.infotecs.vipnet` (.1.3.6.1.4.1.10812.1).

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 51. SNMP-параметры. VIPNET-MIB

OID	Описание
.1.3.6.1.4.1.10812.1.1.1	Шестнадцатеричный идентификатор сети ViPNet.
.1.3.6.1.4.1.10812.1.1.2	Шестнадцатеричный идентификатор сетевого узла ViPNet.
.1.3.6.1.4.1.10812.1.1.3	Имя пользователя узла ViPNet.
.1.3.6.1.4.1.10812.1.1.4	Время работы ViPNet Coordinator HW (сбрасывается каждый раз при перезапуске службы <code>iplircfg</code>).
.1.3.6.1.4.1.10812.1.1.5	Шестнадцатеричный идентификатор, включающий в себя идентификаторы узла ViPNet и сети ViPNet.
.1.3.6.1.4.1.10812.1.1.7	Имя узла ViPNet.
.1.3.6.1.4.1.10812.1.2.1	Количество сетевых интерфейсов.

Таблица объектов, описывающих сетевые интерфейсы (.1.3.6.1.4.1.10812.1.2.2):

.1.3.6.1.4.1.10812.1.2.2.1.1	Номер интерфейса.
.1.3.6.1.4.1.10812.1.2.2.1.2	Системное имя интерфейса.
.1.3.6.1.4.1.10812.1.2.2.1.3	Устаревший параметр, всегда равен 0.
.1.3.6.1.4.1.10812.1.2.2.1.4	Количество пропущенных входящих незашифрованных пакетов.
.1.3.6.1.4.1.10812.1.2.2.1.5	Количество заблокированных входящих незашифрованных пакетов.
.1.3.6.1.4.1.10812.1.2.2.1.6	Количество пропущенных исходящих незашифрованных пакетов.
.1.3.6.1.4.1.10812.1.2.2.1.7	Количество заблокированных исходящих незашифрованных пакетов.
.1.3.6.1.4.1.10812.1.2.2.1.8	Количество пропущенных входящих зашифрованных пакетов.
.1.3.6.1.4.1.10812.1.2.2.1.9	Количество заблокированных входящих зашифрованных пакетов.
.1.3.6.1.4.1.10812.1.2.2.1.10	Количество пропущенных исходящих зашифрованных пакетов.

OID	Описание
.1.3.6.1.4.1.10812.1.2.2.1.11	Количество заблокированных исходящих зашифрованных пакетов.
Таблица объектов, описывающих ViPNet-клиентов, для которых координатор является сервером IP-адресов (.1.3.6.1.4.1.10812.1.4.1)	
.1.3.6.1.4.1.10812.1.4.1.1.1	Шестнадцатеричный идентификатор сетевого узла ViPNet.
.1.3.6.1.4.1.10812.1.4.1.1.2	Имя сетевого узла ViPNet.
.1.3.6.1.4.1.10812.1.4.1.1.3	IP-адрес точки доступа (FirewallIP).
.1.3.6.1.4.1.10812.1.4.1.1.4	TCP/UDP порт точки доступа.
Таблица объектов, описывающих состояние служб iplircfg, failoverd, mftpd, webgui-fcgi-server (.1.3.6.1.4.1.10812.1.7.1):	
.1.3.6.1.4.1.10812.1.7.1.1.1	Индекс службы.
.1.3.6.1.4.1.10812.1.7.1.1.2	Имя службы: • Failover — failoverd, • Iplir — iplircfg, • Mftp — mftpd, • Webgui — webgui-fcgi-server.
.1.3.6.1.4.1.10812.1.7.1.1.3	Состояние службы: • 0 (unknown) — состояние неизвестно, • 1 (shutdown) — работа службы завершена, • 2 (running) — служба работает.
.1.3.6.1.4.1.10812.1.7.1.1.4	Время работы службы.
.1.3.6.1.4.1.10812.1.8.1.0	Режим работы службы failoverd: • 0 (single) — одиночный, • 1 (cluster passive) — режим кластера, узел работает в пассивном режиме, • 2 (cluster active) — режим кластера, узел работает в активном режиме, • 3 (cluster switch) — режим кластера, узел меняет режим работы в текущий момент.
.1.3.6.1.4.1.10812.1.8.2.0	Время последнего переключения службы failover в текущий режим работы.
.1.3.6.1.4.1.10812.1.8.3.0	IP-адрес интерфейса синхронизации другого узла кластера.
.1.3.6.1.4.1.10812.1.8.4.0	Количество сетевых соединений, находящихся в кеше узла кластера.
.1.3.6.1.4.1.10812.1.8.5.0	Системное имя интерфейса синхронизации.
Таблица доступности соединений в канале между узлами кластера (.1.3.6.1.4.1.10812.1.8.6):	
X – индекс таблицы	
.1.3.6.1.4.1.10812.1.8.6.1.1.X	Индекс таблицы.

OID	Описание
.1.3.6.1.4.1.10812.1.8.6.1.2.X	IP-адрес интерфейса синхронизации.
.1.3.6.1.4.1.10812.1.8.6.1.3.X	Статус доступности узла: 1 — доступен, 2 — недоступен.
Таблица с информацией о ПАК (.1.3.6.1.4.1.10812.1.10.1)	
.1.3.6.1.4.1.10812.1.10.1.1	Запись таблицы
.1.3.6.1.4.1.10812.1.10.1.1.1.1	Индекс строки
.1.3.6.1.4.1.10812.1.10.1.1.2.1	Наименование параметра: Serial number
.1.3.6.1.4.1.10812.1.10.1.1.3.1	Серийный номер

SNMP-агент ViPNet Coordinator HW может отправлять оповещения:

Таблица 52. SNMP-оповещения. VIPNET-MIB

OID	Описание
.1.3.6.1.4.1.10812.1.0.1	Запуск службы <code>iplircfg</code> .
.1.3.6.1.4.1.10812.1.0.2	Остановка службы <code>iplircfg</code> .
.1.3.6.1.4.1.10812.1.0.3	Запуск службы <code>mftpd</code> .
.1.3.6.1.4.1.10812.1.0.4	Остановка службы <code>mftpd</code> .
.1.3.6.1.4.1.10812.1.0.7	Запуск службы <code>failoverd</code> .
.1.3.6.1.4.1.10812.1.0.8	Остановка службы <code>failoverd</code> .
.1.3.6.1.4.1.10812.1.0.9	Переключение службы <code>failoverd</code> из пассивного режима в активный.
.1.3.6.1.4.1.10812.1.0.10	Запуск службы <code>webgui</code> .
.1.3.6.1.4.1.10812.1.0.11	Остановка службы <code>webgui</code> .
.1.3.6.1.4.1.10812.1.0.12	Потеря связи с пассивным узлом кластера.
.1.3.6.1.4.1.10812.1.0.13	Восстановление связи с пассивным узлом кластера.
.1.3.6.1.4.1.10812.1.0.16	Изменение IP-адреса и порта точки доступа к ViPNet-клиенту.

MonitoringAgent-MIB

MonitoringAgent-MIB (.1.3.6.1.4.1.10812.4) включает в себя объекты, предназначенные для получения информации о характеристиках SNMP-агента (например, версия, уникальный идентификатор).

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 53. SNMP-параметры. *MonitoringAgent-MIB*

OID	Описание
.1.3.6.1.4.1.10812.4.3.1	Версия агента мониторинга.
.1.3.6.1.4.1.10812.4.3.2	Идентификатор инсталляции агента мониторинга.

System-MonitoringAgent-MIB

System-MonitoringAgent-MIB (.1.3.6.1.4.1.10812.4.20) включает в себя объекты, предназначенные для получения информации об устройстве, на котором запущен SNMP-агент.

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 54. SNMP-параметры. *System-MonitoringAgent-MIB*

OID	Описание
.1.3.6.1.4.1.10812.4.20.5.0	Идентификатор устройства в HEX-формате.
.1.3.6.1.4.1.10812.4.20.9.0	Уникальный идентификатор устройства в виде строки.



Список служб в составе ПО ViPNet Coordinator HW

Ниже приведен список служб в составе ПО ViPNet Coordinator HW, для которых можно настроить фильтрацию вывода событий системного журнала (см. [Просмотр системного журнала](#)):

Службы ViPNet:

- `bonding_helper` — скрипт управления модуля ядра, реализующего функционал агрегированных каналов;
- `dgdd` — служба DGD policy routing;
- `drviplir` — драйвер перехвата сетевого трафика (модуль ядра);
- `failover`, `failoverd` — служба слежения за основными службами ViPNet и организации кластера;
- `http_proxy` — управляющие скрипты прокси-сервера squid;
- `l2overip` — драйвер, организующий прозрачное соединение сегментов сети на уровне L2 (модуль ядра);
- `icap` — служба для проверки трафика, передаваемого через прокси-сервер, внешним антивирусом;
- `lplircfg` — служба, управляющая настройками драйвера перехвата сетевого трафика, ведения журнала IP пакетов, обновления справочно-ключевой информации;
- `lplirpasswd` — утилита проверки паролей пользователя и администратора ViPNet;
- `ltscrip` — крипто-драйвер шифрования IP пакетов (модуль ядра);
- `ltskrniface` — интерфейс драйвер перехвата сетевого трафика;
- `ltswd` — watch-dog драйвер службы слежения за основными службами;

- `mftpd`, `mftpd` — служба транспортного модуля ViPNet;
- `ntp_service` — управляющий скрипт службы NTP;
- `rvpn_shell` — командный интерпретатор ViPNet;
- `session_watcher` — служба, отслеживающая сессии пользователей ViPNet;
- `unmerge` — утилита распаковки дистрибутива справочников и ключей;
- `webgui-fcgi-server` — служба веб-интерфейса ViPNet Coordinator HW;
- `snmpd`, `vipnetsnmp` — служба SNMP;
- `vupgrade` — утилита обновления ПО ViPNet Coordinator HW.

Службы Linux:

- `acpid` (ACPI event daemon) — служба для реакции на ACPI события, например, реакция на нажатие кнопки питания;
- `bonding` — модуль ядра, реализующий функционал агрегированных каналов;
- `chat` — скрипт управления службы установки соединения по модему;
- `crond` — служба для запуска задач по расписанию;
- `conntrackd` — служба для отслеживания соединений;
- `conntrackd_tracker` — служба для отслеживания состояния `conntrackd`;
- `hostapd` — служба wi-fi AP;
- `login` — служба для аутентификации пользователя и открытия его сессии;
- `named` — служба DNS;
- `ntpd` — служба NTP;
- `pcscd` — служба pcsc для работы с токенами;
- `pppd` — служба установки соединения по модему;
- `sshd` — служба SSH;
- `squid` — служба прокси-сервера squid;
- `syslogd` — служба системного журнала;
- `udhcpd` — служба DHCP-клиента;
- `dhcpcd` — служба DHCP-сервера;
- `dhcrelay` — служба агента-ретранслятора DHCP-запросов;
- `upsd` — служба ИБП;
- `upsmon` — служба монитора ИБП (отключен);
- `vmunix` — ядро системы;
- `watchquagga` — служба слежения за сервисом динамической маршрутизации;
- `wpa_cli` — служба Wi-Fi-клиента;

- `wpa_supplicant` — скрипт управления службой Wi-Fi-клиента;
- `xinetd` — служба, запускающая сетевые серверные процессы;
- `zebra` — служба динамической маршрутизации;
- `zdhcpd` — служба протокола DHCP динамической маршрутизации;
- `ospfd` — служба протокола OSPF динамической маршрутизации.



Поддерживаемые MIME-типы

- **Application — Внутренний формат прикладной программы**
 - application/atom+xml: Atom
 - application/EDI-X12: EDI X12 (RFC 1767)
 - application/EDIFACT: EDI EDIFACT (RFC 1767)
 - application/json: JavaScript Object Notation JSON (RFC 4627)
 - application/javascript: JavaScript (RFC 4329)
 - application/octet-stream: двоичный файл без указания формата (RFC 2046)
 - application/ogg: Ogg (RFC 5334)
 - application/pdf: Portable Document Format, PDF (RFC 3778)
 - application/postscript: PostScript (RFC 2046)
 - application/soap+xml: SOAP (RFC 3902)
 - application/font-woff: Web Open Font Format
 - application/xhtml+xml: XHTML (RFC 3236)
 - application/xml-dtd: DTD (RFC 3023)
 - application/xop+xml: XOP
 - application/zip: ZIP
 - application/gzip: Gzip
 - application/x-bittorrent: BitTorrent

- application/x-tex: TeX
- **audio — Аудио**
 - audio/basic: mulaw аудио, 8 кГц, 1 канал (RFC 2046)
 - audio/L24: 24-bit Linear PCM аудио, 8-48 кГц, 1-N каналов (RFC 3190)
 - audio/mp4: MP4
 - audio/aac: AAC
 - audio/mpeg: MP3 или др. MPEG (RFC 3003)
 - audio/ogg: Ogg Vorbis, Speex, Flac или др. аудио (RFC 5334)
 - audio/vorbis: Vorbis (RFC 5215)
 - audio/x-ms-wma: Windows Media Audio
 - audio/x-ms-wax: Windows Media Audio перенаправление
 - audio/vnd.rn-realaudio: RealAudio
 - audio/vnd.wave: WAV (RFC 2361)
 - audio/webm: WebM
- **image — Изображение**
 - image/gif: GIF (RFC 2045 и RFC 2046)
 - image/jpeg: JPEG (RFC 2045 и RFC 2046)
 - image/pjpeg: JPEG
 - image/png: Portable Network Graphics (RFC 2083)
 - image/svg+xml: SVG
 - image/tiff: TIFF (RFC 3302)
 - image/vnd.microsoft.icon: ICO
 - image/vnd.wap.wbmp: WBMP
- **message — Сообщение**
 - message/http: (RFC 2616)
 - message/imdn+xml: IMDN (RFC 5438)
 - message/partial: E-mail (RFC 2045 и RFC 2046)
 - message/rfc822: E-mail; EML файлы, MIME файлы, MHT файлы, MHTML файлы (RFC 2045 и RFC 2046)
- **model — Для 3D моделей**
 - model/example: (RFC 4735)
 - model/iges: IGS файлы, IGES файлы (RFC 2077)
 - model/mesh: MSH файлы, MESH файлы (RFC 2077), SILO файлы

- model/vrml: WRL файлы, VRML файлы (RFC 2077)
- model/x3d+binary: X3D ISO стандарт для 3D компьютерной графики, X3DB файлы
- model/x3d+vrml: X3D ISO стандарт для 3D компьютерной графики, X3DV VRML файлы
- model/x3d+xml: X3D ISO стандарт для 3D компьютерной графики, X3D XML файлы
- **multipart — Составные файлы**
 - multipart/form-data: MIME Webform (RFC 2388)
 - multipart/signed: (RFC 1847)
 - multipart/encrypted: (RFC 1847)
 - multipart/mixed: MIME E-mail (RFC 2045 и RFC 2046)
 - multipart/alternative: MIME E-mail (RFC 2045 и RFC 2046)
 - multipart/related: MIME E-mail (RFC 2387 и используемое MHTML (HTML mail))
- **text — Текст**
 - text/cmd: команды
 - text/css: Cascading Style Sheets (RFC 2318)
 - text/csv: CSV (RFC 4180)
 - text/html: HTML (RFC 2854)
 - text/javascript (Obsolete): JavaScript (RFC 4329)
 - text/plain: текстовые данные (RFC 2046 и RFC 3676)
 - text/php: Скрипт языка PHP
 - text/xml: Extensible Markup Language (RFC 3023)
- **video — Видео**
 - video/mpeg: MPEG-1 (RFC 2045 и RFC 2046)
 - video/mp4: MP4 (RFC 4337)
 - video/ogg: Ogg Theora или другое видео (RFC 5334)
 - video/quicktime: QuickTime
 - video/webm: WebM
 - video/x-ms-wmv: Windows Media Video
 - video/x-flv: FLV
 - video/3gpp: .3gpp .3gp
 - video/3gpp2: .3gpp2 .3g2
- **vnd — Вендорные файлы**
 - application/vnd.oasis.opendocument.text: OpenDocument
 - application/vnd.oasis.opendocument.spreadsheet: OpenDocument

- application/vnd.oasis.opendocument.presentation: OpenDocument
- application/vnd.oasis.opendocument.graphics: OpenDocument
- application/vnd.ms-excel: Microsoft Excel файлы
- application/vnd.openxmlformats-officedocument.spreadsheetml.sheet: Microsoft Excel 2007 файлы
- application/vnd.ms-powerpoint: Microsoft Powerpoint файлы
- application/vnd.openxmlformats-officedocument.presentationml.presentation: Microsoft Powerpoint 2007 файлы
- application/msword: Microsoft Word файлы
- application/vnd.openxmlformats-officedocument.wordprocessingml.document: Microsoft Word 2007 файлы
- application/vnd.mozilla.xul+xml: Mozilla XUL файлы
- application/vnd.google-earth.kml+xml: KML файлы (например, для Google Earth)
- **x — Нестандартные файлы**
 - application/x-www-form-urlencoded: Form Encoded Data
 - application/x-dvi: DVI
 - application/x-latex: LaTeX файлы
 - application/x-font-ttf: TrueType (не зарегистрированный MIME-тип, но наиболее часто используемый)
 - application/x-shockwave-flash: Adobe Flash
 - application/x-stuffit: Stuffit
 - application/x-rar-compressed: RAR
 - application/x-tar: Tarball
 - text/x-jquery-tmpl: jQuery
 - application/x-javascript
- **x-pkcs — PKCS**
 - application/x-pkcs12: p12 файлы
 - application/x-pkcs12: pfx файлы
 - application/x-pkcs7-certificates: p7b файлы
 - application/x-pkcs7-certificates: spc файлы
 - application/x-pkcs7-certreqresp: p7r файлы
 - application/x-pkcs7-mime: p7c файлы
 - application/x-pkcs7-mime: p7m файлы
 - application/x-pkcs7-signature: p7s файлы



Использование ViPNet Coordinator HW в качестве МЭ типа А четвертого класса защиты

Рекомендации по использованию ViPNet Coordinator HW в качестве межсетевого экрана типа «А» четвертого класса защиты:

- Запретите трафик от нежелательных приложений. Для этого [создайте сетевые фильтры, запрещающие передачу данных через порты, которые используют эти приложения](#).
Разрешите трафик от доверенных приложений. Для этого создайте сетевые фильтры, разрешающие передачу данных через порты, которые используют эти приложения.
- Для ассоциации пользователей сети и IP-адресов:
 - для каждого сетевого узла задайте статический IP-адрес;
 - занесите в специальный журнал ФИО физических лиц, имеющих учетные записи на сетевых узлах;
- организуйте фиксацию времени работы пользователей на сетевых узлах, где зарегистрировано несколько учетных записей, в специальном журнале.



Возможные неполадки

Блокирование медиа-сессии SIP-соединения при завершении сигнальной сессии по тайм-ауту

В SIP-клиентах некоторых производителей сигнальная сессия SIP-соединения может завершаться через 300 сек. при отсутствии активности. В этом случае ViPNet Coordinator HW блокирует медиа-сессию установленного SIP-соединения. Чтобы избежать блокирования медиа-сессии, установите в настройках SIP-клиента отправку сообщений `Keep Alive` каждые 30 сек. У разных производителей SIP-клиентов этот параметр может называться `re-INVITE`, `OPTIONS`, `INFO` или `NOTIFY`.

Для новых вариантов ключей отображается старая дата обновления

После смены варианта ключей сетевого узла и пользователя ViPNet Coordinator HW информация о дате обновления ключей на ViPNet Coordinator HW может быть неактуальной. Это происходит, когда в ViPNet УКЦ администратор сети ViPNet сначала создал новые ключи сетевого узла, а потом ключи пользователя. После отправки и применения нового варианта ключей на ViPNet Coordinator HW команда `iplir show key-info` в разделе **Current personal key info > Current personal key update date** будет отображать дату предыдущего обновления.

Следующие инструкции предназначены для администратора сети ViPNet.

Чтобы устранить проблему:

- 1 В ViPNet УКЦ создайте новые ключи сетевого узла ViPNet Coordinator HW и отправьте их в ViPNet ЦУС.
- 2 В ViPNet ЦУС отправьте новые ключи на сетевой узел ViPNet Coordinator HW.
- 3 На ViPNet Coordinator HW выполните команду `iplir show key-info` и убедитесь, что дата обновления правильная.

Правильный порядок изменения варианта ключей:

- 1 В ViPNet УКЦ примените новый вариант ключей для сетевого узла и пользователей ViPNet Coordinator HW.
- 2 Создайте и передайте в ЦУС ключи пользователя ViPNet Coordinator HW.
- 3 Создайте и передайте в ЦУС ключи сетевого узла ViPNet Coordinator HW.
- 4 В ViPNet ЦУС отправьте обновления ключей на ViPNet Coordinator HW.
- 5 На ViPNet Coordinator HW выполните команду `iplir show key-info` и убедитесь, что дата обновления правильная.

Подробнее см. документ «ViPNet Удостоверяющий и ключевой центр. Руководство администратора», раздел «Изменение вариантов персонального ключа пользователя и ключей узла».

Истек срок действия пароля администратора

Если у вас истек срок действия пароля администратора и вы не можете авторизоваться на ViPNet Coordinator HW, обратитесь к администратору ViPNet ЦУС или Prime. Администратор создаст новый пароль и отправит его на координатор в составе обновлений справочников и ключей.

Если обновления на ViPNet Coordinator HW не применяются, обратитесь в службу технической поддержки [ИнфоТеКС](#).

При переносе клиента за другой координатор обновления из ЦУСа не применяются

При переносе клиента за другой координатор администратор ViPNet Administrator ЦУС отправляет обновления на узлы ViPNet. Для старого координатора, на котором клиент был зарегистрирован до изменения, автоматически будет сформировано отложенное обновление с временем применения, заданным в настройках программы ViPNet ЦУС (по умолчанию — 24 часа). По истечении этого срока обновления будут применены.

Подробнее см. документ «ViPNet Центр управления сетью 4. Руководство администратора», раздел «Перенос клиента на другой координатор».

Проблема с обновлением ПО в кластере с ограничением по IP-адресам

В ранних версиях ViPNet Coordinator HW кластер горячего резервирования мог настраиваться в режиме с ограничением адресов. При обновлении ПО на узлах кластера в файлах конфигурации остаются несовместимые записи о пассивных IP-адресах, имеющих одну из масок подсети: 255.255.255.255, 255.255.255.254, 0.0.0.0. При попытке обновить ПО на узле кластера обновление не будет принято, а в ViPNet ЦУС или Prime будет отправлено сообщение Обновление не прошло.

Чтобы устранить проблему:

- 1 Остановите систему защиты от сбоев:

```
hostname# failover stop
```

- 2 Просмотрите записи системного журнала, отфильтрованные по службе vupgrade:

```
hostname# machine show logs filtered vupgrade
```

- 3 Если в списке сообщений имеется строка `Software upgrade failed`, совпадающая по времени с моментом неудачной попытки обновления, то на обоих узлах кластера:

- 3.1 Сбросьте конфигурацию всех сетевых интерфейсов:

```
hostname# inet ifconfig all reset
```

- 3.2 Выполните настройку сетевых интерфейсов и кластера горячего резервирования согласно разделу [Настройка системы защиты от сбоев](#).

- 4 Повторно выполните обновления ПО на обоих узлах кластера.

Проблемы при работе системы защиты от сбоев

В ViPNet ЦУС или Prime справочники и ключи имеют статус Приняты, но на кластере изменения не применились

Иногда применение обновления справочников и ключей может совпасть по времени с переключением кластера, что может привести к отправке в ViPNet ЦУС или Prime некорректного статуса обновления. Если в момент применения обновления было переключение кластера, то повторно отправьте из ViPNet ЦУС или Prime обновление справочников и ключей.

Роль (функция) Failover задана, но в командном интерпретаторе отсутствуют команды для работы с кластером

Если в ViPNet ЦУС или Prime роль (функция) Failover была добавлена не при первичной установке дистрибутива ключей, а при обновлении ключей, то команды для работы с кластером могут не появиться в командном интерпретаторе.

В таком случае выйдите из командного интерпретатора (с помощью команды `exit`) и войдите снова.

Циклическая перезагрузка

Если ViPNet Coordinator HW автоматически перезагружается сразу после загрузки и не дает возможности авторизоваться, причиной могут быть:

- ошибки в системных файлах;
- неисправность аппаратной платформы;
- ошибки обновления ПО;
- удаленная смена пользователя ViPNet Coordinator HW;
- некорректная смена мастер-ключей;
- некорректная настройка ViPNet Coordinator HW;
- некорректная настройка кластера горячего резервирования.

После трех перезагрузок подряд автоматически запускается система восстановления, которая может восстановить системные файлы, если это необходимо.

Если после четырех автоматических перезагрузок подряд ViPNet Coordinator HW по-прежнему не загружается и не дает возможности выполнять команды:

- 1 Подключитесь к ViPNet Coordinator HW с помощью консоли и дождитесь появления сообщений:

```
Cyclic reboot detected
If you want to start VPN later, press ENTER
```

- 2 Нажмите клавишу **Enter**.

Появятся сообщения:

```
User has confirmed starting VPN later
VPN start after <600> seconds
```

- 3 Когда появится приглашение ввода имени пользователя, введите имя пользователя `user` и пароль.

Появится сообщение:

```
Late start VPN was disabled
```

и вы получите доступ к командному интерпретатору.

- 4 Выполните экспорт системного журнала и передайте его в [службу технической поддержки ИнфоТеКС](#).
- 5 Если у вас есть предположения о причине возникновения циклической перезагрузки, вы можете самостоятельно изменить настройки ViPNet Coordinator HW.

Циклическая перезагрузка ViPNet Coordinator HW после смены пользователя в ViPNet ЦУС или Prime

После инициализации ViPNet Coordinator HW, смена пользователя сетевого узла в ViPNet ЦУС или Prime с обновлением справочников и ключей, приводит к циклической перезагрузке ViPNet Coordinator HW.



Внимание! Менять пользователя сетевого узла в ViPNet ЦУС или Prime запрещено.

Чтобы восстановить работоспособность ViPNet Coordinator HW, выполните сброс к заводским настройкам. После этого вы можете выполнить инициализацию ViPNet Coordinator HW с дистрибутивом ключей нового пользователя.

Некорректные настройки параметров работы системы защиты от сбоев

При обновлении с предыдущих версий ViPNet Coordinator HW может уйти в циклическую перезагрузку, если в файле `failover.ini` указаны некорректные параметры `checktime` и `timeout`.

Если после 4 автоматических перезагрузок подряд ViPNet Coordinator HW по-прежнему не загружается и не дает возможности выполнять команды:

- 1 Подключитесь к ViPNet Coordinator HW с помощью консоли и дождитесь появления сообщений:

```
Cyclic reboot detected
If you want to start VPN later, press ENTER
```

- 2 Нажмите **Enter**.

В результате появятся сообщения:

```
User has confirmed starting VPN later
VPN start after <600> seconds
```


- 3 Когда появится приглашение ввода имени пользователя, введите имя пользователя `user` и пароль.

В результате появится сообщение:

```
Late start VPN was disabled
```

и вы получите доступ к командному интерпретатору.

- 4 Откройте конфигурационный файл `failover.ini` для редактирования:

```
hostname# failover config edit
```

В конфигурационном файле `failover.ini` замените значение параметров `checktime` и `timeout` секции `[network]`.



Внимание! Значение параметра `checktime` должно быть больше чем значение параметра `timeout`.

- 1 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).
- 2 Перезагрузите ViPNet Coordinator HW.

Если изменения не помогли выполните экспорт системного журнала и передайте его в [службу технической поддержки ИнфоТекС](#).

Нарушена целостность конфигурационных файлов

Если целостность конфигурационных файлов нарушена, то при запуске служб появится сообщение об ошибке `Error checking product integrity:...`

Для восстановления работоспособности ViPNet Coordinator HW, выполните:

- Восстановите настройки ViPNet Coordinator HW из копии конфигурации VPN (см. [Восстановление настроек из копии конфигурации VPN](#)):

```
hostname# admin config load <имя> [<версия>]
```
- Если нет сохраненных копий конфигурации, перезагрузите ViPNet Coordinator HW:

```
hostname# machine reboot
```

Циклическая перезагрузка узла кластера после смены пароля пользователя

Если пассивный узел кластера циклически перезагружается после смены пароля пользователя, возможно, пароль на узлах был изменен не в том порядке.



Примечание. Чтобы изменить пароль пользователя на ViPNet Coordinator HW, работающих в режиме кластера горячего резервирования, сначала измените его на пассивном узле кластера, а затем на активном узле. Пароли, задаваемые на обоих узлах, должны совпадать.

Чтобы восстановить работу пассивного узла:

1 Экспортируйте справочники, ключи и настройки с работоспособного узла.

2 На пассивном узле:

2.1 Сбросьте настройки к заводским.

2.2 Импортируйте справочники, ключи и настройки.

2.3 Поменяйте IP-адрес на интерфейсе синхронизации:

```
hostname# inet ifconfig <интерфейс> address <IP-адрес> netmask <маска подсети>
```

2.4 Завершите работу службы системы защиты от сбоев:

```
hostname# failover stop
```

2.5 Откройте на редактирование файл `failover.ini`:

```
hostname# failover config edit
```

2.6 В секции `secondconfig` поменяйте IP-адрес интерфейса синхронизации в параметре `activeip`.

2.7 Сохраните изменения (**Ctrl+O**, затем нажмите **Enter**) и закройте редактор (**Ctrl+X**).

2.8 Запустите службу системы защиты от сбоев:

```
hostname# failover start
```

Блокируется защищенный транзитный трафик

Защищенный транзитный трафик может блокироваться при обновлении на версию 4.5.0 и выше. Это связано с тем, что изменился принцип обработки трафика.

Раньше при получении IP-пакета он проверялся на признаки защищенного трафика, затем проверялся IP-адрес назначения и IP-пакет обрабатывался соответствующими фильтрами. Если IP-пакет был защищенным, и IP-адрес назначения не принадлежал ViPNet Coordinator HW, он передавался дальше, своему серверу соединения.



Рисунок 75. Обработка транзитного трафика на версиях ниже 4.5.0

Теперь при получении IP-пакета сначала проверяется IP-адрес назначения: если он не принадлежит ViPNet Coordinator HW и IP-пакет не является туннелируемым, он обрабатывается транзитными фильтрами открытой сети, вне зависимости от того, защищенный он или открытый (см. [Принципы фильтрации](#)).

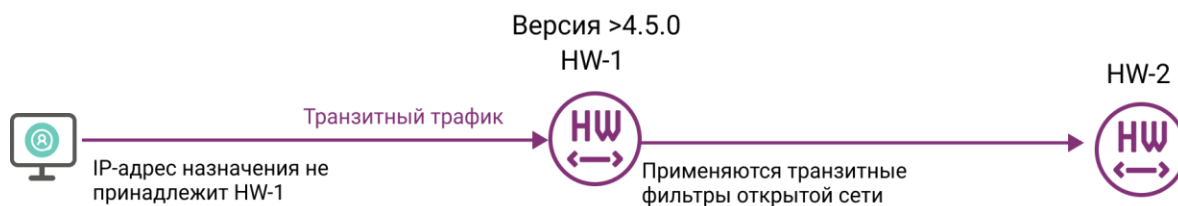


Рисунок 76. Обработка транзитного трафика на версиях выше 4.5.0

Поэтому некоторые защищенные транзитные IP-пакеты могут блокироваться при поступлении на ViPNet Coordinator HW. Чтобы пропустить такие пакеты [добавьте разрешающие транзитные фильтры открытой сети](#).

Блокируется исходящий трафик от ViPNet Coordinator HW в режиме «Со статической трансляцией адресов»

Исходящий защищенный трафик может блокироваться при обновлении до версии 4.5.0 и выше. Это связано с тем, что в новых версиях поменялся принцип обработки исходящих пакетов, когда ViPNet Coordinator HW находится в [режиме «Со статической трансляцией адресов»](#).

В предыдущих версиях параметр `port`, указанный в собственной секции `[id]` ViPNet Coordinator HW, использовался как порт источника при отправке UDP-пакетов, а также как порт назначения при их получении.

В новых версиях IP-пакеты отправляются не с определенного порта, а порт выбирается из диапазона от `port` до `port+255` (например, если в собственной секции `[id]` указан `port=55777`, то для исходящих пакетов будет выбран порт из диапазона 55777–56032).



Примечание. Динамический выбор порта помогает повысить производительность ViPNet Coordinator HW. Раньше создавалась одна сетевая сессия, внутри которой были все соединения. Теперь на каждое соединение создается своя сессия, таким образом реализуется многопоточность.

Поэтому если на устройстве со статическим NAT были добавлены правила обработки трафика с указанием определенного порта источника, пакеты, отправленные с ViPNet Coordinator HW, могут блокироваться.

Избежать блокировки исходящего трафика можно двумя способами:

- На устройстве со статическим NAT добавьте правила пропускающие UDP-пакеты с трансляцией адреса источника на внешний IP-адрес устройства NAT:
 - адрес источника: IP-адрес доступа ViPNet Coordinator HW;
 - порт источника: от `port` до `port+255` (`port` — параметр в собственной секции `[id]` ViPNet Coordinator HW);
 - адрес назначения: любой;
 - порт назначения: любой;
- Переведите ViPNet Coordinator HW в режим «С динамической трансляцией адресов».

Получено сообщение от сервера мониторинга о недостатке дискового пространства

Заполнение дискового пространства служебных разделов `/cache`, `/mnt/sysfs` и `/mnt/vipnet` на 100% — штатная работа ViPNet Coordinator HW. Заполнение этих разделов не влияет на работоспособность координатора.

Сообщение не является критическим. Не рекомендуем собирать и анализировать данную информацию.



Термины и сокращения

BDR (Backup Designated Router)

Резервный выделенный маршрутизатор, управляющий рассылкой объявлений о состоянии канала в сетях с маршрутизирующим протоколом OSPF.

CEF (Common Event Format)

Открытый текстовый формат описания событий. Позволяет агрегировать события различных типов сетевых устройств и приложений в едином формате.

COM-консоль

Ноутбук, подключенный к COM-порту, который используется для локальной настройки ViPNet Coordinator HW.

DAD (Duplicate address detection)

Метод проверки уникальности IP-адреса с помощью отправки специального ARP-запроса с указанием проверяемого IP-адреса и ожиданием ответа от этого IP-адреса. Результатом является получение от устройства с указанным IP-адресом ответа. Если в течение определенного времени ответ не поступил, считается, что такой адрес в сети не используется.

DGD (Dead Gateway Detection)

Технология проверки доступности шлюза путем отправки эхо-запросов через определенный промежуток времени. Если в течение определенного времени ответ не поступил, считается, что данный шлюз недоступен. Данная технология может определять политику маршрутизации в зависимости от доступности шлюзов.

DHCP-relay

Функция ретрансляции запросов и ответов от (к) внешнего DHCP-сервера в локальную сеть.

DR (designated router)

Выделенный маршрутизатор, управляющий рассылкой объявлений о состоянии канала в сетях с маршрутизирующим протоколом OSPF.

DSCP-метка

Информация о приоритете обработки IP-пакета, указанная в заголовке IP-пакета.

ICAP

Протокол для расширения функциональности прокси-серверов. Чаще всего используется для внедрения функций антивирусной проверки и контентной фильтрации трафика, проходящего через прокси-сервер.

ICAP-сервер

Сервер, выполняющий обработку трафика по протоколу [ICAP](#).

L2OverIP

Технология, которая позволяет организовать защиту удаленных сегментов сети, использующих одно и то же адресное пространство, на канальном уровне модели OSI. В результате узлы из разных сегментов сети смогут взаимодействовать друг с другом в одном широковещательном домене. В основе технологии лежит перехват на канальном уровне модели OSI Ethernet-кадров, отправленных из одного сегмента сети в другой.

MTU (Maximum Transmission Unit)

Максимальный размер полезного блока данных одного пакета, который может быть передан протоколом через сетевой интерфейс без фрагментации.

OSPF (Open Shortest Path First)

Протокол динамической маршрутизации, основанный на технологии отслеживания состояния канала для нахождения кратчайшего маршрута. Распространяет информацию о доступных маршрутах внутри автономной системы.

PPP (Point-to-Point Protocol)

Протокол канального уровня, использующийся для установления прямой связи между двумя узлами сети.

SNMP (Simple Network Management Protocol)

Интернет-протокол для управления устройствами в IP-сетях на основе архитектур TCP/UDP. Используется в системах сетевого управления для контроля подключенных к сети устройств. Позволяет собирать данные с контролируемого устройства и отправлять их управляющему элементу, на котором выполняется контроль.

SNMP-агент

Программное обеспечение, которое запускается на устройстве и передает информацию о его работе на специальный сервер по протоколу SNMP.

TCP-туннель

Способ соединения клиентов ViPNet, находящихся во внешних сетях, со своим сервером соединений, а затем и с другими узлами сети ViPNet по протоколу TCP. Используется в том случае, если соединение по протоколу UDP заблокировано провайдерами услуг интернета.

TCP-туннель настраивается на координаторе, который является для клиента сервером соединений.

ViPNet Policy Manager

Программа для централизованного управления политиками безопасности узлов защищенной сети ViPNet.

ViPNet Prime

ПО для централизованного управления решениями ViPNet. Позволяет управлять конфигурацией сети (включая устройства, пользователей и лицензии), централизованно обновлять ПО ViPNet и выполнять мониторинг состояния сети ViPNet.

Включает в себя основные функциональные модули:

- ViPNet VPN — модуль управления топологией сети, регистрирует защищаемые устройства и задает связи между ними.
- ViPNet Network Visibility System — модуль мониторинга состояния сети ViPNet и входящих в нее устройств.
- ViPNet Policy Management — модуль централизованного управления политиками безопасности узлов сети ViPNet.

ViPNet StateWatcher

Программный комплекс для наблюдения за состоянием узлов сетей ViPNet, мониторинга событий безопасности, происходящих на сетевых узлах, своевременного выявления неполадок в работе узлов и оперативного оповещения пользователей о возникающих проблемах.

ViPNet Центр управления сетью (ЦУС)

Программа, входящая в состав программного обеспечения ViPNet Administrator. Предназначена для создания и управления конфигурацией сети и позволяет решить следующие основные задачи:

- построение виртуальной сети (сетевые объекты и связи между ними, включая межсетевые);
- изменение конфигурации сети;
- формирование и рассылка справочников;
- рассылка ключей узлов и ключей пользователей;
- формирование информации о связях пользователей для Удостоверяющего и ключевого центра;
- задание полномочий пользователей сетевых узлов ViPNet.

VLAN

Виртуальная локальная компьютерная сеть, представляет собой группу сетевых узлов с общим набором требований, которые взаимодействуют так, как если бы они были подключены к широковещательному домену, независимо от их физического местонахождения. VLAN имеет те же свойства, что и физическая локальная сеть, но позволяет узлам группироваться вместе, даже если они не находятся в одной физической сети.

Автономная система

Один или несколько сегментов сети, в которых выполняется маршрутизация по одному протоколу (OSPF, IGRP, EIGRP, IS-IS, RIP, BGP, Static). Также может трактоваться как домен маршрутизации — группа маршрутизаторов сети, работающих по одинаковым протоколам маршрутизации.

Агрегированный сетевой интерфейс

Логический [сетевой интерфейс](#), образованный из нескольких физических интерфейсов Ethernet, объединенных на канальном уровне сетевой модели OSI.

Административная дистанция

Характеристика [маршрута](#). Позволяет определить меру доверия к маршруту. Задается для любого маршрута в виде целого числа в диапазоне от 1 до 255.

Администратор сети ViPNet

Лицо, отвечающее за управление сетью ViPNet, создание и обновление справочников и ключей для сетевых узлов ViPNet, настройку межсетевого взаимодействия с доверенными сетями.

Базовый виртуальный адрес

Является точкой отсчета при назначении виртуальных адресов для каждого из реальных адресов узла. Если в данный момент узел виден по виртуальному адресу, его адресом доступа считается

либо базовый виртуальный адрес, либо вторичный виртуальный адрес, соответствующий первому в списке реальному адресу.

Вариант персонального ключа пользователя

Номер персонального ключа пользователя из резервного набора персональных ключей (РНПК). Изменение варианта персонального ключа пользователя означает, что номер персонального ключа был увеличен на единицу и для создания новых ключей пользователя будет использоваться следующий персональный ключ из РНПК.

Вес

Параметр, который задается для шлюза в статическом [маршруте](#) в виде целого числа в диапазоне от 1 до 255. Позволяет настроить балансировку IP-трафика между шлюзами в одинаковый адрес назначения. Определяет долю IP-трафика, который должен передаваться по маршруту на указанный шлюз.

Виртуальная защищенная сеть

Технология, позволяющая создать логическую сеть, чтобы обеспечить множественные сетевые соединения между компьютерами или локальными сетями через существующую физическую сеть. Уровень доверия к такой виртуальной сети не зависит от уровня доверия к физическим сетям благодаря использованию средств криптографии (шифрования, аутентификации) и средств персонального и межсетевого экранирования.

Виртуальный IP-адрес

IP-адрес, который приложения на сетевом узле ViPNet (А) используют для обращения к ресурсам сетевого узла ViPNet (Б) или туннелируемых им узлов вместо реального IP-адреса узла. Виртуальные IP-адреса узлу ViPNet (Б) назначаются непосредственно на узле А. На других узлах узлу ViPNet (Б) могут быть назначены другие виртуальные адреса. Узлу ViPNet (Б) назначается столько виртуальных адресов, сколько реальных адресов имеет данный узел. При изменении реальных адресов у узла Б выделенные ему виртуальные адреса не изменяются. Виртуальные адреса туннелируемых узлов привязываются к реальным адресам этих узлов и существуют, пока существует данный реальный адрес. Использование виртуальных адресов позволяет избежать конфликта реальных IP-адресов в случае, если сетевые узлы ViPNet работают в локальных сетях с пересекающимся адресным пространством, а также использовать эти адреса для аутентификации удаленных узлов в приложениях ViPNet.

Внутренняя сеть

Локальная сеть, где находятся рассматриваемые узлы, которая отделена от внешней сети межсетевым экраном.

Динамический сетевой интерфейс

Разновидность **сетевых интерфейсов**, который добавляется в процессе работы при наступлении некоторого события (например, при подключении встроенного или USB-модема, предоставляющего данный интерфейс).

Динамические интерфейсы объединяются в группы по типу интерфейса. Поэтому иногда может встречаться термин «групповой динамический интерфейс».

Существуют следующие группы динамических интерфейсов:

- `ppp` — группа интерфейсов для подключения к мобильной сети через встроенный модем;
- `pppoe` — группа интерфейсов для подключения к сети провайдера по PPPoE;
- `wifi` — группа интерфейсов для подключения к беспроводной сети Wi-Fi с помощью внешних адаптеров Wi-Fi. В эту группу не входит встроенный интерфейс `wlan0` (для исполнений ПАК со встроенным модулем Wi-Fi).

Дистрибутив ключей

Файл с расширением `*.dst`, создаваемый в программе ViPNet Удостоверяющий и ключевой центр или ViPNet Prime для каждого пользователя сетевого узла ViPNet. Содержит справочники, ключи и файл лицензии для первичного запуска и последующей работы программы ViPNet на сетевом узле. Для работы программы ViPNet дистрибутив ключей необходимо установить на сетевой узел.

Доверенная сеть

Сеть ViPNet, с узлами которой узлы собственной сети ViPNet осуществляют защищенное взаимодействие.

Домен коллизий

Часть сети Ethernet, все узлы которой конкурируют за общую разделяемую среду передачи и, следовательно, каждый узел которой может создать коллизию с любым другим узлом этой части сети.

Сеть Ethernet, построенная на повторителях, всегда образует один домен коллизий. Мосты, коммутаторы и маршрутизаторы делят сеть Ethernet на несколько доменов коллизий.

Защищенный узел

Сетевой узел, на котором установлено программное обеспечение ViPNet с функцией шифрования трафика на сетевом уровне.

Источник бесперебойного питания (ИБП)

Автоматическое электронное устройство с аккумуляторной батареей, предназначенное для бесперебойного кратковременного снабжения электрической энергией с целью корректного завершения работы и сохранения данных в случае резкого падения или отсутствия входного питающего напряжения системы.

Класс сетевого интерфейса

Признак, определяющий назначение сетевого интерфейса. В ViPNet Coordinator HW интерфейсам можно назначить следующие классы: `access`, `trunk`, `slave`.

По умолчанию сетевому интерфейсу назначен класс `access`. Если требуется, чтобы интерфейс Ethernet или агрегированный интерфейс обрабатывал трафик из нескольких VLAN, ему необходимо назначить класс `trunk`. Чтобы объединить несколько интерфейсов Ethernet в агрегированный интерфейс, каждому из таких интерфейсов необходимо предварительно назначить класс `slave`.

Кластер горячего резервирования

Кластер горячего резервирования состоит из двух взаимосвязанных ViPNet Coordinator HW, один из которых (активный) выполняет функции координатора сети ViPNet, а другой (пассивный) находится в режиме ожидания. В случае сбоев, критичных для работоспособности ПО ViPNet Coordinator HW на активном ViPNet Coordinator HW, пассивный ViPNet Coordinator HW переключается в активный режим для выполнения функций. При этом сбойный ViPNet Coordinator HW перезагружается и становится пассивным.

Клиент (ViPNet-клиент)

Сетевой узел ViPNet, который является начальной или конечной точкой передачи данных. В отличие от координатора клиент не выполняет функции маршрутизации трафика и служебной информации.

Ключ защиты

Ключ, на котором шифруется другой ключ.

Ключ обмена

Симметричный ключ, известный отправителю и получателю зашифрованной информации, которой обмениваются сетевые узлы ViPNet. Используется для зашифрования и расшифрования передаваемых данных.

Ключи узла ViPNet

Совокупность ключей, с использованием которых производится шифрование трафика, служебной информации и писем программы ViPNet Деловая почта.

Командный интерпретатор

Командная оболочка, предназначенная для администрирования программного обеспечения ViPNet Coordinator HW с помощью ряда специальных команд.

Координатор (ViPNet-координатор)

Сетевой узел ViPNet, представляющий собой компьютер с установленным программным обеспечением координатора (ViPNet Coordinator) или программно-аппаратный комплекс. В сети ViPNet-координатор выполняет серверные функции, а также маршрутизацию трафика и служебной информации.

Маршрут

Путь следования IP-трафика при передаче в сети от одного узла другому.

Маршрут по умолчанию

Путь следования IP-пакетов, для которых не был найден подходящий маршрут в таблице маршрутизации.

Маршрутизатор-сосед

OSPF-маршрутизатор, находящиеся в одной области маршрутизации с другими маршрутизаторами этого типа.

Маршрутизация

Процесс выбора пути для передачи информации в сети.

Межсетевое взаимодействие

Информационное взаимодействие, организованное между сетями ViPNet. Позволяет узлам различных сетей ViPNet обмениваться информацией по защищенным каналам. Для организации взаимодействия между узлами различных сетей ViPNet администраторы этих сетей обмениваются межсетевой информацией.

Межсетевой экран

Устройство на границе локальной сети, служащее для предотвращения несанкционированного доступа из одной сети в другую. Межсетевой экран проверяет весь входящий и исходящий IP-трафик, после чего принимается решение о возможности дальнейшего направления трафика к пункту назначения. Межсетевой экран обычно преобразует внутренние IP-адреса в адреса, доступные из внешней сети (выполняет NAT).

Метрика адреса доступа

Определяет задержку (в миллисекундах) отправки тестовых пакетов при выполнении опроса узла для определения доступности адреса. Предназначена для задания приоритета использования каналов связи.

Метрика маршрута

Параметр, определяющий приоритет маршрута передачи IP-трафика.

Модуль ViPNet Network Visibility System

Функциональный модуль системы управления ViPNet Prime для сбора информации о состоянии устройств.

Модуль ViPNet Policy Management

Функциональный модуль системы ViPNet Prime для управления политиками безопасности устройств защищенной сети ViPNet.

Область маршрутизации

Одна или несколько IP-сетей, в которых происходит обмен информацией по определенному протоколу, в частности, по протоколу [OSPF](#).

Протокол OSPF рассматривает межсетевую среду как множество областей, соединенных друг с другом через некоторую базовую область (backbone area). Для идентификации областей каждой из них выделяется специальный идентификатор (area ID), представляющий собой целое число в десятичном формате.

Открытый узел

Узел без ПО ViPNet с функцией шифрования трафика на сетевом уровне, расположенный в сети «за координатором».

Пароль администратора сетевого узла ViPNet

Пароль для входа на сетевом узле ViPNet в режим администратора, в рамках которого становятся доступны дополнительные возможности настройки приложений ViPNet. Пароль администратора сетевого узла ViPNet может быть создан администратором сети ViPNet в программе ViPNet Удостоверяющий и ключевой центр.

Пароль пользователя

Индивидуальный пароль пользователя для работы в приложениях ViPNet на сетевом узле ViPNet. Первоначально создается администратором сети ViPNet в программе ViPNet Удостоверяющий и ключевой центр или ViPNet Prime. Этот пароль может быть изменен пользователем на сетевом узле ViPNet.

Перераспределение маршрутов

Обмен маршрутной информацией между двумя различными маршрутизирующими протоколами.

Персональный ключ пользователя

Главный ключ защиты ключей, к которым имеет доступ пользователь. Действующий персональный ключ необходимо хранить в безопасном месте.

Прокси-сервер

Программа, транслирующая соединения по некоторым протоколам из внутренней сети во внешнюю и выступающая при этом как посредник между клиентами и сервером.

Резервный набор персональных ключей (РНПК)

Набор из нескольких запасных персональных ключей, которые администратор УКЦ создает для пользователя. Имя этого файла имеет маску `AAAA.pk`, где `AAAA` — идентификатор пользователя ViPNet в рамках своей сети. Используется для удаленного обновления ключей пользователя при их компрометации и при смене мастер-ключа персональных ключей.

Сервер IP-адресов

Функциональность координатора, обеспечивающая регистрацию, рассылку и предоставление информации о состоянии защищенных узлов.

Сервер соединений

Функциональность координатора, обеспечивающая соединение клиентов друг с другом в случае, если они находятся в разных подсетях и не могут соединиться напрямую. Для каждого клиента можно выбрать свой сервер соединений. По умолчанию сервер соединений для клиента также является сервером IP-адресов.

Сетевой интерфейс

Физическое или виртуальное устройство для подключения компьютера к сети. С помощью сетевого интерфейса компьютер принимает и передает IP-пакеты. Физическим интерфейсом может служить сетевая плата, модем и другие подобные устройства, виртуальным — агрегированный интерфейс, интерфейс для VLAN.

Сетевой узел ViPNet

Сетевой узел с ПО ViPNet, зарегистрированный в ViPNet Prime VPN.

Сеть ViPNet

Логическая сеть, организованная с помощью программного обеспечения ViPNet и представляющая собой совокупность защищенных узлов ViPNet. Сеть ViPNet имеет наложенную маршрутизацию, обеспечивающую взаимодействие узлов сети. Каждая сеть ViPNet имеет свой уникальный номер.

Справочники и ключи

Справочники, [ключи узла](#) и ключи пользователя.

Статический сетевой интерфейс

[Сетевой интерфейс](#), для работы которого требуется задать секцию `[adapter]` в файле `iplir.conf` с описанием параметров этого интерфейса. К таким интерфейсам относятся физические (Ethernet) и виртуальные (VLAN) интерфейсы.

Стоимость маршрута

Количество издержек, которые возникнут при отправке IP-пакета в сеть назначения через тот или иной шлюз. Стоимость маршрута обратно пропорциональна его пропускной способности канала связи.

Трансляция сетевых адресов (NAT)

Технология, позволяющая преобразовывать IP-адреса и порты, используемые в одной сети, в адреса и порты, используемые в другой.

Транспортная квитанция

Файл, оповещающий отправителя о невозможности доставки транспортного конверта MFTP.

Транспортный конверт

Зашифрованная информация служб или приложений, доставляемая на защищенные узлы ViPNet транспортным сервером MFTP.

Транспортный сервер MFTP

Компонент координатора, обеспечивающий маршрутизацию транспортных конвертов MFTP между узлами сети ViPNet.

Туннелирование

Технология для защиты соединений между устройствами локальных сетей, которые связаны через интернет или другие публичные сети. Шифрование трафика устройств выполняется координаторами, установленными на границах локальных сетей.

Туннелируемый узел

Узел, на котором не установлено программное обеспечение ViPNet с функцией шифрования трафика на сетевом уровне, но его трафик на потенциально опасном участке сети зашифровывается и расшифровывается на координаторе, за которым он стоит.

Туннелирующий координатор

Координатор, который осуществляет туннелирование.

Файл подкачки

Файл подкачки представляет собой дополнительную виртуальную память, которая позволяет увеличить объем памяти, доступной для использования на компьютере.

Шлюз

Устройство, предназначенное для соединения двух сетей с разными канальными протоколами. Перед передачей данных из одной сети в другую шлюз их преобразует, обеспечивая совместимость протоколов.

Шлюзовой координатор

Координатор, через который происходит обмен транспортными конвертами между сетями ViPNet, установившими межсетевое взаимодействие. Шлюзовые координаторы назначаются в ViPNet ЦУС или Prime каждой сети при организации взаимодействия между двумя различными сетями ViPNet.