

ViPNet Coordinator KB 4

Настройка с помощью командного интерпретатора

Версия продукта: 4.3.3

infotecs

© АО «ИнфоТеКС», 2024

ФРКЕ.465614.001РЭ2

Версия продукта 4.3.3, документ обновлен 28.02.2024

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 **10**
О документе 11
Для кого предназначен документ 11
Соглашения документа 11
Общие сведения о ViPNet Coordinator KB 13
Обратная связь 14

Глава 1. Работа с командным интерпретатором **15**
Основные сведения по работе с командным интерпретатором 16
Интерфейс командного интерпретатора 16
Сокращенный ввод команд 16
Автодополнение команд 17
Контекстная справка 17
Сочетания клавиш 18
Локальное подключение к ViPNet Coordinator KB с помощью консоли 19
Аутентификация пользователя на ViPNet Coordinator KB 19
Включение режима администратора ViPNet Coordinator KB 20
Удаленное подключение к ViPNet Coordinator KB по протоколу SSH 22
Мониторинг неактивных удаленных сессий 23

Глава 2. Обновление программного обеспечения **24**
Подготовка к обновлению программного обеспечения 25
Особенности обновления программного обеспечения 25
Процедура обновления программного обеспечения 27
Обновление программного обеспечения на кластере горячего резервирования 29

Глава 3. Резервное копирование и восстановление настроек **31**
Назначение экспорта и импорта справочников, ключей и настроек 32
Экспорт справочников, ключей и настроек 33
Импорт справочников, ключей и настроек 35

Глава 4. Настройка системных параметров **36**
Настройка даты и времени 37
Контроль целостности файлов ViPNet Coordinator KB 39
Процедура самотестирования и сброс настроек к заводскому состоянию 40

Использование резервного раздела 41

Глава 5. Настройка подключения к сети 42

Настройка сетевых интерфейсов 43

Назначение дополнительных IP-адресов 45

Поддержка технологии VLAN 46

Пример организации обработки трафика из нескольких VLAN 47

Удаление конфигурации VLAN 48

Использование агрегированных сетевых интерфейсов 50

Создание агрегированного сетевого интерфейса 50

Режимы работы агрегированного сетевого интерфейса 53

Примеры организации агрегированного канала связи 55

Организация агрегированного канала связи между ViPNet Coordinator KB и коммутатором 55

Организация агрегированного канала между двумя ViPNet Coordinator KB 58

Глава 6. Настройка VPN 61

Общие принципы настройки VPN 62

Подключение к сети через межсетевой экран 63

Режимы подключения к сети через межсетевой экран 63

Настройка режима «Без использования межсетевого экрана» 64

Настройка режима «Со статической трансляцией адресов» 65

Настройка режима «С динамической трансляцией адресов» 66

Особенности использования ViPNet Coordinator KB в качестве межсетевого экрана 69

Принципы назначения виртуальных адресов 70

Настройка параметров видимости узлов 73

Настройка туннелируемых адресов 75

Задание виртуальных адресов для туннелируемых узлов в автоматическом режиме 76

Задание виртуальных адресов в ручном режиме 77

Настройка видимости туннелируемых узлов 78

Настройка IP-адресов доступа к узлу и их приоритета 80

Настройка туннелирования на канальном уровне по технологии L2OverIP 82

Общее описание технологии L2OverIP 82

Организация защиты соединения между удаленными сегментами сети на канальном уровне модели OSI 84

Примеры настройки туннелирования 86

Организация защищенного взаимодействия открытых узлов (туннелирование) 86

Ограничение доступа туннелируемых узлов 88

Организация туннелей между защищенными и открытыми узлами ... 89
Управление копиями конфигурации VPN ... 92
Виды копий конфигурации VPN ... 92
Создание копии конфигурации VPN ... 92
Просмотр списка копий конфигурации VPN ... 93
Восстановление настроек из копии конфигурации VPN ... 94
Удаление копии конфигурации VPN ... 94
Смена мастер-ключей ... 96

Глава 7. Настройка сетевых фильтров ... 99
Основные принципы фильтрации трафика ... 100
Виды трафика ... 100
Принципы фильтрации трафика ... 101
Общие сведения о сетевых фильтрах ... 104
Группы объектов ... 106
Системные группы объектов ... 107
Пользовательские группы объектов по умолчанию ... 108
Создание пользовательской группы объектов ... 109
Просмотр групп объектов ... 110
Удаление пользовательских групп объектов ... 111
Создание сетевого фильтра ... 112
Адрес отправителя ... 113
Адрес получателя ... 115
Протокол ... 116
Расписание ... 116
Действие ... 117
Примеры создания сетевых фильтров ... 117
Просмотр сетевых фильтров ... 119
Изменение сетевого фильтра ... 121
Удаление сетевого фильтра ... 122

Глава 8. Настройка правил трансляции сетевых адресов ... 123
Трансляция сетевых адресов (NAT) ... 124
Трансляция адреса назначения ... 125
Трансляция адреса источника ... 125
Создание правила трансляции адресов ... 127
Просмотр, изменение, удаление правил трансляции сетевых адресов ... 129
Примеры использования правил трансляции адресов ... 131
Организация доступа пользователей к интернету ... 131

Размещение общедоступного сервера в демилитаризованной зоне 132
Использование дополнительных IP-адресов на интерфейсе одиночного сервера 133
Обеспечение доступа удаленных клиентов к серверу по технологии PPTP VPN 135

Глава 9. Тонкая настройка межсетевого экрана 137
Обработка фрагментированных пакетов в ViPNet Coordinator KB 138
Настройка антиспуфинга 140
Блокирование веб-сайтов по доменным именам с помощью сетевых фильтров 142
Настройка дополнительных параметров межсетевого экрана 143
Обработка сетевого трафика в соответствии с приоритетом 145

Глава 10. Настройка обработки прикладных протоколов 146
О прикладных протоколах 147
Поддерживаемые прикладные протоколы 150
Настройка параметров обработки прикладных протоколов 151

Глава 11. Настройка сетевых служб 154
Настройка параметров DHCP-сервера 155
Настройка DHCP-relay 158
Настройка DNS-сервера 160
Настройка параметров NTP-сервера 162
Пример организации работы клиентов с локальным или удаленным DHCP-сервером. 165
Организация работы клиентов с локальным DHCP-сервером 165
Организация работы клиентов с удаленным DHCP-сервером 166
Организация работы клиентов удаленного офиса с корпоративными DNS- и NTP-серверами 168

Глава 12. Настройка маршрутизации 170
Общие сведения о маршрутизации 171
Принципы формирования таблиц маршрутизации в ViPNet Coordinator KB 172
Принципы формирования таблиц маршрутизации в ViPNet Coordinator KB 175
Просмотр общей таблицы маршрутизации 178
Настройка статической маршрутизации 180
Создание статических маршрутов 180
Удаление статического маршрута 181
Настройка балансировки трафика 181
Просмотр статических маршрутов 182
Настройка динамической маршрутизации 184
Настройка параметров динамических маршрутов от протокола DHCP 184

Настройка административной дистанции для маршрутов DHCP-протокола... 185
Настройка метрики для маршрутов протокола DHCP ... 185
Изменение метрики по умолчанию для маршрутов от протокола DHCP ... 186
Просмотр настроек DHCP в режиме клиента ... 186
Общие сведения о протоколе OSPF ... 187
Настройка параметров динамической маршрутизации по протоколу OSPF ... 189
Настройка протокола OSPF ... 190
Настройка перераспределения маршрутов ... 191
Просмотр настроек протокола OSPF ... 192
Просмотр информации базы данных состояний каналов связи по протоколу OSPF ... 193
Просмотр информации о соседних маршрутизаторах ... 193
Просмотр маршрутов, сформированных по протоколу OSPF ... 194
Настройка функции MultiWAN ... 195
Резервирование каналов доступа в Интернет ... 196
Разделение каналов доступа в интернет для распределения (балансировки) нагрузки на сеть ... 197
Резервирование каналов доступа в интернет с балансировкой трафика ... 199
Обеспечение связи между прокси-сервером и его клиентами, находящимися в разных сетях ... 201

Глава 13. Обеспечение отказоустойчивости ... 203
Настройка системы защиты от сбоев ... 204
Назначение и принципы работы ... 204
Работа системы защиты от сбоев в одиночном режиме ... 204
Работа системы защиты от сбоев в режиме кластера горячего резервирования ... 205
Развертывание кластера горячего резервирования ... 207
Типовая схема организации кластера ... 210
Развертывание кластера совместно с DHCP-сервером и DHCP-relay ... 214
Запуск и завершение работы системы защиты от сбоев ... 214
Переключение кластера в одиночный режим работы ... 215
Замена сбойного пассивного узла кластера на новый без прерывания трафика... 216
Просмотр информации о работе системы защиты от сбоев ... 217
Текущее состояние системы защиты от сбоев ... 217
Журнал переключений ... 219
Работа кластера совместно с коммутационным оборудованием ... 220
Организация электропитания от ИБП ... 221

Глава 14. Настройка транспортного сервера MFTP ... 223

Назначение и функции транспортного модуля ... 224
Выбор канала передачи конвертов ... 225
Настройка взаимодействия с модулем почтового обмена ... 226
Настройка протоколирования событий транспортного сервера ... 227

Глава 15. Ведение и просмотр журналов ... 228
Просмотр журнала регистрации IP-пакетов ... 229
Экспорт журнала регистрации IP-пакетов ... 236
Просмотр журнала транспортных конвертов MFTP ... 238
Работа с системным журналом ... 239
Просмотр системного журнала ... 239
Настройка параметров ведения системного журнала ... 240
Экспорт системного журнала на USB-носитель ... 242
Просмотр журнала приема и применения обновлений ... 243
Просмотр журнала переключений кластера горячего резервирования ... 244
Просмотр журнала СКЗИ ... 245

Глава 16. Мониторинг ViPNet Coordinator KB ... 246
Мониторинг с помощью программного комплекса ViPNet StateWatcher ... 247
Мониторинг по протоколу SNMP ... 248
Описание SNMP-параметров для ПО ViPNet Coordinator KB ... 250
Поддерживаемые базы управляющей информации SNMP ... 253

Приложение A. Базы управляющей информации (MIB) для протокола SNMP ... 255
MIB-II System ... 255
IF-MIB ... 259
IP-MIB ... 262
TCP-MIB ... 268
UDP-MIB ... 271
HOST-RESOURCES-MIB ... 272
UCD-SNMP-MIB ... 278
SNMP-FRAMEWORK-MIB ... 282
VIPNET-MIB ... 283
VIPNETDNSD-MIB ... 285
VIPNETKB-MIB ... 286
MonitoringAgent-MIB ... 286

Приложение B. Сетевые фильтры по умолчанию ... 288

Приложение C. Пользовательские группы протоколов по умолчанию 292

Приложение D. Типы событий в журнале регистрации IP-пакетов 294

Приложение E. События системного журнала, связанные с аутентификацией и настройкой оборудования 300

Приложение F. События журнала СКЗИ 308

Приложение G. Глоссарий 310



Введение

О документе 11

Общие сведения о ViPNet Coordinator KB 13

Обратная связь 14

О документе

В документе описаны основные сценарии настройки программно-аппаратного комплекса ViPNet Coordinator KB 4 (далее — ViPNet Coordinator KB) с помощью командного интерпретатора, а также работа с журналами ViPNet Coordinator KB.

Для кого предназначен документ

Документ предназначен для технических специалистов, отвечающих за настройку и эксплуатацию ViPNet Coordinator KB. Для работы с документом требуется наличие базовых знаний о топологии Ethernet-сетей, сетевых службах и криптографической защите информации.

Соглашения документа

Обозначение	Описание
	**Внимание!** Содержит критически важную информацию
	**Примечание.** Содержит рекомендательную информацию
	**Совет.** Содержит полезные приемы и хорошие практики
Название	Название элемента интерфейса: окна, вкладки, поля, кнопки, ссылки
Клавиша+Клавиша	Сочетание клавиш: нажмите первую клавишу и, не отпуская ее, 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.

При описании команд в данном документе используются следующие условные обозначения:

- Команды, которые могут быть выполнены в режиме администратора, содержат приглашение с символом `#`. Например:

  ```
  hostname# команда
  ```

- Команды, которые могут быть выполнены в режиме пользователя, содержат приглашение с символом `>`. Например:

  ```
  hostname> команда
  ```

- Обязательные параметры, которые должны быть заданы, заключены в угловые скобки. Например:

  ```
  команда <параметр>
  ```

- Необязательные параметры или ключевые слова заключены в квадратные скобки. Например:

  ```
  команда <обязательный параметр> [необязательный параметр]
  ```

- Если при вводе команды можно указать один из нескольких параметров, допустимые варианты заключены в фигурные скобки и разделены вертикальной чертой. Например:

  ```
  команда {вариант-1 | вариант-2}
  ```

Общие сведения о ViPNet Coordinator KB

ViPNet Coordinator KB — это криптошлюз, обеспечивающий информационную безопасность и конфиденциальность при передаче информации, а также безопасный доступ пользователей защищенной сети к ресурсам сетей общего пользования.

Представляет собой средство криптографической защиты информации в IP-сетях на базе специализированной аппаратной платформы и специального программного обеспечения ViPNet ViPNet Coordinator KB (далее — ПО ViPNet Coordinator KB), которое функционирует под управлением адаптированной ОС GNU/Linux.

ViPNet Coordinator KB используется в составе комплекса программных продуктов ViPNet и выполняет роль координатора ViPNet, а также выполняет ряд дополнительных функций.

Полное описание технических характеристик и выполняемых функций ViPNet Coordinator KB 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.

  Канал поддержки в Telegram: 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 компании ИнфоТеКС регулируется политикой ответственного разглашения.

1

Работа с командным интерпретатором

Основные сведения по работе с командным интерпретатором 16

Локальное подключение к ViPNet Coordinator KB с помощью консоли 19

Удаленное подключение к ViPNet Coordinator KB по протоколу SSH 22

Основные сведения по работе с командным интерпретатором

Все операции по настройке ViPNet Coordinator KB вы можете выполнить с помощью командного интерпретатора. Работа в командном интерпретаторе возможна при локальном подключении к ViPNet Coordinator KB (см. Локальное подключение к ViPNet Coordinator KB с помощью консоли на стр. 19).

Командный интерпретатор запускается автоматически после успешной аутентификации пользователя (см. Аутентификация пользователя на ViPNet Coordinator KB на стр. 19). После запуска командный интерпретатор находится в режиме пользователя (доступен ограниченный набор команд управления, обозначается символом «>» в приглашении командного интерпретатора). Для управления ViPNet Coordinator KB требуется перейти в режим администратора (см. Включение режима администратора ViPNet Coordinator KB на стр. 20).

Чтобы вернуться в режим пользователя, выполните команду `exit`.

Чтобы выйти из командного интерпретатора, выполните команду `exit`. В случае выхода из командного интерпретатора на консоли снова появится приглашение для аутентификации пользователя.

Интерфейс командного интерпретатора

Командный интерпретатор принимает текстовые команды, состоящие из слов, разделенных пробелами и состоящими из латинских и специальных символов. Для облегчения ввода команд в этом интерпретаторе предусмотрены следующие средства:

- сокращенный ввод команд (на стр. 16);
- автодополнение команд (на стр. 17);
- контекстная справка (на стр. 17);
- сочетания клавиш (на стр. 18).

Сокращенный 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 введенных символов есть ошибочные, то вся команда считается ошибочной, даже если ее можно однозначно определить по первым правильно введенным символам (то есть в приведенном выше примере команда `iplor sta` будет ошибочной).

Автодополнение команд

Если введенные 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 задаете (например, имена создаваемых объектов).

Контекстная справка

Контекстная справка позволяет 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            control command for Failover daemon
iplir               control command for IpLir daemon
webui               control WebUI service
mftp                commands for managing the 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
machine             a group of system commands
ups
firewall            firewall's objects commands
alg                 control the properties of the application-level gateway
hostname> _
```

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

В случае ввода символа «?» в процессе набора команды, командный 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 ?
show                a group of commands for viewing system settings
reboot              reboot the software and hardware appliance
```

```
halt                turn off the software and hardware appliance
hostname> machine _
```

После информации о вариантах завершения команды отображается приглашение командного интерпретатора с ранее введенной командой для редактирования. Редактировать 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 все 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 KB с помощью консоли

Для работы с ViPNet Coordinator KB с помощью консоли подключите к нему монитор и клавиатуру и выполните первоначальную настройку ViPNet Coordinator KB. Если первоначальная настройка уже была проведена, то выполните аутентификацию пользователя (см. Аутентификация пользователя на ViPNet Coordinator KB на стр. 19).



Примечание. Если истекает или уже истек срок действия персонального ключа пользователя или ключевого блокнота ДСДР, то на экране монитора появится предупреждающее сообщение:

`> User keys expired on <Date>`, где `<Date>` — дата истечения срока действия.

При возникновении такого сообщения обратитесь к администратору ViPNet Coordinator KB для обновления ключей.

Если требуется выполнить настройку параметров ViPNet Coordinator KB, перейдите в режим администратора (см. Включение режима администратора ViPNet Coordinator KB на стр. 20).

Аутентификация пользователя на ViPNet Coordinator KB

Для начала работы в командном интерпретаторе ViPNet Coordinator KB необходимо пройти процедуру аутентификации. Для этого вам понадобится токен, на котором должен быть сохранен ваш персональный ключ пользователя и PIN-код доступа к этому токену.



Примечание. Токен с персональным ключом пользователя и PIN-кодом к нему вы можете получить у администратора сети ViPNet.

Командный интерпретатор загружается сразу после загрузки ПО ViPNet Coordinator KB. На приглашение командного интерпретатора пройти аутентификацию выполните следующие действия:

1 Подключите к одному из USB-разъемов ViPNet Coordinator KB токен, на который сохранен ваш персональный ключ пользователя.



Внимание! Не отключайте токен в процессе работы с ViPNet Coordinator KB, поскольку это может привести к ошибке обмена ключевой информацией и последующей блокировке командного интерпретатора. Всегда завершайте сеанс работы перед отключением токена.

2 Когда отобразится сообщение `Enter PIN`, введите PIN-код доступа к подключенному токену.

   Если введен неверный PIN-код, аутентификация не будет выполнена. Если вы три раза подряд ввели неправильный PIN-код, то клавиатура будет заблокирована на несколько секунд, при этом после каждой неудачной попытки время задержки будет увеличиваться.

   Задержка нужна для предотвращения подбора пароля методом перебора, все неуспешные попытки ввода PIN-кода фиксируются в системном журнале. Счетчик попыток обнуляется после успешного ввода PIN-кода либо после десятой ошибки.

   В случае успешной аутентификации командный интерпретатор будет запущен в режиме пользователя. Вы можете начать работу с ViPNet Coordinator KB.

О том, как перейти в режим администратора, см. раздел Включение режима администратора ViPNet Coordinator KB (на стр. 20).

Для завершения сеанса работы с ViPNet Coordinator KB выполните команду `exit`, отключите токен и убедитесь, что сессия корректно завершилась, о чем будет свидетельствовать появление сообщения `Enter PIN`.

Включение режима администратора ViPNet Coordinator KB

Чтобы перейти в режим администратора:

1 Выполните команду `enable`.

2 Введите пароль администратора ViPNet Coordinator KB. Процесс ввода пароля не отображается на экране. Если введен неверный пароль, на консоли появится соответствующее сообщение и приглашение для повторной авторизации.



Примечание. Пароль администратора ViPNet Coordinator KB вы можете получить у администратора сети ViPNet.

Для предотвращения подбора пароля методом перебора в ViPNet Coordinator KB 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 KB 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 — на консоли появится символ `#` в строке приглашении. В этом режиме вы можете изменять те настройки, которые были недоступны в режиме пользователя.

На ViPNet Coordinator KB могут авторизоваться три типа администраторов:

- администратор ViPNet Coordinator KB;
- администратор группы сетевых узлов ViPNet;
- администратор сети ViPNet.

Чтобы вернуться в режим пользователя, используйте сочетание клавиш **Ctrl+D** или выполните команду `exit`.



Примечание. Пароль администратора ViPNet Coordinator KB можно изменить в программе ЦУС, либо локально в режиме администратора при помощи команды:

```
hostname# admin passwd
```

Удаленное подключение к ViPNet Coordinator KB по протоколу SSH

Удаленное подключение к ViPNet Coordinator KB по протоколу SSH допускается только из командного интерпретатора другого ViPNet Coordinator KB, входящего в ту же сеть ViPNet и для которого созданы соответствующие разрешающие правила межсетевого экрана. Для работы по протоколу SSH в командном интерпретаторе ViPNet Coordinator KB имеется встроенный SSH-клиент.

Для подключения к удаленному ViPNet Coordinator KB по протоколу SSH вам потребуются следующие параметры подключения, которые можно получить у администратора сети ViPNet:

- IP-адрес удаленного ViPNet Coordinator KB или его ViPNet ID (идентификатор узла в сети ViPNet);
- имя пользователя и пароль удаленного ViPNet Coordinator KB;
- пароль администратора удаленного ViPNet Coordinator KB для включения режима администратора.

Чтобы подключиться к удаленному ViPNet Coordinator KB по протоколу SSH, выполните следующие действия:

1. На узле, с которого вы будете удаленно подключаться к ViPNet Coordinator KB, выполните аутентификацию, используя заданный администратором сети ViPNet способ аутентификации.
2. Выполните команду `hostname# inet ssh`, указав необходимые параметры подключения и пароль. Подробнее об использовании команды `inet ssh` см. в документе «Справочник команд и конфигурационных файлов».
3. Для включения режима администратора на удаленном ViPNet Coordinator KB выполните команду `enable` и введите пароль администратора удаленного ViPNet Coordinator KB.

   Если на удаленном ViPNet Coordinator KB запущена другая сессия в режиме администратора, то прервите ее (введите `Yes` и нажмите **Enter**). Иначе попытка удаленного подключения будет завершена.

   Если в течение 30 секунд вы не подтвердите подключение к ViPNet Coordinator KB в режиме администратора, то удаленная сессия также будет завершена.

Примечание. Если во время удаленной сессии с ViPNet Coordinator KB вы не будете выполнять никаких действий в течение 30 минут, то сессия будет прервана.



Если на ViPNet Coordinator KB истекают или уже истекли сроки действия персонального ключа пользователя или ключей защиты ключей обмена, то на консоли появится соответствующее сообщение. Обратитесь к администратору сети ViPNet для обновления ключей.

Мониторинг неактивных удаленных сессий

В ViPNet Coordinator KB количество удаленных сессий не ограничено, поэтому при одновременном удаленном подключении по протоколу SSH большого количества пользователей нагрузка на ViPNet Coordinator KB значительно увеличивается. Во избежание сбоев в работе производится мониторинг активности пользователей в удаленных сессиях.

Проверка активности пользователя в каждой удаленной сессии осуществляется один раз в 5 минут. Если пользователь не выполняет никаких действий в течение 30 минут, его сессия 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 KB в режиме кластера горячего резервирования указанные настройки передаются с активного узла кластера на пассивный.



2

Обновление программного обеспечения

Подготовка к обновлению программного обеспечения 25

Процедура обновления программного обеспечения 27

Обновление программного обеспечения на кластере горячего резервирования 29

Подготовка к обновлению программного обеспечения

Обновление специального программного обеспечения на ViPNet Coordinator KB выполняется локально. Прежде чем начать локальное обновление ПО ViPNet Coordinator KB выполните следующие действия:

1 Ознакомьтесь с особенностями и процедурой обновления программного обеспечения (на стр. 27).

2 У администратора сети ViPNet получите файл обновления `*.lzh` и запишите его на USB-носитель, отформатированный в одной из следующих файловых систем: `fat32, ext3`. Полный путь к файлу обновления не должен содержать специальные символы и пробел.



Примечание. На один USB-носитель можно записать несколько файлов обновления, предназначенных для разных исполнений ViPNet Coordinator KB. При выполнении локального обновления производится анализ содержимого USB-носителя и становится доступным для использования только тот дистрибутив, который соответствует данному исполнению ViPNet Coordinator KB.

3 Выполните процедуру самотестирования ViPNet Coordinator KB с помощью команды:

```
hostname# machine self-test
```

4 Выполните экспорт справочников, ключей и настроек ViPNet Coordinator KB в файл экспорта `*.vbe`. Это позволит вам восстановить их на ViPNet Coordinator KB, если обновление ПО пройдет некорректно.

Особенности обновления программного обеспечения

При обновлении ПО ViPNet Coordinator KB необходимо учитывать следующие особенности:

- В версии 4.x для аутентификации пользователя используется учетная запись `user`, тогда как в более ранних версиях — учетная запись vipnet. После обновления ПО до версии 4.x используйте для аутентификации новую учетную запись.
- Возможно понижение версии ViPNet Coordinator KB до версии 4.3.0. Процедура понижения версии не отличается от процедуры обновления ПО. При необходимости перейти на более старую версию ПО ViPNet Coordinator KB обратитесь к представителю АО «ИнфоТеКС».
- При обновлении ViPNet Coordinator KB с версии 4.2.x до версии 4.3.x из конфигурационного файла `failover.ini` будут удалены параметры `afterifconf`, `beforeifconf` и `passiveip`, так как они больше не используются.

- Перед обновлением с версии 4.2.x до 4.3.x убедитесь, в наличии не менее 1 Гб свободной оперативной памяти с помощью команды:

  ```
  hostname> machine show memory
  ```

- Если значение свободной памяти, отображаемое в столбце `free`, менее 1 Гб, то перезагрузите ViPNet Coordinator KB перед обновлением.

- При обновлении ПО ViPNet Coordinator KB с версии 4.2.x до версии 4.3.0 будет потеряна связь между обновляемым ViPNet Coordinator KB и защищенными узлами сети ViPNet. Чтобы восстановить связи:

  4.1 Перезапустите службу защиты от сбоев с помощью команд:

  ```
  hostname> failover stop
  ```

  ```
  hostname> failover start
  ```

  4.2 Повторно установите ключевой блокнот ДСДР. Подробнее см. в разделе «Установка ключевого блокнота ДСДР» в документе «Подготовка к работе».



Внимание! Перед обновлением ПО ViPNet Coordinator KB с версии 4.2.x до версии 4.3.x убедитесь в наличии диска ДСДР.

Процедура обновления программного обеспечения

Внимание!



- При использовании кластера горячего резервирования обновление ПО ViPNet Coordinator KB должно быть произведено на обоих узлах кластера (см. Обновление программного обеспечения на кластере горячего резервирования на стр. 29).
- Не отключайте токен и питание ViPNet Coordinator KB до полного окончания процесса обновления ПО. Это может привести к неработоспособности ViPNet Coordinator KB.

Для обновления ПО ViPNet Coordinator KB выполните следующие действия:

1 На ViPNet Coordinator KB выполните:

```
hostname# admin upgrade software usb
```

В результате выполнения этой команды появится предложение подключить USB-носитель.

2 Подключите USB-носитель к USB-разъему ViPNet Coordinator KB и подтвердите подключение.

Если USB-носитель не будет найден, появится соответствующее сообщение, выполнение команды завершится, процедура обновления ПО остановится.

3 На USB-носителе автоматически будет произведен поиск файлов `*.lzh`:

- Если файлов нет, появится соответствующее сообщение и выполнение команды завершится, а процедура обновления ПО остановится.
- Если файлы найдены, то будет проверено соответствуют ли они данному исполнению ViPNet Coordinator KB, если ни один файл не соответствует данному исполнению, процедура обновления ПО остановится.
- На экран выведется пронумерованный список отобранных файлов. Выберите файл, в имени которого указана нужная версия обновления ПО ViPNet Coordinator KB.

4 Введите номер выбранного файла обновления и нажмите **Enter**.

Будет проверена совместимость выбранной версии обновления ПО ViPNet Coordinator KB с версией ПО ДНСД. Если версия обновления ПО не совместима с версией ПО ДНСД, процедура обновления ПО остановится.

5 Введите технологический ключ контроля целостности (ТККС) из бюллетеня на обновление ПО.

- На экран будет выведено значение MAC (Message Authentication Code) обновления ПО.
- Подтвердите, что выведенное на экран значение MAC соответствует значению, указанному в бюллетене на обновление ПО, для чего введите `Yes`.
- В случае если выведенное на экран значение MAC не соответствует значению, указанному в бюллетене, введите `No`, при этом процедура обновления ПО будет отменена.

6 Дождитесь появления сообщения о результате обновления.

7 После обновления подтвердите перезагрузку ViPNet Coordinator KB, чтобы обновление вступило в силу.



Примечание. После успешного обновления ViPNet Coordinator KB до версии 4.3.0 убедитесь в наличии связи с узлами сети ViPNet. Если связь с узлами отсутствует, то может потребоваться установка новой серии ключевого блокнота ДСДР.

Обновление программного обеспечения на кластере горячего резервирования

Для обновления ПО ViPNet Coordinator KB на узлах кластера горячего резервирования выполните следующие действия:

1. Выключите сетевой интерфейс, подключенный к каналу синхронизации кластера, с помощью команды:

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

2. Отсоедините кабель от сетевого интерфейса синхронизации.

3. Обновите ПО на пассивном узле (так же, как при работе в одиночном режиме):

   ```
   hostname# admin upgrade software usb
   ```

   Информацию о текущем режиме ViPNet Coordinator KB можно получить с помощью команды `hostname> failover show info`.

4. После обновления перезагрузите пассивный узел. Для этого в ответ на предложение перезагрузки нажмите Enter.

   Убедитесь в стабильной работе ViPNet Coordinator KB:

   - Проверьте текущее состояние служб ViPNet с помощью команды:

     ```
     hostname> failover show info
     ```

     Все службы должны быть запущены, система защиты от сбоев должна работать в режиме кластера.

   - Убедитесь, что в течение 15 минут пассивный узел не перезагружается.

5. Перезагрузите активный узел:

   ```
   hostname> machine reboot
   ```

   В результате пассивный узел (с обновленным ПО) перейдет в активный режим, а узел со старой версией ПО окажется в пассивном режиме.

6. Обновите ПО на втором узле, который стал пассивным:

   ```
   hostname# admin upgrade software usb
   ```

7. После обновления перезагрузите пассивный узел и убедитесь в его стабильной работе (как на шаге 4).

8. Подсоедините кабель канала синхронизации к сетевому интерфейсу.

9. Включите сетевой интерфейс, подключенный к каналу синхронизации кластера:

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

10 Убедитесь в функционировании канала синхронизации. Для этого измените какую-либо настройку в конфигурационном файле на активном узле и через некоторое время проверьте наличие этих же изменений на пассивном узле. Чтобы изменения попали на пассивный узел, необходимо включить резервирование группы конфигурационных файлов в файле `failover.ini` в секции `[sendconfig]`.



Внимание! При обновлении ПО ViPNet Coordinator KB с версии 4.2.X до версии 4.3.X из конфигурационного файла `failover.ini` будут удалены параметры `afterifconf`, `beforeiconf` и `passiveip`, отвечающие за настройку сетевых интерфейсов узлов кластера, так как они больше не используются. Поэтому, после обновления ПО на узлах кластера необходимо заново выполнить настройку сетевых интерфейсов (см. Развертывание кластера горячего резервирования на стр. 207).

3

Резервное копирование и восстановление настроек

Назначение экспорта и импорта справочников, ключей и настроек 32

Экспорт справочников, ключей и настроек 33

Импорт справочников, ключей и настроек 35

Назначение экспорта и импорта справочников, ключей и настроек

Вы можете использовать экспорт и импорт справочников, ключей и настроек ViPNet Coordinator KB в следующих ситуациях:

- Сохранение и восстановление данных на ViPNet Coordinator KB в случае некорректного обновления ПО.
- Перенос справочников, ключей и настроек с одного сервера ViPNet Coordinator KB на другой, например, в случае замены аппаратной платформы. Это позволяет упростить подготовку нового ViPNet Coordinator KB к работе, так как не требуется выполнять установку справочников и ключей и необходимые настройки вручную.

  В этом случае оба сервера ViPNet Coordinator KB должны иметь одинаковые исполнение и тип лицензии. Кроме этого, по завершении импорта сервер, 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 KB узлов (информация ключевого блокнота ДСДР не входит в состав экспортируемой информации);
- резервный набор персональных ключей (файл `*.pk`, при наличии);
- файлы, содержащие информацию о связанных с ViPNet Coordinator KB узлах (справочники);
- настройки защищенной сети (конфигурационный файл `iplir.conf`);
- настройки сетевых интерфейсов (IP-адрес, маска подсети, конфигурационный файл `iplir.conf-<интерфейс>`);
- настройки маршрутизации и статические маршруты, если такие созданы;
- настройки часового пояса;
- сетевые фильтры и правила трансляции сетевых адресов;
- настройки параметров обработки прикладных протоколов;
- настройки системы защиты от сбоев (конфигурационный файл `failover.ini`);
- настройки транспортного модуля (конфигурационный файл `mftp.conf`);
- настройки протоколирования;
- настройки дополнительных сервисов (DHCP-сервера и других);
- настройки взаимодействия с источником бесперебойного питания (UPS);
- настройки туннелирования канального уровня (L2OverIP).

Экспорт справочников, ключей и настроек

Экспорт справочников, ключей и настроек требуется выполнять в следующих случаях:

- Перед обновлением ПО на ViPNet Coordinator KB (см. Обновление программного обеспечения на стр. 24), а также периодически для возможности восстановления данных в случае сбоев в работе.

  Примечание. Периодичность резервирования данных на ViPNet Coordinator KB определяется регламентом безопасности вашей организации.

- При переносе справочников, ключей и настроек на другой ViPNet Coordinator KB аналогичного исполнения (например, при замене аппаратной платформы).

Вы можете выполнить экспорт справочников, ключей и настроек только при непосредственном доступе к ViPNet Coordinator KB с использованием монитора и клавиатуры. Для этого вам понадобится USB-носитель, отформатированный в одну из поддерживаемых файловых систем: FAT32, ext2, ext3 или ext4.

Примечание. Во избежание получения неработоспособного файла экспорта рекомендуется предварительно выполнить процедуру самотестирования ViPNet Coordinator KB с помощью команды `hostname# machine self-test`.

Для корректного восстановления настроек в режиме кластера горячего резервирования экспорт справочников и ключей необходимо выполнять одновременно на двух узлах кластера.

Для экспорта справочников, ключей и настроек:

1 Завершите работу служб `iplircfg` и `mftpd` с помощью команд:

```
hostname> iplir stop
hostname> mftp stop
```

2 Выполните сохранение файла экспорта:

```
hostname# admin export keys binary-encrypted usb
```

При успешном выполнении команды файл экспорта будет сформирован и сохранен в каталоге `/tmp/vipnet`.

Примечание. Имя файла экспорта формируется по следующему шаблону: `<название ViPNet Coordinator KB>-<идентификатор узла>-<дата экспорта>.vbe`,

где `<название ViPNet Coordinator KB>` — наименование исполнения без последних двух символов.

Файл экспорта автоматически зашифровывается на текущем пароле пользователя.

3 Перенесите сформированный файл экспорта на USB-носитель:

3.1 Подготовьте USB-носитель, отформатированный в одной из файловых систем: `Fat32`, `ext3`.

3.2 Подключите подготовленный USB-носитель к одному из USB-разъемов ViPNet Coordinator KB в ответ на предложение:

```
Put <имя файла экспорта> file onto USB drive.

Insert USB drive and press Enter.
```

3.3 Появится пронумерованный список всех обнаруженных устройств. Выберите USB-носитель для переноса на него файла экспорта. Для этого в ответ на сообщение `Select target partition [1-1] or 0 to abort` введите цифру, соответствующую USB-носителю в списке, и нажмите **Enter**.

3.4 При успешной записи файла экспорта на USB-носитель появится сообщение с разрешением извлечь устройство:

```
You may remove the USB drive.
```

Внимание! Необходимо дождаться этого сообщения прежде, чем извлечь USB-носитель из разъема.

4 Запустите службы `iplircfg` и `mftpd`:

```
hostname> iplir start

hostname> mftp 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 KB (см. Обновление программного обеспечения на стр. 24).
- Для загрузки справочников, ключей и настроек с одного ViPNet Coordinator KB на другой (например, при замене аппаратной платформы).

Перед началом импорта убедитесь в наличии файла *.vbe, полученного в результате экспорта справочников, ключей и настроек на действующем ViPNet Coordinator KB (на стр. 33). Вместе с файлом *.vbe вам должен быть предоставлен пароль, на котором он зашифрован. Данный пароль потребуется ввести при выполнении импорта данных из файла экспорта *.vbe.



Примечание. При экспорте файл *.vbe автоматически зашифровывается на текущем пароле пользователя.

Кроме этого, если импорт справочников, ключей и настроек выполняется на ViPNet Coordinator KB, на котором уже установлены справочники и ключи, то удалите их перед началом импорта с помощью команды:

```
hostname# admin keys remove
```

Иначе, импорт данных будет невозможен.

Для импорта справочников, ключей и настроек на ViPNet Coordinator KB выполните все действия из нижеприведенной таблицы в предложенном порядке.

Таблица 1. Последовательность импорта справочников, ключей и настроек

	Действие	Ссылка
☐	Инициируйте импорт справочников, ключей и настроек на ViPNet Coordinator KB	Начало установки
☐	Укажите часовой пояс, дату и время	Настройка часового пояса, даты и времени
☐	Выберите файл экспорта *.vbe	Установка дистрибутива ключей на ViPNet Coordinator KB
☐	Перезагрузите ViPNet Coordinator KB	Завершение установки
☐	Установите ключевой блокнот ДСДР	Установка ключевого блокнота ДСДР

4

Настройка системных параметров

Настройка даты и времени 37

Контроль целостности файлов ViPNet Coordinator KB 39

Процедура самотестирования и сброс настроек к заводскому состоянию 40

Использование резервного раздела 41

Настройка даты и времени

Для корректного взаимодействия ViPNet Coordinator KB с другими защищенными узлами ViPNet, необходимо правильно настроить системные дату и время.



Внимание! Если системные дата и время заданы неверно, защищенные соединения с другими узлами ViPNet могут блокироваться.

Рекомендуется настроить синхронизацию системного времени по протоколу NTP.

Чтобы настроить системные дату и время вручную, выполните:



Примечание. При работе в режиме кластера горячего резервирования установка системных даты, времени и часового пояса вручную (команды `machine set date`, `machine set timezone`) возможна только на активном узле кластера. Параметры времени и часового пояса на пассивном узле кластера синхронизируются с активным узлом и недоступны для изменения.

1 Чтобы просмотреть текущее время:

```
hostname> machine show date
```

2 Чтобы изменить часовой пояс:

```
hostname# machine set timezone {<Континент/Пояс> | <UTC>}
```



Примечание. Название континента и часового пояса должны начинаться с прописной буквы. Например, чтобы установить часовой пояс Москвы, выполните:

```
hostname# machine set timezone Europe/Moscow
```

Чтобы посмотреть список всех часовых поясов, введите данную команду без параметра.

3 Чтобы изменить системное время:

3.1 Остановите службы `iplircfg`, `failoverd` и `mftpd`:

```
hostname> iplir stop
hostname> failover stop
hostname> mftp stop
```

3.2 Настройте 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.

3.3 Запустите службы `iplircfg`, `failoverd` и `mftpd`:

```
hostname> iplir start
```

```
hostname> failover start
hostname> mftp start
```

Контроль целостности файлов ViPNet Coordinator KB

Во избежание возникновения сбоев ПО целостность важных файлов ViPNet Coordinator KB регулярно проверяется с помощью встроенных скриптов. Контролируются следующие компоненты:

- Ядро ОС и образы модулей ViPNet Coordinator KB, используемые при загрузке ПО.
- Системные службы и библиотеки.
- Конфигурационные файлы служб ViPNet Coordinator KB.
- Справочники и ключи ViPNet.

Проверка целостности файлов автоматически выполняется в следующих случаях:

- Обновление ПО ViPNet Coordinator KB.
- Загрузка ПО ViPNet Coordinator KB.
- Периодическая проверка с интервалом 12 часов.

Кроме того, вы можете проверить целостность конфигурационных файлов вручную. Для этого запустите процедуру самотестирования (см. Процедура самотестирования и сброс настроек к заводскому состоянию на стр. 40).

Нарушением целостности файлов также считается отсутствие любого конфигурационного файла или файла с контрольными суммами.

При несовпадении контрольных сумм в ходе периодической проверки ViPNet Coordinator KB автоматически перезагружается.

При несовпадении контрольных сумм в ходе проверки во время 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 (см. Использование резервного раздела на стр. 41), файл будет автоматически восстановлен. При этом будет выведено соответствующее сообщение. При восстановлении ViPNet Coordinator KB может быть перезагружен.
- Если резервная копия искаженного конфигурационного файла не найдена, будет выведено соответствующее сообщение. После этого службы, отвечающие за работу в сети ViPNet, не будут запущены либо будут остановлены и сетевые интерфейсы будут выключены.
- Если при загрузке обнаружено нарушение целостности ядра ОС или одного из образов модулей ViPNet Coordinator KB, и соответствующая резервная копия не найдена, то будет выведено соответствующее сообщение и загрузка ViPNet Coordinator KB завершится. Для выключения ViPNet Coordinator KB нажмите любую клавишу.

Процедура самотестирования и сброс настроек к заводскому состоянию

Для подтверждения неизменности ПО ViPNet Coordinator KB в процессе эксплуатации вам необходимо периодически проводить регламентный контроль целостности ViPNet Coordinator KB. Важной частью регламентного контроля является процедура самотестирования ViPNet Coordinator KB.

Чтобы запустить процедуру самотестирования, выполните:

```
hostname# machine self-test
```



Примечание. Вы можете выполнить эту команду только при подключении к ViPNet Coordinator KB через обычную консоль. Выполнение команды в удаленной SSH-сессии невозможно.

По завершению процедуры самотестирования будет рассчитана и выведена на экран контрольная сумма ПО ViPNet Coordinator KB.

- Если выведенная на экран контрольная сумма ПО ViPNet Coordinator KB совпадает с контрольной суммой, зафиксированной при предыдущем регламентном контроле, то в ответ на сообщение `Is it correct? [Yes;No]` введите `Yes` и нажмите **Enter**. В результате будет создан новый ключ регламентного контроля, при использовании которого будет произведен расчет контрольной суммы ПО ViPNet Coordinator KB. Для проведения следующего регламентного контроля вам нужно зафиксировать новое значение рассчитанной контрольной суммы.

  По завершении процедуры самотестирования необходимо перезагрузить ViPNet Coordinator KB. Для этого в ответ на сообщение `Press enter to reboot` нажмите **Enter**.

- Если выведенная на экран контрольная сумма ПО не совпадает с контрольной суммой, зафиксированной при предыдущем регламентном контроле, то в ответ на сообщение `Is it correct? [Yes;No]` введите `No` и нажмите **Enter**. ViPNet Coordinator KB автоматически запустит выполнение процедуры сброса настроек к заводскому состоянию. Для продолжения нажмите **Enter**. После завершения процедуры ViPNet Coordinator KB будет находится в состоянии, при котором вы можете заново выполнить процедуру ввода ViPNet Coordinator KB в эксплуатацию.

Подробное описание процедуры регламентного контроля описано в документе «ViPNet Coordinator KB 4. Правила пользования», входящем в комплект поставки ViPNet Coordinator KB, в разделе «Регламентный контроль».

Использование резервного раздела

Для возможности оперативного восстановления критически важных компонентов ViPNet Coordinator KB, целостность которых регулярно проверяется (см. Контроль целостности файлов ViPNet Coordinator KB на стр. 39), такие файлы периодически копируются на специальный резервный раздел накопителя ViPNet Coordinator KB.

Копии файлов сохраняются в резервный раздел `/mnt/reserv` того накопителя ViPNet Coordinator KB, с которого не происходит загрузка ПО. При проверке целостности критически важных файлов искаженные файлы автоматически восстанавливаются из своих резервных копий.

Копирование файлов в резервный раздел выполняется в следующих случаях:

- Успешная проверка целостности файлов при первоначальной установке и обновлении ПО либо при выполнении контроля целостности (см. Контроль целостности файлов ViPNet Coordinator KB на стр. 39).
- Успешная проверка целостности при загрузке ViPNet Coordinator KB.
- Обновление справочников и ключей.
- Изменение пароля пользователя.

Если два ViPNet Coordinator KB работают в режиме кластера горячего резервирования, то резервные копии файлов регулярно копируются с активного сервера на пассивный. При изменении справочников и ключей на кластере, копии справочников и ключей обновляются как на активном, так и на пассивном узле.



5

Настройка подключения к сети

Настройка сетевых интерфейсов 43

Назначение дополнительных IP-адресов 45

Поддержка технологии VLAN 46

Использование агрегированных сетевых интерфейсов 50

Настройка сетевых интерфейсов

Сетевым интерфейсам Ethernet, установленным в системе, присваиваются имена `eth0`, `eth1` и так далее (по количеству таких интерфейсов в системе). Чтобы настроить подключение к сети, задайте параметры сетевых интерфейсов. Для этого выполните следующие действия:

1 По умолчанию сетевому интерфейсу назначен класс `access`. При необходимости смените класс с помощью команды:

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

Если требуется, чтобы данный сетевой интерфейс обрабатывал трафик из нескольких VLAN (см. Поддержка технологии VLAN на стр. 46), назначьте ему класс `trunk`.

Если вы хотите объединить данный интерфейс с другими в составе агрегированного интерфейса (см. Использование агрегированных сетевых интерфейсов на стр. 50), назначьте ему класс `slave`.

2 Выполните одно из действий:

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

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

  Также для режима DHCP вы можете задать дополнительные параметры:

  - включить или выключить автоматическое получение адресов DNS-серверов:

    ```
    hostname# inet ifconfig <имя интерфейса> dhcp dns {on | off}
    ```

  - включить или выключить автоматическое получение адресов NTP-серверов:

    ```
    hostname# inet ifconfig <имя интерфейса> dhcp ntp {on | off}
    ```

  - включить или выключить автоматическое получение маршрутов от DHCP-сервера:

    ```
    hostname# inet ifconfig <имя интерфейса> dhcp route {on | off}
    ```

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

    ```
    hostname# inet ifconfig <имя интерфейса> dhcp route-metric {<1-255> |
    none}
    ```

  По умолчанию для всех дополнительных параметров режима DHCP установлено значение `on (none` для метрики маршрутов, полученных от DHCP-сервера`)`.

  Вы можете задать ряд параметров, которые будут присваиваться маршрутам DHCP-сервера (см. Настройка параметров динамических маршрутов от протокола DHCP на стр. 184). Вы можете просмотреть параметры, которые заданы для режима DHCP на всех сетевых интерфейсах, (см. Просмотр настроек DHCP в режиме клиента на стр. 186).



Внимание! Если ни для одного включенного сетевого интерфейса ViPNet Coordinator KB не было задано автоматическое получение маршрутов по DHCP, добавьте статический маршрут по умолчанию (см. Создание статических маршрутов на стр. 180).

- Чтобы присвоить сетевому интерфейсу статический IP-адрес, выполните:

  ```
  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.

3 По умолчанию для передачи данных через сетевой интерфейс используется размер MTU, равный 1500 байт. Чтобы оптимизировать передачу данных через интерфейс, вы можете изменить размер MTU. Для этого задайте новое значение MTU из диапазона 1280–9000 байт с помощью команды:

```
hostname# inet ifconfig <имя интерфейса> MTU {<значение MTU> | auto}
```

Чтобы вернуть значение по умолчанию, используйте параметр `auto`.

4 Включите сетевой интерфейс:

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

5 При необходимости задайте дополнительные IP-адреса для сетевого интерфейса (см. Назначение дополнительных IP-адресов на стр. 45).

6 При необходимости выполните настройку маршрутизации (см. Настройка маршрутизации на стр. 170).



Внимание! Если хотя бы на одном используемом сетевом интерфейсе ViPNet Coordinator KB задано автоматическое получение маршрутов от DHCP-сервера, проверьте, что был получен маршрут по умолчанию. Если маршрут по умолчанию не был получен, будет выведено соответствующее предупреждение. В этом случае задайте статический маршрут по умолчанию (см. Создание статических маршрутов на стр. 180).

Назначение дополнительных IP-адресов

Назначение дополнительных IP-адресов (alias) для сетевых интерфейсов ViPNet Coordinator KB удобно при развертывании сетей некоторых топологий. Например, с помощью дополнительных IP-адресов вы можете в рамках своей сети организовать логическую подсеть, узлам которой, в отличие от узлов остальной сети, будет разрешен доступ в Интернет.

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

Чтобы добавить дополнительный IP-адрес:

1 Проверьте, что на интерфейсе задан статический основной IP-адрес и класс `access`:

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

В выводе команды:

- Назначен IP-адрес, например:

  ```
  inet 192.168.0.1 netmask 255.255.255.0 broadcast 192.168.0.255
  ```

- В строке `Configured by DHCP` значение `no`.
- В строке `Class` значение `access`.

Если на интерфейсе не настроен статический IP-адрес, выберите другой интерфейс или измените настройки (см. Настройка сетевых интерфейсов на стр. 43).

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.

Поддержка технологии VLAN

Вы можете использовать ViPNet Coordinator KB для обработки трафика в физической локальной сети, разделенной 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).

Чтобы настроить обработку трафика из нескольких VLAN:

1 Подключите один из сетевых интерфейсов ViPNet Coordinator KB 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 KB — 128 (как физических, так и виртуальных, включая loopback).

Максимальное количество виртуальных интерфейсов в конфигурации сети ViPNet — 9.

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.

- 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 созданный виртуальный интерфейс в конфигурацию защищенной сети ViPNet:

```
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 выключено.

Чтобы включить прохождение трафика через интерфейс, выполните:

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

6 Измените значение MTU, если хотите оптимизировать передачу данных через интерфейс.

7 Включите физический интерфейс, к которому подключен коммутатор:

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

При этом автоматически будут включены созданные виртуальные интерфейсы.

Теперь ViPNet Coordinator KB сможет обрабатывать трафик из виртуальных сетей на соответствующих виртуальных интерфейсах.

Пример организации обработки трафика из нескольких VLAN

Рассмотрим пример организации обработки трафика из нескольких VLAN.

Допустим, имеется три подсети, которые стоят за координатором, обеспечивающим выход в интернет. У координатора свободен для подключения только один физический интерфейс, поэтому для подключения подсетей используется VLAN для каждой подсети.

На рисунке ниже приведена схема использования ViPNet Coordinator KB в сети с тремя VLAN, которым присвоены номера 10, 20 и 30.

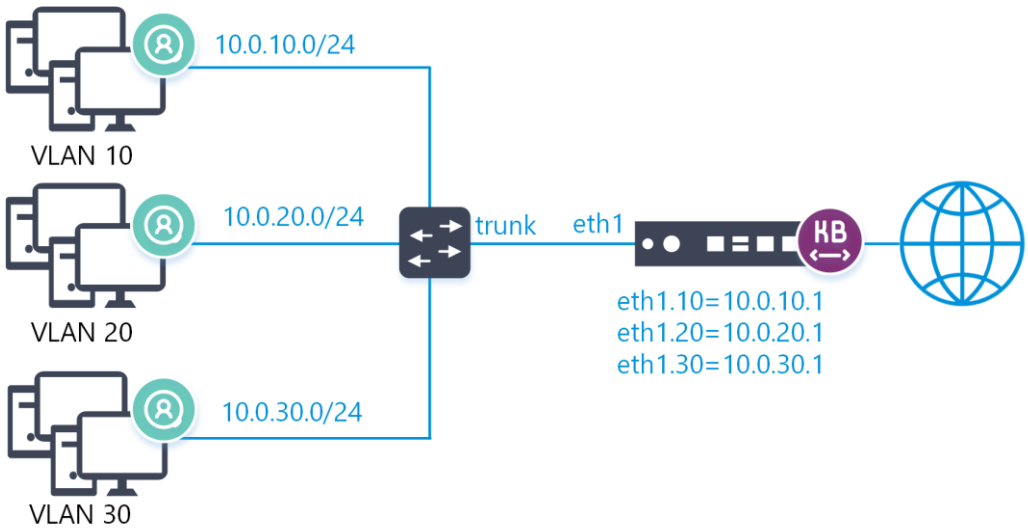


Рисунок 1. Схема использования ViPNet Coordinator KB в сети с VLAN

Чтобы настроить обработку трафика из нескольких VLAN:

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

2 Установите для порта коммутатора, к которому подключен интерфейс ViPNet Coordinator KB, режим `trunk`.

3 Назначьте интерфейсу `eth1` класс `trunk`, чтобы интерфейс обрабатывал трафик из нескольких VLAN:

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

4 Создайте виртуальные интерфейсы `eth1.10`, `eth1.20`, `eth1.30`, которые будут соответствовать виртуальным сетям с номерами 10, 20 и 30:

```
hostname# inet ifconfig eth1 vlan add 10
hostname# inet ifconfig eth1 vlan add 20
hostname# inet ifconfig eth1 vlan add 30
```

5 Задайте параметры созданных 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
hostname# inet ifconfig eth1.30 address 10.0.30.1 netmask 255.255.255.0
```

6 Добавьте новые виртуальные интерфейсы в конфигурационный файл `iplir.conf`:

```
hostname# iplir adapter add eth1.10 traffic on
hostname# iplir adapter add eth1.20 traffic on
hostname# iplir adapter add eth1.30 traffic on
```

Если вы не используете параметр `traffic on`, прохождение трафика через создаваемый виртуальный интерфейс будет выключено. Позже можно включить прохождение трафика через виртуальный интерфейс:

```
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`, `eth1.30`.

После произведенных настроек ViPNet Coordinator KB сможет обрабатывать трафик из трех виртуальных сетей на соответствующих виртуальных интерфейсах.

Удаление конфигурации VLAN

При необходимости удаления конфигурации сетевого интерфейса на котором была организована обработка трафика из нескольких VLAN выполните следующие действия:

1 Завершите работу управляющей службы `iplircfg`:

```
hostname> iplir stop
```

2 Откройте редактор конфигурационного файла `iplir.conf`:

```
hostname# iplir config
```

Удалите секции `[adapter]`, описывающие созданные виртуальные интерфейсы (см. документ «Справочник команд и конфигурационных файлов»).

3. В секции `[adapter]` с описанием сетевого интерфейса, на котором была настроена конфигурация VLAN, присвойте параметру `allowtraffic` значение `on`.
4. Чтобы сохранить изменения в конфигурационном файле, нажмите сочетание клавиш **Ctrl+O**. Затем нажмите **Enter**.
5. Нажмите сочетание клавиш **Ctrl+X**, чтобы закрыть конфигурационный файл.
6. Измените класс сетевого интерфейса, на котором была настроена конфигурация VLAN:

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

7. Запустите управляющую службу:

   ```
   hostname> iplir start
   ```

В результате выполнения описанных действий конфигурация VLAN будет корректно удалена.

Использование агрегированных сетевых интерфейсов

Чтобы увеличить пропускную способность сетевых интерфейсов ViPNet Coordinator KB или повысить надежность канала связи, вы можете объединить несколько физических сетевых интерфейсов в один логический — агрегированный сетевой интерфейс. При этом соответствующие каналы связи объединяются на канальном уровне сетевой модели OSI.

Например, если вам нужен канал связи с пропускной способностью выше 1 Гбит/с, а на вашем исполнении ViPNet Coordinator KB есть только гигабитные сетевые интерфейсы, вы можете объединить два или три из них в один агрегированный. При этом даже если один из объединенных интерфейсов выйдет из строя, агрегированный канал связи продолжит работать.

Кроме того, вы можете использовать агрегированный сетевой интерфейс только для резервирования канала связи, без увеличения его пропускной способности. В этом случае 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 (ЦОД).

Чтобы задать, как будет распределяться нагрузка по подчиненным физическим интерфейсам, необходимо выбрать один из режимов работы агрегированного сетевого интерфейса (см. Режимы работы агрегированного сетевого интерфейса на стр. 53).

Создание агрегированного сетевого интерфейса



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

Чтобы создать агрегированный сетевой интерфейс:

1. Выберите физические интерфейсы ViPNet Coordinator KB, 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 (см. Режимы работы агрегированного сетевого интерфейса на стр. 53), затем создайте агрегированный канал связи:

`hostname# inet bonding add <номер> mode <режим> slaves <интерфейс 1> [<интерфейс 2>] [<интерфейс 3>]`, где:

- `<номер>` — номер агрегированного сетевого интерфейса. Вы можете создать до трех агрегированных интерфейсов с номерами 0, 1 или 2. В результате выполнения команды создается агрегированный сетевой интерфейс с именем `bond<номер>`.
- `<режим>` — режим работы агрегированного интерфейса.
- `<интерфейс 1>`, `<интерфейс 2>`, `<интерфейс 3>` — имена физических интерфейсов, подчиненных создаваемому агрегированному сетевому интерфейсу. Вы можете задать до трех подчиненных интерфейсов. 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 miimon <частота в миллисекундах>`,

где `<имя агрегированного интерфейса>` — `bond{0,1,2}`.

5 Настройте дополнительные параметры для режимов работы агрегированного интерфейса. Подробнее в разделе Режимы работы агрегированного сетевого интерфейса (на стр. 53).

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

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

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

  Также для режима DHCP вы можете задать дополнительные параметры:

  - включить или выключить автоматическое получение адресов DNS-серверов:

    `hostname# inet ifconfig <имя агрегированного интерфейса> dhcp dns {on | off}`

  - включить или выключить автоматическое получение адресов NTP-серверов:

    `hostname# inet ifconfig <имя агрегированного интерфейса> dhcp ntp {on | off}`

  - включить или выключить автоматическое получение маршрутов от DHCP-сервера:

```
hostname# inet ifconfig <имя агрегированного интерфейса> dhcp route {on |
off}
```

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

```
hostname# inet ifconfig <имя агрегированного интерфейса> dhcp route-metric
{<1-255> | none}
```

По умолчанию всем дополнительным параметрам режима DHCP установлено значение `on` (`none` для метрики маршрутов, полученных от DHCP-сервера).

Вы можете задать ряд параметров, которые будут присваиваться маршрутам DHCP-сервера (см. Настройка параметров динамических маршрутов от протокола DHCP на стр. 184). При необходимости вы можете просмотреть параметры, которые заданы для режима DHCP на всех сетевых интерфейсах (см. Просмотр настроек DHCP в режиме клиента на стр. 186).

- Назначьте агрегированному сетевому 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 сетевой интерфейс в конфигурацию ViPNet Coordinator KB:

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

где `<имя агрегированного интерфейса>` — `bond{0,1,2}`.

Если вы не используете параметр `traffic on`, прохождение трафика через создаваемый агрегированный интерфейс будет выключено. Вы можете позже включить прохождение трафика через этот интерфейс:

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

Чтобы агрегированный интерфейс обрабатывал трафик из нескольких VLAN, назначьте ему класс `trunk` (см. Поддержка технологии VLAN на стр. 46). По умолчанию созданному агрегированному сетевому интерфейсу назначается класс `access`.

8 Последовательно включите все подчиненные физические интерфейсы:

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

9 Включите агрегированный сетевой интерфейс:

```
hostname# inet ifconfig <агрегированный интерфейс> 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.

Режимы работы агрегированного сетевого интерфейса

При создании агрегированного сетевого интерфейса необходимо указать режим его работы, наиболее подходящий для решения ваших задач. В ViPNet Coordinator KB предусмотрено несколько режимов, позволяющих по-разному распределять нагрузку между подчиненными сетевыми интерфейсами.

Таблица 2. Режимы работы агрегированного интерфейса

Режим	Описание
`balance-rr`	Режим, подходящий как для балансировки нагрузки на подчиненных физических интерфейсах, так и для защиты от сбоев. Может применяться в сетях с простой топологией. В этом режиме исходящие IP-пакеты, попадающие на агрегированный интерфейс, отправляются через подчиненные интерфейсы поочередно: первый пакет отправляется через один подчиненный интерфейс, второй пакет — через следующий подчиненный интерфейс и так далее.
`balance-xor`	Режим, предназначенный для защиты от сбоев и распределения нагрузки таким образом, чтобы IP-пакеты от одного и того же отправителя к одному и тому же получателю всегда отправлялись через один и тот же подчиненный сетевой интерфейс. В этом режиме для определения подчиненного физического интерфейса, через который отправляется пакет, используется специальная хэш-функция, алгоритм вычисления которой можно задать с помощью команды: `inet ifconfig <имя агрегированного интерфейса> bonding xmit-hash-policy <алгоритм>` Вы можете задать один из следующих алгоритмов: • `layer2` — в алгоритме используются MAC-адреса отправителя и получателя пакета, таким образом, одинаковыми считаются сетевые узлы с одинаковыми MAC-адресами; • `layer2+3` — в алгоритме используются MAC-адреса отправителя и получателя, а также IP-адреса отправителя и получателя (для протокола IPv4), таким образом, одинаковыми считаются сетевые узлы с одинаковыми MAC-адресами и IP-адресами; • `layer3+4` — в алгоритме используются IP-адреса отправителя и получателя, а также номера портов TCP и UDP (при наличии), таким образом, одинаковыми считаются сетевые узлы с одинаковыми IP-адресами, а также портами TCP или UDP.
`balance-tlb`	Режим, предназначенный для балансировки нагрузки на подчиненных физических интерфейсах и рекомендуемый к использованию при передаче большого числа IP-пакетов разного размера, когда более простые режимы не распределяют нагрузку равномерно.

Режим	Описание
	В этом режиме ведется подсчет размера исходящих пакетов, переданных через каждый из подчиненных интерфейсов, и на основе этого выполняется выбор интерфейса, через который будет передан пакет, попавший на агрегированный сетевой интерфейс. **Примечание.** Для работы агрегированного сетевого интерфейса в режиме `balance-tlb` необходимо, чтобы все подчиненные физические интерфейсы были подключены к сети через коммутатор.

`802.3ad`	Режим динамического агрегирования с использованием протокола LACP, предназначен для комплексной балансировки нагрузки. В этом режиме агрегированный сетевой интерфейс работает следующим образом:
	• Среди подчиненных однотипных физических интерфейсов формируются группы — «агрегаторы», скорость передачи данных на интерфейсах которых одинакова. Параметры физических интерфейсов `mtu`, `duplex` и `autoneg`, объединяемых в группу, должны совпадать.
	• Активный агрегатор выбирается в соответствии с режимом выбора, задаваемым с помощью команды:
	`inet ifconfig <интерфейс> bonding ad-select <режим>`
	Вы можете выбрать один из приведенных ниже режимов выбора:
	`stable` — режим, при котором первоначально выбирается агрегатор с наибольшей суммарной пропускной способностью подчиненных физических интерфейсов, а в дальнейшем выбор нового агрегатора выполняется только в случае сбоя всех подчиненных интерфейсов текущего агрегатора.
	`bandwidth` — режим, при котором первоначально выбирается агрегатор с наибольшей пропускной способностью подчиненных физических интерфейсов, а в дальнейшем, при добавлении, удалении или сбое подчиненных физических интерфейсов, в агрегаторах производится перегруппировка подчиненных физических интерфейсов и выполняется выбор нового агрегатора.
	`count` — режим, при котором первоначально выбирается агрегатор с наибольшим количеством подчиненных физических интерфейсов, а в дальнейшем, при добавлении, удалении или сбое подчиненных физических интерфейсов, в агрегаторах производится перегруппировка подчиненных физических интерфейсов и выполняется выбор нового агрегатора.
	• Внутри активного агрегатора подчиненный физический интерфейс, через который отправляются исходящие пакеты, выбирается аналогично режиму `balance-xor`.
	• С другим сетевым оборудованием происходит обмен LACPDU-пакетами с периодичностью, задаваемой с помощью команды:
	`inet ifconfig <интерфейс> bonding lacp-rate <slow | fast>`
	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.

Режим	Описание
	Примечание. Механизм MC-LAG (MLAG) в данном режиме не поддерживается. Поэтому физические порты разных ViPNet Coordinator KB нельзя использовать в одном агрегированном интерфейсе.
`active-backup`	Режим, предназначенный для защиты от сбоев, но не для балансировки нагрузки на подчиненных физических интерфейсах. В этом режиме один из подчиненных физических интерфейсов назначается основным (автоматически или явно с помощью команды `inet ifconfig bonding primary`), и все исходящие IP-пакеты отправляются через него. При этом, в случае сбоя на основном подчиненном интерфейсе пакеты будут отправляться через другие подчиненные интерфейсы.
`broadcast`	Режим предоставляет наибольшую защиту от сбоев. В этом режиме IP-пакеты, попадающие на агрегированный сетевой интерфейс, отправляются через все подчиненные физические 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.

Примеры организации агрегированного канала связи

Организация агрегированного канала связи между ViPNet Coordinator KB и коммутатором

Рассмотрим пример организации агрегированного канала для взаимодействия координатора ViPNet Coordinator KB, имеющего четыре физических сетевых интерфейса, и коммутатора. Данную схему можно использовать чтобы с небольшими издержками обеспечить отказоустойчивость и избежать эффекта "бутылочного горлышка" при передаче данных между координатором и коммутатором.

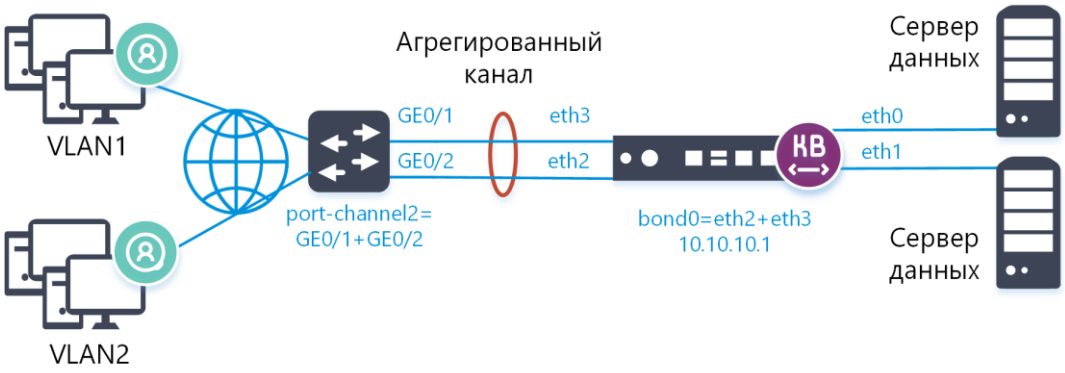


Рисунок 2. Агрегированный канал между ViPNet Coordinator KB и коммутатором

К сетевым интерфейсам `eth0` и `eth1` ViPNet Coordinator KB подключены два сервера. ViPNet-клиенты подключаются к ним через коммутатор. Объединим на ViPNet Coordinator KB интерфейсы `eth2` и `eth3` в агрегированный интерфейс `bond0` чтобы организовать отказоустойчивый канал связи.

При передаче пакетов от ViPNet-клиентов через коммутатор MAC-адрес, IP-адрес и порт сетевого узла отправителя не подменяется, поэтому в данной схеме можно использовать режимы, связанные с распределением нагрузки по MAC-адресам, IP-адресам и портам: `balance-xor`, `balance-tlb` и `802.3ad`. Подробнее о режимах работы агрегированных интерфейсов см. в разделе "Режимы работы агрегированного интерфейса (см. Режимы работы агрегированного сетевого интерфейса на стр. 53)".

Используем режим `802.3ad`. В этом режиме для определения подчиненного физического интерфейса, через который отправляется пакет, используется специальная 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 описанный агрегированный канал связи:

1. Установите для физических интерфейсов `eth2` и `eth3` класс `slave`:

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

2. Задайте режим работы создаваемого агрегированного интерфейса `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 новый агрегированный интерфейс в конфигурацию ViPNet Coordinator KB:

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

   При выполнении команды вы можете не использовать параметр `traffic on`, в этом случае прохождение трафика через создаваемый агрегированный интерфейс будет выключено. Чтобы позже включить прохождение трафика через агрегированный интерфейс, выполните:

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

5 Задайте для созданного агрегированного интерфейса IP-адрес:

```
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 сетевые интерфейсы коммутатора, к которым подключены ViPNet-клиенты, а также интерфейсы, которые будут подключены к агрегированному каналу, в виртуальную сеть VLAN с определенным номером (в примере VLAN имеет номер 116).

2 Создайте логический интерфейс `port-channel` (в примере он имеет номер 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 два физических интерфейса (в примере имена этих интерфейсов — `GigabitEthernet0/1` и `GigabitEthernet0/2`), которые будут входить в состав логического интерфейса `port-channel2`. Эти физические интерфейсы 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
```

В результате координатор будет доступен по заданному IP-адресу, а 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. Этим обеспечивается отказоустойчивость полученного агрегированного канала.

Организация агрегированного канала между двумя ViPNet Coordinator KB

Рассмотрим пример организации агрегированного канала связи для взаимодействия двух координаторов ViPNet Coordinator KB, имеющих по четыре физических сетевых интерфейса.

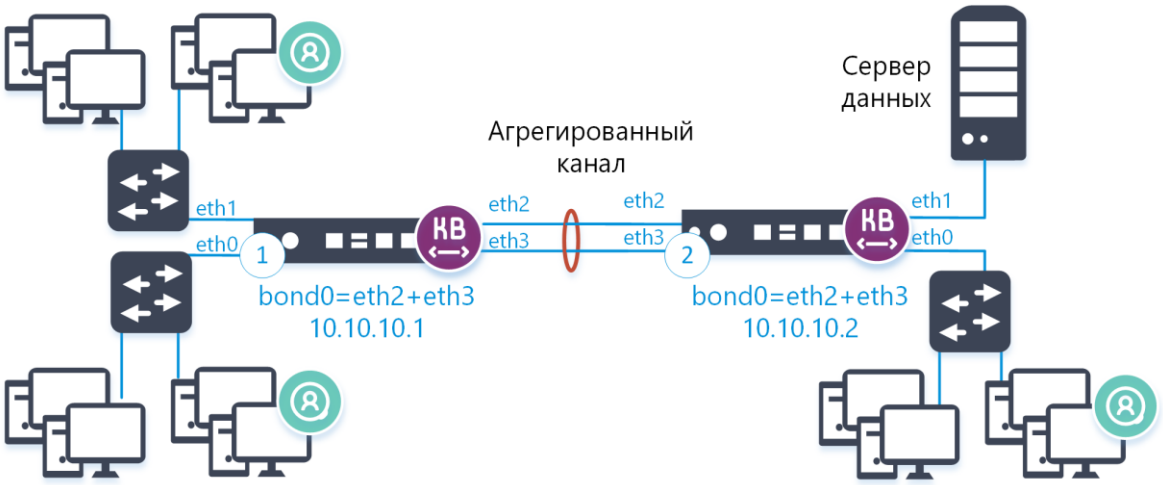


Рисунок 3. Агрегированный канал связи между двумя ViPNet Coordinator KB

К сетевым интерфейсам `eth0` и `eth1` ViPNet Coordinator KB 1, а также к интерфейсу `eth0` ViPNet Coordinator KB 2 через коммутаторы подключены ViPNet-клиенты и открытые узлы. К интерфейсу `eth1` ViPNet Coordinator KB 2 подключен сервер данных. Чтобы обеспечить обмен информацией между перечисленными сетевыми узлами, между ViPNet Coordinator KB 1 и ViPNet Coordinator KB 2 организуем отказоустойчивый канал связи.

Для этого объединим на каждом координаторе сетевые интерфейсы `eth2` и `eth3` в агрегированный сетевой интерфейс `bond0`.

Так как при передаче IP-пакетов через ViPNet Coordinator KB MAC-адрес, порт и IP-адрес сетевого узла отправителя подменяются, режимы работы агрегированных каналов, связанные с распределением нагрузки по MAC-адресам, IP-адресам и портам, использовать нельзя. Таким образом, можно использовать только следующие режимы работы агрегированных интерфейсов (подробнее о режимах работы см. в разделе "Режимы работы агрегированного интерфейса (см. Режимы работы агрегированного сетевого интерфейса на стр. 53)"):

- `balance-rr`
- `active-backup`
- `broadcast`

Будем использовать режим `balance-rr`. В этом режиме исходящие пакеты, попадающие на агрегированный сетевой 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 связи 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.

Для настройки агрегированного канала:

1 На обоих ViPNet Coordinator KB:

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 Задайте режим работы создаваемого агрегированного интерфейса `balance-rr`:

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

1.3 Добавьте новый агрегированный интерфейс в конфигурацию ViPNet Coordinator KB:

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

2 Задайте для созданных агрегированных интерфейсов IP-адреса:

- На ViPNet Coordinator KB 1:

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

- На ViPNet Coordinator KB 2:

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

3 Соедините сетевой интерфейс `eth2` ViPNet Coordinator KB 1 с интерфейсом `eth2` ViPNet Coordinator KB 2.

4 Соедините сетевой интерфейс `eth3` ViPNet Coordinator KB 1 с интерфейсом `eth3` ViPNet Coordinator KB 2.

5 На обоих ViPNet Coordinator KB:

5.1 Последовательно включите все подчиненные физические интерфейсы:

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

5.2 Включите агрегированный сетевой интерфейс:

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

В результате координаторы будут доступны по заданным IP-адресам, и трафик между ними будет идти по агрегированному каналу связи, состоящему из двух подчиненных физических каналов. В случае отказа одного из подчиненных каналов оставшийся продолжает работу. При возобновлении работы канала работа агрегированного канала продолжается в режиме `balance-rr`.

6

Настройка VPN

Общие принципы настройки VPN 62

Подключение к сети через межсетевой экран 63

Принципы назначения виртуальных адресов 70

Настройка туннелируемых адресов 75

Настройка IP-адресов доступа к узлу и их приоритета 80

Настройка туннелирования на канальном уровне по технологии L2OverIP 82

Примеры настройки туннелирования 86

Управление копиями конфигурации VPN 92

Смена мастер-ключей 96

Общие принципы настройки VPN

Настройка параметров VPN производится путем редактирования конфигурационного файла `iplir.conf`. Большинство параметров в этом файле задается автоматически при установке справочников и ключей на ViPNet Coordinator KB — параметрам присваиваются значения, заданные при настройке сети ViPNet и отдельных сетевых узлов в ViPNet Центр управления сетью (подробнее см. в документе «ViPNet Центр управления сетью 4. Руководство администратора»).

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

- Настройка параметров сетевого подключения координатора (на стр. 63).
- Принципы назначения виртуальных адресов (на стр. 70).
- Настройка параметров видимости узлов (на стр. 73).
- Настройка туннелируемых адресов (на стр. 75).
- Настройка IP-адресов доступа к узлу и их приоритета (на стр. 80).

Подключение к сети через межсетевой экран

Если ViPNet Coordinator KB работает во внутренней сети с частными IP-адресами и на границе этой сети установлен межсетевой экран или другое устройство, осуществляющее трансляцию сетевых адресов, то для нормальной работы с другими узлами, находящимися во внешней сети или стоящими за другими межсетевыми экранами, необходимо настроить один из режимов работы через межсетевой экран.

Настройка режима выполняется путем редактирования конфигурационного файла `iplir.conf`. Перед редактированием этого файла необходимо завершить работу управляющей службы командой `hostname> iplir stop`, а после окончания редактирования, чтобы все изменения вступили в силу, — вновь запустить его командой `hostname> iplir start`.

Режимы подключения к сети через межсетевой экран

Для подключения ViPNet Coordinator KB к внешней сети предусмотрены следующие режимы:

1. «Без использования межсетевого экрана» — ViPNet Coordinator KB имеет прямое подключение к внешней сети.
2. «Со статической трансляцией адресов» — подключение через внешний межсетевой экран (устройство) с трансляцией сетевых адресов (NAT), на котором возможна настройка статических правил трансляции сетевых адресов.
3. «С динамической трансляцией адресов» — подключение через внешний межсетевой экран (устройство), которое выполняет динамическую трансляцию сетевых адресов (наиболее распространенный способ подключения к внешней сети).

Если ViPNet Coordinator KB имеет непосредственное подключение к внешней сети, то есть может быть доступен напрямую со стороны любых других сетевых узлов (например, имеет публичный IP-адрес), то для него следует выбрать режим «Без использования межсетевого экрана» (на стр. 64).

Если ViPNet Coordinator KB имеет частный IP-адрес, который может быть недоступен для других сетевых узлов в соответствии с общими правилами маршрутизации (то есть на выходе во внешнюю сеть установлен межсетевой экран или другое устройство NAT), то необходимо выбрать один из режимов подключения к сети через межсетевой экран:

- Режим «Со статической трансляцией адресов» (на стр. 65) выбирается в случае, если на выходе во внешнюю сеть установлен межсетевой экран или иное NAT-устройство, на котором можно настроить статические правила трансляции сетевых адресов, обеспечивающие взаимодействие

с определенным внутренним IP-адресом сети по протоколу UDP с заданным портом. Данный режим задан по умолчанию.

Также режим «Со статической трансляцией адресов» следует использовать, 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 же на координаторах будет установлен режим «Без использования межсетевого экрана», то для удаленного узла, использующего один из этих координаторов как сервер соединений, в качестве IP-адреса доступа к другому узлу может зарегистрироваться некорректный IP-адрес (при этом доступ к нему будет потерян).

- Режим «С динамической трансляцией адресов» (на стр. 66) выбирается в случае, 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 KB 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 осуществляется напрямую по IP-адресу узла.

Настройка режима «Без использования межсетевого экрана»

Для настройки работы ViPNet Coordinator KB 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**) и закройте редактор (**Ctrl+X**).
7. Запустите управляющую службу:

   ```
   hostname# iplir start
   ```

Настройка режима «Со статической трансляцией адресов»

По умолчанию в ViPNet Coordinator KB заданы параметры подключения к сети для случая, когда он установлен за внешним межсетевым экраном (устройством) с трансляцией сетевых адресов (NAT), на котором можно настроить статические правила NAT.

В этом случае на межсетевом экране должны быть заданы статический IP-адрес ViPNet Coordinator KB и порт доступа к нему. Для клиентов локальной сети ViPNet Coordinator KB следует использовать в качестве сервера соединений.

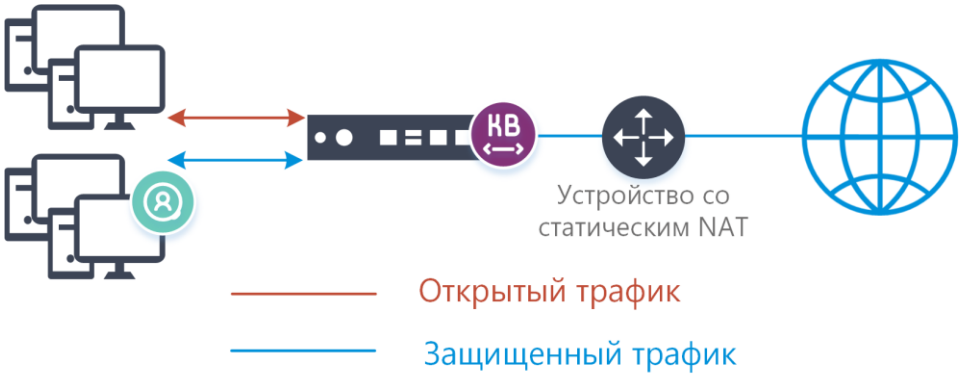


Рисунок 4. Подключение координатора через межсетевой экран со статической трансляцией адресов

Кроме того, на ViPNet Coordinator KB необходимо настроить маршрутизацию на внешний межсетевой экран (шлюз по умолчанию или маршруты для удаленных подсетей).

На межсетевом экране (или устройстве NAT) должны быть заданы следующие статические правила NAT:

- Пропускать исходящие UDP-пакеты с IP-адресом и портом ViPNet Coordinator KB (порт источника, описан как параметр `port` в собственной секции `[id]` конфигурационного файла `iplir.conf`) на любой внешний адрес и порт (с подменой адреса источника на внешний адрес устройства NAT).
- Пропускать и перенаправлять входящие UDP-пакеты с портом назначения, заданным в собственной секции `[id]` конфигурационного файла `iplir.conf`, на IP-адрес ViPNet Coordinator KB.

Если вы изменили настройки ViPNet Coordinator KB по умолчанию, то для подключения к сети через межсетевой экран со статическим NAT:

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`;
- `proxyid` в значение `0x00000000`;
- `fixfirewall` в значение:
  - `on` — если на внешнем межсетевом экране с NAT есть возможность настроить статические правила для входящих IP-пакетов, т.е. перенаправлять пакеты, у которых задан IP-адрес ViPNet Coordinator KB и порт назначения. Для данного режима укажите:

    `firewallip` — IP-адрес внешнего межсетевого экрана с NAT;

    `port` — порт доступа к внешнему межсетевому экрану.
  - `off` — если внешний IP-адрес межсетевого экрана с NAT в процессе работы может измениться, то на других сетевых узлах IP-адрес доступа к ViPNet Coordinator KB 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 KB (порт источника) и приходят на данный узел (порт назначения).
7. Сохраните изменения (**Ctrl+O**) и закройте редактор (**Ctrl+X**).
8. Запустите управляющую службу:

   ```
   hostname# iplir start
   ```

Информация об IP-адресе межсетевого экрана и порте доступа сообщается программой всем остальным узлам, с которыми связан ViPNet Coordinator KB.

Настройка режима «С динамической трансляцией адресов»

Если ViPNet Coordinator KB подключен к внешней сети через устройство NAT, выполняющее трансляцию сетевых адресов, 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.

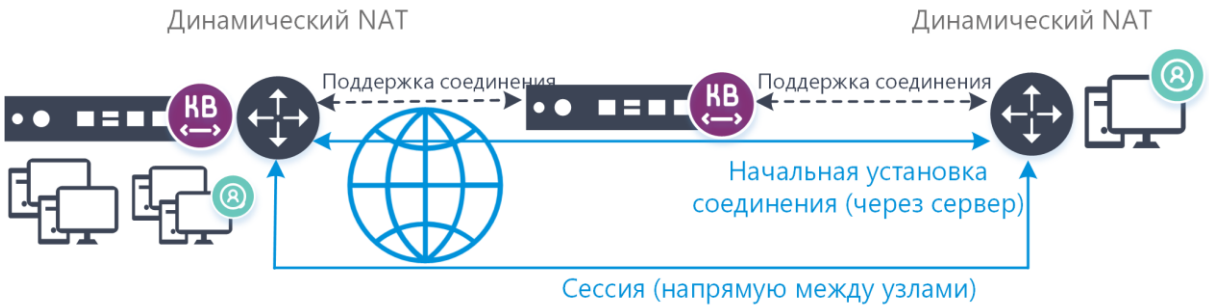


Рисунок 5. Подключение координатора через межсетевой экран с динамической трансляцией адресов

Организация соединений между узлами осуществляется 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 осуществляет 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 KB 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`;
   - в параметре `port` укажите значение из диапазона 1–65535 (по умолчанию — 55777).

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 KB будет устанавливать соединение с другими узлами.



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

При подключении через межсетевой экран 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**) и закройте редактор (**Ctrl+X**).

7 Запустите управляющую службу:

```
hostname# iplir start
```



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

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

Особенности использования ViPNet Coordinator KB в качестве межсетевого экрана

В случае, когда ViPNet Coordinator KB, установленный в сети ViPNet, выполняет функцию межсетевого экрана и в этой же сети есть другой координатор, к которому физически подключены клиенты, то для корректного взаимодействия клиентов с другой сетью через ViPNet Coordinator KB, который установлен в другой сети, выполните следующие настройки:

1 Администратор сети ViPNet в ЦУС должен установить для всех клиентов в сети в качестве межсетевого экрана ViPNet Coordinator KB, подробное описание см. в документе «ViPNet Центр управления сетью 4. Руководство администратора» в разделе «Настройка подключения через координатор в качестве межсетевого экрана».

2 Пользователь клиента, для которого ViPNet Coordinator KB выполняет роль сервера соединений, в настройках ПО ViPNet Client должен установить флажок **Весь трафик направлять через сервер соединений**, подробное описание см. в руководстве пользователя для конкретного ПО ViPNet Client, которое установлено на клиентском узле, в разделе «Настройка подключения к защищенной сети».



Внимание! Для данной конфигурации сети гарантируется корректная работоспособность клиентов с ПО ViPNet Client версии 4.0.0 и выше.

---

3 Администратор ViPNet Coordinator KB должен установить виртуальную видимость с ViPNet Coordinator KB для клиентов, которые находятся в одной подсети, но подключены к другому координатору. Для этого на ViPNet Coordinator KB в конфигурационном файле `iplir.conf` в секции `[id]` соответствующих клиентов присвойте параметру `tunnelvisibility` значение `virtual`, подробное описание см. в документе «Справочник команд и конфигурационных файлов» в разделе «Секция [id]».

Принципы назначения виртуальных адресов

Виртуальные адреса назначаются всем защищенным и туннелируемым сетевым узлам, с которыми есть связь. Необходимость использования виртуальных адресов обусловлена тем, что сетевые узлы, стоящие за разными межсетевыми экранами, могут иметь одинаковые IP-адреса в своих частных сетях, и в случае обращения к ним по реальным адресам могла бы возникнуть неоднозначность. Для узлов, находящихся в одном широковещательном сегменте, виртуальные адреса также назначаются, но не используются по умолчанию (при необходимости эти узлы можно переключить на использование виртуальных адресов).

Параметры, необходимые для назначения виртуальных адресов, задаются в секции `[virtualip]` конфигурационного файла `iplir.conf`.

При появлении сетевого узла в списке связей ему назначается виртуальный IP-адрес (поз. 2 на рисунке ниже), который привязан к первому реальному адресу (поз. 1 на рисунке ниже). Такой виртуальный IP-адрес называется базовым, и он является точкой отсчета при назначении виртуальных адресов для каждого из реальных адресов узла.

Остальные виртуальные адреса узла сети ViPNet, соответствующие каждому из реальных адресов, называются вторичными виртуальными адресами или просто виртуальными адресами, и указываются в параметре `ip` соответствующей секции `[id]` через запятую после реального адреса. Каждому узлу сети ViPNet выделяется не более 15 вторичных виртуальных адресов.

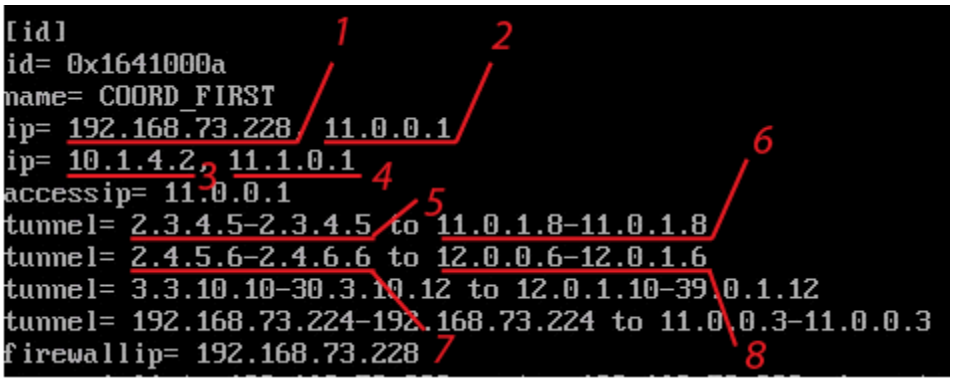


Рисунок 6: Пример секции [id] конфигурационного файла iplir.conf

На схеме показаны:

1. Реальный адрес сетевого узла ViPNet.
2. Базовый виртуальный IP-адрес сетевого узла ViPNet.
3. Второй реальный IP-адрес сетевого узла ViPNet.
4. Вторичный виртуальный IP-адрес сетевого узла ViPNet.
5. Реальный адрес туннелируемого узла (см. Настройка туннелируемых адресов на стр. 75).

6 Виртуальный IP-адрес туннелируемого узла.

7 Диапазон реальных адресов туннелируемых узлов.

8 Диапазон виртуальных адресов туннелируемых узлов.

Четыре октета, составляющие IP-адрес, используются при назначении виртуальных адресов следующим образом:

- Первый октет всех виртуальных адресов имеет одинаковое значение, соответствующее первому октету стартового виртуального адреса `startvirtualip` секции `[virtualip]` конфигурационного файла `iplir.conf`.
- Второй октет характеризует реальные адреса сетевого узла и используется для назначения вторичных виртуальных адресов узла.
- Третий и четвертый октеты характеризуют сетевой узел. Они одинаковы для всех виртуальных адресов данного узла и различны для виртуальных адресов, принадлежащих разным узлам.

То есть при назначении виртуального адреса для следующего сетевого узла сначала увеличивается четвертый октет, а когда четвертый октет достигает максимального значения (254), увеличивается третий октет. Когда значения третьего и четвертого октетов становятся равными 255 и 254 соответственно, второй октет увеличивается на 16.

Таблица 3. Пример назначения виртуальных адресов для 65025 сетевых узлов

Порядковый номер сетевого узла	Реальный IP-адрес сетевого узла	Виртуальный IP-адрес сетевого узла
№1	192.168.0.1	11.0.0.1
…	…	…
№254	192.168.0.254	11.0.0.254
№255	192.168.1.1	11.0.1.1
…	…	…
№65024	192.168.255.254	11.0.255.254
№65025	192.169.0.1	11.16.0.1

Если какой-либо сетевой узел имеет несколько реальных адресов, то при назначении виртуального адреса для следующего реального адреса того же сетевого узла увеличивается второй октет, но не более чем для 15 вторичных виртуальных адресов:

Таблица 4. Пример назначения вторичных виртуальных адресов трем сетевым интерфейсам узла

Сетевой интерфейс	Реальный IP-адрес	Виртуальный IP-адрес
Ethernet 0	192.169.0.1	11.16.0.1
Ethernet 1	88.88.88.88	11.17.0.1

Сетевой интерфейс	Реальный IP-адрес	Виртуальный IP-адрес
Ethernet 2	200.0.0.1	11.18.0.1

Текущий IP-адрес доступа к сетевому узлу определяется автоматически и содержится в параметре `accessip` соответствующей секции `[id]` конфигурационного файла `iplir.conf`. Если в данный момент узел виден по виртуальному адресу, то его IP-адресом доступа считается базовый виртуальный адрес.

Вторичные виртуальные адреса могут использоваться, если нужно обратиться к конкретному реальному адресу данного сетевого узла, который виден по виртуальным адресам, а не к узлу вообще. Такая необходимость может возникнуть, например, если на данном сетевом узле работает приложение, которое ожидает сетевые запросы только по одному из IP-адресов сетевого узла.

На схеме ниже показан защищенный сервер с двумя сетевыми интерфейсами Ethernet 0 и Ethernet 1. На защищенном сервере настроен веб-сервис, который обрабатывает подключения по внешнему IP-адресу 88.88.88.88 на Ethernet 1.

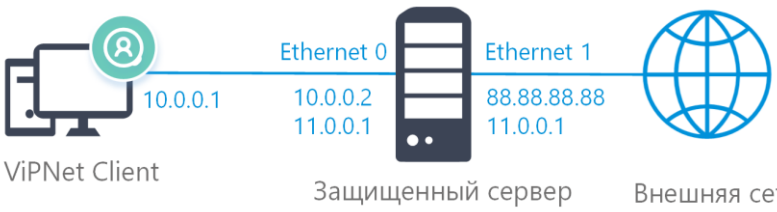


Рисунок 7. Схема использования вторичного виртуального адреса

Если настроена виртуальная видимость защищенного сервера, то обращение с сетевого узла ViPNet Client к защищенному серверу будет происходить по виртуальным адресам. При этом 11.0.0.1 — базовый виртуальный IP-адрес защищенного сервера, соответствующий реальному адресу 10.0.0.2, а 11.1.0.1 — вторичный виртуальный IP-адрес защищенного сервера, соответствующий реальному адресу 88.88.88.88, к которому ViPNet Client подключается для взаимодействия с веб-сервисом.

IP-пакет, отправленный с ViPNet Client на 11.1.0.1, после обработки драйвером iplir на ViPNet Client пойдет на реальный адрес 10.0.0.2 сетевого интерфейса Ethernet 0, затем, после обработки драйвером `iplir`, он будет направлен на адрес 88.88.88.88 и таким образом будет передан веб-сервису.

При обновлениях адресных справочников, а также изменениях в списке реальных адресов для какого-либо узла, сетевые узлы сохраняют свои базовые виртуальные адреса, а вновь добавленные сетевые узлы и реальные IP-адреса получают новые свободные виртуальные адреса (с учетом ограничения на максимально возможный IP-адрес, регулируемого параметром `maxvirtualip` секции `[virtualip]` конфигурационного файла `iplir.conf`).

Настройка параметров видимости узлов

Для связи ViPNet Coordinator KB с защищенными узлами могут использоваться реальные или виртуальные IP-адреса. При этом:

- сетевые узлы, находящиеся в одном и том же широковещательном сегменте, по умолчанию обращаются друг к другу по реальным IP-адресам;
- туннелируемые узлы по умолчанию доступны по реальным IP-адресам;
- остальные узлы сети ViPNet по умолчанию доступны по виртуальным IP-адресам.

Вы можете изменить настройки адресов видимости с помощью параметра `visibility` в секциях `[id]` и параметров в секции `[visibility]` конфигурационного файла `iplir.conf`.

Видимость каждого защищенного узла определяется 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_auto`, `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]`.

Индивидуальные настройки видимости узлов приоритетнее настроек видимости сетей, а настройки видимости сетей приоритетнее настроек видимости по умолчанию.

Чтобы настроить параметры видимости узлов, выполните следующие действия:

1. Завершите работу управляющей службы:

   ```
   hostname# iplir stop
   ```

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

   ```
   hostname# iplir config
   ```

3. Чтобы настроить параметры видимости 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.

Идентификаторы сетей указываются в шестнадцатеричном формате с префиксом 0x. В одной секции [visibility] можно задать несколько параметров subnet_real. При этом в каждом параметре можно указать либо один, либо несколько идентификаторов через запятую.



Внимание! Один и тот же идентификатор сети ViPNet можно указывать только в одном из параметров `subnet_auto`, `subnet_real` или `subnet_virtual`.

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

- `auto` — определять видимости 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 `iplircfg`:

```
hostname# iplir start
```

Настройка туннелируемых адресов

Технология туннелирования позволяет защищать трафик открытых узлов корпоративной сети на потенциально опасном участке сети или включать открытые узлы в защищенную сеть без установки на эти узлы ПО ViPNet. Настройки туннелирования открытых узлов, которые задает администратор сети ViPNet, хранятся на координаторах сети ViPNet, выполняющих туннелирование этих узлов.

Туннелирование также используют для защиты распределенной сетевой инфраструктуры. Инкапсуляция трафика в туннеле ViPNet позволяет защищать как взаимодействие открытых узлов через публичные сети (например, по протоколу BGP), так и другие туннели, не использующие шифрование (например, GRE-туннели).

Обычно параметры туннелирования рассылаются на сетевые узлы ViPNet в составе справочников. Если туннелируемые IP-адреса координатора заданы в ViPNet Центр управления сетью, то другие узлы получат информацию об этом автоматически. Если туннелируемые IP-адреса заданы на координаторе вручную, эти адреса также необходимо указать вручную на каждом узле, который будет подключаться к этим туннелируемым узлам посредством сети ViPNet.

В случае если настройки были сначала заданы вручную на узлах, а затем пришли настройки из ЦУСа, то настройки сравниваются и при этом:

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

Чтобы избежать конфликтов IP-адресов, узлы в сети ViPNet могут обращаться друг к другу не только по реальным, но и по виртуальным IP-адресам. При этом:

- сетевые узлы, находящиеся в одной и той же подсети, по умолчанию обращаются друг к другу по реальным IP-адресам;
- туннелируемые узлы по умолчанию доступны по реальным IP-адресам;
- остальные узлы сети ViPNet по умолчанию доступны по виртуальным IP-адресам.

После изменения настроек туннелирования может возникнуть конфликт между реальными IP-адресами узлов из подсети вашего ViPNet Coordinator KB и реальными IP-адресами узлов, туннелируемых удаленным координатором. Чтобы устранить такой конфликт, задайте видимость туннелируемых узлов по виртуальным IP-адресам в конфигурационном файле `iplir.conf` в секции `[id]` координатора, туннелирующего узлы с конфликтующими реальными IP-адресами (см. Настройка видимости туннелируемых узлов на стр. 78).

Задание виртуальных адресов для туннелируемых узлов может производиться в одном из следующих режимов:

- в автоматическом режиме (по умолчанию) — при этом задаются реальные адреса туннелируемых узлов, а виртуальные адреса для туннелируемых узлов назначаются автоматически;
- в ручном режиме (см. Задание виртуальных адресов в ручном режиме на стр. 77) — при этом реальные адреса туннелируемых узлов и виртуальные адреса для туннелируемых узлов задаются на узлах вручную.

Настройка видимости туннелируемых узлов (на стр. 78) по реальным или виртуальным адресам доступна при любом режиме задания виртуальных адресов для туннелируемых узлов.

Задание виртуальных адресов для туннелируемых узлов в автоматическом режиме



Примечание. Мы рекомендуем настраивать туннелирование централизованно через ViPNet Центр управления сетью, в этом случае параметры туннелирования будут рассылаться на сетевые узлы ViPNet в составе справочников и ключей.

Настраивать туннелирование на координаторе рекомендуется только когда централизованная настройка туннелирования в ViPNet Центр управления сетью не используется, иначе настройки туннелирования на координаторе будут заменены настройками из ЦУСа.

Вы можете задать реальные адреса туннелируемых узлов на координаторе вручную, а соответствующие виртуальные адреса для туннелируемых узлов будут заданы в автоматическом режиме. Для этого выполните следующие действия:

1 Завершите работу управляющей службы:

```
hostname# iplir stop
```

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

```
hostname# iplir config
```

3 Убедитесь, что в секции `[misc]` параметр `tunnel_virt_assignment` установлен в значение `auto`.

4 В секции `[id]` измените значение параметра `tunnel` следующим образом:

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

где:

`ip1` и `ip2` — начальный и конечный реальные адреса диапазона туннелируемых узлов.



Внимание! На всех туннелируемых узлах сети необходимо настроить статические маршруты для защищаемого трафика через ViPNet Coordinator KB или указать ViPNet Coordinator KB в качестве шлюза по умолчанию.

Например, чтобы координатор туннелировал адреса с 192.168.201.5 по 192.168.201.10, а виртуальные адреса для этого диапазона реальных адресов были назначены автоматически, укажите:

```
tunnel= 192.168.201.5-192.168.201.10
```

5 IP-адреса `<ip1>-<ip2>`, заданные в параметре `tunnel`, также укажите на каждом узле, который будет работать с этими туннелируемыми узлами посредством сети ViPNet.

6 При необходимости в секции `[virtualip]` задайте начальный и конечный адреса для формирования виртуальных адресов (по умолчанию для одиночных туннелируемых адресов: `startvirtualip= 11.0.0.1`, `maxvirtualip= 11.0.255.254`, для диапазонов туннелируемых узлов параметр определяется автоматически: `starttunnelvirtualip= <x+1>.0.0.1`, где `x` — первый октет `maxvirtualip`).

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

8 Запустите управляющую службу:

9 `hostname# iplir start`

При переходе в автоматический режим назначения виртуальных адресов все виртуальные адреса для туннелируемых узлов будут переназначены. Чтобы обращение к узлам, туннелируемым ViPNet Coordinator KB, происходило по виртуальным адресам, настройте на каждом узле, который будет работать с этими туннелируемыми узлами, видимость по виртуальным адресам (см. Настройка видимости туннелируемых узлов на стр. 78).

Задание виртуальных адресов в ручном режиме

Вы можете вручную задать реальные и виртуальные адреса туннелируемых узлов на координаторе. Для этого:

1 Остановите управляющую службу:

```
hostname# iplir stop
```

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

```
hostname# iplir config
```

3 Убедитесь, что в секции `[misc]` для параметра `tunnel_virt_assignment` установлено значение `manual`.

4 В секции `[id]` измените значение параметра `tunnel` следующим образом:

`tunnel = <ip1>-<ip2> to <ip3>-<ip4>`, где `ip1` и `ip2` — начальный и конечный реальные адреса диапазона туннелируемых узлов, `<ip3>-<ip4>` — диапазон виртуальных адресов, которые соответствуют реальным адресам из диапазона `<ip1>-<ip2>`, и которые будут использоваться вместо реальных адресов туннелируемых узлов, если на узле, который к ним

обращается, настроена видимость по виртуальным адресам (см. Настройка видимости туннелируемых узлов на стр. 78).

Например, в случае когда диапазон `<ip1>-<ip2>` принадлежит к частной сети и такие же IP-адреса уже есть в локальной сети данного координатора. Параметр `ip4` можно не указывать, он формируется путем прибавления к `ip3` разницы между `ip2` и `ip1`.



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

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

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



Внимание! Когда ViPNet Coordinator KB туннелирует компьютеры из локальной сети, на всех туннелируемых компьютерах необходимо с помощью статических маршрутов направить защищаемый трафик через ViPNet Coordinator KB или указать ViPNet Coordinator KB в качестве шлюза по умолчанию.

IP-адреса `<ip1>-<ip2>`, указанные в параметре `tunnel`, будут туннелироваться другими координаторами.

5 Укажите `<ip3>-<ip4>`, туннелируемые ViPNet Coordinator KB, на каждом узле, который будет работать с этими туннелируемыми узлами посредством сети ViPNet.

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

7 Запустите управляющую службу:

```
hostname# iplir start
```

Настройка видимости туннелируемых узлов

Вы можете настроить видимость туннелируемых узлов по реальным или виртуальным IP-адресам для ViPNet Coordinator KB. Для этого:

1 Остановите управляющую службу:

```
hostname# iplir stop
```

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

3 В секции `[id]`, которая относится к туннелирующему узлу, в параметре `tunnelvisibility` укажите тип видимости туннелируемых узлов: `real` — реальные IP-адреса или `virtual` — виртуальные IP-адреса.



Внимание! Если значение параметра `tunnelvisibility` в секции `[id]`, которая относится к туннелирующему узлу, не указано, то видимость туннелируемых узлов будет определяться параметром `tunneldefault` секции `[visibility]`.

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

5 Запустите управляющую службу:

6 `hostname# iplir start`

Настройка IP-адресов доступа к узлу и их приоритета

Каждый узел сети ViPNet взаимодействует с другим узлом, с которым у него есть связь, по определенному IP-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.

При наличии разных каналов для связи между узлами может возникнуть ситуация, когда для взаимодействия будет выбран менее предпочтительный канал (более дорогостоящий или менее быстродействующий). Например, если координатор, находящийся в центральном офисе организации, для выхода в интернет использует одновременно два канала связи, предоставляемых разными провайдерами, то в результате автоматического определения IP-адреса доступа может быть выбран менее предпочтительный канал связи, а более предпочтительный канал будет простаивать. Во избежание такой ситуации, вы можете задать приоритеты каналов связи, исходя из стоимости их обслуживания, быстродействия или других параметров.

Приоритеты каналов связи задаются путем настройки метрик маршрута на стороне того сетевого узла, которому необходимо получить доступ к защищенному сегменту сети. Например, в случае взаимодействия узлов филиала и центрального офиса, если в центральном офисе используется одновременно два канала связи, то настройки производятся на стороне сетевого узла филиала.

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

- IP-адрес с наименьшей метрикой маршрута 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 IP-адресов с заданными метриками маршрута 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 маршрута задана вручную.
- После установления соединения с узлом определение доступности других его IP-адресов выполняется только при разрыве текущего соединения.

Настройка IP-адресов доступа для узлов, связанных с координатором ViPNet Coordinator KB, осуществляется в соответствующих секциях `[id]` конфигурационного файла `iplir.conf`. Каждый

IP-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 Центр управления сетью. В этом случае они будут указаны в соответствующих секциях конфигурационного файла `iplir.conf`. Для таких адресов вы также можете настроить приоритет, изменив соответствующие значения метрик маршрута (по умолчанию они определяются автоматически).

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

1. Остановите управляющую службу:

   ```
   hostname> iplir stop
   ```

2. Откройте конфигурационный 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 Administrator автоматически, в параметре `accessiplist` задайте метрики маршрутов для этих адресов в соответствии с рекомендациями, описанными выше.

4. Сохраните изменения (**Ctrl+O**) 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 адреса, условный номер этого интерфейса и тип регистрации данного IP-адреса доступа узла в виде:

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

Настройка туннелирования на канальном уровне по технологии L2OverIP

Общее описание технологии L2OverIP

В ViPNet Coordinator KB реализована поддержка технологии L2OverIP, 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 так, как будто они находятся в одном сегменте с прямой видимостью по MAC-адресам. Такая защита может потребоваться, например, для организации работы территориально распределенных ЦОДов, локальные сети которых объединены высокоскоростным каналом связи и представляют собой единый домен коллизий.



Внимание! Функцию L2OverIP нельзя использовать для объединения сегментов, которые соединены какими-то другими каналами связи. Сегменты должны быть полностью разобщены. Иначе это может парализовать работу всей сети.

Технология L2OverIP предполагает взаимодействие между узлами нескольких удаленных сегментов сети через ViPNet Coordinator KB, которые установлены на границе этих сегментов. В основе технологии лежит перехват на канальном уровне модели OSI Ethernet-кадров, отправленных из одного сегмента сети в другой. ViPNet Coordinator KB осуществляет перехват Ethernet-кадров, отправленных из его сегмента сети в другой, их упаковку в IP-пакеты специального формата и передачу этих IP-пакетов другому ViPNet Coordinator KB по защищенному каналу. ViPNet Coordinator KB, получивший IP-пакеты специального формата, извлекает из них исходные кадры и передает получателям в своем сегменте.



Примечание. В большинстве практически важных случаев нужного результата можно достичь, объединяя не более 252 IP-адресов во всех объединяемых сегментах сети. Многократное превышение этого количества может привести к серьезному снижению производительности сети в объединяемых сегментах.

С увеличением числа IP-адресов в одном широковещательном сегменте сети значительно возрастает количество широковещательных служебных пакетов, что приводит к снижению пропускной способности сети и увеличению нагрузки на ViPNet Coordinator KB.

С помощью функции L2OverIP вы можете объединить несколько сегментов сети, в том числе сегменты, состоящие из виртуальных локальных сетей (VLAN). Возможны следующие варианты объединения:

- сегменты без VLAN, находящиеся за сетевыми интерфейсами класса `access`;
- VLAN из одного сегмента и виртуальная сеть VLAN из другого сегмента;
- все 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, IP-адреса виртуальных сетей VLAN должны совпадать);
- одна из VLAN сегмента и сегмент без VLAN, находящийся за сетевым интерфейсом класса `access`.



Примечание. Текущая версия ViPNet Coordinator KB позволяет объединить не более 31 сегмента сети.

При использовании функции L2OverIP ViPNet Coordinator KB 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, номера портов задаются в настройках L2OverIP. Порт, заданный для собственного сегмента сети, называется локальным. Порты, заданные на ViPNet Coordinator KB в других сегментах, называются удаленными. Трафик самого ViPNet Coordinator KB 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.

Функцию L2OverIP можно использовать в кластере горячего резервирования. В этом случае настройку функции L2OverIP достаточно выполнить на активном узле, на пассивный узел они передаются автоматически, если включено резервирование конфигурационных файлов. При этом включить функциональность можно только на узле кластера, находящемся в активном режиме, на пассивном узле он всегда выключен и будет включен только при переходе узла в активный режим.

При работе в кластере горячего резервирования таблица MAC-адресов не передается пассивному узлу, и при переключении пассивного узла в активный режим его таблица пуста. Это может привести к увеличению времени конвергенции сети после смены активного узла — до тех пор, пока узлы из разных сегментов сети, взаимодействующие друг с другом, не отправят заново ARP-запросы.

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 KB, установленному 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 KB.

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

1 Включите режим администратора (см. Включение режима администратора ViPNet Coordinator KB на стр. 20).

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

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



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

3 Задайте параметры локального сегмента сети, указав уникальный номер порта и IP-адрес внешнего интерфейса с помощью команды:

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

Каждому сегменту необходимо назначить свой номер порта из интервала 1–31.

4 Задайте параметры удаленного сегмента сети, указав его номер порта и актуальный IP-адрес видимости удаленного ViPNet Coordinator KB с помощью команды:

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

В зависимости от настройки видимости удаленного ViPNet Coordinator KB укажите его реальный или виртуальный IP-адрес.



Примечание. При указании виртуального адреса удаленного ViPNet Coordinator KB убедитесь, что этот виртуальный адрес является базовым (см. Принципы назначения виртуальных адресов на стр. 70). Базовые виртуальные адреса имеют 0 во втором октете (`x.0.x.x`).

5 В случае необходимости вы можете удалить из настроек параметры другого сегмента. Для этого выполните:

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

Если IP-адрес видимости другого сегмента изменился, выполните:

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

6 При необходимости измените время жизни MAC-адреса в таблице MAC-адресов виртуального коммутатора при отсутствии трафика, поступающего от этого адреса. Для этого выполните:

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

Время жизни MAC-адреса задается в секундах. Значение по умолчанию — 300 секунд. При необходимости вы можете задать значение из интервала 60–86400.

7 Добавьте сетевой фильтр защищенной сети, разрешающий любые соединения по протоколу ETHERIP:

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

8 Включите функцию L2OverIP:

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

9 Для просмотра текущего состояния и настроек функции L2OverIP выполните:

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

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



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

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 случай взаимодействия двух офисов через интернет. В одном из офисов находится сервер, к которому обращаются пользователи из другого офиса, и необходимо, чтобы весь обмен данными через интернет происходил с шифрованием трафика. При этом установка ПО ViPNet непосредственно на участвующие в информационном обмене компьютеры невозможна или по каким-либо причинам нежелательна. Чтобы решить эту задачу, в каждом из офисов устанавливается ViPNet Coordinator KB, который выступает в роли криптошлюза для компьютеров локальной сети.

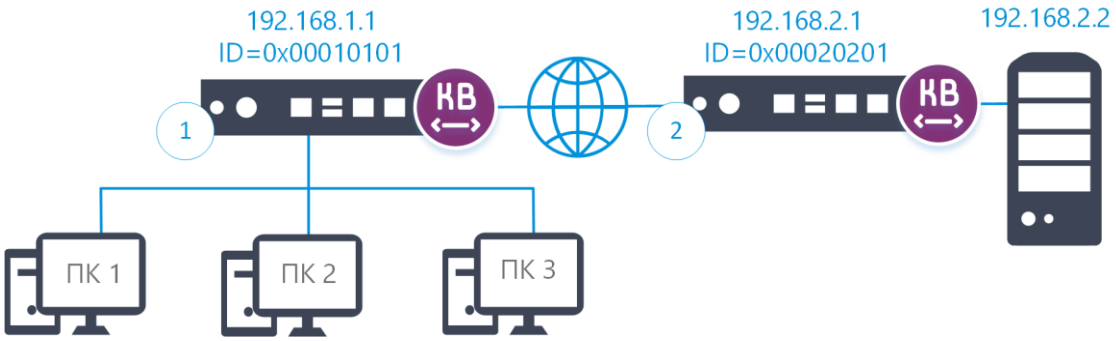


Рисунок 8. Организация туннелей с использованием ViPNet Coordinator KB

Каждый из ViPNet Coordinator KB имеет как минимум два сетевых интерфейса, один из которых имеет публичный IP-адрес и подключен к интернету (внешний интерфейс), а второй имеет частный IP-адрес и подключен к локальной сети офиса (внутренний интерфейс). Пусть заданы следующие сетевые настройки:

- ViPNet Coordinator KB 1 имеет частный IP-адрес 192.168.1.1 и идентификатор 0x00010101 в сети ViPNet;
- компьютеры 1, 2, 3, подключенные к ViPNet Coordinator KB 1, имеют IP-адреса с 192.168.1.2 по 192.168.1.4;
- ViPNet Coordinator KB 2 имеет частный IP-адрес 192.168.2.1 и идентификатор 0x00020201 в сети ViPNet;
- сервер, подключенный к ViPNet Coordinator KB 2, имеет IP-адрес 192.168.2.2.

Чтобы настроить работу туннелей:

1. На компьютерах 1, 2, 3 в качестве шлюза по умолчанию задайте частный IP-адрес ViPNet Coordinator KB 1, на сервере — частный IP-адрес ViPNet Coordinator KB 2.
2. На каждом ViPNet Coordinator KB задайте IP-адреса компьютеров, которые туннелирует он сам и противоположный ViPNet Coordinator KB (см. ниже).
3. Убедитесь, что в настройках каждого ViPNet Coordinator KB разрешен трафик для заданных туннелируемых узлов. По умолчанию на ViPNet Coordinator KB разрешен трафик для всех туннелируемых узлов, но эта настройка могла быть изменена. Подробнее см. в разделе Ограничение доступа туннелируемых узлов (на стр. 88).

Адреса туннелируемых узлов можно задать двумя способами:

- Централизованно в ViPNet Administrator. В этом случае для каждого ViPNet Coordinator KB задаются только адреса узлов, которые туннелирует он сам. После рассылки обновлений справочников и ключей заданные адреса появятся в настройках обоих ViPNet Coordinator KB. Этот способ является предпочтительным.
- Локально на ViPNet Coordinator KB путем редактирования конфигурационного файла `iplir.conf` (подробнее см. в документе «Справочник команд и конфигурационных файлов», в разделе «Файл iplir.conf»).

Для локальной настройки туннелей

1. На ViPNet Coordinator KB 1:
   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]` (содержащую параметр `id= 0x00010101`) добавьте параметр:
   ```
   tunnel= 192.168.1.2-192.168.1.4 to 192.168.1.2-192.168.1.4
   ```

   Примечание. Если на вашем ViPNet Coordinator KB виртуальные адреса для туннелируемых узлов задаются автоматически (по умолчанию), то вместо этой строки вы можете записать только первую ее часть:
   ```
   tunnel= 192.168.1.2-192.168.1.4
   ```
   В этом случае строка будет автоматически дополнена дублирующимся IP-адресом после запуска управляющей службы. Это справедливо для всех строк, начинающихся с `tunnel=`.

   1.4 В секцию `[id]`, описывающую ViPNet Coordinator KB 2 (содержащую параметр `id= 0x00020201`), добавьте параметр:
   ```
   tunnel= 192.168.2.2-192.168.2.2 to 192.168.2.2-192.168.2.2
   ```
   1.5 Сохраните изменения (**Ctrl+O**) и закройте редактор (**Ctrl+X**).
   1.6 Запустите управляющую службу:

```
hostname# iplir start
```

2 На ViPNet Coordinator KB 2:

2.1 Завершите работу управляющей службы:

```
hostname# iplir stop
```

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

```
hostname# iplir config
```

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

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

2.4 В секцию `[id]`, описывающую ViPNet Coordinator KB 1 (содержащую параметр `id= 0x00010101`), добавьте параметр:

```
tunnel= 192.168.1.2-192.168.1.4 to 192.168.1.2-192.168.1.4
```

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

2.6 Запустите управляющую службу:

```
hostname# iplir start
```

В результате компьютеры 1, 2, 3 смогут обращаться к серверу по адресу 192.168.2.2. При этом на участке между ViPNet Coordinator KB 1 и 2 пакеты будут передаваться в зашифрованном виде.

Ограничение доступа туннелируемых узлов

В приведенном выше примере доступ к серверу разрешен всем открытым компьютерам, туннелируемым ViPNet Coordinator KB 1, так как по умолчанию на ViPNet Coordinator KB 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 ViPNet Coordinator KB 1 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 ViPNet Coordinator KB 1 добавьте фильтры:

```
hostname# firewall tunnel add 1 src 192.168.1.2 dst 192.168.2.2 drop
hostname# firewall tunnel add 2 src 192.168.2.2 dst 192.168.1.2 drop
```

Таким образом, будет разрешен трафик в обоих направлениях только между туннелируемыми компьютерами 2, 3 и ViPNet Coordinator KB 2, 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 KB необходимо создать фильтры туннелируемых узлов, ограничив их протоколом TCP и номером порта 80:

- На ViPNet Coordinator KB 1 вместо приведенных выше фильтров туннелируемых узлов создайте следующие:

  ```
  hostname# firewall tunnel add src 192.168.1.3-192.168.1.4 dst 192.168.2.1 tcp dport 80 pass
  hostname# firewall tunnel add src 192.168.2.1 dst 192.168.1.3-192.168.1.4 tcp sport 80 pass
  ```

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

  ```
  hostname# firewall tunnel delete src @any dst @tunneledip pass
  hostname# firewall tunnel delete src @tunneledip dst @any pass
  hostname# firewall tunnel add src 192.168.2.2 dst 192.168.1.1 tcp sport 80 pass
  hostname# firewall tunnel add src 192.168.1.1 dst 192.168.2.2 tcp dport 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.

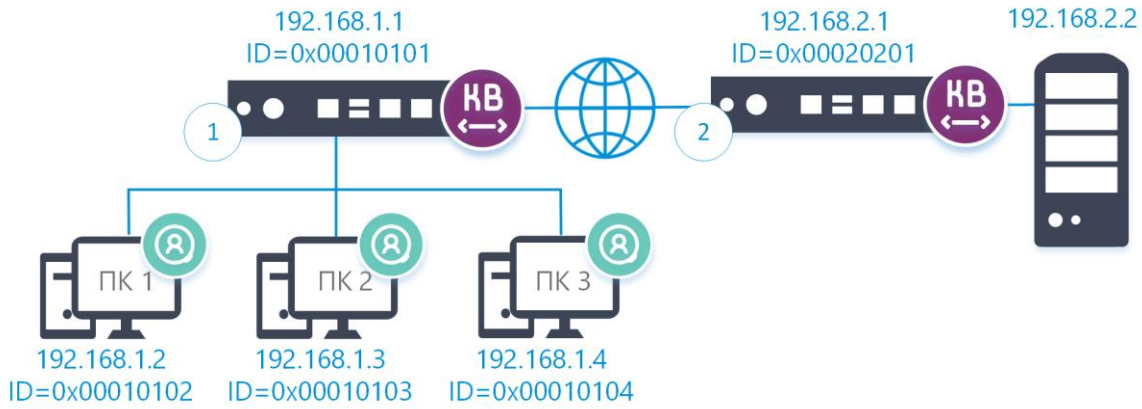


Рисунок 9. Схема организации туннелей между защищенными и открытыми узлами

Пусть компьютерам 1, 2, 3 присвоены в сети ViPNet идентификаторы 0x00010102, 0x00010103, 0x00010104 соответственно.

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

- Централизованно с помощью ViPNet Administrator. В этом случае задайте для ViPNet Coordinator KB 2 в качестве IP-адреса туннелируемого соединения IP-адрес сервера и разошлите справочники на ViPNet Coordinator KB 2 и все связанные с ним узлы вашей сети ViPNet.

  Мы рекомендуем использовать этот способ.

- Локально путем редактирования конфигурационного файла `iplir.conf` ViPNet Coordinator KB 2 и настройки программы ViPNet Client for Windows на компьютерах 1, 2, 3.

Чтобы организовать туннель локально:

1 На ViPNet Coordinator KB 2 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= 192.168.2.2-192.168.2.2 to 192.168.2.2-192.168.2.2
  ```

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

  1.5 Запустите управляющую службу:

  ```
  hostname# iplir start
  ```

2 Если в настройках ViPNet Coordinator KB 2 для туннелируемых узлов задан фильтр по умолчанию, то никакие дополнительные настройки не нужны. Иначе на ViPNet Coordinator KB 2 необходимо задать фильтры туннелируемых узлов, разрешающие трафик между сервером и компьютерами 1, 2, 3:

  2.1 Просмотрите фильтры туннелируемых узлов, заданных на ViPNet Coordinator KB 2:

  ```
  hostname# firewall tunnel show
  ```

2.2 Удалите фильтры туннелируемых узлов:

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

2.3 Добавьте следующие фильтры туннелируемых узлов:

```
hostname# firewall tunnel add src 192.168.2.2 dst
0x00010102,0x00010103,0x00010104 pass
```

```
hostname# firewall tunnel add src 0x00010102,0x00010103,0x00010104 dst
192.168.2.2 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 ViPNet Coordinator KB 2 в качестве IP-адреса для туннелирования IP-адрес 192.168.2.2 (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 192.168.2.2. При этом на участке между ViPNet Coordinator KB 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 ViPNet Coordinator KB 2 вместо приведенных выше фильтров туннелируемых узлов добавьте следующие:

```
hostname# firewall tunnel add src 192.168.2.2 dst 0x00010103,0x00010104 pass
```

```
hostname# firewall tunnel add src 0x00010103,0x00010104 dst 192.168.2.2 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 192.168.2.2 dst 0x00010103,0x00010104 tcp sport 80
pass
```

```
hostname# firewall tunnel add src 0x00010103,0x00010104 dst 192.168.2.2 tcp dport 80
pass
```

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

Управление копиями конфигурации VPN

Виды копий конфигурации VPN

В ViPNet Coordinator KB существует возможность создания копий конфигурации VPN, которые позволяют при необходимости осуществлять возврат соответствующего программного обеспечения к другому сохраненному состоянию (например, к состоянию до выполнения настроек или до приема обновленных справочников и ключей).

Копия конфигурации VPN — это совокупность параметров настроек ViPNet Coordinator KB. Существуют два вида копий конфигурации VPN:

- частичная — включает в себя следующие конфигурационные файлы ПО ViPNet Coordinator KB: `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.
- Вручную по команде. В этом случае всегда создаются только полные копии конфигурации VPN.

Создание копии конфигурации VPN



Внимание! По умолчанию перед обновлением ПО или справочников и ключей автоматически создается частичная копия конфигурации VPN (см. Виды копий конфигурации VPN на стр. 92).

Для создания копии текущей конфигурации VPN:

1 Завершите работу служб `iplircfg`, `failoverd` и `mftpd`:

```
hostname# iplir stop
hostname# failover stop
hostname# mftp stop
```

2 Сохраните копию конфигурации:

```
hostname# admin config save <имя копии конфигурации>
```

В имени копии конфигурации VPN могут использоваться только буквы латинского алфавита, цифры, символы «-» и «_». Если имя копии конфигурации VPN состоит из нескольких слов, заключите его в кавычки (например: «settings_copy_1»).

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 KB>) exists. Overwrite it (Yes/No)?
```

Выполните одно из действий:

- Чтобы подтвердить перезапись, введите y.
- Чтобы создать конфигурацию с другим именем, введите n.

4 Запустите службы iplircfg, failoverd и mftpd:

```
hostname# iplir start
hostname# failover start
hostname# mftp start
```

Созданную копию конфигурации VPN вы можете использовать для восстановления настроек ViPNet Coordinator KB (см. Восстановление настроек из копии конфигурации VPN на стр. 94).

Вы также можете просмотреть список всех созданных копий конфигурации VPN.

Просмотр списка копий конфигурации VPN

Для просмотра списка ранее созданных копий конфигурации VPN (см. Создание копии конфигурации VPN на стр. 92), выполните:

```
hostname# admin config list
```

В результате отобразится список, каждая строка которого содержит следующую информацию:

- имя;
- версия ViPNet Coordinator KB, 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-2017-10-10.

Если копия конфигурации VPN уже использовалась для восстановления настроек ViPNet Coordinator KB, 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 KB:

```
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 KB>],
```

указав имя копии конфигурации VPN и версию ViPNet Coordinator KB, в которой она была создана. Если имя копии конфигурации VPN состоит из нескольких слов, заключите его в кавычки.

3 Если текущая версия ViPNet Coordinator KB выше версии ViPNet Coordinator KB, 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 KB:

```
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 KB, введите `n` и нажмите **Enter**. Для подтверждения действия введите фразу `Yes, do as I say`.

5 Дождитесь завершения процесса восстановления настроек ViPNet Coordinator KB.

6 Запустите службы и драйверы ViPNet Coordinator KB:

```
hostname# iplir start
hostname# failover start
hostname# mftp start
```

Теперь вы можете продолжить работу с ViPNet Coordinator KB.

Удаление копии конфигурации VPN

Чтобы удалить ненужные копии конфигурации VPN:

```
hostname# admin config delete <имя копии конфигурации> [<версия ViPNet Coordinator KB>],
```

указав имя копии конфигурации VPN и версию ПО ViPNet Coordinator KB, 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 KB, укажите номер этой версии. Например:

```
hostname# admin config delete rollback-2020-08-* 4.2.0
```

Смена мастер-ключей

Смена мастер-ключей влечет за собой смену ключей обмена, ключей защиты и РНПК и может быть плановой или внеплановой. Плановую смену мастер-ключей производят с периодичностью один раз в 15 месяцев. Внеплановую смену мастер-ключей производят при компрометации ключей или при достижении максимального значения варианта персонального ключа пользователя. Подробнее о работе с мастер-ключами см. в разделе «Работа с мастер-ключами» документа «ViPNet Удостоверяющий и ключевой центр 4.6. Руководство администратора».

После смены мастер-ключей администратор сети ViPNet выпускает в ViPNet Administrator отложенные обновления справочников и ключей и рассылает их на все узлы ViPNet, соблюдая порядок рассылки.

Для корректной доставки обновлений администратор сети ViPNet:

1 Рассылает обновления справочников и ключей на все ViPNet-клиенты кроме рабочей станции, где установлено ПО ViPNet Administrator. Перед следующим шагом администратор сети ViPNet проверяет успешность доставки обновлений на узлы и, если обнаруживает ошибки, то принимает меры для обеспечения доставки и применения обновлений.



Внимание! Если в настройках ViPNet Administrator для узла ViPNet-клиент стоит минимальный уровень полномочий, то удаленно обновить персональный ключ будет невозможно. В этом случае требуется локальное обновление справочников и ключей.

2 Рассылает обновления справочников и ключей на все ViPNet-координаторы и проверяет успешность доставки и применения обновлений. Если удаленная рассылка и применение обновлений на координатор невозможна, то администратор сети ViPNet обеспечивает локальное обновление справочников и ключей.

3 Обновляет справочники и ключи на рабочей станции, где установлено ПО ViPNet Administrator.

Особенности смены мастер-ключей в сетях с ViPNet Coordinator KB

Очередность смены мастер-ключей

Для обеспечения надежного обновления справочников и ключей рекомендуем обновлять мастер ключи в два этапа:

1 Смените мастер-ключ персональных ключей, обеспечьте доставку и применение обновлений на узлах ViPNet.

2 Смените мастер-ключи ключей обмена и защиты, обеспечьте доставку и применение обновлений на узлах ViPNet.

Смена мастер-ключей на ViPNet Coordinator KB, работающем в кластере

Если вы используете ViPNet Coordinator KB в кластере горячего резервирования, то после смены мастер-ключей важно обеспечить доставку и применение обновлений справочников и ключей на оба узла кластера. Иначе при переключении кластера необновленный пассивный узел станет активным и связь с кластером ViPNet Coordinator KB прервется.

- Перед отправкой обновления справочников и ключей убедитесь, что пассивный узел кластера включен. Если это условие нельзя обеспечить во время обновления справочников и ключей, то после включения пассивного узла дождитесь репликации настроек.
- Время ожидания репликации настроек определяется удвоенным значением параметра `sendtime` в файле `failover.ini`. Например, если задано значение `sendtime=64`, то для репликации настроек подождите 128 секунд.

Обновление токена после удаленного обновления справочников и ключей

После смены мастер ключей при удаленном обновлении справочников и ключей на ViPNet Coordinator KB необходимо обновить персональный ключ на токене. Для этого подключите существующий токен к ViPNet Coordinator KB и введите PIN-код. После успешной аутентификации персональный ключ на токене будет обновлен автоматически.

Смена мастер-ключей на ViPNet Coordinator KB с несколькими пользователями

Если для ViPNet Coordinator KB было создано несколько пользователей, то при смене мастер-ключей для таких узлов обновление персонального ключа на токенах всех пользователей невозможно. Перед сменой мастер-ключей в ПО ViPNet Administrator удалите для таких узлов всех пользователей кроме одного.

Локальное обновление справочников и ключей на ViPNet Coordinator KB после смены мастер-ключей

Иногда удаленное обновление справочников и ключей на ViPNet Coordinator KB может быть затруднено или невозможно. В таких случаях после смены мастер-ключей обновите справочники и ключи на ViPNet Coordinator KB локально:

1. Получите у администратора сети ViPNet USB-накопитель с обновленным дистрибутивом ключей (файл `*.dst`) и новый токен.
2. Подключитесь к командному интерпретатору ViPNet Coordinator KB, используя для аутентификации текущий токен.
3. Перейдите в режим администратора. Для этого выполните команду `enable` и введите текущий пароль администратора ViPNet Coordinator KB.
4. Запустите локальное обновление справочников и ключей:

   ```
   hostname# admin keys update
   ```

   В ответ на запрос `Insert USB drive with application initialization file...` подключите полученный USB-накопитель и нажмите **Enter**.
5. Введите номер файла с обновленным дистрибутивом ключей, затем пароль к нему.

6. После запроса `Extract the current token and insert token with new personal key` отключите текущий токен и подключите новый.
7. Введите PIN-код от нового токена и дождитесь завершения обновления справочников и ключей.
8. Для завершения обновления в ответ на запрос подтвердите перезагрузку ViPNet Coordinator KB.

Локальное обновление справочников и ключей на кластере ViPNet Coordinator KB после смены мастер-ключей

Если пароль пользователя `user` совпадает с паролем для ключевого дистрибутива, то после смены мастер-ключей выполните локальное обновление справочников и ключей на активном узле. Пассивный узел будет обновлен автоматически.

Если пароли не совпадают, обновите справочники и ключи на обоих ViPNet Coordinator KB в следующем порядке:

1. На пассивном узле остановите службу защиты от сбоев:

   ```
   hostname# failover stop
   ```

2. Отключите Ethernet-кабель от интерфейсов синхронизации кластера.
3. Выполните локальное обновление справочников и ключей на пассивном узле.
4. Выполните локальное обновление справочников и ключей на активном узле.
5. Соедините интерфейсы синхронизации кластера Ethernet-кабелем.
6. На пассивном узле включите службу защиты от сбоев:

   ```
   hostname# failover start
   ```

7

Настройка сетевых фильтров

Основные принципы фильтрации трафика 100

Общие сведения о сетевых фильтрах 104

Группы объектов 106

Создание сетевого фильтра 112

Просмотр сетевых фильтров 119

Изменение сетевого фильтра 121

Удаление сетевого фильтра 122

Основные принципы фильтрации трафика

Виды трафика

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

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

Открытый трафик может быть локальным или транзитным:

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

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

- открытый локальный трафик — между открытым узлом и ViPNet Coordinator KB;
- защищенный трафик — трафик между узлами сети ViPNet;
- туннелируемый трафик — трафик между двумя открытыми (или открытым и защищенным) узлами, зашифрованный в пределах сети ViPNet;

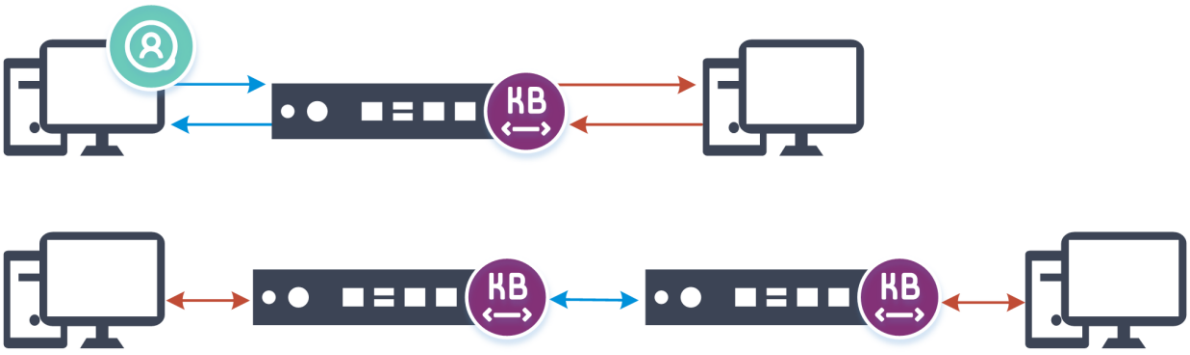


Рисунок 10. Туннелируемый трафик

- открытый транзитный трафик — трафик между открытыми (или открытым и защищенным) узлами, проходящий через координатор, который не является ни отправителем, ни получателем IP-пакетов.

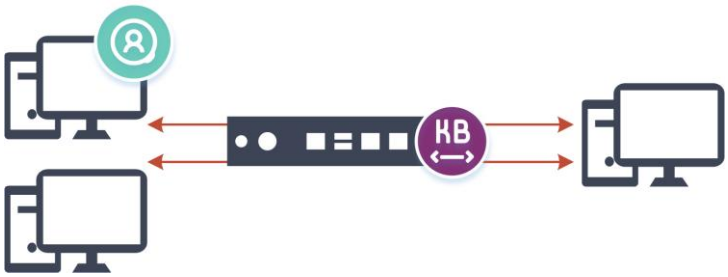


Рисунок 11. Открытый транзитный трафик

Для каждого вида трафика можно настроить соответствующие сетевые фильтры.

Принципы фильтрации трафика

Чтобы правильно настроить сетевые фильтры, ознакомьтесь с принципами и схемой фильтрации трафика в ViPNet Coordinator KB:

- 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 KB обрабатывает входящий трафик следующим образом:

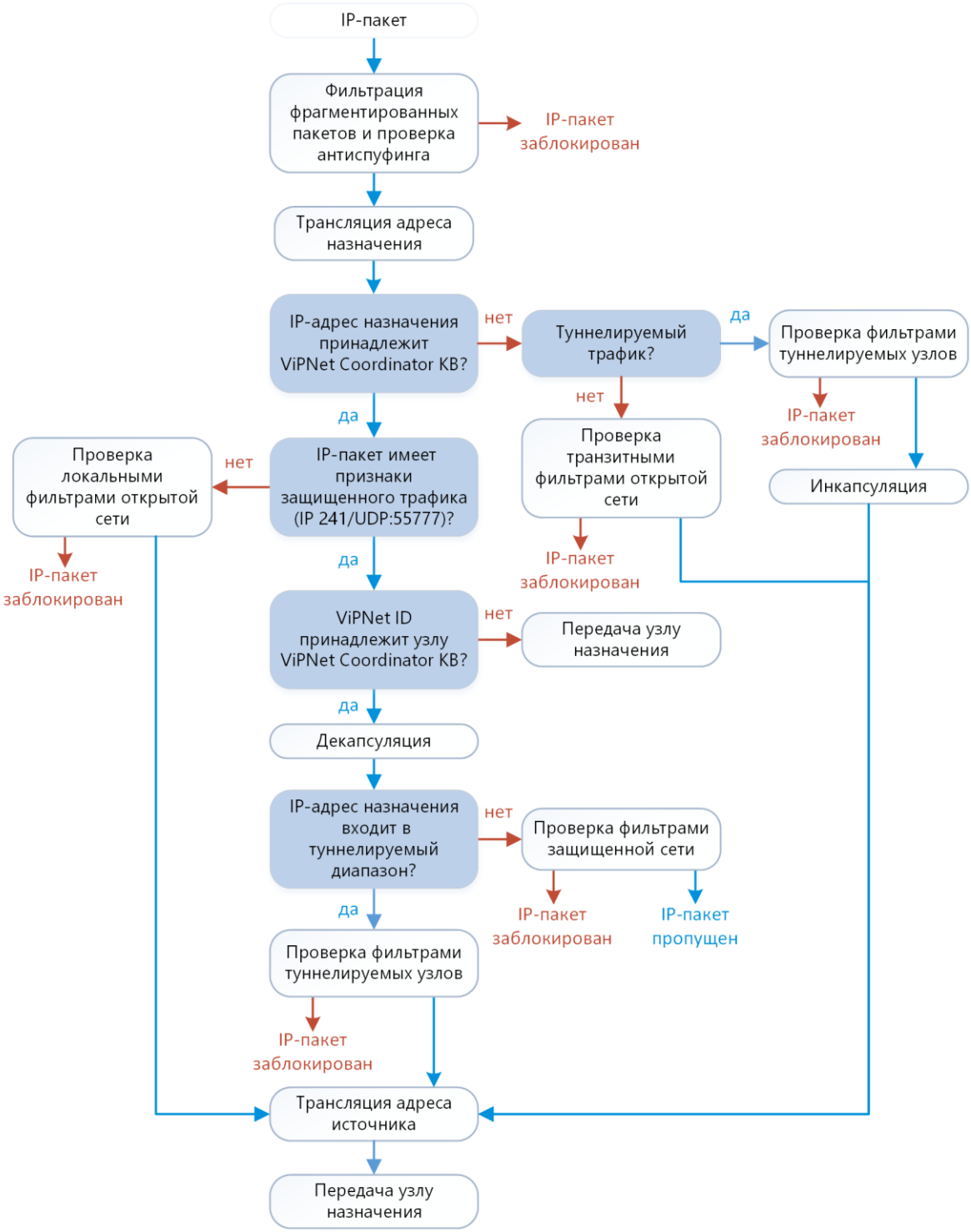


Рисунок 12. Схема фильтрации трафика

IP-пакет, поступивший на сетевой интерфейс ViPNet Coordinator KB, 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 на стр. 140). 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 на стр. 125):

- IP-пакет удовлетворяет условиям правила трансляции адресов.
- Включена трансляция прикладного протокола, к которому относится IP-пакет.

4 Если IP-пакет адресован ViPNet Coordinator KB 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 KB, 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 KB, 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 KB, IP-пакет проверяется фильтрами туннелируемых узлов. В результате IP-пакет блокируется или отправляется на туннелируемый узел после трансляции адреса источника (см. Трансляция адреса источника на стр. 125).
  - Если IP-адрес назначения в исходном IP-пакете не входит в туннелируемый диапазон ViPNet Coordinator KB, IP-пакет проверяется фильтрами защищенной сети, после чего блокируется или пропускается.

5 Если IP-пакет адресован ViPNet Coordinator KB и не имеет признаков защищенного трафика, IP-пакет проверяется локальными фильтрами открытой сети. В результате IP-пакет блокируется или проходит следующие проверки.

6 Если IP-пакет адресован не ViPNet Coordinator KB, а другому узлу, проверяется принадлежность IP-пакета туннелируемому трафику:

- Если IP-адрес отправителя входит в туннелируемый диапазон ViPNet Coordinator KB, 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-пакет не принадлежит туннелируемому трафику, он проверяется транзитными фильтрами открытой сети. В результате IP-пакет блокируется или отправляется получателю после трансляции адреса источника.

Примечание. При настройке фильтрации трафика необходимо учитывать, что TCP-порт 2047 и UDP-порт 2046 являются служебными портами ViPNet Coordinator KB. Поэтому входящий открытый трафик, поступающий на эти порты, блокируется.

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 KB 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 KB 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, 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 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.

Служебные фильтры создаются в ViPNet Coordinator KB автоматически, имеют самый высокий приоритет, то есть применяются в первую очередь, и недоступны для редактирования. После служебных фильтров применяются фильтры, поступившие в составе политик безопасности из ПО ViPNet Policy Manager. Эти фильтры также недоступны для редактирования. Затем применяются предустановленные фильтры и фильтры, заданные пользователем. Их можно изменять и удалять. Наименьший приоритет имеет блокирующий фильтр по умолчанию, который нельзя ни изменить, ни удалить.

Последовательность применения сетевых фильтров в порядке убывания приоритета представлена ниже.

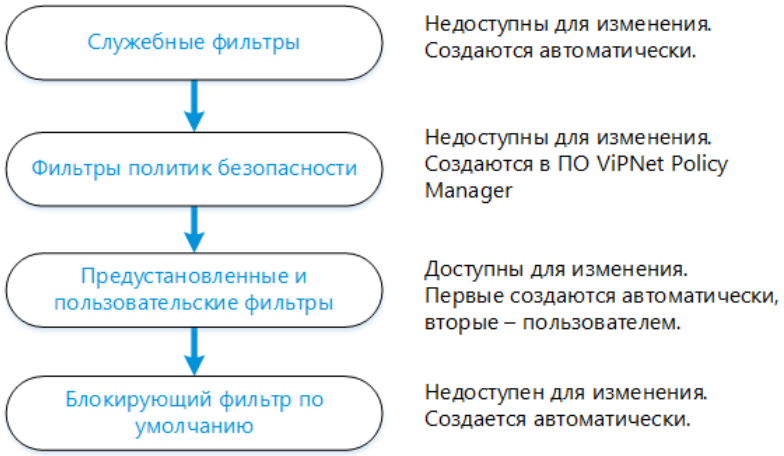


Рисунок 13. Последовательность применения сетевых фильтров

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 KB. Каждая группа объединяет несколько объектов одного типа (например, IP-адреса или сетевые интерфейсы). Группы можно указывать при задании параметров 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 и правилах трансляции сетевых адресов.

Каждая группа объектов относится к одному из следующих видов:

- Системные группы объектов (на стр. 107) — настроенные по умолчанию группы с фиксированными именами, которые могут использоваться для задания адресов отправителей и получателей IP-пакетов в сетевых фильтрах (см. Создание сетевого фильтра на стр. 112) и правилах трансляции адресов (см. Создание правила трансляции адресов на стр. 127), а также при создании пользовательских групп объектов (см. Создание пользовательской группы объектов на стр. 109). Такие группы объектов не отображаются в списках групп (см. Просмотр групп объектов на стр. 110), их нельзя ни изменить, ни удалить.
- Группы объектов из ViPNet Policy Manager — группы, получаемые в составе политик безопасности, и используемые в соответствующих 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 (на стр. 108). Такие группы можно использовать для задания параметров сетевых фильтров (см. Создание сетевого фильтра на стр. 112) и правил трансляции адресов (см. Создание правила трансляции адресов на стр. 127), а также при создании других пользовательских групп объектов (см. Создание пользовательской группы объектов на стр. 109). При необходимости их можно изменять и удалять (см. Удаление пользовательских групп объектов на стр. 111).

Системные группы объектов

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

Таблица 5. Системные группы объектов

Имя системной группы объектов	Описание и область применения группы
`allclients` (Все клиенты)	Все клиенты, с которыми у узла есть связь. Можно использовать в фильтрах защищенной сети и фильтрах туннелируемых узлов для задания адреса отправителя для входящих соединений узла или адреса получателя для исходящих соединений узла.
`allcoordinators` (Все координаторы)	Все координаторы, с которыми у узла есть связь. Можно использовать в фильтрах защищенной сети и фильтрах туннелируемых узлов для задания адреса отправителя для входящих соединений узла или адреса получателя для исходящих соединений узла.
`broadcast` (Широковещательные адреса)	Все широковещательные адреса. Можно использовать в фильтрах защищенной и открытой сетей для идентификации широковещательных адресов получателей. Использование других адресов совместно с этой группой недопустимо.
`local` (Мой узел)	Собственный узел ViPNet Coordinator KB. Можно использовать в фильтрах защищенной и открытой сетей для задания адреса отправителя для исходящих соединений узла или адреса получателя для входящих соединений узла. Использование других адресов совместно с этой группой недопустимо.
`remote` (Другие узлы)	Другие узлы (любые, кроме собственного). Можно использовать в фильтрах защищенной и открытой сетей для задания адреса отправителя для входящих соединений узла или адреса получателя для исходящих соединений узла. Использование других адресов совместно с этой группой недопустимо.
`tunneledip` (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 защищенной сети — все узлы сети ViPNet;
- для фильтров туннелируемых узлов — все IP-адреса или все узлы сети ViPNet;
- 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 KB 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.
- Группы протоколов, которые наиболее часто используются при создании сетевых фильтров. Они перечислены в приложении (см. Пользовательские группы протоколов по умолчанию на стр. 292).
- Группы расписаний:
  - `Workdays` — рабочие дни недели (с понедельника по пятницу).
  - `Weekends` — выходные дни недели (суббота и воскресенье).



Примечание. О том, как просмотреть пользовательские группы объектов, созданные на узле, см. в разделе Просмотр групп объектов (на стр. 110).

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:
  - `ip-object` — IP-адреса или доменные имена;
  - `vpn-object` — узлы сети ViPNet;
  - `interface-object` — сетевые интерфейсы;
  - `service-object` — протоколы;
  - `schedule-object` — расписания;
- имя группы объектов должно быть уникальным и не должно содержать пробелов и символов «"»;
- в параметрах `<состав>` и `<исключения>` указываются объекты, входящие и не входящие в группу соответственно, причем:
  - IP-адреса и идентификаторы узлов сети ViPNet указываются через запятую или в виде диапазона (подробнее см. в разделах Адрес отправителя (на стр. 113) и Адрес получателя (на стр. 115));
  - сетевые интерфейсы разделяются пробелом, и перед именем каждого сетевого интерфейса необходимо указать слово `interface`;
  - протоколы или расписания указываются в соответствии с синтаксисом, описанным в разделах Протокол (на стр. 116) и Расписание (на стр. 116).



Примечание. Если при вводе команды была допущена синтаксическая ошибка, то в результате выполнения такой команды появится сообщение с указанием слова, содержащего ошибку. Исправьте ошибку и выполните команду еще раз.

В результате будет создана группа объектов, и вы сможете использовать ее для задания параметров сетевых фильтров (см. Создание сетевого фильтра на стр. 112) и правил трансляции сетевых адресов (см. Создание правила трансляции адресов на стр. 127).

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, выполните одну из команд:

- для просмотра всех групп объектов:

  ```
  hostname> firewall object show
  ```

- для просмотра групп объектов определенного типа (см. Создание пользовательской группы объектов на стр. 109):

  ```
  hostname> firewall <тип объектов> show
  ```

  Например, для просмотра всех групп IP-адресов:

  ```
  hostname> firewall ip-object show
  ```

В результате выполнения команды будет отображена таблица, содержащая те или иные группы объектов. Пример таблицы групп IP-адресов представлен на рисунке ниже.

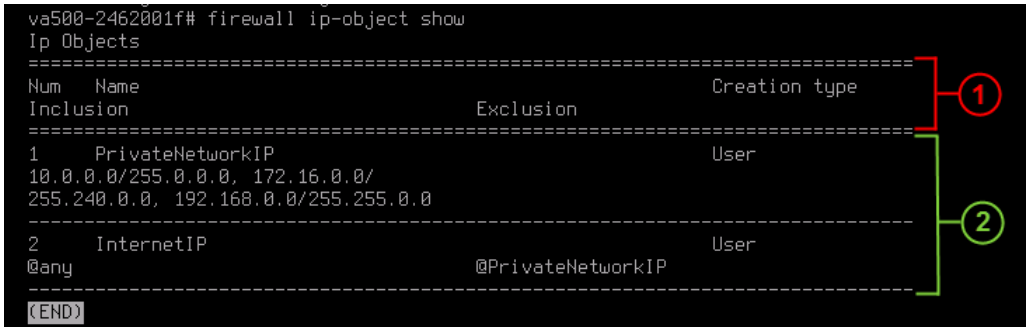


Рисунок 14. Просмотр групп IP-адресов

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 — `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>
```



Примечание. 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 KB, необходимо сначала удалить все сетевые фильтры, в которых содержится эта группа.

В результате группа объектов будет удалена, и вы больше не сможете использовать ее для задания параметров сетевых фильтров, правил трансляции сетевых адресов и при создании других групп объектов.

Создание сетевого фильтра

Для каждого вида трафика создаются отдельные сетевые фильтры, которые хранятся в соответствующей таблице (см. Просмотр сетевых фильтров на стр. 119).

Чтобы создать сетевой фильтр, выполните:

`hostname# firewall <тип> add [<номер>] [rule <имя>] src <адрес отправителя> dst <адрес получателя> [<протокол>] [<расписание>] <действие>`, указав:

- `<тип>` — тип создаваемого сетевого фильтра:
  - `local` — локальный фильтр открытой сети;
  - `forward` — транзитный фильтр открытой сети;
  - `vpn` — фильтр защищенной сети;
  - `tunnel` — фильтр туннелируемых узлов;
  - `nat` — правило трансляции сетевых адресов (см. Трансляция сетевых адресов (NAT) на стр. 124).



Примечание. Если в ViPNet Центр управления сетью запрещено использование туннелирования на узле ViPNet Coordinator KB, 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 порядковый номер фильтра, тем выше его приоритет (наиболее приоритетный фильтр имеет номер 1). При создании фильтра с порядковым номером, меньшим максимального номера в таблице фильтров, номера фильтров, следующих за этим фильтром, автоматически увеличиваются на единицу (то есть их приоритет понижается).
- `<имя>` — имя сетевого фильтра. Если имя состоит из нескольких слов, разделенных пробелом, заключите его в кавычки (например, `rule "number one"`).



Примечание. 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 (на стр. 113).

- Адрес получателя (на стр. 115).
- Протокол (на стр. 116).
- Расписание (на стр. 116).
- Действие (на стр. 117).



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 на стр. 119), изменить (см. Изменение сетевого фильтра на стр. 121) или удалить его (см. Удаление сетевого фильтра на стр. 122).



Примечание. Если при вводе команды была допущена синтаксическая ошибка, то в результате выполнения такой команды появится сообщение с указанием слова, содержащего ошибку. Исправьте ошибку и выполните команду еще раз.

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 ее жизни (TTL, «Time to live»), полученные в ответ на такой запрос, сохраняются в кэше. По истечении времени ее жизни отправляется новый запрос на DNS-сервер для актуализации данных в кэше. 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 диапазон 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 (см. Группы объектов на стр. 106). 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;
- 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 (см. Группы объектов на стр. 106). 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`.

- 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 (см. Адрес отправителя на стр. 113), 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 (на стр. 107):
  - 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`.
  - 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 этих параметров сначала необходимо указать параметр `sport`, затем — параметр `dport`. Например: `tcp sport 1024-65535 dport 1024;`
  - 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 (на стр. 108), в виде:

  `service @<имя группы>`

  Например: `service @DNS`.

Расписание

Расписание задает время действия сетевого фильтра и не является его обязательным параметром.

Примечание. Интервалы времени в расписании указываются для часового пояса (временной зоны) UTC, которое может отличаться от настроенного на ViPNet Coordinator KB. Если расписание не задано, то фильтр действует постоянно.

Расписание описывается одной из следующих лексем:

- `daily <чч:мм>-<чч:мм>` — фильтр действует ежедневно в течение заданного интервала времени. Время указывается в 24-часовом формате: `чч` — часы, `мм` — минуты.

- `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.
- `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.
- `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 (на стр. 108).

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.

Примеры создания сетевых фильтров

Чтобы разрешить отправку FTP-запросов со своего узла открытому узлу с IP-адресом 192.168.2.4, выполните:

```
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, выполните:

```
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, блокирующий IP-пакеты от защищенного узла с идентификатором 0x1a12000a в защищенную сеть с идентификатором 0x1b14, выполните:

```
hostname# firewall vpn add src 0x1a12000a dst 0x1b14 drop
```

Чтобы создать сетевой 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, выполните:

```
hostname# firewall forward add src 192.168.0.1 dst 192.168.30.3 pass
```

Чтобы создать сетевой фильтр, разрешающий отправку IP-пакетов от туннелируемого узла с адресом 192.168.0.1 на защищенный узел с идентификатором 0x1234abab, выполните:

```
hostname# firewall tunnel add src 192.168.0.1 dst 0x1234abab pass
```

Чтобы создать сетевой фильтр, разрешающий отправку IP-пакетов от туннелируемого узла с адресом 192.168.0.1 туннелируемому узлу с адресом 192.168.2.3, выполните:

```
hostname# firewall tunnel add src 192.168.0.1 dst 192.168.2.3 pass
```

Просмотр сетевых фильтров

Вы можете просмотреть сетевые фильтры, заданные на узле. 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 на стр. 129).

- для просмотра сетевых фильтров определенного типа (см. Создание сетевого фильтра на стр. 112):

  ```
  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 (на стр. 113), адрес получателя (на стр. 115), протокол (на стр. 116) или действие (на стр. 117) 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 осуществляется 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.

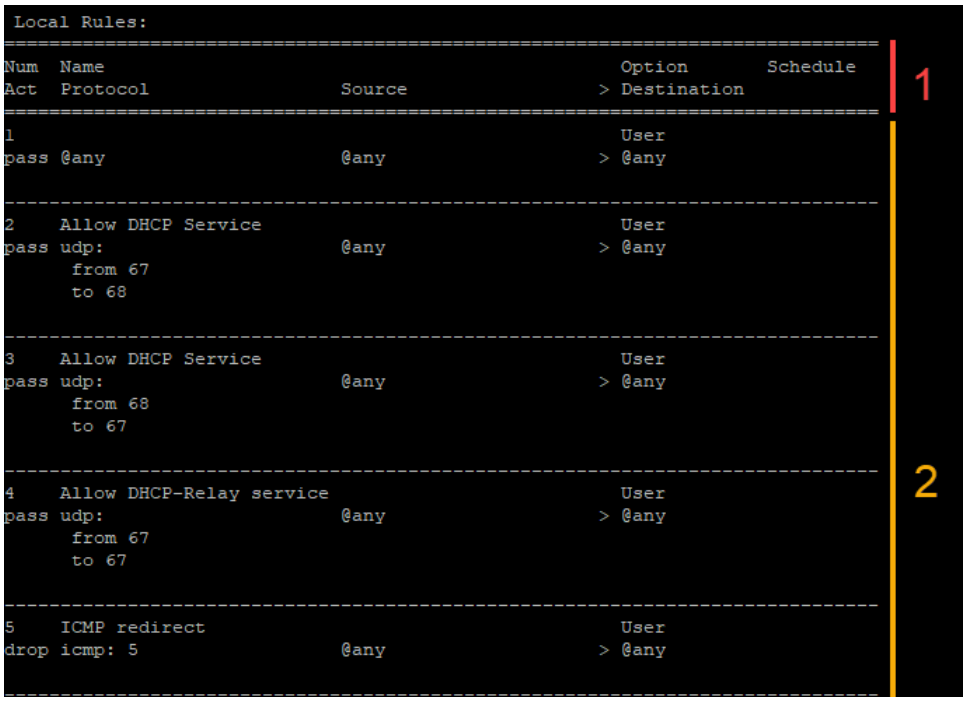


Рисунок 15. Просмотр локальных фильтров открытой сети

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 — `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 отредактировать 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:

```
hostname# firewall <тип> change append <номер> <параметры>
```

указав в качестве `<параметры>` адрес отправителя (на стр. 113), адрес получателя (на стр. 115), протокол (на стр. 116) или расписание (на стр. 116). При необходимости вы можете указать несколько параметров.



Примечание. Если при вводе команды была допущена синтаксическая ошибка, то в результате выполнения такой команды появится сообщение с указанием слова, содержащего ошибку. 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 на стр. 119).

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 его номер из столбца `Num`.
  - Если найден один фильтр, подтвердите его удаление: введите `y`.

После удаления изменятся номера (приоритеты) фильтров. Номера фильтров с более низким приоритетом уменьшатся на единицу.



8

Настройка правил трансляции сетевых адресов

Трансляция сетевых адресов (NAT) 124

Создание правила трансляции адресов 127

Просмотр, изменение, удаление правил трансляции сетевых адресов 129

Примеры использования правил трансляции адресов 131

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 KB может выполнять:

- Трансляцию адреса источника (маскарадинг (masquerading) или динамическая трансляция).

  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 (форвардинг портов (port forwarding), статическая трансляция).

  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 на стр. 127). 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 16. 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 следующие 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 на стр. 125).

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 17. 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 KB 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 на стр. 129).

Примечание. Правила трансляции IP-адресов работают только при наличии транзитных фильтров открытой сети (см. Создание сетевого фильтра на стр. 112), пропускающих трафик от узлов, к которым будет применяться трансляция адресов.

Чтобы создать правило трансляции адресов, выполните:

`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 (на стр. 113) и адрес получателя (на стр. 115) 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 (на стр. 116) и расписание (на стр. 116) 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 была допущена синтаксическая ошибка, то в результате выполнения такой команды появится сообщение с указанием слова, содержащего ошибку. 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 на стр. 129).

Примечание. Созданное правило трансляции сетевых адресов будет работать для новых соединений. Для уже установленных соединений правило будет работать после истечения времени жизни соединения.

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 KB.

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, протокол или действие правила (см. Создание правила трансляции адресов на стр. 127). При необходимости вы можете указать несколько параметров.

Внимание! Поиск правил трансляции сетевых адресов для отображения осуществляется по строгому совпадению с указанными параметрами.

В результате выполнения команды будет отображена таблица, содержащая те или иные правила трансляции сетевых адресов. Пример такой таблицы показан на рисунке ниже.

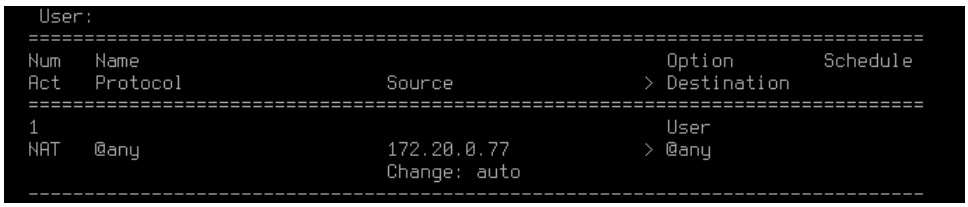


Рисунок 18. Просмотр правил трансляции сетевых адресов

Таблица содержит следующие столбцы:

- `Num` — порядковый номер правила, определяющий его приоритет.
- `Name` — имя правила.
- `Option` — категория правила: для правил из ViPNet Policy Manager — `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 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 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 (на стр. 121) и Удаление сетевого фильтра (на стр. 122).

Примеры использования правил трансляции адресов

Организация доступа пользователей к интернету

С помощью ViPNet Coordinator KB 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 KB.



Рисунок 19. Организация доступа пользователей к Интернету с помощью ViPNet Coordinator KB

На приведенной схеме локальная сеть имеет адресное пространство 192.168.1.0/24. Один интерфейс ViPNet Coordinator KB 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 KB 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 KB:

   ```
   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:
   - чтобы пользователи могли работать в интернете с использованием любого протокола:

     ```
     hostname# firewall forward add src 192.168.1.0/24 dst @InternetIP pass
     ```

   - 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 сетевые 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 узлы сети. Чтобы защитить локальную сеть, ее разбивают на сегменты, размещая серверы в демилитаризованной зоне . 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 KB. На рисунке ниже приведена схема сети с веб-сервером, размещенным в демилитаризованной зоне (DMZ), и ViPNet Coordinator KB с тремя сетевыми интерфейсами, установленным на границе локальной сети.

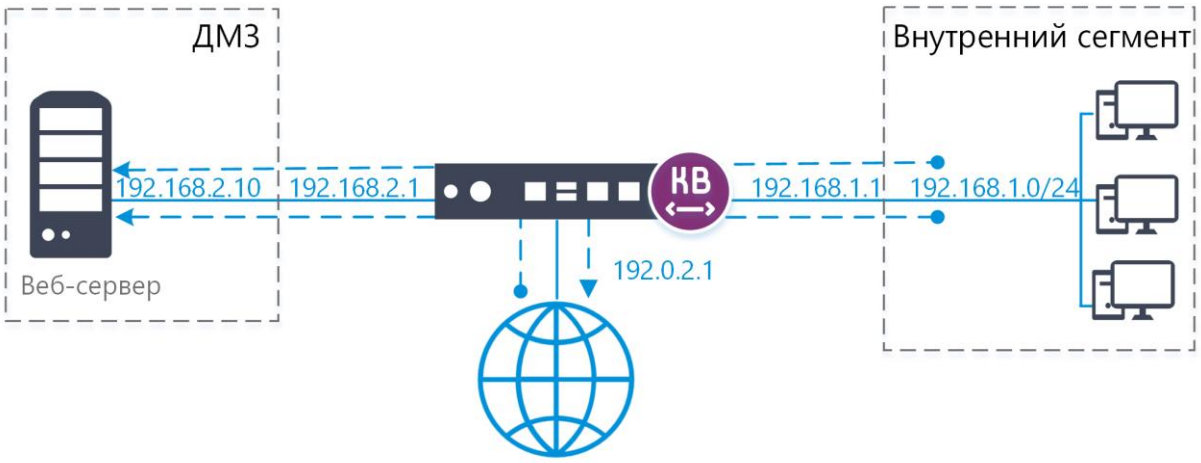


Рисунок 20. Использование ViPNet Coordinator KB для размещения общедоступного сервера в DMZ

Два сетевых интерфейса ViPNet Coordinator KB (`eth0` и `eth1`) имеют частные IP-адреса и подключены к сегментам локальной сети, а сетевой интерфейс `eth2` имеет публичный IP-адрес и подключен к Интернету. Пусть заданы следующие сетевые настройки:

- компьютеры во внутреннем сегменте имеют адреса `192.168.1.0/24`;
- веб-сервер имеет адрес `192.168.2.10`;
- сетевые 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 DMZ;
- из Интернета к веб-серверу.

Чтобы установление таких соединений было возможно, на ViPNet Coordinator KB необходимо создать сетевые фильтры транзитных IP-пакетов и правила трансляции сетевых адресов. Для этого в командном интерпретаторе выполните следующие действия:

1 Включите режим администратора (см. Включение режима администратора ViPNet Coordinator KB на стр. 20).

2 Создайте транзитные фильтры открытой сети:

   - фильтр, разрешающий соединения из внутреннего сегмента с любыми IP-адресами:

     ```
     hostname# firewall forward add src 192.168.1.0/24 dst @any pass
     ```

   - фильтр, разрешающий соединения с веб-сервером по протоколу HTTP с любых IP-адресов:

     ```
     hostname# firewall forward add src @any dst 192.168.2.10 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 192.168.1.0/24 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 192.168.2.10
     ```

В результате пользователи локальной сети и внешние пользователи смогут обращаться к веб-серверу по адресу 192.168.2.10. Пользователи локальной сети также будут иметь доступ в интернет, при этом их компьютеры будут защищены от атак из интернета.

Использование дополнительных IP-адресов на интерфейсе одиночного сервера

Рассмотрим пример использования дополнительных IP-адресов на одном из сетевых интерфейсов координатора ViPNet Coordinator KB, работающего в одиночном режиме.

Пусть в локальной сети находятся серверы, предоставляющие различные сервисы (например, почтовый сервер, веб-сервер и FTP-сервер). Требуется, чтобы эти сервисы были доступны из внешней сети, при этом серверы не имеют публичных адресов. Для решения этой задачи на границе локальной сети устанавливается ViPNet Coordinator KB, к которому подключаются серверы.

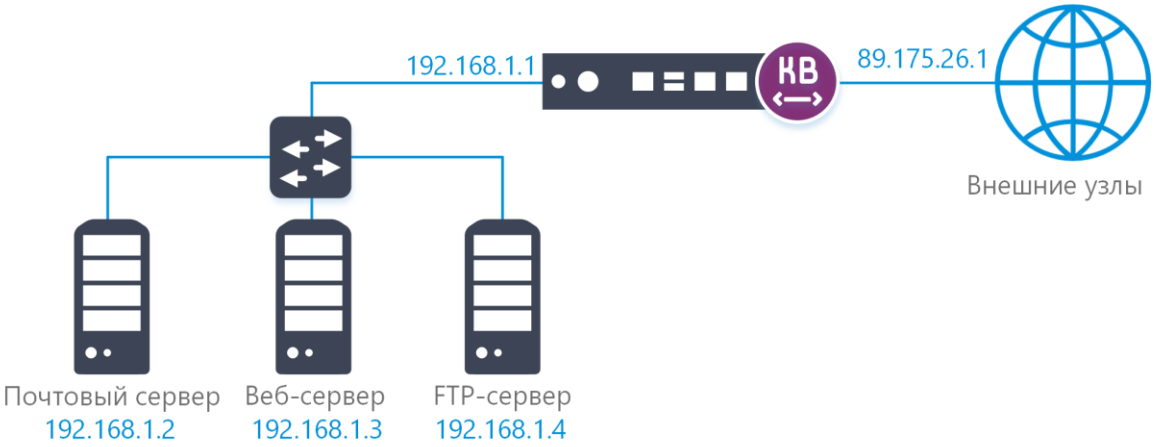


Рисунок 21. Подключение серверов к ViPNet Coordinator KB

ViPNet Coordinator KB имеет два сетевых интерфейса, один из которых имеет реальный IP-адрес и подключен к интернету, а другой — имеет частный IP-адрес и подключен к локальной сети. Пусть координатор имеет внешние адреса с 89.175.26.1 по 89.175.26.4, внутренний адрес 192.168.1.1, а подключенные к нему серверы имеют частные адреса с 192.168.1.2 по 192.168.1.4. Предполагается, что координатор работает в режиме **Со статической трансляцией адресов**.

Чтобы обеспечить доступ к серверам из внешней сети:

1 На внешнем интерфейсе `eth0` задайте дополнительные 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
hostname# inet ifconfig eth0 address add 89.175.26.4 netmask 255.255.255.0
```

2 Создайте правила трансляции адреса получателя:

```
hostname# firewall nat add src @InternetIP dst 89.175.26.2 tcp dport 80 change dst
192.168.1.2:25
hostname# firewall nat add src @InternetIP dst 89.175.26.3 tcp dport 81 change dst
192.168.1.3:80
hostname# firewall nat add src @InternetIP dst 89.175.26.4 tcp dport 82 change dst
192.168.1.4:21
```

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
hostname# firewall forward add src @InternetIP dst 192.168.1.4 tcp dport 21 pass
```

При указанных настройках к почтовому серверу можно будет обращаться по адресу 89.175.26.2 и номеру порта 80, к веб-серверу — по адресу 89.175.26.3 и номеру порта 81, к FTP-серверу — по адресу 89.175.26.4 и номеру порта 82. На схеме ниже показана организация доступа к серверам с использованием дополнительных адресов на внешнем интерфейсе координатора.

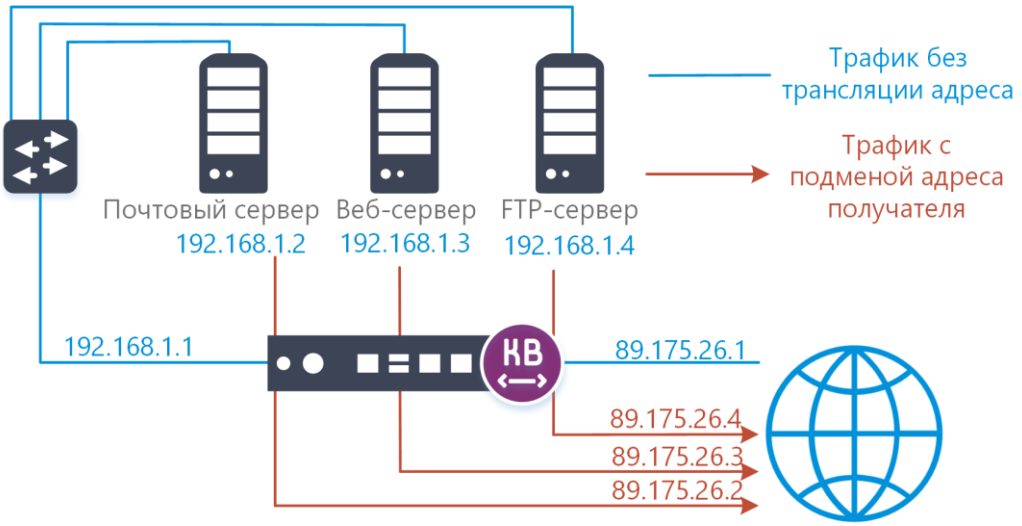


Рисунок 22. Схема организации доступа к серверам с помощью дополнительных IP-адресов

Обеспечение доступа удаленных клиентов к серверу по технологии PPTP VPN

ViPNet Coordinator KB 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 KB, 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 KB.

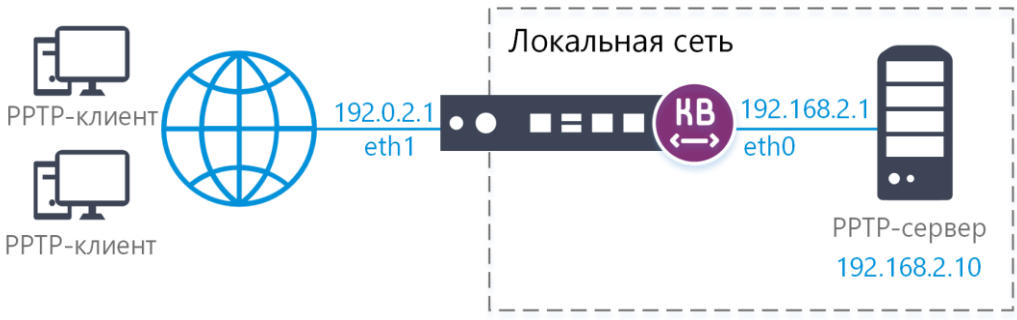


Рисунок 23. Схема взаимодействия PPTP-клиентов с PPTP-сервером через ViPNet Coordinator KB

На приведенной схеме один интерфейс ViPNet Coordinator KB (`eth0`) подключен к локальной сети и имеет частный адрес 192.168.2.1, другой интерфейс (`eth1`) подключен к интернету и имеет публичный адрес 192.0.2.1. PPTP-сервер имеет в локальной сети адрес 192.168.2.10.

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 KB 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
```

В результате произведенных настроек удаленные PPTP-клиенты смогут взаимодействовать с PPTP-сервером, установленным в локальной сети.

9

Тонкая настройка межсетевого экрана

Обработка фрагментированных пакетов в ViPNet Coordinator KB 138

Настройка антиспуфинга 140

Блокирование веб-сайтов по доменным именам с помощью сетевых фильтров 142

Настройка дополнительных параметров межсетевого экрана 143

Обработка сетевого трафика в соответствии с приоритетом 145

Обработка фрагментированных пакетов в ViPNet Coordinator KB

Сетевые протоколы содержат механизмы управления размером IP-пакетов, позволяющие сетевым узлам уменьшать размер IP-пакета при необходимости (например, для избежания излишней фрагментации). В то же время, ViPNet Coordinator KB при шифровании трафика добавляет к каждому IP-пакету служебный заголовок (выполняет инкапсуляцию), вследствие чего размер сформированного ViPNet-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 KB предусмотрены встроенные средства решения этой проблемы, описанные ниже.

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 KB 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 файла `iplir.conf` (подробнее см. в документе «Справочник команд и конфигурационных файлов»).

Обработка IP-пакетов с флагом Don't fragment (DF)

Рассмотрим случай, когда ViPNet Coordinator KB туннелирует сетевые узлы, которые отправляют через него IP-пакеты с размером, не превышающим размер MTU, заданный на внешнем сетевом интерфейсе ViPNet Coordinator KB. При шифровании трафика размер IP-пакета может превысить MTU, и в этом случае IP-пакет будет фрагментирован на более мелкие.

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 KB его пропустит и сбросит флаг DF (который не восстановится при последующей декапсуляции).
- Если размер IP-пакета после инкапсуляции превышает MTU, то ViPNet Coordinator KB его отбросит (при этом в журнале IP-пакетов не будет информации об этом событии) 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 следующих 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 (согласно RFC 4459), то такой трафик будет пропущен.

Блокировка входящих фрагментированных пакетов

В ряде случаев фрагментированные IP-пакеты используются как средство сетевой атаки «отказ в обслуживании» (DDoS). В качестве меры защиты от такой атаки используется опция межсетевого экрана ViPNet Coordinator KB `block-fragmented-packets` (см. Настройка дополнительных параметров межсетевого экрана на стр. 143), которая блокирует все входящие фрагментированные IP-пакеты на всех сетевых интерфейсах.

Настройка антиспуфинга

В ViPNet Coordinator KB реализована функция антиспуфинга, то есть блокирования входящих IP-пакетов от отправителей, IP-адреса которых недопустимы на данном сетевом интерфейсе узла. Антиспуфинг работает только для открытого трафика, поскольку для защищенного трафика IP-адрес отправителя не имеет значения. Открытые пакеты сначала проверяются правилами антиспуфинга, а затем уже обрабатываются сетевыми фильтрами.

Основная задача антиспуфинга — это 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 IP-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 KB выключен. Для повышения уровня безопасности сети включите его:

1 Проверьте текущее состояние антиспуфинга:

```
hostname# iplir option get antispoofing
```

2 Включите антиспуфинг в ViPNet Coordinator KB:

```
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 6. 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, при этом 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.

Блокирование веб-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 KB 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 KB 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 vpn add 2 rule "Deny DNS vpn" src @local dst @any service @DNS drop
   ```

   ```
   hostname# firewall vpn add 3 rule "Deny DNS vpn" src @any dst @local 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 KB.

4. Задайте IP-адрес ViPNet Coordinator KB в качестве DNS-сервера для клиентов защищенной сети.

После этого вы можете создавать запрещающие сетевые фильтры для доменных имен веб-сайтов, например:

```
hostname# firewall forward add src @any dst facebook.com drop
```

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` — состояние функции антиспуфинга. Возможные значения:
  - `off` (по умолчанию) — антиспуфинг выключен.
  - `on` — антиспуфинг включен.
- `block-fragmented-packets` — состояние функции блокирования входящих фрагментированных IP-пакетов, которые принимаются по всем сетевым интерфейсам. Возможные значения:
  - `off` (по умолчанию) — блокирование фрагментированных пакетов выключено.
  - `on` — блокирование фрагментированных пакетов включено.
- `block-other-protocols` — состояние функции блокирования IP-пакетов, передаваемых по всем протоколам, кроме IP, ARP и RARP. Возможные значения:
  - `off` (по умолчанию) — пропускаются пакеты, передаваемые по всем проколам, при этом обрабатываются только IP-пакеты.
  - `on` — блокируются пакеты, передаваемые по всем протоколам, кроме IP, ARP и RARP. ARP- и RARP-пакеты пропускаются всегда, поскольку это необходимо для корректной работы протокола IP.
- `cleanup-interval` — время, по истечении которого производится удаление соединений с истекшим временем жизни. Указывается в десятых долях секунды, по умолчанию — 20.
- `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).
- `dynamic-ports` — диапазон портов, используемых для динамической трансляции IP-адресов. Выводится в формате `<Начало диапазона>-<Конец диапазона>` (по умолчанию 4096–65535).
- `dynamic-timeouts` — состояние режима динамических тайм-аутов соединений. Возможные значения: `off` (по умолчанию) или `on`.

Данный режим используется для противодействия флуд-атакам. Когда количество соединений достигает определенного процента от максимума, тайм-ауты всех соединений уменьшаются на определенную величину. Эта величина тем больше, чем ближе число соединений к максимуму. При этом тайм-ауты не уменьшаются ниже определенного минимума. Когда количество соединений уменьшается до определенного процента от максимального, значения тайм-аутов восстанавливаются до исходной величины.

- `max-connections` — максимальное количество параллельно установленных соединений. Значение данного параметра зависит от исполнения ViPNet Coordinator KB.

Таблица 7. Значения параметра max-connections для разных аппаратных платформ ViPNet Coordinator KB

Аппаратная платформа ViPNet Coordinator KB	Значение по умолчанию	Максимальное значение
KB100 N1	150000	150000
KB100 Q1/Q2	500000	500000
KB1000 Q6	800000	1000000
KB1000 Q8	2000000	4000000
KB2000 Q4	2500000	3000000
KB2000 Q5	4000000	8000000
KB5000 Q1	6000000	6500000
KB5000 Q2	8000000	16000000

Для изменения настройки какого-либо параметра межсетевого экрана в ViPNet Coordinator KB выполните:

```
hostname# iplir option set <параметр> <значение>
```

Обработка сетевого трафика в соответствии с приоритетом

В ViPNet Coordinator KB реализована поддержка протокола классификации сетевого трафика DiffServ. Использование этого протокола предполагает, что в заголовок каждого IP-пакета может быть добавлена DSCP-метка, задающая приоритет обработки пакета.

Когда на ViPNet Coordinator KB поступают IP-пакеты с DSCP-метками, по значению метки определяется принадлежность каждого IP-пакета к одному из 8 классов приоритета. IP-пакеты, принадлежащие к классу с более высоким приоритетом, всегда обрабатываются раньше пакетов, принадлежащих к менее приоритетным классам.

При этом, при зашифровании и расшифровании (инкапсуляциии и декапсуляции) IP-пакета DSCP-метка перемещается соответственно из закрытой в открытую или из открытой в закрытую часть IP-пакета. Поэтому в случае, когда на ViPNet Coordinator KB приходит открытый IP-пакет с DSCP-меткой, ViPNet Coordinator KB его шифрует и отправляет далее получателю, по пути следования IP-пакета его DSCP-метка может быть снята или изменена и останется такой после расшифрования.

ViPNet Coordinator KB поддерживает следующие политики обработки трафика с учетом приоритета в соответствии с RFC 2474 и RFC 2475:

- Assured Forwarding — гарантированная переадресация.
- Class Selector — политика, обеспечивающая обратную совместимость с полем IP Precedence.
- Default PHB (Best Effort) — негарантированная доставка.

ViPNet Coordinator KB гарантирует обработку трафика в соответствии с его приоритетом в том случае, если на сетевом оборудовании (например, коммутаторе), подключенном к ViPNet Coordinator KB, поддерживается эта функция, а также включено управление потоком передачи данных (Ethernet Flow Control).

10

Настройка обработки прикладных протоколов

О прикладных протоколах 147

Поддерживаемые прикладные протоколы 150

Настройка параметров обработки прикладных протоколов 151

О прикладных протоколах

Прикладные протоколы обеспечивают работу пользовательских сетевых сервисов, например, IP-телефонии, DNS-службы, FTP-службы. При использовании данных служб IP-адреса часто передаются в теле IP-пакета. 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 на случайно выбранный порт. Поскольку номер порта заранее неизвестен, то для IP-пакетов невозможно создать разрешающий сетевой фильтр, следовательно, соединение будет заблокировано.

ViPNet Coordinator KB может выполнять роль шлюза прикладного уровня, которая позволяет решить перечисленные проблемы, обеспечивая:

- Подмену виртуального IP-адреса защищенного узла в теле IP-пакета на реальный.
- Статическую и динамическую трансляцию IP-адреса при передаче транзитного трафика (только для протокола FTP).
- Включение сетевого фильтра, пропускающего IP-пакеты, для дополнительного соединения на случайно выбранный порт, который используется прикладным протоколом. При этом для установления управляющего соединения с открытыми узлами на узле необходимо задать соответствующие фильтры открытой сети.

Вы можете использовать эту функцию для всех видов трафика. Список поддерживаемых прикладных протоколов см. в разделе Поддерживаемые прикладные протоколы (на стр. 150).

Обработка прикладных протоколов не предполагает автоматического разрешения на установление управляющего соединения с открытыми узлами. Установление управляющего соединения с открытыми узлами выполняется в соответствии с фильтрами трафика.

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).

Установление TCP-соединений по протоколу FTP происходит в следующем порядке:

1. Клиент создает 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:
   - номер порта, открытого клиентом для получения данных с сервера (активный режим);

- запрос на номер порта сервера, к которому будет подключаться клиент (пассивный режим).

3 В зависимости от используемого режима передачи данных, соединение устанавливается одним из двух способов:

- сервер создает дополнительное соединение для передачи данных на указанный клиентом порт (активный режим);
- сервер по управляющему соединению передает клиенту номер порта, к которому клиент создает дополнительное соединение (пассивный режим).

Таким образом, в активном режиме соединение для передачи данных создает сервер, а в пассивном — клиент.

Для установления управляющего и дополнительного соединений в активном или пассивном режиме работы протокола FTP:

- Создайте фильтр открытой сети, разрешающий исходящее TCP-соединение на 21-й порт FTP-сервера.
- Для разрешения дополнительного соединения в активном режиме работы включите обработку протокола FTP (см. Настройка параметров обработки прикладных протоколов на стр. 151), которая добавляет необходимый сетевой фильтр.
- Для разрешения дополнительного соединения в пассивном режиме работы настройки не требуются.

Обработка протокола SIP

Протокол SIP предназначен для организации сеансов связи (мультимедийных конференций, телефонных соединений) и распределения мультимедийной информации.

SIP-клиент отправляет запрос (например, приглашение для начала сеанса связи, подтверждение приема ответа на запрос, завершение сеанса связи) другому SIP-клиенту с указанием его SIP-адреса. В зависимости от способа установления соединения запрос направляется второму клиенту либо напрямую, либо с участием прокси-сервера SIP, либо с участием сервера переадресации. Вызываемый клиент передает вызывающему клиенту ответ на запрос (например, информацию об успешной обработке запроса, отклонении вызова, ошибке при обработке запроса).

Для организации сеанса связи протокол SIP регламентирует установление соединений TCP и UDP через порт 5060.

Чтобы установить сеанс связи между SIP-клиентами, убедитесь, что включена обработка протокола SIP (см. Настройка параметров обработки прикладных протоколов на стр. 151), и создайте фильтр открытой сети, разрешающий входящее и исходящее соединение по протоколам TCP и UDP на порт 5060 (если хотя бы один из SIP-клиентов является открытым узлом).

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.

При организации сеансов связи между SIP клиентами действуют следующие ограничения:

- Между SIP-клиентами нельзя использовать статическую или динамическую трансляцию IP-адресов.
- Если хотя бы один из SIP-клиентов является открытым узлом, то для всех защищенных узлов должна быть настроена видимость по реальным IP-адресам.

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 удаленным 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 Administrator укажите корпоративный DNS-сервер.
   - **2.2** На самих клиентах настройте автоматическую регистрацию на DNS-сервере.
3. На туннелирующем координаторе включите обработку протокола.

Поддерживаемые прикладные протоколы



Внимание! Обработка прикладного протокола в сетевом трафике, проходящем трансляцию сетевых адресов назначения, поддерживается только для прикладного протокола FTP в случае, когда FTP-клиент создает соединение с ViPNet Coordinator KB, который перенаправляет трафик к FTP-серверу по открытому каналу.

В ViPNet Coordinator KB 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 KB.
- H.323 — для передачи мультимедийных данных в IP-сетях (IP-телефония, видео- и аудиоконференции).

  Передача мультимедийных данных с использованием протокола H.323 возможна только между телекоммуникационными серверами, туннелируемыми ViPNet Coordinator KB. При этом на координаторе должен быть разрешен весь туннелируемый трафик, поскольку между телекоммуникационными серверами могут устанавливаться дополнительные соединения.
- 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 KB.

  Примечание. ViPNet Coordinator KB 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.
- 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.

Список поддерживаемых прикладных протоколов изменить нельзя.

Настройка параметров обработки прикладных протоколов

По умолчанию в ViPNet Coordinator KB включена обработка всех поддерживаемых прикладных протоколов для открытого и защищенного трафика (см. Поддерживаемые прикладные протоколы на стр. 150). Для обработки заданы наиболее часто используемые сетевые протоколы и порты.

Примечание. Для непрерывной работы прикладных сервисов при штатной работе ViPNet Coordinator KB не рекомендуется завершать работу службы `algd`.

При завершении работы службы `algd`, если управляющая служба iplircfg запущена, то установленные до этого момента соединения могут продолжать функционировать до 5 минут. Установление новых соединений при выключенной службе `algd` невозможно.

Чтобы просмотреть текущие настройки обработки прикладных протоколов:

```
hostname> alg show
```

В результате отобразится таблица, показанная ниже.

```
+==============+========+================================================+======+
|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 24. Просмотр параметров прикладных протоколов

Таблица содержит следующие столбцы:

- `Service` — обрабатываемый прикладной протокол.
- `Protocol` — сетевой протокол для обработки.
- `Ports` — порты для обработки.
- `On/Off` — состояние обработки: `on` — включена, `off` — выключена.

Чтобы настроить и включить обработку прикладного протокола:

```
hostname# alg module <прикладной протокол> process <сетевой протокол> <порты> on
```

где:

- `<прикладной протокол>` — один из поддерживаемых прикладных протоколов: `dns`, `ftp`, `h323`, `sccp` или `sip`;
- `<сетевой протокол>` — один из сетевых протоколов для обработки выбранного прикладного протокола: `tcp` или `udp`;

Примечание. Для обработки конкретного прикладного протокола должен быть выбран определенный сетевой протокол, см. таблицу ниже.

Таблица 8. Поддержка прикладных протоколов сетевыми протоколами

Прикладной протокол	Сетевой протокол	
	TCP	UDP
FTP		заблокирован
DNS	заблокирован	
H.323		заблокирован
SCCP (Skinny)		заблокирован
SIP		

- порт, диапазон портов либо список портов или диапазонов портов, разделенных запятой.

Внимание! Нулевое значение порта означает выключение обработки прикладного протокола для указанного сетевого протокола.

В результате на узле будет включена обработка прикладного протокола, и вы сможете работать с соответствующими приложениями.

Примечание. Выбранные параметры обработки прикладных протоколов должны соответствовать параметрам, указанным в настройках соответствующих приложений (FTP-клиент, DNS-клиент, SIP-клиент и так далее).

Чтобы изменить ранее выбранный сетевой протокол или порты для обработки какого-либо прикладного протокола, необходимо настроить и включить обработку этого прикладного протокола заново. Например, если кроме стандартных портов для обработки данных, передаваемых по протоколу SIP, используются еще несколько портов для протокола UDP, то их можно добавить с помощью команды:

```
hostname# alg module sip process udp 10000,21-65 on
```

Чтобы выключить обработку прикладного протокола:

- для одного сетевого протокола — задайте для этого сетевого протокола порт, равный нулю, и включите обработку. Например:

```
hostname# alg module sip process tcp 0 on
```

- для всех сетевых протоколов:

```
hostname# alg module <прикладной протокол> process off
```



Внимание! При выключении обработки прикладного протокола работа соответствующих сетевых сервисов на защищенном узле будет невозможна.

11

Настройка сетевых служб

Настройка параметров DHCP-сервера 155

Настройка DHCP-relay 158

Настройка DNS-сервера 160

Настройка параметров NTP-сервера 162

Пример организации работы клиентов с локальным или удаленным DHCP-сервером 165

Организация работы клиентов удаленного офиса с корпоративными DNS- и NTP-серверами 168

Настройка параметров DHCP-сервера

В состав ViPNet Coordinator KB 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 сбросить настройки DHCP-сервера, выполните:

```
hostname# inet dhcp server reset
```

Чтобы запустить DHCP-сервер на одном из сетевых интерфейсов, выполните следующие действия:

1 Укажите сетевой интерфейс, на котором будет работать DHCP-сервер, с помощью команды:

```
hostname# inet dhcp server add interface <имя интерфейса>
```



Примечание. Рабочий сетевой интерфейс для DHCP-сервера должен быть основным, а также должен быть включен и иметь статический IP-адрес. В качестве рабочего интерфейса для DHCP-сервера можно указать Ethernet-, VLAN- или агрегированный сетевой интерфейс. Максимальное число рабочих интерфейсов не может превышать 128.

2 Задайте диапазон IP-адресов для распределения сервером:

```
hostname# inet dhcp server add range <начальный IP-адрес> <конечный IP-адрес>
{interface <имя рабочего интерфейса> | remote subnet <подсеть> mask <маска
подсети>}
```

Диапазон IP-адресов должен принадлежать адресному пространству рабочего интерфейса и не должен пересекаться с уже существующим диапазоном IP-адресов, выделяемых клиентам. В случае указания `remote subnet` проверяется принадлежность диапазона `<начальный IP-адрес> - <конечный IP-адрес>` к указанной подсети `<подсеть>`.



Примечание. DHCP-сервер может выделять клиентам любые диапазоны IP-адресов. Однако в локальной сети, маршрутизируемой в сеть Интернет, рекомендуется выделять IP-адреса только из диапазонов, установленных стандартом о распределении адресов в частных IP-сетях (RFC 1918): `10.0.0.0/8`, `172.16.0.0/12`, `192.168.0.0/16`. При попытке задать диапазон не из приведенного списка, в консоли будет выдано предупреждение, но настройки будут приняты.

3 Чтобы задать IP-адрес шлюза по умолчанию, выполните:

```
hostname# inet dhcp server add router <IP-адрес> {interface <имя интерфейса> |
remote subnet <подсеть> mask <маска подсети> | host <имя узла>}
```

IP-адрес шлюза должен принадлежать адресному пространству рабочего интерфейса и не должен входить в диапазон IP-адресов, выделяемых клиентам. В случае указания `remote` не проводятся проверки принадлежности адреса шлюза `<IP-адрес>` к подсети интерфейса `<имя интерфейса>`, но проверяется принадлежность к указанной подсети `<subnet>`.

4 Чтобы задать IP-адрес DNS-сервера, выполните:

```
hostname# inet dhcp server add dns <IP-адрес> [{interface <имя интерфейса> |
remote subnet <подсеть> mask <маска> | host <имя узла>}]
```

5 Чтобы задать IP-адрес NTP-сервера, выполните:

```
hostname# inet dhcp server add ntp <IP-адрес> [{interface <имя интерфейса> |
remote subnet <подсеть> mask <маска> | host <имя узла>}]
```

6 Вы можете задать другие параметры, передаваемые DHCP-сервером своим клиентам (либо клиентам удаленной подсети при использовании стороннего DHCP-relay), такие как: имя домена, маску подсети, IP-адреса WINS- и TFTP-сервера, широковещательный IP-адрес, время аренды IP-адресов и другие. Полный перечень команд DHCP-сервера см. в документе «Справочник команд и конфигурационных файлов».

7 Вы также можете задать различные параметры DHCP-сервера для клиентов удаленной сети. При этом необходимо сначала добавить эту сеть в качестве удаленной подсети для DHCP-сервера.

Например, чтобы задать адрес DNS-сервера 192.168.15.41 для клиентов другой сети с адресом 192.168.1.0/24, выполните следующие действия:

7.1 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
```



Примечание. Задать можно только тот сетевой интерфейс, который был добавлен в список DHCP ранее командой:

```
hostname# inet dhcp server add interface <имя интерфейса>
```

7.2 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
```

8 Чтобы включить или выключить автоматический запуск DHCP-сервера при загрузке ViPNet Coordinator KB, выполните:

```
hostname# inet dhcp server mode {on | off}
```

По умолчанию автоматический запуск DHCP-сервера выключен.

9 Чтобы запустить DHCP-сервер или завершить его работу, выполните:

```
hostname# inet dhcp server {start | stop}
```



Внимание! После запуска DHCP-сервера запрещено изменение настроек используемого локального сетевого интерфейса, а попытка изменить настройки будет заблокирована.

Перед изменением настроек используемого интерфейса завершите работу DHCP-сервера.

10 Чтобы просмотреть текущие параметры DHCP-сервера, выполните:

```
hostname> inet show dhcp server
```

11 Чтобы просмотреть список клиентов DHCP-сервера, выполните:

```
hostname> inet show dhcp server lease [{last | all | full}]
```

Более подробное описание команд работы с DHCP-сервером см. в документе «Справочник команд и конфигурационных файлов».

Настройка DHCP-relay

ViPNet Coordinator KB 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.

Агент DHCP-relay используется, когда сетевые узлы не могут обращаться напрямую к DHCP-серверу. Например, в разветвленной сети, включающей в себя несколько 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 KB, 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 KB, 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 параметров видимости узлов на стр. 73). Если между ними настроена видимость по виртуальным адресам, ответы от DHCP-сервера будут отправляться на реальный адрес ViPNet Coordinator KB, 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 KB 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 KB, 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 KB:

```
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 KB:

```
hostname# inet dhcp relay reset
```

Настройка DNS-сервера

В состав ViPNet Coordinator KB 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.

  Примечание. Если режим DHCP установлен на нескольких интерфейсах ViPNet Coordinator KB, 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 KB.

  Настройте 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 KB 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 KB:

  ```
  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 параметров NTP-сервера

В состав ViPNet Coordinator KB входит локальный NTP-сервер, который может использоваться для синхронизации времени на самом ViPNet Coordinator KB и на других сетевых узлах (NTP-клиентах).

По умолчанию локальный NTP-сервер настроен таким образом, что при наличии подключения к Интернету он может осуществлять синхронизацию времени с использованием публичных NTP-серверов:

- `ntp1.vniiftri.ru`
- `ntp2.vniiftri.ru`
- `ntp.ix.ru`

Этот список можно дополнить другими NTP-серверами (публичными или корпоративными).



Внимание! Для синхронизации времени используйте доверенные NTP-серверы из защищенной сети.

Такая необходимость может возникнуть в случае отсутствия доступа к Интернету или при наличии более близкого и менее нагруженного NTP-сервера (например, корпоративного). Если адрес дополнительного NTP-сервера известен заранее, то его можно задать в процессе первичной инициализации.



Примечание. Если в качестве дополнительного NTP-сервера используется защищенный узел, видимый по реальному адресу, то для успешной синхронизации времени с таким сервером при загрузке системы рекомендуется в конфигурационном файле `iplir.conf` в секции `[id]` для этого защищенного узла установить параметр `visibility` в значение `real`. Описание секции и параметра см. в документе «Справочник команд и конфигурационных файлов», в разделе «Файл iplir.conf».

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.

Примечание. Если режим DHCP установлен на нескольких сетевых интерфейсах ViPNet Coordinator KB, то результат синхронизации времени на ViPNet Coordinator KB и на NTP-клиентах не определен, так как неизвестно, какой сетевой интерфейс будет включен первым и какой список дополнительных NTP-серверов получит ViPNet Coordinator KB.

Настройте 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.

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.

- 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 KB:

  `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 удается 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 KB используется в качестве NTP-сервера для внутренней сети. Иначе клиенты ViPNet Coordinator KB не будут получать информацию о точном времени при потере соединения ViPNet Coordinator KB 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. Если ViPNet Coordinator KB не удалось подключиться ни к одному из NTP-серверов, локальный NTP-сервер запускается с тем временем, которое было задано при первоначальной настройке. Для таких случаев вы можете добавить в список NTP-серверов выбранный NTP-сервер локальной сети, с которым ViPNet Coordinator KB сможет синхронизировать время в случае невозможности подключения к внешним NTP-серверам.

- 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
```

Пример организации работы клиентов с локальным или удаленным DHCP-сервером

ViPNet Coordinator KB может выполнять функцию DHCP-сервера или функцию агента DHCP-relay. Эти возможности позволяют организовать работу клиентов с DHCP-сервером с помощью ViPNet Coordinator KB двумя способами:

- Использовать ViPNet Coordinator KB в качестве DHCP-сервера для узлов своей сети (см. Организация работы клиентов с локальным DHCP-сервером на стр. 165).
- Использовать ViPNet Coordinator KB в качестве посредника между узлами своей сети и удаленным DHCP-сервером.

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 KB:

1 Остановите DHCP-сервер, если он был запущен:

```
hostname# inet dhcp server stop
```

2 Подключите один из интерфейсов ViPNet Coordinator KB к локальной сети.

3 Установите на этом интерфейсе IP-адрес, принадлежащий адресному пространству локальной сети.

4 Задайте этот интерфейс для работы DHCP-сервера:

```
hostname# inet dhcp server add interface <имя интерфейса>
```

5 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.

6 Задайте срок (время аренды), на который клиентам предоставляются IP-адреса:

```
hostname# inet dhcp server add default-lease-time <время аренды>
```

7 Задайте IP-адрес шлюза по умолчанию, если его необходимо передавать клиентам:

```
hostname# inet dhcp server add router <IP-адрес> interface <имя интерфейса>
```

8 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».

9 Проверьте текущие параметры DHCP-сервера:

```
hostname# inet show dhcp server
```

10 Включите интерфейс, на котором будет работать DHCP-сервер:

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

11 Включите автоматический запуск DHCP-сервера при загрузке ViPNet Coordinator KB:

```
hostname# inet dhcp server mode on
```

12 Запустите DHCP-сервер:

```
hostname# inet dhcp server start
```



Внимание! Вместе с IP-адресом DHCP-сервер всегда 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 KB. Если эти службы не запущены, то клиенты не смогут работать с доменными именами и синхронизировать свое время.

Организация работы клиентов с удаленным DHCP-сервером

Пусть у организации есть несколько офисов, при этом в центральном установлен DHCP-сервер. Клиентам, находящимся в филиале, требуется динамически выделять IP-адреса, но устанавливать в филиале свой DHCP-сервер нецелесообразно.

Взаимодействие клиентов филиала с DHCP-сервером в центральном офисе можно организовать, используя ViPNet Coordinator KB в качестве агента DHCP-relay. Пусть в филиале имеется две подсети с адресными пространствами 10.0.0.0/24 и 10.0.1.0/24. В каждой подсети есть узлы, которые должны получать IP-адреса от DHCP-сервера, находящегося в центральном офисе и имеющего адрес 172.16.1.10.

Внимание! 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 KB, выступающим в качестве агента DHCP-relay, и координатором, туннелирующим DHCP-сервер, настройте видимость по реальным IP-адресам (см. Настройка параметров видимости узлов на стр. 73). Если между этими координаторами настроена видимость по виртуальным адресам, ответы от DHCP-сервера будут отправляться на реальный адрес ViPNet Coordinator KB, выступающего в качестве агента DHCP-relay, и не будут доставлены.

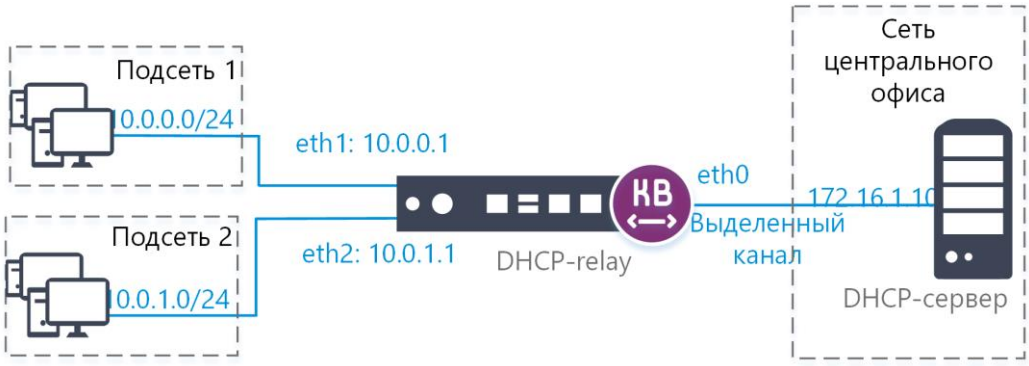


Рисунок 25. Схема использования ViPNet Coordinator KB в качестве агента DHCP-relay

Задайте на DHCP-сервере для каждой подсети диапазон выделяемых адресов и маршрут в сторону ViPNet Coordinator KB. На ViPNet Coordinator KB 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 KB 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 KB:

```
hostname# inet dhcp relay mode on
```

   По умолчанию автоматический запуск DHCP-relay выключен.

7 Запустите DHCP-relay:

```
hostname# inet dhcp relay 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 KB в качестве DNS- и NTP-серверов. Клиенты удаленного офиса будут направлять запросы на разрешение доменных имен и на синхронизацию времени на ViPNet Coordinator KB.
2. На ViPNet Coordinator KB 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 KB:
   ```
   hostname# inet dns mode on
   ```
3. На ViPNet Coordinator KB 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 KB:

```
hostname# inet ntp mode on
```

12

Настройка маршрутизации

Общие сведения о маршрутизации 171

Принципы формирования таблиц маршрутизации в ViPNet Coordinator KB 172

Просмотр общей таблицы маршрутизации 178

Настройка статической маршрутизации 180

Настройка динамической маршрутизации 184

Настройка функции MultiWAN 195

Общие сведения о маршрутизации

ViPNet Coordinator KB поддерживает функции 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 маршрутизаторе и хранящихся в таблице маршрутизации.

Маршруты могут быть:

- Статические — попадают в таблицу маршрутизации в явном виде. 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 разветвленной и неоднородной топологией. При этом на всех маршрутизаторах в сети настройте использование определенного протокола динамической маршрутизации, по которому они будут обмениваться информацией о доступных им сетях, автоматически строить доступные маршруты в каждую сеть и выбирать из них наилучшие.

Кроме автоматического формирования таблиц маршрутизации, динамическая маршрутизация позволяет:

- Автоматически выбирать по определенным критериям наилучший маршрут из нескольких доступных.
- Организовать защиту от сбоев. В случае сбоя какого-либо маршрутизатора автоматически выбирается другой наилучший маршрут и загружается в таблицу маршрутизации.
- Организовать динамическую балансировку нагрузки передаваемого IP-трафика.

ViPNet Coordinator KB может маршрутизировать IP-трафик с использованием следующих видов маршрутов:

- статических, в том числе на основе маршрутов с несколькими шлюзами;
- динамических, формируемых по протоколу DHCP;
- динамических, формируемых протоколом OSPF.

Принципы формирования таблиц маршрутизации в ViPNet Coordinator KB

Для хранения маршрутов в ViPNet Coordinator KB используются таблицы маршрутизации:

- Routing Information Base (далее — RIB) — полная таблица маршрутизации, которая содержит все статические маршруты, созданные администратором ViPNet Coordinator KB, и все динамические маршруты, полученные по протоколам DHCP и OSPF.
- Forwarding Information Base (далее — FIB) — таблица маршрутизации, которая содержит только наилучшие статические и динамические маршруты в каждую сеть, выбранные из RIB. Таблица загружается в ядро системы и используется при маршрутизации IP-трафика.
- Пользовательские таблицы маршрутизации, содержат заданные вручную статические маршруты.

Все новые маршруты, независимо от способа их создания, попадают в RIB. Каждый маршрут имеет следующие характеристики:

- Источник получения маршрута — определяет, каким образом сформирован маршрут: с использованием статического правила или по протоколу динамической маршрутизации.
- Административная дистанция — определяет приоритет маршрута от каждого источника. Используется, когда в таблице маршрутизации задано несколько маршрутов в одну и ту же сеть. Чем меньше административная дистанция, тем более приоритетным считается маршрут.

  Административная дистанция задается для каждого статического маршрута. В случае динамических маршрутов административная дистанция задается не для конкретного маршрута, а для всего протокола сразу. Поэтому все динамические маршруты, добавленные в RIB этим протоколом, имеют одинаковую дистанцию.

  Административная дистанция для протоколов DHCP и OSPF не может быть одинаковой, чтобы маршруты этих протоколов не перемешивались. Также административная дистанция статических правил маршрутизации не может совпадать с дистанцией, заданной для какого-либо протокола динамической маршрутизации.

Примечание. В ViPNet Coordinator KB для каждого типа маршрутов по умолчанию задана следующая административная дистанция:

- для статических маршрутов (Static) — 10;
- для маршрутов DHCP-сервера — 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.

Для маршрутов DHCP-сервера задается общая административная дистанция.

- Метрика маршрута — определяет приоритет внутри списка динамических маршрутов, сформированных и добавленных в RIB определенным протоколом динамической маршрутизации. Чем меньше метрика маршрута, тем выше его приоритет.

  Для протокола DHCP метрики маршрутов 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.

  Если не заданы метрики для маршрутов, полученных по протоколу DHCP, то используется метрика по умолчанию.

  Для динамических маршрутов, формируемых протоколом OSPF, метрики маршрутов формируются самим протоколом, поэтому их задавать не нужно.

- Вес — определяет долю IP-трафика, который будет проходить по маршруту через указанный шлюз. Используется только в статических маршрутах и задается 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 трафика на стр. 181). Поэтому вес задается тогда, когда создается несколько статических маршрутов с одинаковым IP-адресом назначения и разными шлюзами.

Таблица FIB формируется на основе содержания таблицы RIB и включает в себя только наилучшие маршруты в каждую сеть. Если в результате заполнения таблицы RIB в ней оказывается несколько маршрутов в одну сеть, то наилучший маршрут определяется по следующим правилам:

- Если в RIB присутствуют статические маршруты с разной дистанцией, то в FIB загружается маршрут с наименьшей дистанцией. Например, имеется два маршрута в сеть 10.0.5.0 с маской подсети 255.255.255.0. Для первого маршрута задана административная дистанция — 10, для второго — 30. В результате в FIB попадет первый маршрут.
- Если в RIB присутствуют статические маршруты с одинаковой дистанцией, то в FIB загружаются все маршруты.
- Если в RIB присутствуют динамические маршруты, то в FIB загружается маршрут с наибольшим приоритетом (то есть наименьшей метрикой, заданной протоколом динамической маршрутизации). Если приоритеты равны, то в FIB попадают все маршруты.

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

суммируются — объединяются в один маршрут с несколькими шлюзами (multi-hop route). При просмотре такие маршруты отображаются в виде одного маршрута с несколькими шлюзами.

Если маршрут был удален из RIB, то он также удаляется из FIB. При этом по вышеописанным правилам выбирается новый наилучший маршрут в ту же сеть из имеющихся в RIB и немедленно загружается в FIB.

Принципы формирования таблиц маршрутизации в ViPNet Coordinator KB

Для хранения маршрутов в ViPNet Coordinator KB используются таблицы маршрутизации:

- Routing Information Base (далее — RIB) — полная таблица маршрутизации, которая содержит все статические маршруты, созданные администратором ViPNet Coordinator KB, и все динамические маршруты, полученные по протоколам DHCP и OSPF.
- Forwarding Information Base (далее — FIB) — таблица маршрутизации, которая содержит только наилучшие статические и динамические маршруты в каждую сеть, выбранные из RIB. Таблица загружается в ядро системы и используется при маршрутизации IP-трафика.
- Пользовательские таблицы маршрутизации, содержат заданные вручную статические маршруты.

Все новые маршруты, независимо от способа их создания, попадают в RIB. Каждый маршрут имеет следующие характеристики:

- Источник получения маршрута — определяет, каким образом сформирован маршрут: с использованием статического правила или по протоколу динамической маршрутизации.
- Административная дистанция — определяет приоритет маршрута от каждого источника. Используется, когда в таблице маршрутизации задано несколько маршрутов в одну и ту же сеть. Чем меньше административная дистанция, тем более приоритетным считается маршрут.

  Административная дистанция задается для каждого статического маршрута. В случае динамических маршрутов административная дистанция задается не для конкретного маршрута, а для всего протокола сразу. Поэтому все динамические маршруты, добавленные в RIB этим протоколом, имеют одинаковую дистанцию.

  Административная дистанция для протоколов DHCP и OSPF не может быть одинаковой, чтобы маршруты этих протоколов не перемешивались. Также административная дистанция статических правил маршрутизации не может совпадать с дистанцией, заданной для какого-либо протокола динамической маршрутизации.

Примечание. В ViPNet Coordinator KB для каждого типа маршрутов по умолчанию задана следующая административная дистанция:

- для статических маршрутов (Static) — 10;
- для маршрутов DHCP-сервера — 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.

Для маршрутов DHCP-сервера задается общая административная дистанция.

- Метрика маршрута — определяет приоритет внутри списка динамических маршрутов, сформированных и добавленных в RIB определенным протоколом динамической маршрутизации. Чем меньше метрика маршрута, тем выше его приоритет.

  Для протокола DHCP метрики маршрутов 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.

  Если не заданы метрики для маршрутов, полученных по протоколу DHCP, то используется метрика по умолчанию.

  Для динамических маршрутов, формируемых протоколом OSPF, метрики маршрутов формируются самим протоколом, поэтому их задавать не нужно.

- Вес — определяет долю IP-трафика, который будет проходить по маршруту через указанный шлюз. Используется только в статических маршрутах и задается 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 трафика на стр. 181). Поэтому вес задается тогда, когда создается несколько статических маршрутов с одинаковым IP-адресом назначения и разными шлюзами.

Таблица FIB формируется на основе содержания таблицы RIB и включает в себя только наилучшие маршруты в каждую сеть. Если в результате заполнения таблицы RIB в ней оказывается несколько маршрутов в одну сеть, то наилучший маршрут определяется по следующим правилам:

- Если в RIB присутствуют статические маршруты с разной дистанцией, то в FIB загружается маршрут с наименьшей дистанцией. Например, имеется два маршрута в сеть 10.0.5.0 с маской подсети 255.255.255.0. Для первого маршрута задана административная дистанция — 10, для второго — 30. В результате в FIB попадет первый маршрут.
- Если в RIB присутствуют статические маршруты с одинаковой дистанцией, то в FIB загружаются все маршруты.
- Если в RIB присутствуют динамические маршруты, то в FIB загружается маршрут с наибольшим приоритетом (то есть наименьшей метрикой, заданной протоколом динамической маршрутизации). Если приоритеты равны, то в FIB попадают все маршруты.

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

суммируются — объединяются в один маршрут с несколькими шлюзами (multi-hop route). При просмотре такие маршруты отображаются в виде одного маршрута с несколькими шлюзами.

Если маршрут был удален из RIB, то он также удаляется из FIB. При этом по вышеописанным правилам выбирается новый наилучший маршрут в ту же сеть из имеющихся в RIB и немедленно загружается в FIB.

Просмотр общей таблицы маршрутизации

В процессе или после настройки маршрутизации вы можете просмотреть список всех существующих маршрутов — содержимое RIB (см. Принципы формирования таблиц маршрутизации в ViPNet Coordinator KB на стр. 172). Для этого выполните:

```
hostname> inet show routing
```

В результате будет выдана информация о маршрутах в следующем виде:

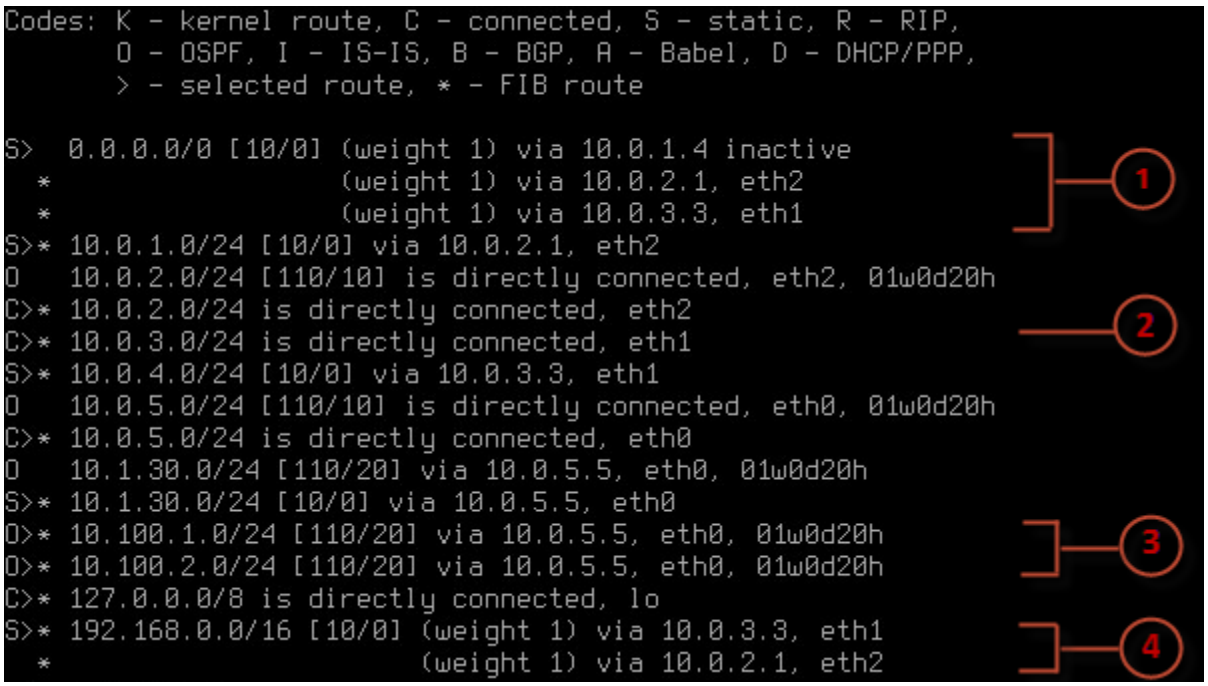


Рисунок 26. Пример таблицы маршрутизации

Цифрами на рисунке обозначены:

1 Статический маршрут по умолчанию. Выбран как наилучший. 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 (см. Создание статических маршрутов на стр. 180).

---

2 Маршрут в подсеть `10.0.3.0` с маской `255.255.255.0`, к которой подключен один из сетевых интерфейсов ViPNet Coordinator KB. Выбран как наилучший.

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 наилучшие.

4 Статический маршрут в подсеть `192.168.0.0` с маской `255.255.0.0`. Выбран как наилучший. 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 9. Атрибуты маршрутов

Атрибут	Расшифровка в командном интерпретаторе	Пояснение
K	kernel route	Маршрут, заданный вручную в ядре системы
C	connected	Маршрут в подсеть, к которой подключен один из сетевых интерфейсов ViPNet Coordinator KB
S	static	Статический маршрут
R	RIP	Маршрут протокола динамической маршрутизации RIP
O	OSPF	Маршрут протокола динамической маршрутизации OSPF
I	IS-IS	Маршрут протокола динамической маршрутизации IS-IS
A	Babel	Маршрут протокола динамической маршрутизации Babel
D	DHCP	Маршрут, предоставленный DHCP-сервером
>	selected route	Наилучший маршрут. Отображается в том случае, если в таблице есть несколько маршрутов в одну и ту же сеть с разной административной дистанцией
*	FIB route	Маршрут, загруженный в таблицу FIB и используемый для 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 KB не используются.

Настройка статической маршрутизации

Если ViPNet Coordinator KB функционирует в сети, конфигурация которой никогда не меняется или в которой используется минимальное количество путей для доставки IP-пакетов, то вы можете настроить маршрутизацию на ViPNet Coordinator KB вручную с помощью статических маршрутов. Кроме этого, настройка маршрутизации вручную также может потребоваться, когда нет возможности использовать в сети маршрутизаторы, работающие по протоколам динамической маршрутизации, или в следующих случаях:

- Чтобы направлять часть IP-трафика всегда по конкретным маршрутам (например, есть требование направлять IP-трафик по самому надежному каналу связи), или направлять интернет-трафик через конкретный шлюз.
- Если нужен статический маршрут по умолчанию на случай, когда работа по протоколам динамической маршрутизации невозможна.
- Чтобы распределять передачу IP-трафика по нескольким маршрутам.

Создание статических маршрутов

Вы можете создать маршрут по умолчанию или ряд статических маршрутов следующим образом:

- Чтобы добавить маршрут по умолчанию:

```
hostname# inet route add default next-hop <IP-адрес шлюза> [distance <1-255>
[weight <1-255>]]
```

- Чтобы добавить статический маршрут:

```
hostname# inet route add <IP-адрес назначения> next-hop <IP-адрес шлюза> [netmask
<маска подсети>] [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).

В обеих командах вы можете задать необязательные параметры `distance` — административную дистанцию маршрута и `weight` — вес шлюза.

Административная дистанция задается, чтобы назначить приоритет маршруту. Если вы не укажете значение административной дистанции, то оно будет задано автоматически (`distance=70`). Если указанное значение `distance` совпадает со значением административной дистанции, заданной

протоколом динамической маршрутизации (см. Настройка административной дистанции для маршрутов DHCP-протокола на стр. 185), то 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 создается несколько маршрутов в одну сеть с разными шлюзами, между которыми требуется производить балансировку нагрузки. Подробнее см. раздел Настройка балансировки IP-трафика (см. Настройка балансировки трафика на стр. 181).

В результате маршрут будет добавлен в RIB (см. Принципы формирования таблиц маршрутизации в ViPNet Coordinator KB на стр. 172). Если маршрут будет определен как наилучший, то он также будет загружен в FIB, после чего начнет использоваться при маршрутизации IP-трафика.

При необходимости вы можете удалить созданные маршруты.

Удаление статического маршрута

При необходимости вы можете удалить маршрут по умолчанию или статический маршрут:

- Чтобы удалить маршрут по умолчанию:

  ```
  hostname# inet route delete default next-hop <IP-адрес шлюза>
  ```

- Чтобы удалить статический маршрут:

  ```
  hostname# inet route delete <IP-адрес назначения> [netmask <маска подсети>] next-hop <IP-адрес шлюза>
  ```

Если при вводе команды вы не указали маску подсети или IP-адрес шлюза, и в процессе проверки будет установлено, что в таблице RIB (см. Принципы формирования таблиц маршрутизации в ViPNet Coordinator KB на стр. 172) есть несколько маршрутов в указанную сеть, то будет выдан список всех маршрутов. Для удаления всех этих маршрутов введите `y` и нажмите **Enter**. Аналогичная ситуация возникнет при удалении маршрута с несколькими шлюзами.

В результате маршрут или все маршруты с указанными параметрами будут удалены из RIB. Если маршруты присутствуют в таблице FIB, то из нее они также будут удалены.

Примечание. Если требуется удалить все статические маршруты, включая все маршруты по умолчанию, воспользуйтесь командой:

```
hostname# inet route clear
```

Настройка балансировки трафика

Чтобы ускорить процесс отправки IP-пакетов в сеть назначения, создайте несколько статических маршрутов в данную сеть через разные шлюзы и настроить балансировку IP-трафика между этими маршрутами. Это позволит для разных TCP-соединений отправлять IP-пакеты по разным маршрутам, что повысит скорость передачи IP-пакетов в сеть назначения. Балансировка трафика

настраивается с помощью одного из параметров маршрутов — веса (`weight`). При этом все маршруты, которые будут участвовать в балансировке трафика, должны иметь одинаковую административную дистанцию, то есть иметь одинаковый приоритет.



Примечание. Маршруты в одну сеть назначения и с одинаковой дистанцией объединяются в один маршрут с несколькими шлюзами (multi-hop route). При просмотре общей таблицы маршрутизации они отображаются как один маршрут с несколькими шлюзами (см. Принципы формирования таблиц маршрутизации в ViPNet Coordinator KB на стр. 172).

Для настройки балансировки трафика:

1. Создайте маршруты (см. Создание статических маршрутов на стр. 180) с одинаковым IP-адресом назначения и с разными шлюзами.
2. Задайте каждому маршруту одинаковую административную дистанцию.
3. Задайте в каждом маршруте вес шлюзу. Трафик будет распределяться между шлюзами в соответствии с соотношением их весов. Например:
   - если вы создаете 2 маршрута и в каждом маршруте вес шлюза равен 1, то трафик будет разделен между этими маршрутами поровну, то есть 50% трафика будет передаваться по одному маршруту, 50% — по второму;
   - если вы создаете 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 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 next-hop 10.0.1.1 netmask 255.255.255.0 distance 20
weight 1

hostname# inet route add 10.0.5.0 next-hop 10.0.4.3 netmask 255.255.255.0 distance 20
weight 1
```



Примечание. Если при создании маршрута вес не указан, то шлюзу автоматически будет назначен вес равный 1. Вес не может быть равен 0.

Балансировка IP-трафика будет осуществляться, только если созданные маршруты признаны наилучшими и присутствуют в таблице FIB (см. Принципы формирования таблиц маршрутизации в ViPNet Coordinator KB на стр. 172).

Просмотр статических маршрутов

Если вы хотите просмотреть список всех статических маршрутов, которые были созданы и присутствуют в RIB (см. Принципы формирования таблиц маршрутизации в ViPNet Coordinator KB на стр. 172):

```
hostname> inet show routing static
```

В результате выполнения команды появится список маршрутов:

```
Destination      Netmask          Next hop         Distance Weight
-----------      -------          --------         -------- ------
0.0.0.0          0.0.0.0          10.0.1.4         10       1
0.0.0.0          0.0.0.0          10.0.2.1         10       1
0.0.0.0          0.0.0.0          10.0.3.3         10       1
10.0.1.0         255.255.255.0    10.0.2.1         10       1
10.0.4.0         255.255.255.0    10.0.3.3         10       1
10.1.30.0        255.255.255.0    10.0.5.5         10       1
192.168.0.0      255.255.0.0      10.0.2.1         10       1
192.168.0.0      255.255.0.0      10.0.3.3         10       1
```

Рисунок 27. Просмотр статических маршрутов

В списке для каждого маршрута указаны: IP-адрес назначения, маска сети, шлюз, административная дистанция маршрута и вес шлюза.

Настройка динамической маршрутизации

Если в вашей сети поддерживается работа по протоколу DHCP или OSPF, то вы можете настроить на ViPNet Coordinator KB динамическую маршрутизацию. Данные настройки позволят автоматически создавать таблицы маршрутизации на основе маршрутов, формируемых по данным протоколам, и распространять их между другими маршрутизаторами в рамках одной сети.



Внимание! Настройку динамической маршрутизации должен выполнять опытный пользователь, понимающий принципы работы протоколов динамической маршрутизации, в том числе протокола OSPF. Приведенная в руководстве информация носит справочный характер.

Если в вашей сети используются другие протоколы динамической маршрутизации (например BGP), то вы можете защитить взаимодействие между маршрутизаторами с помощью технологии туннелирования.

Настройка параметров динамических маршрутов от протокола DHCP

Если на каком-либо сетевом интерфейсе ViPNet Coordinator KB включен режим DHCP и настроено автоматическое получение маршрутов от DHCP-сервера (см. Настройка сетевых интерфейсов на стр. 43), то в процессе маршрутизации будут использоваться эти маршруты. В этом случае выполните следующие дополнительные настройки:

1. Настройте административную дистанцию для протокола DHCP (см. Настройка административной дистанции для маршрутов DHCP-протокола на стр. 185). Это позволит определить приоритет маршрутов DHCP-сервера среди всех маршрутов, имеющихся в таблице RIB (статических маршрутов, маршрутов протокола OSPF, если такие есть).
2. Если режим DHCP включен на нескольких сетевых интерфейсах, настройте метрики маршрутов протокола DHCP на каждом сетевом интерфейсе, на котором включен этот режим (см. Настройка метрики для маршрутов протокола DHCP на стр. 185). Это позволит определить приоритет среди маршрутов DHCP-сервера в одну и ту же сеть, полученных с разных сетевых интерфейсов.

Настройка административной дистанции для маршрутов DHCP-протокола

Если на ViPNet Coordinator KB настроено взаимодействие с DHCP-сервером, от которого поступают динамические маршруты для маршрутизации IP-трафика, а также настроена статическая маршрутизация (см. Настройка статической маршрутизации на стр. 180) или динамическая маршрутизация по протоколу OSPF, 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 KB 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 (на стр. 186).

Настройка метрики для маршрутов протокола DHCP

Если на ViPNet Coordinator KB 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 наилучшим, чтобы добавить его в FIB (см. Принципы формирования таблиц маршрутизации в ViPNet Coordinator KB на стр. 172) и использовать при маршрутизации. Параметром, позволяющим выбрать наилучший маршрут, выступает метрика маршрута. Ее 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 KB вы можете задать или удалить метрику для маршрутов DHCP-сервера на конкретном сетевом интерфейсе (далее — специфичная метрика). Если специфичная метрика не задана, то вместо нее используется метрика по умолчанию.

- Чтобы добавить специфичную метрику на сетевом интерфейсе, выполните:

  ```
  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 на стр. 43).
  - VLAN (см. Поддержка технологии VLAN на стр. 46).
  - Агрегированный интерфейс (см. Использование агрегированных сетевых интерфейсов на стр. 50).

  При этом должны выполняться следующие условия:

  - на нескольких сетевых интерфейсах включен режим DHCP;
  - на этих интерфейсах настроен параметр автоматического получения маршрутов от DHCP-сервера.

  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 специфичную метрику, заданную на сетевом интерфейсе, выполните:

  ```
  hostname# inet ifconfig <имя интерфейса> dhcp route-metric none
  ```

  После удаления специфичной метрики на этом интерфейсе будет использоваться метрика по умолчанию.

При необходимости вы можете посмотреть, на каких сетевых интерфейсах включен режим DHCP и какие метрики заданы. Подробнее см. раздел «Просмотр настроек DHCP в режиме клиента» (см. Просмотр настроек DHCP в режиме клиента на стр. 186).

Изменение метрики по умолчанию для маршрутов от протокола DHCP

Если для маршрутов, полученных от DHCP-сервера, не задавались специфичные метрики, то для определения приоритета маршрутов этого протокола будет использоваться метрика по умолчанию. Первоначально метрика по умолчанию равна 70. При необходимости вы можете изменить это значение. Для этого выполните:

```
hostname# inet dhcp client route-default-metric <1-255>
```

Просмотр настроек DHCP в режиме клиента

Вы можете просмотреть следующие настройки DHCP в режиме клиента:

- административная дистанция, которая задана для маршрутов, поступающих от DHCP-сервера (см. Настройка административной дистанции для маршрутов DHCP-протокола на стр. 185);
- метрика по умолчанию (см. Настройка метрики для маршрутов протокола DHCP на стр. 185);
- 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, выполните:

```
hostname> inet show dhcp client
```

В результате выполнения команды указанные настройки будут отражены в следующем виде:

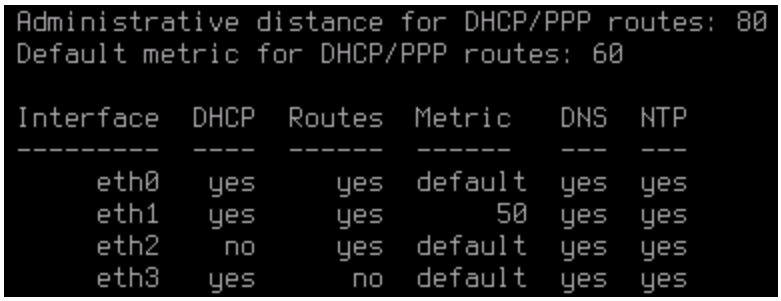


Рисунок 28. Просмотр настроек DHCP-режима

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 KB имеет некоторые ограничения. Из спецификации RFC 2328 не поддерживаются:

- изменение пропускной способности reference-bandwidth — cost, auto-cost;
- аутентификация: простая на пароле (simple password authentication) и криптографическая (cryptographic authentication);
- настройка метрик маршрутов;
- перераспределение (redistribute) маршрутов, кроме заданных статически и полученных по DHCP;
- настройка параметров взаимодействия соседних маршрутизаторов — HelloInterval и DeadInterval;
- настройка для фильтрации LSA других типов областей (area type) кроме backbone: standard, stub, totally stubby, not-so-stubby;
- настройка типа сети (network types): p2p, Broadcast, Non-Broadcast, Non-Broadcast Multi-Access;
- 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 передается 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, соединенных 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 определенную 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 передается через маршрутизатор, который располагается на границе этих областей. Если автономная система взаимодействует с другой автономной системой, то информация передается через маршрутизатор, который располагается на границе систем. Такой подход позволяет уменьшить объем 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) — берет 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).

Настройка параметров динамической маршрутизации по протоколу OSPF



Совет. Настройка динамической маршрутизации по протоколу OSPF имеет смысл в том случае, если в сети, в которой установлен ViPNet Coordinator KB, одновременно выполняются следующие условия:

- используются другие OSPF-маршрутизаторы и они напрямую связаны с ViPNet Coordinator KB;
- используется протокол OSPF версии 2;
- политика безопасности организации не запрещает передачу IP-трафика по протоколу OSPF между сетевыми узлами.

Для настройки маршрутизации по протоколу OSPF необходимо:

- включить использование протокола OSPF;
- указать сети, к которым подключен ViPNet Coordinator KB и которые будут участвовать в маршрутизации по протоколу OSPF, и области маршрутизации;
- создать необходимые для работы маршрутизации разрешающие сетевые фильтры.

При настройке динамической маршрутизации по протоколу OSPF следует учитывать следующие рекомендации и ограничения:

- В случае сбоя на одном из маршрутов для переключения на альтернативный маршрут может потребоваться около 10 секунд после обнаружения потери связи с удаленным координатором.
- Если ваш ViPNet Coordinator KB непосредственно связан с другими координаторами ViPNet, рекомендуется в конфигурационном 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.

При необходимости вы также можете настроить перераспределение маршрутов по протоколу OSPF.

Настройка протокола OSPF

Для настройки протокола OSPF:

1 Включите использование протокола:

```
hostname# inet ospf mode on
```

2 Укажите сеть, в которой осуществляется обмен информацией по протоколу OSPF:

```
hostname# inet ospf network add <IP-адрес назначения> netmask <маска сети> area
<0-4294967295>
```

К данной сети должен быть подключен сетевой интерфейс ViPNet Coordinator KB.

С помощью параметра `area` задайте область маршрутизации, в которую входит ViPNet Coordinator KB (см. Общие сведения о протоколе OSPF на стр. 187).



Примечание. Вы можете указать не более 128 сетей, в которых осуществляется обмен информацией по протоколу OSPF.

---

3 Создайте на ViPNet Coordinator KB следующие сетевые фильтры открытой сети:

- Фильтры, разрешающие входящий однонаправленный (unicast) и многоадресный (multicast) IP-трафик по протоколу OSPF от всех соседних OSPF-маршрутизаторов, с которыми взаимодействует ViPNet Coordinator KB:

  ```
  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
  ```

  Фильтры должны быть созданы для каждого соседнего OSPF-маршрутизатора в области, в которой находится ViPNet Coordinator KB.

  Примечание. В фильтрах вместо service @OSPF вы можете указать proto 89.

- Фильтр, разрешающий исходящий трафик ViPNet Coordinator KB по протоколу OSPF на любой IP-адрес назначения:

  ```
  hostname# firewall local add 3 rule "Allow OSPF outcoming" src @local dst @any
  service @OSPF pass
  ```

Подробнее о создании сетевых фильтров см. в разделе Создание сетевого фильтра.

4 Если между координаторами сети ViPNet установлены связи на уровне пользователей, разрешите трафик протокола OSPF в защищенной сети:

```
hostname# firewall vpn add 1 rule "Allow OSPF outcoming" src @local dst @any
service @OSPF pass
```

```
hostname# firewall vpn add 2 rule "Allow OSPF incoming" src @any dst @local
service @OSPF pass
```

Внимание! Если вы не настроите указанные фильтры, то ViPNet Coordinator KB не сможет работать по протоколу OSPF (см. Общие сведения о протоколе OSPF на стр. 187).

Впоследствии вы можете отказаться от использования протокола OSPF для маршрутизации:

```
hostname# inet ospf mode off
```

Если в подсети прекращено использование протокола OSPF, удалите ее:

```
hostname# inet ospf network delete <IP-адрес назначения> netmask <маска подсети> area
<0-4294967295>
```

Настройка перераспределения маршрутов

В одной разветвленной сети может быть образовано нескольких автономных систем по причине использования в разных подсетях разных протоколов маршрутизации. Например, в одной подсети может использоваться статическая маршрутизация, в другой — динамическая маршрутизация по протоколу OSPF. В результате этого образуются две автономные системы. Чтобы системы могли взаимодействовать друг с другом, на маршрутизаторе, который установлен на границе этих систем, требуется настроить перераспределение маршрутов. В результате произойдет обмен маршрутной информацией, получаемой из этих систем, и они будут иметь связь друг с другом.

На рисунке ниже показана схема взаимодействия нескольких автономных систем с использованием координаторов ViPNet для распространения маршрутов.

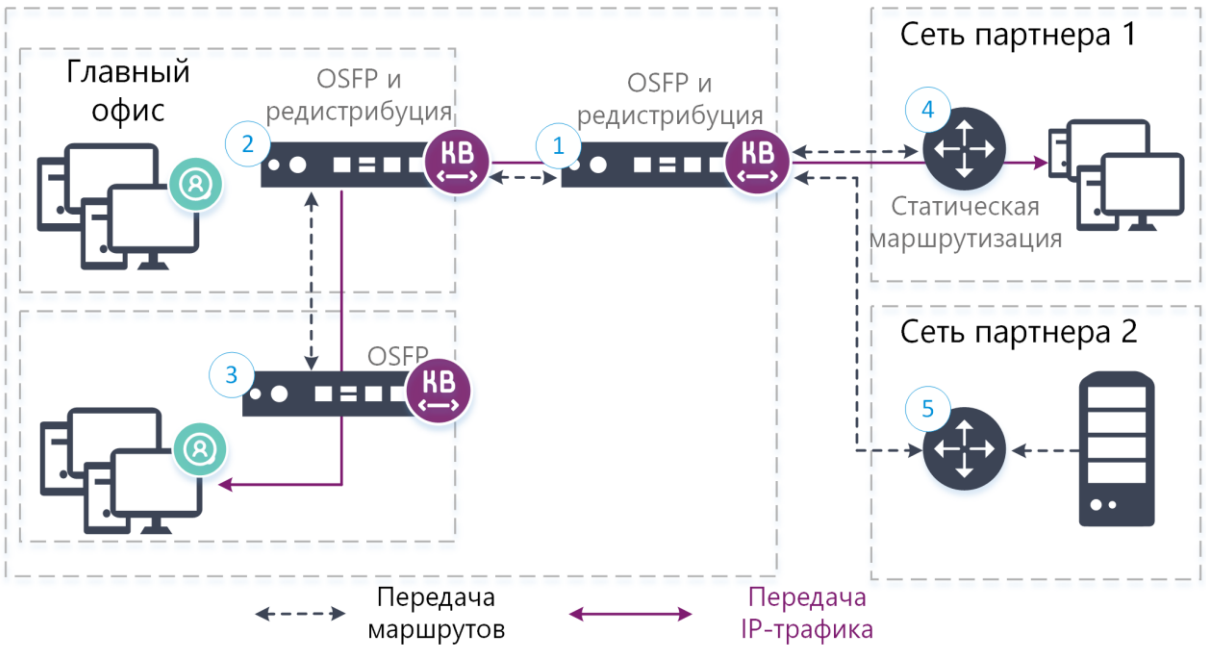


Рисунок 29. Пример топологии сети с настроенным перераспределением маршрутов

В приведенном примере рассматривается 4 взаимодействующих подсети, 2 из которых входят в одну автономную систему, поскольку они работают по одному протоколу OSPF, остальные — в две другие автономные системы, поскольку они работают по протоколам Static и DHCP соответственно. При этом маршрутизатор 1 является граничным маршрутизатором (см. Общие сведения о

протоколе OSPF на стр. 187), поскольку он расположен на границе трех автономных систем. Чтобы координаторы 2 и 3 могли получить информацию о маршрутах, используемых на маршрутизаторах 4 и 5, и, соответственно, чтобы узлы сети филиала и сетей партнеров могли обмениваться IP-трафиком друг с другом, требуется настроить перераспределение маршрутов на маршрутизаторе 1. После настройки маршруты, полученные из сетей партнеров (от маршрутизаторов 4 и 5) на маршрутизаторе 1, будут передаваться на маршрутизатор 2, а с маршрутизатора 2 — на маршрутизатор 3. В результате маршрутизатор 3 будет знать статические маршруты, прописанные в сети партнера 1, и маршруты, выдаваемые DHCP-сервером, в сети партнера 2.

Чтобы включить перераспределение маршрутов на ViPNet Coordinator KB, выполните:

```
hostname# inet ospf redistribute add {static | dhcp}
```

указав один из параметров:

- `static` — включение перераспределения статических маршрутов;
- `dhcp` — включение перераспределения маршрутов от DHCP-сервера.

Если перераспределение указанного типа маршрутов было включено ранее, появится соответствующее сообщение.

Чтобы выключить перераспределение маршрутов, выполните:

```
hostname# inet ospf redistribute delete {static | dhcp}
```

В параметрах протокола OSPF указывается, настроено перераспределение маршрутов или нет. Подробнее см. раздел Просмотр настроек протокола OSPF (на стр. 192).

Просмотр настроек протокола OSPF

Вы можете просмотреть следующие параметры настройки протокола OSPF:

- статус использования протокола OSPF: включен или выключен;
- список сетей, в которых ViPNet Coordinator KB осуществляет маршрутизацию по протоколу OSPF (IP-адреса, маски подсетей и области маршрутизации);
- статус перераспределения маршрутов: доступно или недоступно.

Чтобы посмотреть указанные настройки, выполните:

```
hostname> inet show ospf configuration
```

В результате выполнения команды указанные настройки будут отображены в следующем виде:

```
OSPF protocol is enabled
OSPF networks defined:
Destination      Netmask          OSPF Area
-----------      -------          ---------
10.0.2.0         255.255.255.0    1
10.0.5.0         255.255.255.0    0
Redistribution of DHCP routes is enabled.
```

Рисунок 30. Просмотр настроек протокола OSPF

Просмотр информации базы данных состояний каналов связи по протоколу OSPF

Кроме настроек протокола OSPF вы также можете просмотреть информацию о состоянии каналов связи между всеми маршрутизаторами, работающими по протоколу OSPF. Данная информация содержится в базе данных (link state database) (см. Общие сведения о протоколе OSPF на стр. 187) и синхронизируется на всех OSPF-маршрутизаторах. Именно на ее основе вычисляются динамические маршруты протокола OSPF. Для просмотра информации базы данных состояний каналов связи, выполните:

```
hostname> inet show ospf database
```

В результате выполнения команды отобразится содержимое базы данных в следующем виде:

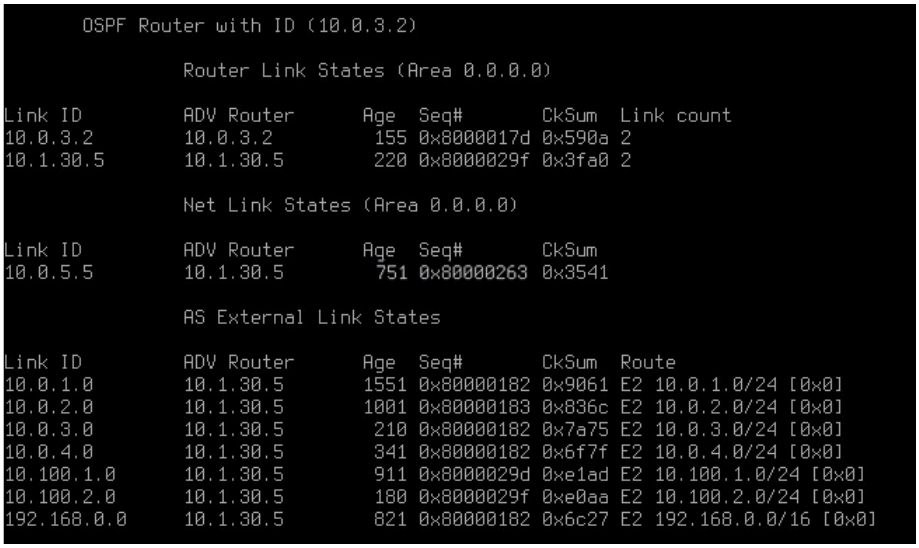


Рисунок 31. Просмотр базы данных состояний связи

Просмотр информации о соседних маршрутизаторах

При составлении маршрутов по протоколу OSPF участвуют все OSPF-маршрутизаторы, находящиеся в области маршрутизации ViPNet Coordinator KB. При необходимости вы можете просмотреть сведения о маршрутизаторах-соседях. Для этого выполните:

```
hostname> inet show ospf neighbour
```

В результате выполнения команды отобразится информация обо всех соседних OSPF-маршрутизаторах:

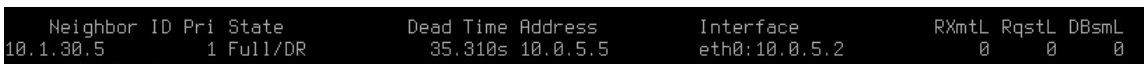


Рисунок 32. Просмотр информации о соседних OSPF-маршрутизаторах

По каждому такому 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 на стр. 187) (в случае, приведенном на рисунке выше, 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

Если вы хотите просмотреть список динамических маршрутов, сформированных по протоколу OSPF и присутствующих в RIB, выполните:

```
hostname> inet show routing ospf
```

В результате выполнения команды появится списки маршрутов в трех разделах:

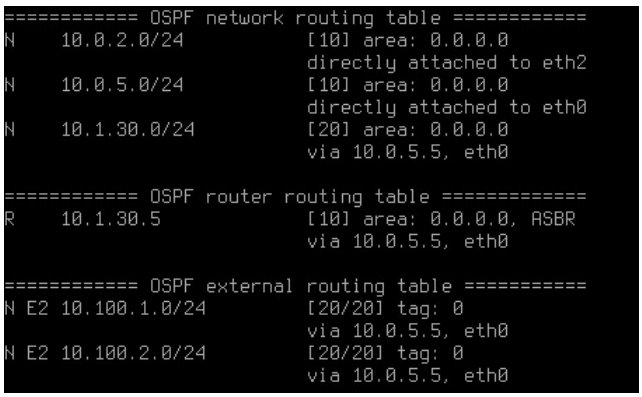


Рисунок 33. Просмотр маршрутов протокола OSPF

В каждом из разделов отображаются следующие маршруты:

- `OSPF network routing table` — маршруты, которые были сформированы по протоколу OSPF в той области маршрутизации, к которой подключен ViPNet Coordinator KB. Если ViPNet Coordinator KB является межобластным пограничным маршрутизатором (ABR) (см. Общие сведения о протоколе OSPF на стр. 187), то в этом разделе будут присутствовать маршруты из всех областей, которые он объединяет.
- `OSPF router routing table` — маршруты до маршрутизаторов-соседей, которые находятся в области маршрутизации ViPNet Coordinator KB и выполняют перераспределение маршрутов (см. Настройка перераспределения маршрутов на стр. 191).
- `OSPF external routing table` — маршруты в другие автономные системы, поступившие при перераспределении от соседних OSPF-маршрутизаторов.

Настройка функции MultiWAN

Если для доступа ViPNet Coordinator KB 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 KB реализована технология MultiWAN, позволяющая настроить поведение ViPNet Coordinator KB в случае недоступности одного из интернет-шлюзов. Эта технология представлена в ViPNet Coordinator KB следующими функциями:

- Пользовательские таблицы маршрутизации. 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.

Внимание! При настройке функции MultiWAN:

- 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 KB с узлом, с которого вы настраиваете ViPNet Coordinator KB.
- Не используйте одинаковые IP-адреса для проверки шлюзов DGD и для параметров `testip` при работе в кластере горячего резервирования — 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.

Ниже приведены основные сценарии использования функции MultiWAN.

Резервирование каналов доступа в Интернет

В этом примере сеть ViPNet имеет выход в Интернет через двух провайдеров: основного (шлюз GW-1) и запасного (шлюз GW-2). Чтобы в случае потери соединения со шлюзом одного провайдера переключаться на другой шлюз, необходимо настроить резервирование каналов доступа в Интернет с помощью MultiWAN.

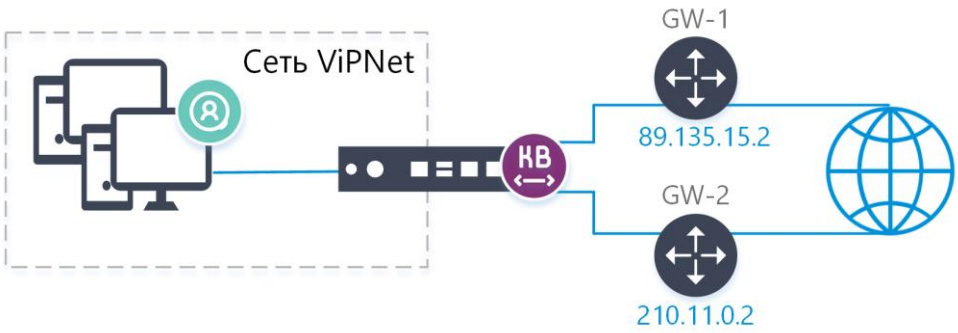


Рисунок 34. Пример схемы резервирования каналов доступа в Интернет

Для настройки резервирования выполните следующие действия на ViPNet Coordinator KB:

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:

```
inet dgd mode off

inet dgd mode on
```

Так вы обеспечите маршрутизацию трафика между двумя сетями ViPNet даже в случае недоступности одного из провайдеров.

Разделение каналов доступа в интернет для распределения (балансировки) нагрузки на сеть

В этом примере рассматриваются две сети ViPNet (сеть 1 и сеть 2), в каждой из которых работает ViPNet Coordinator KB (-1 в сети 1 и -2 в сети 2), за которым стоят защищенные, туннелируемые и открытые узлы. Обе сети имеют выход в интернет через шлюзы двух провайдеров (GW-1 и GW-2). Необходимо обеспечить доступ к ресурсам обеих сетей через разные каналы связи (шлюзы), чтобы распределить нагрузку на сеть.

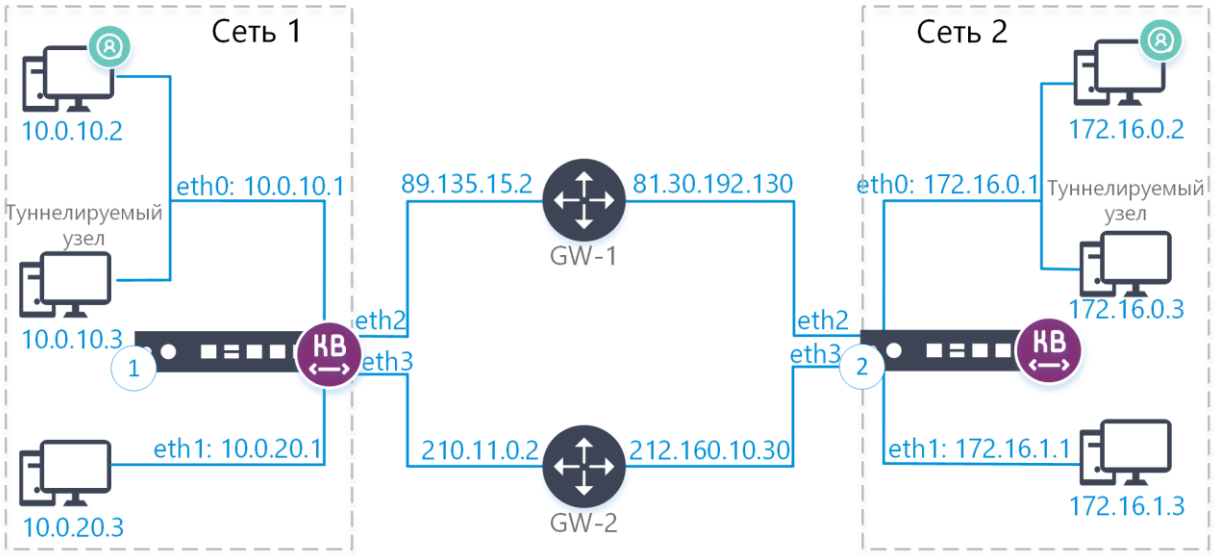


Рисунок 35. Схема маршрутизации между двумя ViPNet Coordinator KB

Допустим, что для этой схемы выполнены предварительные условия:

- На маршрутизаторах GW-1 и GW-2 настроена обычная статическая маршрутизация:
  - маршрут в сеть 1 через -1;
  - маршрут в сеть 2 через -2.
- На клиентских узлах обеих сетей настроены маршруты по умолчанию:
  - на защищенных и туннелируемых узлах сети 1 — через адрес интерфейса `eth0` -1;
  - на открытых узлах сети 1 — через адрес интерфейса `eth1` -1;
  - на защищенных и туннелируемых узлах сети 2 — через адрес интерфейса `eth0` -2;
  - на открытых узлах сети 2 — через адрес интерфейса `eth1` -2.
- На обоих ViPNet Coordinator KB для клиентских узлов необходимо настроить трансляцию адресов в IP-адрес исходящих интерфейсов (`eth2`, `eth3`).

Чтобы настроить маршрутизацию по различным каналам для приведенной схемы, на обоих ViPNet Coordinator KB :

1 Создайте таблицы маршрутизации IP-пакетов через шлюзы GW-1 и GW-2:
   - на -1:

     ```
     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
     ```
   - на -2:

     ```
     hostname# inet route add default next-hop 81.30.192.130 table 1024 name GW-1
     hostname# inet route add default next-hop 212.160.10.30 table 1025 name GW-2
     ```

2 Настройте политики маршрутизации для подсетей, в которых находятся защищенные и туннелируемые узлы:
   - на -1:

```
hostname# inet policy rule add isp-both 1025 match address from 10.0.10.0/24
hostname# inet policy rule add isp-both 1025 table name GW-1
```

- на -2:

```
hostname# inet policy rule add isp-both 1025 match address from 172.16.0.0/24
hostname# inet policy rule add isp-both 1025 table name GW-1
```

3 Настройте политики маршрутизации для подсетей, в которых находятся открытые узлы:

- на -1:

```
hostname# inet policy rule add isp-both 1026 match address from 10.0.20.0/24
hostname# inet policy rule add isp_both 1026 table name GW-2
```

- на -2:

```
hostname# inet policy rule add isp-both 1026 match address from 172.16.1.0/24
hostname# inet policy rule add isp-both 1026 table name GW-2
```

4 Настройте политики маршрутизации для всего остального трафика:

- на -1:

```
hostname# inet policy rule add isp-both 1027 table name GW-1
```

- на -2:

```
hostname# inet policy rule add isp-both 1027 table name GW-1
```

5 Добавьте правила маршрутизации по таблице по умолчанию Main (default) для работы подключенных напрямую к ViPNet Coordinator KB сетей:

```
hostname# inet policy rule add isp-both 1024 table default
```

6 Активируйте созданные политики маршрутизации:

```
hostname# inet policy active isp-both
```

В результате трафик с защищенных и туннелируемых узлов обеих сетей ViPNet будет маршрутизироваться через шлюз провайдера GW-1, а трафик с открытых узлов — через шлюз провайдера GW-2, что уменьшит нагрузку на сетевые каналы.

Резервирование каналов доступа в интернет с балансировкой трафика

В предыдущем сценарии была обеспечена балансировка нагрузки сети, но в случае неработоспособности одного из шлюзов часть пользователей сети потеряет доступ к интернету и другой сети ViPNet. Чтобы в случае потери соединения со шлюзом одного интернет-провайдера обеспечить переключение на другой шлюз, выполните следующие действия (после выполнения действий из предыдущего сценария):

1 Добавьте правила маршрутизации для таблицы по умолчанию Main (default) для обоих провайдеров. Для этого на каждом ViPNet Coordinator KB выполните команды:

```
hostname# inet policy rule add isp-gw-1 1024 table default
```

```
hostname# inet policy rule add isp-gw-2 1024 table default
```

Данные правила будут выполняться в первую очередь за счет высокого приоритета (1024).



Примечание. Чтобы не возникло проблем с прохождением трафика по маршрутам таблицы по умолчанию, приоритет у правил маршрутизации трафика по таблице по умолчанию должен быть выше, чем у общих правил маршрутизации, применяющихся для всего трафика.

---

2 Настройте политики маршрутизации, по которым все IP-адреса будут маршрутизироваться по таблице маршрутизации GW-1 (в случае, когда работает только шлюз интернет-провайдера GW-1) и по таблице GW-2 (в случае, когда работает только шлюз интернет-провайдера GW-2). Для этого на каждом ViPNet Coordinator KB выполните команды:

```
hostname# inet policy rule add isp-gw-1 1025 table name GW-1
hostname# inet policy rule add isp-gw-2 1025 table name GW-2
```

Данные правила будут выполняться в последнюю очередь за счет более низкого приоритета (1025).

3 Запустите службу DGD на обоих ViPNet Coordinator KB:

```
hostname# inet dgd mode on
```

4 Настройте проверку доступности шлюзов провайдеров GW-1 и GW-2:

- на -1:

```
hostname# inet dgd next-hop add GW-1-hop address 89.135.15.2 check icmp
hostname# inet dgd next-hop add GW-2-hop address 210.11.0.2 check icmp
```

- на -2:

```
hostname# inet dgd next-hop add GW-1-hop address 81.30.192.130 check icmp
hostname# inet dgd next-hop add GW-2-hop address 212.160.10.30 check icmp
```

5 Создайте правило службы DGD, по которому действует политика маршрутизации isp-both, если доступны шлюзы обоих провайдеров. Для этого на каждом ViPNet Coordinator KB выполните команды:

```
hostname# inet dgd rule add both-rule match-next-hop GW-1-hop up
hostname# inet dgd rule add both-rule match-next-hop GW-2-hop up
hostname# inet dgd rule add both-rule action-priority 10 service route command
policy active isp-both
```

6 Создайте правило службы DGD, по которому действует политика маршрутизации isp-gw-1, если доступен только шлюз провайдера GW-1. Для этого на каждом ViPNet Coordinator KB выполните команды:

```
hostname# inet dgd rule add gw-1-rule match-next-hop GW-1-hop up
hostname# inet dgd rule add gw-1-rule match-next-hop GW-2-hop down
hostname# inet dgd rule add gw-1-rule action-priority 10 service route command
policy active isp-gw-1
```

7 Создайте правило службы DGD, по которому действует политика маршрутизации isp-gw-2, если доступен только шлюз провайдера GW-2. Для этого на каждом ViPNet Coordinator KB выполните команды:

```
hostname# inet dgd rule add gw-2-rule match-next-hop GW-1-hop down
hostname# inet dgd rule add gw-2-rule match-next-hop GW-2-hop up
hostname# inet dgd rule add gw-2-rule action-priority 10 service route command
policy active isp-gw-2
```

8 Создайте правило службы DGD, по которому создается запись в системном журнале, если недоступны шлюзы обоих провайдеров. Для этого на каждом ViPNet Coordinator KB выполните команды:

```
hostname# inet dgd rule add down-rule match-next-hop GW-1-hop down
hostname# inet dgd rule add down-rule match-next-hop GW-2-hop down
hostname# inet dgd rule add down-rule action-priority 10 service log command both-
isp-gws-down
```

После выполнения описанных действий вы обеспечите маршрутизацию трафика между двумя сетями ViPNet даже в случае недоступности шлюза одного из интернет-провайдеров.

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 36. Связь между туннелируемыми и защищенными узлами

В этой схеме необходимо настроить туннелирование всех IP-адресов интернета, чтобы прокси-сервер, расположенный за ViPNet Coordinator KB, 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 Administrator 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:

   ```
   host# 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.

13

Обеспечение отказоустойчивости

Настройка системы защиты от сбоев 204

Организация обеспечения электропитания от ИБП 221

Настройка системы защиты от сбоев

Назначение и принципы работы

Система защиты от сбоев предназначена для контроля работоспособности ПО ViPNet Coordinator KB и создания отказоустойчивого решения на базе узлов ViPNet Coordinator KB. Данная система может работать в одиночном режиме (см. Работа системы защиты от сбоев в одиночном режиме на стр. 204) или в режиме кластера горячего резервирования (см. Работа системы защиты от сбоев в режиме кластера горячего резервирования на стр. 205).

Параметры работы системы защиты задаются в конфигурационном файле `failover.ini`. Подробнее см. в документе «Справочник команд и конфигурационных файлов».

Работа системы защиты от сбоев в одиночном режиме

По умолчанию система защиты от сбоев работает в одиночном режиме и выполняет следующие функции:

- контроль собственной работоспособности;
- контроль работоспособности служб и драйверов ViPNet Coordinator KB, ведение статистики использования системных ресурсов;
- контроль сбоев при обработке IP-пакетов драйвером ViPNet.

Данные функции выполняют watchdog-драйвер `itcswd` и служба `failoverd`, которые работают постоянно в фоновом режиме. При загрузке системы сначала запускается watchdog-драйвер, а затем служба `failoverd`, которая запускает остальные службы и драйверы ViPNet Coordinator KB.

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 KB. 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 драйвера ViPNet 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 удается, служба `failoverd` производит принудительный перезапуск. Если после пяти перезапусков

подряд, работу контролируемой службы восстановить не удалось, она считается неработоспособной, и `failoverd` инициирует перезапуск системы.



Примечание. Система защиты от сбоев контролирует работоспособность только запущенных служб и драйверов ViPNet Coordinator KB.

Работа системы защиты от сбоев в режиме кластера горячего резервирования

Система защиты от сбоев позволяет объединить два ViPNet Coordinator KB 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 напрямую к интерфейсу другого узла кластера, и используется для передачи данных между узлами кластера.

  Наличие прямого 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.

  Через интерфейс синхронизации передаются следующие данные:

  - 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»).
- 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.

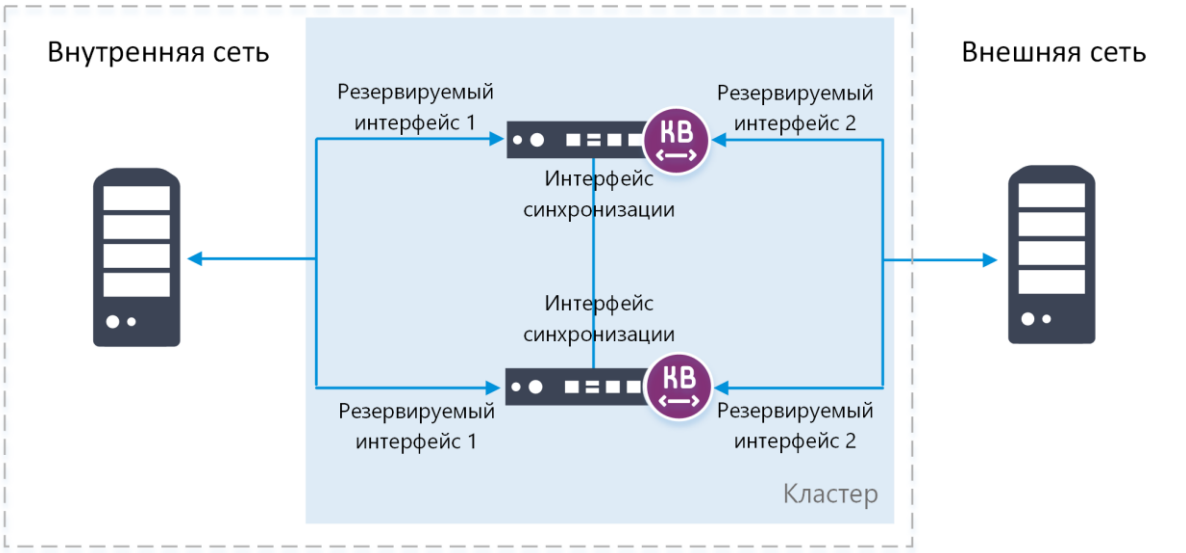


Рисунок 37. Схема кластера горячего резервирования

В процессе работы кластера пассивный узел контролирует работоспособность активного узла кластера, а активный узел контролирует работоспособность своих интерфейсов по приведенным ниже алгоритмам.

Алгоритм проверки работоспособности активного узла кластера:

1. Пассивный узел периодически посылает ARP-запрос на обнаружение дублирующего IP-адреса Duplicate address detection, проверяя присутствует ли в сети интерфейс с IP-адресом активного узла, и ожидает ARP-ответ.
2. В случае отсутствия ответа, пассивный узел кластера переключается в активный режим, и каждый его резервируемый интерфейс получает IP-адрес для работы в активном режиме. При этом активный узел кластера переходит в пассивный режим, а на его резервируемых интерфейсах настроенные IP-адреса сбрасываются.

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, производится дополнительная проверка — через каждый интерфейс посылаются эхо-запросы стабильным объектам сети (например, маршрутизаторам).

3 В случае отсутствия ответов на эхо-запросы, счетчик отказов для соответствующего интерфейса увеличивается на единицу. Если получен хотя бы один ответ или зафиксирован трафик на интерфейсе, счетчик отказов сбрасывается.

4 При достижении счетчиком порогового значения интерфейс считается неработоспособным, и активный узел переключается в пассивный режим, а пассивный узел становится активным.



Примечание. Эхо-запросы для контроля работоспособности сетевых интерфейсов активного узла могут посылаться постоянно (подробнее см. в описании конфигурационного файла `failover.ini` в документе «Справочник команд и конфигурационных файлов»).

Подробнее об алгоритме проверки работоспособности активного узла кластера см. в разделе Типовая схема организации кластера (на стр. 210).

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 в наличии двух ViPNet Coordinator KB одного исполнения, на которых установлены одинаковые справочники, ключи и ключевой блокнот ДСДР, а также одинаковая версия ПО.

2 Если вы разворачиваете кластер на исполнениях KB100, то убедитесь, что для них в ПО ViPNet Центр управления сетью задана роль **Failover KB100**. Для этого на ViPNet Coordinator KB выполните:

```
hostname> version features list
```

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 **Failover KB100** сетевому узлу ViPNet Coordinator KB и последующего обновления справочников ключей на обоих узлах кластера ViPNet Coordinator KB.



Примечание. Если роль **Failover KB100** была добавлена на ViPNet Coordinator KB не при первичной установке, а при обновлении справочников и ключей, то команды для работы с кластером могут не появиться в текущем сеансе командного интерпретатора. В таком случае отключите токен и заново пройдите процедуру аутентификации.

---

3 Настройте параметры сетевых интерфейсов, соблюдая следующие ограничения:

- Перед началом настройки сетевых интерфейсов необходимо отключить от локальной сети оба узла кластера. Это позволит избежать проблем в работе локальной сети и рассылке справочно-ключевой информации в сети ViPNet.
- Для сетевых интерфейсов, задействованных в работе кластера горячего резервирования, должны быть заданы статические IP-адреса. Назначение IP-адресов по протоколу DHCP не допускается.



Примечание. В качестве резервируемых сетевых интерфейсов можно задавать физические, виртуальные или VLAN интерфейсы.

- Во избежание конфликтов при копировании конфигурационных файлов с активного узла на пассивный требуется, чтобы имена сетевых интерфейсов, используемых для доступа к соответствующим сетям, совпадали.
- При настройке параметров в секциях `[channel]` IP-адреса, указанные в параметрах `activeip` и `testip`, должны находиться в одной подсети. При этом для параметра `activeip` необходимо указать маску подсети.
- В соответствующих секциях `[channel]` конфигурационного файла `failover.ini` должны быть указаны сетевые интерфейсы класса `access`, описанные секциями `[adapter]` в конфигурационном файле `iplir.conf`. Подробнее см. документ «Справочник команд и конфигурационных файлов».

4 Организуйте между активным и пассивным узлами кластера канал синхронизации для передачи файлов с целью резервирования, для этого допускается использовать только кросс-кабель, без использования коммутирующего оборудования.



Внимание! IP-адрес интерфейсов синхронизации не должны совпадать с IP-адресами защищенных и туннелируемых узлов, иначе работа канала синхронизации будет нарушена. Если вам необходимо использовать одинаковые адреса, то для разрешения конфликта IP-адресов настройте видимость соответствующих защищенных узлов по виртуальным адресам.

5 Выберите один или несколько стабильных объектов сети (например, маршрутизатор), которые будут использоваться для проверки работоспособности сетевых интерфейсов активного узла кластера.

Для развертывания кластера на каждом из ViPNet Coordinator KB выполните следующие действия:

1 Завершите работу системы защиты от сбоев командой:

```
hostname> failover stop
```

2 Откройте для редактирования конфигурационный файл `failover.ini` с помощью команды:

```
hostname# failover config edit
```

3 В конфигурационном файле `failover.ini` для резервируемых интерфейсов, которые используются для взаимодействия с узлами внутренней и внешней сетей, в соответствующих секциях `[channel]` укажите значения следующих параметров:

- `device` — имя сетевого интерфейса;
- `activeip` — IP-адрес и маска подсети, которые интерфейс будет иметь в активном режиме;
- `testip` — IP-адрес маршрутизатора или другого стабильного объекта сети, находящегося в одной подсети с IP-адресом из параметра `activeip`, и которому будут посылаться эхо-запросы для проверки работоспособности интерфейса.

  При необходимости можно для каждого из сетевых интерфейсов указать несколько параметров `testip`. В этом случае сбоем интерфейса будет считаться ситуация, когда ни от одного из заданных IP-адресов не будет получено ответа.

  Например, чтобы эхо-запросы отправлялись на IP-адреса 192.168.100.34 и 192.168.100.25, добавьте следующие строки:

  ```
  testip = 192.168.100.34
  testip = 192.168.100.25
  ```

- `ident` — текстовая строка, идентифицирующая сетевой интерфейс. Для интерфейсов, подключенных к одинаковым сетям, параметры `ident` должны совпадать.



Внимание! На узлах кластера значения параметров `activeip`, `ident` и `testip` должны быть одинаковыми для сетевых интерфейсов одного канала синхронизации. В качестве параметра `testip` не рекомендуется задавать IP-адрес интерфейса «внутренней петли» (loopback), например `127.0.0.1` или `localhost`, так как в этом случае реальной проверки работоспособности сетевого интерфейса не производится.

- `checkonlyidle` — указание на необходимость проверки только неактивных сетевых интерфейсов. Возможные значения:
  - `yes` (по умолчанию) — активный узел посылает эхо-запросы на сетевые интерфейсы, IP-адреса которых указаны в параметрах `testip`, только если за период опроса IP-адресов на данных интерфейсах не было ни входящих, ни исходящих пакетов (период опроса задается параметром `checktime` в секции `network`);
  - `no` — эхо-запросы на интерфейсы, IP-адреса которых указаны в параметрах `testip`, посылаются всегда.

4. В конфигурационном файле `failover.ini` в секции `[sendconfig]` задайте следующие параметры, контролирующие пересылку файлов с активного узла на пассивный с целью резервирования:
   - `device` — имя сетевого интерфейса, который используется для организации канала синхронизации;
   - `activeip` — IP-адрес интерфейса, подключенного к каналу синхронизации другого узла кластера (который работает в режиме, противоположном режиму данного узла).
5. При необходимости измените другие параметры секции `[sendconfig]` и параметры работы системы защиты от сбоев, относящиеся к посылке IP-пакетов в сеть, в секции `[network]` файла `failover.ini`.

6 Чтобы сохранить изменения в конфигурационном файле `failover.ini`, нажмите сочетание клавиш **Ctrl+O**. Затем нажмите клавишу **Enter**.

7 Чтобы закрыть конфигурационный файл `failover.ini`, нажмите сочетание клавиш **Ctrl+X**.

8 Переведите систему защиты от сбоев в режим кластера горячего:

```
hostname# failover config mode cluster
```

9 Запустите систему защиты от сбоев:

```
hostname# failover start
```



Внимание! Настройку сетевых служб необходимо выполнять после развертывания кластера горячего резервирования и запуска службы защиты от сбоев `failoverd`. Настройка выполняется на активном узле кластера.

Настройки NTP-сервера и DNS-сервера необходимо выполнить на обоих узлах кластера горячего резервирования.

В результате кластер начнет свою работу, и на его узлах будут автоматически созданы нередактируемые сетевые фильтры, необходимые для корректной работы системы защиты от сбоев с выбранными параметрами.

10 Чтобы проверить доступность активного узла кластера, на пассивном узле выполните:

```
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
```

Типовая схема организации кластера

Пример типовой схемы организации кластера горячего резервирования приведен ниже. Сетевые интерфейсы eth0 используются для организации канала синхронизации.

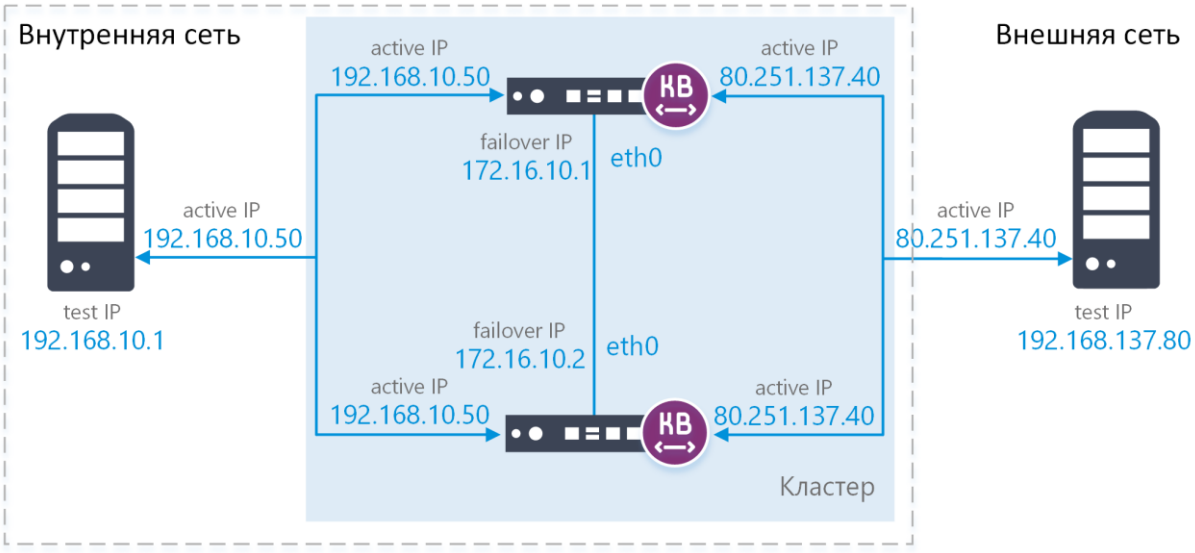


Рисунок 38. Типовая схема организации кластера горячего резервирования

Параметры работы системы защиты от сбоев настраиваются в конфигурационном файле `failover.ini`, содержащем несколько секций. Для настройки кластера предназначены параметры секций `[channel]`, `[network]` и `[sendconfig]`. Подробнее см. в документе «Справочник команд и конфигурационных файлов» в соответствующих разделах..

Внимание! Если в кластере используется функция L2OverIP (см. Настройка туннелирования на канальном уровне по технологии L2OverIP на стр. 82) и требуется проверять состояние его рабочего интерфейса, то в качестве тестового IP-адреса для этого интерфейса необходимо задать IP-адрес из локального сегмента, а также выключить блокирование одноадресных Ethernet-кадров с неизвестным MAC-адресом получателя командой `iplir set l2overip unsolicited-frames broadcast`.

Настоятельно не рекомендуется указывать для рабочего интерфейса L2OverIP тестовый IP-адрес в удаленном сегменте.

В случае использования типовой схемы необходимо соблюдать следующие требования:

- На сетевых интерфейсах, которые задействованы в работе кластера горячего резервирования, должны быть заданы статические IP-адреса. Назначение IP-адресов по протоколу DHCP не допускается.

Примечание. В качестве резервируемых сетевых интерфейсов можно задавать физические, виртуальные или VLAN интерфейсы. В данном сценарии используются только физические сетевые интерфейсы.

- При настройке параметров в секциях `[channel]` IP-адреса, указанные в параметрах `activeip` и `testip`, должны находиться в одной подсети. При этом для параметра `activeip` необходимо указать маску подсети.
- Значения параметров `activeip`, `ident` и `testip` должны быть одинаковыми на обоих узлах кластера для сетевых интерфейсов, используемых для резервирования одного канала.

- Для интерфейсов, подключенных к одинаковым сетям, параметры `ident` должны совпадать на обоих узлах кластера, т.к. по этим параметрам система защиты от сбоев определяет интерфейсы, которые выполняют одинаковые функции на узлах кластера.



Внимание! Настоятельно рекомендуем назначать одинаковые имена сетевым интерфейсам, выполняющим одинаковые функции на узлах кластера. В следующих версиях ViPNet Coordinator KB параметр `ident` не будет использоваться, и отличающиеся имена интерфейсов приведут к нарушению работоспособности сети ViPNet.

Таблица 10. Настройки параметров системы защиты от сбоев для типовой схемы организации кластера

Настройки на первом узле	Настройки на втором узле
[channel]	**[channel]**
`device = eth1`	`device = eth1`
`ident = iface-1`	`ident = iface-1`
`activeip = 192.168.10.50/24`	`activeip = 192.168.10.50/24`
`testip = 192.168.10.1`	`testip = 192.168.10.1`
`checkonlyidle = yes`	`checkonlyidle = yes`
[channel]	**[channel]**
`device = eth2`	`device = eth2`
`ident = iface-2`	`ident = iface-2`
`activeip = 80.251.137.40/24`	`activeip = 80.251.137.40/24`
`testip = 80.251.137.80`	`testip = 80.251.137.80`
`checkonlyidle = yes`	`checkonlyidle = yes`
[network]	**[network]**
`checktime = 10`	`checktime = 10`
`timeout = 2`	`timeout = 2`
`activeretries = 3`	`activeretries = 3`
`channelretries = 3`	`channelretries = 3`
`synctime = 5`	`synctime = 5`
`fastdown = yes`	`fastdown = yes`
[sendconfig]	**[sendconfig]**
`device = eth0`	`device = eth0`
`activeip = 172.16.10.2` (соответствует failover IP второго узла)	`activeip = 172.16.10.1` (соответствует failover IP первого узла)

К такой схеме применяется следующий алгоритм работы активного узла:

1. Каждые `checktime` секунд проверяется работоспособность каждого из указанных в конфигурационном файле сетевых интерфейсов.
2. Если параметр `checkonlyidle` установлен в значение `yes`, то анализируется весь сетевой трафик, проходящий через интерфейс.
3. Если в течение `checktime` через сетевой интерфейс не прошло ни одного IP-пакета, то через этот интерфейс посылаются эхо-запросы на узел с IP-адресом `testip`, и в течение `timeout` ожидаются ответы на эти запросы.
4. Если ни одного ответа нет, то счетчик сбоев сетевого интерфейса увеличивается на единицу, и посылаются новые эхо-запросы. Если на эхо-запрос приходит хотя бы один ответ, счетчик сбоев обнуляется. Если обнуляются счетчики всех интерфейсов, происходит возврат к началу цикла проверки (ожидание в течение `checktime` и так далее). При достижении значения `channelretries` — интерфейс считается неработоспособным, и система перезагружается.



Примечание. Чтобы эхо-запросы на узел с IP-адресом `testip` посылались каждые `checktime` секунд, установите параметр `checkonlyidle` в значение `no`.

Таким образом, максимальное время, которое нужно системе защиты от сбоев для того, чтобы определить неработоспособность сетевого интерфейса, равно:

```
checktime + (timeout * channelretries)
```

К такой схеме применяется следующий алгоритм работы пассивного узла:

1. Каждые `checktime` секунд посылаются ARP-запросы в режиме DAD на узел с IP-адресом `activeip`.
2. По истечении `timeout` проверяется наличие положительного ответа для каждого `activeip` в ARP-таблице и делается вывод о работоспособности соответствующего сетевого интерфейса активного узла.
3. Если ни от одного сетевого 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.

Таким образом, максимальное время, проходящее с момента начала перезагрузки активного узла до обнаружения пассивным узлом этого факта, равно:

```
checktime + (timeout * activeretries)
```

Минимальное общее время неработоспособности системы при сбое равно:

```
checktime * 2 + timeout * (channelretries + activeretries)
```

Реальное время неработоспособности системы при сбое будет несколько больше. Это связано с тем, что после начала перезагрузки сбойного узла система переводит его сетевые интерфейсы в нерабочее состояние не сразу, а через некоторое время, после завершения работы других подсистем (обычно около 30 секунд).

Развертывание кластера совместно с DHCP-сервером и DHCP-relay

Для развертывания кластера горячего резервирования с DHCP-сервером и DHCP-relay на интерфейсах кластера:

1. Разверните кластер (см. Развертывание кластера горячего резервирования на стр. 207) 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 кластера настройте параметры работы DHCP-сервера и DHCP-relay (см. Настройка DHCP-relay на стр. 158).
3. Убедитесь, что параметры DHCP-сервера и DHCP-relay успешно реплицировались на пассивный узел кластера. Для этого на пассивном узле кластера выполните команды:

   ```
   hostname> inet show dhcp server
   hostname> inet show dhcp relay [<номер копии>]
   ```
4. Запустите службы DHCP-сервера и DHCP-relay на активном узле кластера с помощью команд:

   ```
   hostname# inet dhcp server start
   hostname# inet dhcp relay [<номер копии>] start
   ```
5. Включите автоматический запуск DHCP-сервера и DHCP-relay при загрузке ViPNet Coordinator KB:

   ```
   hostname# inet dhcp server mode on
   hostname# inet dhcp relay mode on
   ```

   Переключите режим работы пассивного узла кластера в активный. Для этого перезагрузите активный узел кластера:

   ```
   hostname> machine reboot
   ```
6. Убедитесь, что на ставшем активном узле кластера службы DHCP-сервера и DHCP-relay запущены и имеют верные параметры:

   ```
   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:

```
hostname> failover start
```

Чтобы запустить систему защиты от сбоев на узле кластера горячего резервирования, выполните одну из команд:

- Для запуска в активном режиме:

  ```
  hostname> failover start active
  ```

- Для запуска в пассивном режиме:

  ```
  hostname> failover start passive
  ```



Внимание! Если при запуске на узле кластера горячего резервирования не был указан параметр `active` или `passive`, то система защиты от сбоев будет запущена в том режиме, в котором она находилась до завершения работы.

Перед запуском системы защиты от сбоев на узле кластера горячего резервирования в активном или пассивном режиме убедитесь в том, что второй узел кластера работает в противоположном режиме. Запуск обоих узлов кластера в активном режиме вызовет конфликт IP-адресов и другие неполадки.

Чтобы завершить работу системы защиты от сбоев:

```
hostname> failover stop
```

Переключение кластера в одиночный режим работы

Чтобы переключить кластер горячего резервирования в одиночный режим:

1. На каждом из узлов кластера выполните:

   ```
   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 следующие действия:

   - 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 KB 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 на стр. 207). 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 пассивного 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 KB, который будет включен в состав кластера, отключен от локальной сети.

2 Выполните экспорт справочников, ключей и настроек активного узла кластера (на стр. 33).

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 на стр. 35).

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:

- `<интерфейс>` — имя сетевого интерфейса;
- `<IP-адрес>` — IP-адрес;
- `<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**) и закройте редактор (**Ctrl+X**).

9 Если на новом узле кластера включена функция ежедневной перезагрузки, измените время перезагрузки:

```
hostname# machine set dailyreboot time <ЧЧ:ММ>
```

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:

```
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 KB 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 (`free memory`);
- сведения о текущем состоянии служб ViPNet Coordinator KB:
    - `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 KB 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 4.3.3(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 2023
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 4.3.3 (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 2023
                        * 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 горячего резервирования записываются службой `failoverd` в журнал переключений. Среди таких событий:

- `<BOOT>` — запуск службы системы защиты от сбоев failoverd при загрузке операционной системы;
- `<P_START>` — ручной запуск службы системы защиты от сбоев failoverd в пассивном режиме;
- `<A_START>` — ручной запуск службы системы защиты от сбоев failoverd в активном режиме;
- `<SWITCH>` — переход узла из пассивного режима работы в активный (или наоборот).

Для просмотра записей в журнале переключений узла, выполните:

```
hostname> failover view <dd.mm.yyyy[.hh.mm.ss]> <dd.mm.yyyy[.hh.mm.ss]>,
```

указав начало и конец интервала времени, записи за который вас интересуют. Например:

```
failover view 08.03.2019.00.00.00 23.03.2019.16.37.00
```

В результате выполнения команды выводится следующая информация:

- версии ПО ViPNet Coordinator KB и службы системы защиты от сбоев failoverd;
- идентификатор и имя узла;
- указание на работу системы защиты от сбоев в режиме кластера горячего резервирования;

- 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 4.3.3 (30), daemon 1.5 (1)
Workstation configured for ID 29A0022 (Coordinator_KB)
The workstation works in a cluster mode of protection against failures
Workstation time (utc: 1174916969) Thu Mar 27 17:35:30 2023

09 Mar 2023 12:51:42    <P_START> Start failover daemon in passive mode
22 Mar 2023 12:27:27    <A_START> Start failover daemon in active mode
22 Mar 2023 14:10:35    <A_START> Start failover daemon in active mode
22 Mar 2023 15:30:46    <BOOT> Boot the system
23 Mar 2023 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 совместно с коммутационным оборудованием

На практике часто применяются схемы подключения кластера горячего резервирования к коммутаторам, маршрутизаторам или другому коммутационному оборудованию. Настройки данного оборудования могут влиять на работу кластера горячего резервирования. В частности, администратор такого оборудования может настроить блокирование тех или иных сетевых пакетов, среди которых могут оказаться служебные пакеты, необходимые для корректного функционирования ViPNet Coordinator KB в режиме кластера горячего резервирования.

В связи с этим при использовании коммутационного оборудования с кластером горячего резервирования необходимо убедиться в следующем:

- На 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.
- На оборудовании отключена функция Proxy ARP и пропускаются ARP-запросы от пассивного узла до активного узла и ответы на них.

Данные рекомендации касаются всех сетевых интерфейсов ViPNet Coordinator KB, для которых в файле `failover.ini` существует секция `[channel]`.

Организация электропитания от ИБП

Электропитание ViPNet Coordinator KB должно осуществляться от системы бесперебойного и гарантированного электропитания (СБГЭ). В случае невозможности подключения к СБГЭ, необходимо подключить ViPNet Coordinator KB к источнику бесперебойного питания (ИБП) для защиты от отключений питания и от перебоев в электросети.

Взаимодействие ИБП с ViPNet Coordinator KB происходит через интерфейсный кабель. Если ViPNet Coordinator KB подключен к ИБП, при отключении электроэнергии:

1. ViPNet Coordinator KB начинает питаться от аккумуляторной батареи ИБП.
2. Если заряд аккумуляторной батареи достигает критически низкого уровня (зависит от параметров используемой модели ИБП):
   - **2.1** ИБП посылает ViPNet Coordinator KB сигнал об истощении батареи.
   - **2.2** ViPNet Coordinator KB начинает выключаться и передает на ИБП информацию о том, что для выключения потребуется не менее 180 секунд.
   - **2.3** ИБП корректирует расход оставшегося заряда батареи, чтобы ViPNet Coordinator KB успел штатно завершить работу.

     В результате ViPNet Coordinator KB успевает выключиться до полного разряда батареи в штатном режиме.
3. Если подача электроэнергии восстанавливается до достижения батареей критически низкого уровня заряда, ViPNet Coordinator KB снова начинает питаться от электросети.

К одному ИБП можно подключить несколько устройств. Устройство, к которому подключен интерфейсный кабель ИБП, является главным (режим `master`), остальные — подчиненными (режим `slave`). Главное устройство взаимодействует с ИБП и отвечает за своевременное оповещение по сети подчиненных. Главное устройство должно быть доступно подчиненным по одному из своих IP-адресов.

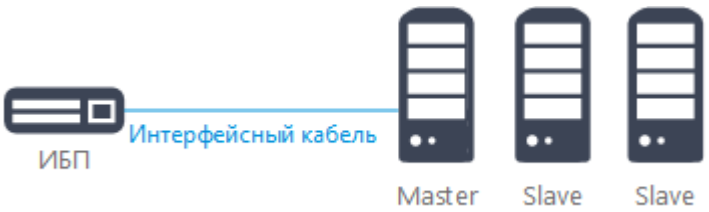


Рисунок 39. Схема подключения нескольких устройств к ИБП

Если к ИБП подключено только одно устройство, то подчиненные отсутствуют.

Если к ИБП подключен кластер горячего резервирования, состоящий из двух ViPNet Coordinator KB, то один ViPNet Coordinator KB будет главным, а другой — подчиненным. При этом никакие другие

устройства к ИБП подключать нельзя. Для настройки взаимодействия узлов кластера с ИБП используйте IP-адреса интерфейсов синхронизации.

Чтобы настроить взаимодействие нескольких ViPNet Coordinator KB с ИБП, настройте каждый ViPNet Coordinator KB в соответствии с его режимом работы.

На главном ViPNet Coordinator KB:

1 Подключите к ViPNet Coordinator KB интерфейсный кабель ИБП.

2 Включите взаимодействие с ИБП:

```
hostname# ups set monitoring on
```

3 Установите режим `master`:

```
hostname# ups set mode master
```

4 Запустите взаимодействие с ИБП:

```
hostname# ups start
```

5 Проверьте взаимодействие 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 KB, а затем снова подключили), то для восстановления взаимодействия с ИБП выполните последовательно команды `ups stop` и `ups start`.

На подчиненном ViPNet Coordinator KB:

1 Включите взаимодействие с ИБП:

```
hostname# ups set monitoring on
```

2 Установите режим `slave`:

```
hostname# ups set mode slave <master_IP>
```

Главный ViPNet Coordinator KB должен быть доступен с подчиненного по IP-адресу, заданному в параметре `<master_IP>`.

При подключении к ИБП кластера горячего резервирования, в качестве `<master_IP>` укажите адрес интерфейса синхронизации первого ViPNet Coordinator KB.

3 Запустите взаимодействие с ИБП:

```
hostname# ups start
```

4 Проверьте взаимодействие с ИБП:

```
hostname# ups show status
```

Если взаимодействие установлено, отобразится информация о текущем состоянии ИБП, подключенного к главному ViPNet Coordinator KB.

14

Настройка транспортного сервера MFTP

Назначение и функции транспортного модуля 224

Выбор канала передачи конвертов 225

Настройка взаимодействия с модулем почтового обмена 226

Настройка протоколирования событий транспортного сервера 227

Назначение и функции транспортного модуля

Транспортный модуль предназначен для надежной и безопасной передачи транспортных конвертов между связанными узлами сети ViPNet посредством протоколов TCP (этот канал передачи конвертов называется MFTP) и SMTP/POP3. Транспортный модуль входит в состав ViPNet Coordinator KB в виде службы `mftpd` и модуля почтового обмена MailTrans. Он обеспечивает выполнение координатором функции транспортного сервера, принимает непосредственное участие в удаленном обновлении справочников и ключей на узлах, а также приеме политик безопасности из ПО ViPNet Policy Manager.

Транспортные конверты представляют собой файлы с данными, которыми обмениваются между собой приложения, входящие в состав ПО ViPNet. Размер конверта, передаваемый транспортным модулем, не может превышать 2 Гбайт (2 097 151 Кбайт). Транспортный модуль передает конверты в соответствии с адресами получателей, прописанными в заголовках этих конвертов.

Для обмена транспортными конвертами на защищенных узлах могут использоваться каналы следующих типов:

- MFTP — при связи по такому каналу узел-отправитель конвертов устанавливает TCP-соединение с узлом-получателем конвертов, проводится взаимная аутентификация узлов и осуществляется передача конвертов между узлами через транспортные серверы. Данный тип канала по умолчанию устанавливается для защищенных сетевых узлов.
- SMTP/POP3 — при связи по такому каналу транспортный модуль переадресует конверты для отправки модулю почтового обмена MailTrans, который передает их через сервер SMTP, а также забирает с сервера POP3 конверты, предназначенные для данного узла. То есть канал SMTP/POP3 позволяет использовать для обмена транспортными конвертами почтовые серверы, что удобно, например, в случае передачи конвертов между сетевыми узлами разных организаций, в которых по каким-либо причинам затруднен выход в Интернет по TCP-каналам (см. Настройка взаимодействия с модулем почтового обмена на стр. 226).

Настройка транспортного модуля выполняется путем редактирования конфигурационного файла `mftp.conf`. Подробнее о параметрах, содержащихся в файле `mftp.conf`, см. в документе «Справочник команд и конфигурационных файлов».

Выбор канала передачи конвертов

По умолчанию для обмена конвертами с защищенными узлами, связанными с ViPNet Coordinator KB, используется канал MFTP. Чтобы изменить тип канала на SMTP/POP3:

1. Завершите работу службы mftpd:

   ```
   hostname> mftp stop
   ```

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

   ```
   hostname# mftp config
   ```

3. В секции `[channel]` нужного узла установите следующие параметры:

   - установите параметр `type` в значение `smtp`;
   - в параметре `reportaddress` укажите адрес электронной почты для отправки исходящих конвертов в формате:

     ```
     <имя пользователя>@<сервер>.<домен>
     ```

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

5. Запустите службу mftpd:

   ```
   hostname> mftp start
   ```

Настройка взаимодействия с модулем почтового обмена

Если для каких-либо узлов в вашей организации предполагается использование каналов SMTP/POP3, настройте параметры взаимодействия транспортного сервера с модулем почтового обмена `mailtrans`:

1. Завершите работу службы `mftpd`:

   ```
   hostname# mftp stop
   ```

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

   ```
   hostname# mftp config
   ```

3. В секции `[mailtrans]` укажите значения следующих параметров:

   - `frommailbox` — адрес электронной почты отправителя SMTP-конвертов в формате: `<имя пользователя>@<сервер>.<домен>`
   - `inputmailbox` — адрес электронной почты, по которому модуль почтового обмена будет забирать конверты по протоколу POP3, в формате: `<имя пользователя>:<пароль>@<IP-адрес POP3-сервера>`
   - `outputmailbox` — IP-адрес SMTP-сервера, на который модуль почтового обмена будет отправлять конверты по протоколу SMTP.

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

5. Запустите службу `mftpd`:

   ```
   hostname# mftp start
   ```

Настройка протоколирования событий транспортного сервера

Служба `mftpd` записывает информацию о событиях, происходящих в процессе работы транспортного сервера, в системный журнал.

Чтобы изменить параметры записи событий транспортного сервера в журнал:

1. Завершите службу `mftpd`:

   ```
   hostname> mftp stop
   ```

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

   ```
   hostname# mftp config
   ```

3. В секции `[debug]` измените значения следующих параметров:

   - `debuglevel` — уровень важности информации, выводимой в журнал. Возможные значения: от `0` до `4` (по умолчанию `3`). Чем выше уровень важности, тем более подробная информация выводится в журнал. Значение параметра `-1` выключает ведение журнала.

     Не рекомендуется задавать уровень детализации выше `3` для постоянного использования, так как эти уровни используются только для диагностики возможных проблем и должны быть включены только по рекомендации специалистов ИнфоТеКС.

   - `debuglogfile` — источник информации, выводимой в журнал, в формате: `syslog:<facility.level>`, где:

     - `facility` — процесс, формирующий информацию. Возможные значения: `kern` (ядро), `user` (пользовательские программы), `mail` (почтовая система) 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
   ```

15

Работа с журналами

Просмотр журнала регистрации IP-пакетов 229

Экспорт журнала регистрации IP-пакетов 236

Просмотр журнала транспортных конвертов MFTP 238

Работа с системным журналом 239

Просмотр журнала приема и применения обновлений 243

Просмотр журнала переключений кластера горячего резервирования 244

Просмотр журнала СКЗИ 245

Просмотр журнала регистрации IP-пакетов

Управляющая 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 заносится 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.

Чтобы просмотреть записи из журнала регистрации IP-пакетов, выполните:

```
hostname# iplir view
```

В результате откроется окно **Set search parameters** для задания параметров поиска записей в журнале.

Примечание. При работе ViPNet Coordinator KB 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.

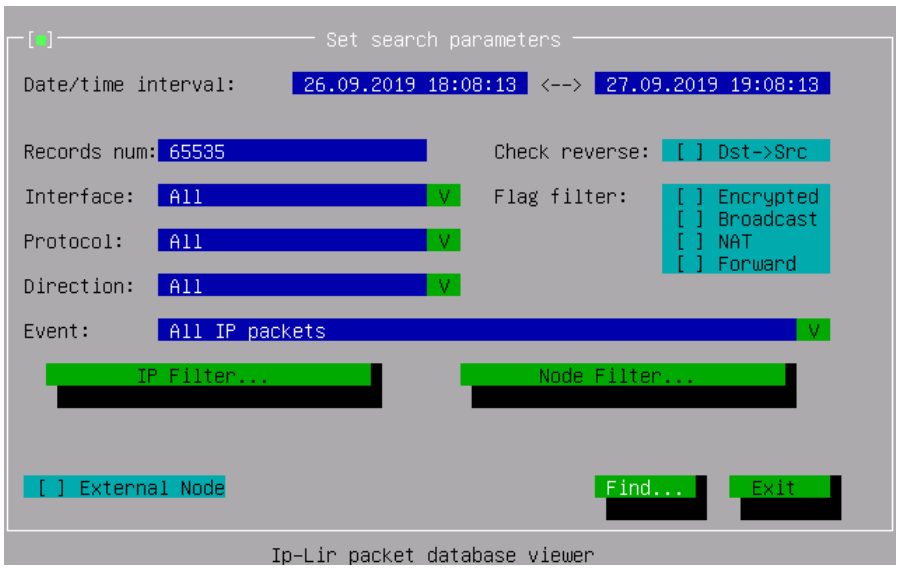


Рисунок 40. Задание параметров поиска записей в журнале регистрации IP-пакетов

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**.

Поле **Records num** задает максимальное количество выводимых записей для каждого сетевого интерфейса. Вы можете ограничить количество выводимых записей для каждого интерфейса, установив значение **Records num**, отличное от заданного по умолчанию (65535). При этом, если вы в параметрах поиска выбрали все сетевые интерфейсы (**All**), то общее количество выводимых записей будет равно сумме всех выводимых записей для каждого интерфейса.

Чтобы выбрать записи из журнала регистрации IP-пакетов для просмотра, выполните следующие действия:

1 Укажите параметры IP-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.
- **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 на стр. 294). По умолчанию — события всех типов (**All**).
- **Check reverse** — включение в журнал ответных IP-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 IP-пакетов:
  - **Encrypted** — зашифрованные пакеты;
  - **Broadcast** — широковещательные пакеты;

- **NAT** — транслированные пакеты;
- **Forward** — транзитные пакеты.

Вы можете выбрать один или несколько видов IP-пакетов, установив соответствующие флажки. Транслированные (**NAT**) и транзитные (**Forward**) пакеты имеет смысл выбирать только для просмотра записей об IP-пакетах на координаторе.

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**);
- 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 — кнопку **Select destination Node**.

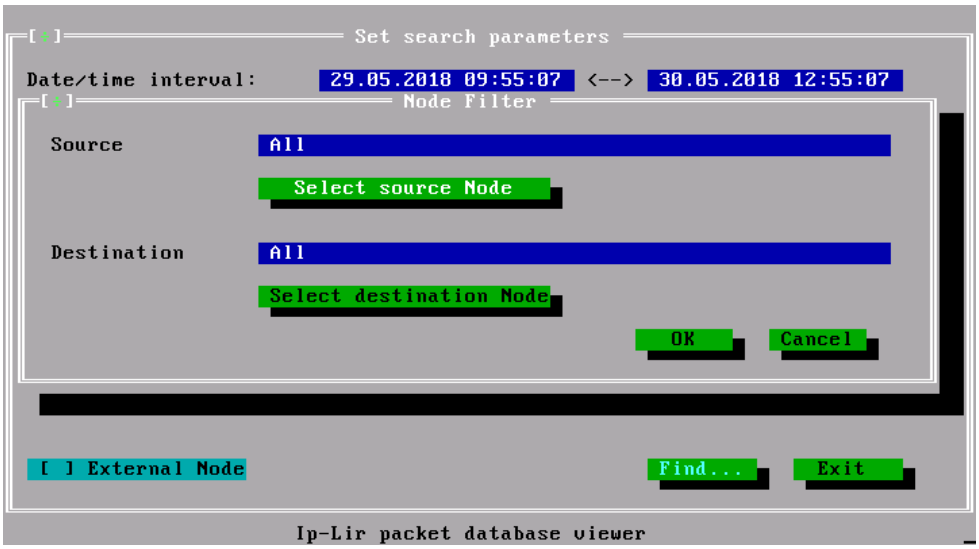


Рисунок 41. Выбор узла отправителя и получателя

В результате откроется таблица со списком защищенных узлов ViPNet, с которыми у данного узла есть связь. В левом столбце таблицы указан шестнадцатеричный идентификатор узла (**ID**), в правом — имя узла (**Name**).

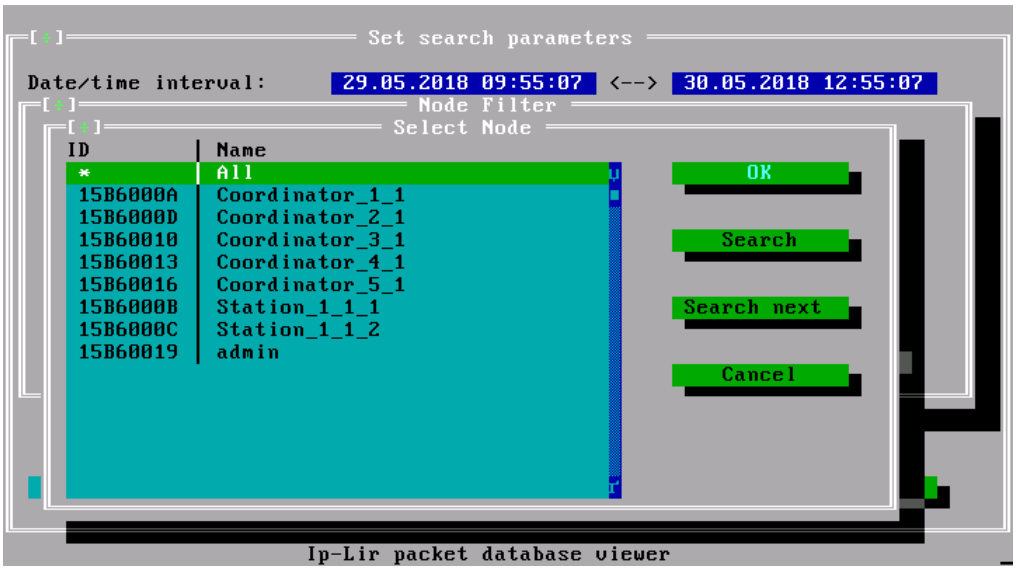


Рисунок 42. Выбор узла отправителя и получателя

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, для этого создайте соответствующий разрешающий фильтр защищенной сети (см. Создание сетевого фильтра на стр. 112). Иначе будет выведено сообщение Can't connect to remote station (errcode -110).

4.1 Установите флажок **External Node** и введите пароль удаленного узла, либо пароль администратора сети ViPNet (группа вся сеть).

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.

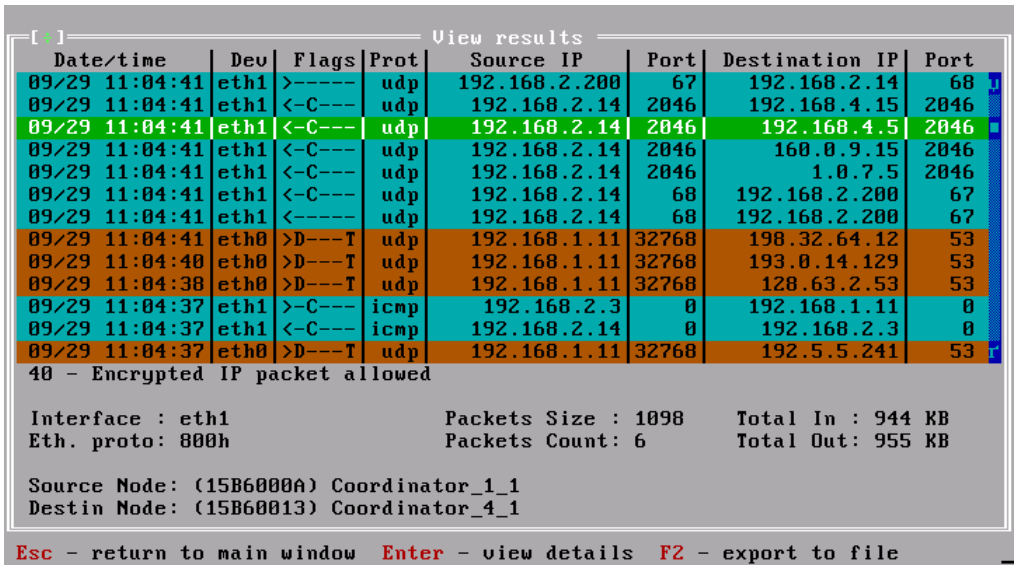


Рисунок 43. Просмотр записей из журнала регистрации IP-пакетов

В окне **View results** вы можете выполнить следующие действия:

- Экспортировать список записей в текстовый файл (например, для последующего анализа). Для этого в окне **View results** нажмите **F2** и укажите имя файла для экспорта.
- Просмотреть краткую информацию об IP-пакете, выбрав его в списке. В под списком записей отобразятся:
  - тип события, которому соответствует IP-пакет (см. Типы событий в журнале регистрации IP-пакетов на стр. 294);
  - **Interface** — имя сетевого интерфейса, на котором был обработан IP-пакет;
  - **Eth. proto** — шестнадцатеричный идентификатор протокола Ethernet;
  - **Packets Size** — суммарный размер пакетов, агрегированных в выбранной записи;

Примечание. Для зашифрованных пакетов в параметре **Packet Size** отображается размер пакетов со всеми служебными заголовками.

  - **Packets Count** — число пакетов, агрегированных в выбранной записи;
  - **Total In** — суммарный размер всех входящих пакетов, отфильтрованных по заданным критериям;
  - **Total Out** — суммарный размер всех исходящих пакетов, отфильтрованных по заданным критериям;

Примечание. Единицы измерения суммарного размера IP-пакетов: **B** — 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).

- Вывести подробную информацию о записи журнала — выберите его в списке и нажмите **Enter**. В окне **Record details** отобразятся:
  - **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

Чтобы экспортировать список записей из журнала регистрации IP-пакетов на USB-носитель:

1 Выполните:

```
hostname# iplir view
```

2 В окне **Set search parameters** задайте параметры для выбора из журнала регистрации IP-пакетов нужных записей и нажмите **Find**.

3 В окне **View results** со списком найденных записей нажмите **F2**.

4 В открывшемся окне введите имя файла, в который вы хотите сохранить журнал, или выберите из списка существующий файл. Затем нажмите **Enter**.

5 Чтобы завершить просмотр журнала регистрации IP-пакетов, нажмите **Esc**. Затем нажмите **Exit**.

6 Выполните:

```
hostname# admin export packetdb usb <имя файла>,
```

указав в качестве параметра имя файла с журналом.

7 Подключите USB-носитель к одному из USB-разъемов ViPNet Coordinator KB и нажмите **Enter**.

8 В списке подключенных к ViPNet Coordinator KB 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** — имя сетевого интерфейса, 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 узла 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.

Просмотр журнала транспортных конвертов MFTP

В процессе работы транспортный сервер (MFTP) записывает информацию о полностью принятых, отправленных, удаленных и поврежденных транспортных конвертах в специальный журнал.

Алгоритм работы с журналом транспортных конвертов основан на использовании ротации. В настройках транспортного сервера задается максимальный размер файла журнала в мегабайтах. При превышении данного размера наиболее ранние записи перезаписываются новыми. Таким образом, задавая максимальный размер журнала, вы можете регулировать число хранимых записей на текущий момент. При изменении максимального размера этого журнала в процессе работы происходит реконструкция журнала транспортных конвертов — в случае уменьшения размера журнала наиболее ранние записи удаляются из него.

Для просмотра журнала транспортных конвертов:

```
hostname# mftp view
```

При просмотре журнала 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.

Работа с системным журналом

В ViPNet Coordinator KB службы `iplircfg`, `mftpd`, `failoverd` записывают информацию о событиях, происходящих в процессе их работы, в системный журнал. Этот журнал позволяет контролировать работоспособность ViPNet Coordinator KB, своевременно выявлять и устранять причины сбоев в его работе.

Запись информации в системный журнал осуществляется системными средствами с использованием протокола `syslog` на самом ViPNet Coordinator KB (локальное протоколирование) или на удаленном сетевом узле (удаленное протоколирование).



Примечание. 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 системного журнала предусмотрена автоматическая ротация файла журнала с целью экономии места на ViPNet Coordinator KB. Ротация выполняется, когда размер файла журнала достигает 1 Гбайт.

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 KB всегда было не более 10 заполненных файлов и 1 текущего файла.

При локальном хранении системного журнала вы можете выполнить с ним следующие действия:

- Просмотреть содержимое журнала, выполнив команду:

  ```
  hostname# machine show logs
  ```

- Экспортировать на USB-носитель (на стр. 242). Файлы журнала экспортируются в виде архива `logs.tar.gz`.

Перечень протоколируемых событий, связанных с аутентификацией и настройкой оборудования, приведен в приложении (на стр. 300).

Просмотр системного журнала

Чтобы просмотреть записи системного журнала, выполните:

```
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 KB.
- `<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 2023 года, и отобразить их в обратном порядке, выполните:

```
hostname# machine show logs reversed since 2023-09-22 14:50:00
```

Настройка параметров ведения системного журнала

Вы можете установить следующие настройки ведения системного журнала:

- Чтобы выбрать место хранения системного журнала, выполните:

  ```
  hostname# machine set loghost {<IP-адрес> | local},
  ```

  указав IP-адрес удаленного сетевого узла, на котором будет храниться журнал, или параметр `local` для локального протоколирования.

  Внимание! Если для удаленного протоколирования будет использоваться открытый узел, на ViPNet Coordinator KB вручную задайте фильтр открытой сети, разрешающий исходящий трафик по протоколу UDP на порт 514 этого открытого узла.

  Не рекомендуется использовать удаленное протоколирование на ViPNet Coordinator KB 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 KB 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 по каналу резервирования.

- Чтобы отключить ведение системного журнала, выполните:

  ```
  hostname# machine set loghost null
  ```



Внимание! Отключать ведение журнала не рекомендуется.

- Чтобы изменить уровень детализации 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 детализации, выбранный по умолчанию, не рекомендуется. При повышении уровня детализации необходимо учитывать, что размер файла журнала будет быстро расти, вследствие чего, будет чаще производиться его ротация при достижении максимального размера.

- Чтобы сообщения о работе службы выводились на консоль командного интерпретатора, выполните:

  `hostname# debug on <facility level>`,

  указав выбранные значения параметров `facility` и `level`.

  Параметры `facility` и `level` 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:

    ```
    hostname# debug on daemon debug
    ```

  - В конфигурационном файле другой службы в секции `[debug]` задайте параметры протоколирования, указав другое значение `facility`, например:

    ```
    [debug]
    debuglogfile = syslog:local0.debug
    debuglevel = 3
    ```

    И на другом терминале выполните:

    ```
    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:

```
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 системного журнала на USB-носитель

1 На ViPNet Coordinator KB выполните:

```
hostname# admin export logs usb
```

При выполнении команды журнал архивируется в файл экспорта `logs_<stamp>.tar.gz`, где `<stamp>` автоматически формируется из идентификатора устройства, версии ПО с учетом номера сборки, ViPNet ID узла, названия, даты и времени создания журнала.



Примечание. Чтобы прервать экспорт во время архивирования, нажмите **Ctrl+C**.

Если на внутреннем накопителе есть сохраненные журналы СКЗИ, то будет предложено задать пароль этим журналам.

2 Подключите USB-носитель к USB-разъему ViPNet Coordinator KB и нажмите **Enter**.

3 В ответ на сообщение `Select target partition [1-X] or 0 to abort:` введите номер раздела USB-носителя, на который будет записан файл архив журналов и нажмите **Enter**.
`X` — количество обнаруженные на USB-носителе разделов.

Чтобы отменить операцию, для этого введите `0` и нажмите **Enter**.

Просмотр журнала приема и применения обновлений

Если администратор сети ViPNet вносит какие-либо изменения в структуру сети ViPNet или настройки отдельных сетевых узлов, например, создает новые связи между ними, то автоматически изменяются справочники и ключи узла ЦУС.

Обновления справочников и ключей узла ЦУС создаются администратором сети ViPNet с помощью ПО ViPNet Administrator и могут быть автоматически отправлены на сетевые узлы ViPNet, которых коснулись изменения.

На ViPNet Coordinator KB поступившие обновления справочников и ключей узла принимаются в автоматическом режиме. Информация о принятых обновлениях записывается в журнал обновления справочников и ключей. В этот журнал записывается время приема, тип и результат обновления.

Для просмотра журнала приема и применения обновлений справочников и ключей выполните:

```
hostname# iplir show keys-upgrade-log
```

Просмотр журнала переключений кластера горячего резервирования

Журнал переключений кластера горячего резервирования регистрирует события старта, остановки и смены режима узла кластера.

Для просмотра журнала выполните:

`hostname> failover view <дата начала> <дата окончания>`, указав интервал времени, в котором необходимо просмотреть записи переключении кластера горячего резервирования.

Просмотр журнала СКЗИ

В журнале СКЗИ регистрируются события, связанные с выполнением ViPNet Coordinator KB криптографических функций.

В процессе работы ViPNet Coordinator KB в журнале СКЗИ регистрируются следующие события:

- начальная инициализация ДСЧ ДНСД;
- ввод ключей контроля целостности (РККЦ, ПККЦ, ТККЦ);
- проведение диагностического контроля ДНСД;
- запись эталонных значений контрольных сумм СПО ViPNet Coordinator KB;
- ввод ключей ключевого блокнота ДСДР, в том числе автоматическая смена и попытка ввода устаревших и недействительных серий;
- факт аутентификации пользователя (см. Аутентификация пользователя на ViPNet Coordinator KB на стр. 19);
- факт включения режима администратора (см. Включение режима администратора ViPNet Coordinator KB на стр. 20);
- проведение контроля целостности;
- срабатывание датчика защиты от несанкционированного доступа;
- снижение заряда батареи до критического значения;
- вызов функции экстренного стирания справочников, ключей и ключевого блокнота ДСДР;
- отклонение напряжения в электрической сети за допустимые значения;
- архивирование системного журнала;
- очистка системного журнала.

Для просмотра журнала СКЗИ выполните:

```
hostname# admin show dnsd log
```

Примеры протоколируемых событий, связанных с выполнением ViPNet Coordinator KB криптографических функций, приведены в приложении (см. События журнала СКЗИ на стр. 308).

16

Мониторинг ViPNet Coordinator KB

Мониторинг с помощью программного комплекса ViPNet StateWatcher 247

Мониторинг по протоколу SNMP 248

Мониторинг с помощью программного комплекса ViPNet StateWatcher

Вы можете осуществлять мониторинг работы ViPNet Coordinator KB с помощью программного комплекса мониторинга защищенных сетей ViPNet StateWatcher (далее ViPNet StateWatcher). Данный комплекс предназначен для наблюдения за состоянием узлов сети ViPNet (в том числе ViPNet Coordinator KB). ViPNet StateWatcher также собирает информацию о состоянии установленных на узлах компонентов ПО ViPNet, о текущей загруженности аппаратного обеспечения узлов, о состоянии сетевых интерфейсов, о статусе связей между узлами.

В случае возникновения сбоев или неполадок на узлах, вы будете оперативно проинформированы об этом. Вы можете настроить удобный вам способ оповещения о событиях: непосредственно в веб-интерфейсе ViPNet StateWatcher (визуальные и звуковые оповещения, оповещения на карте и информационной панели), по электронной почте (в том числе ViPNet Деловая почта), по SMS, по протоколу Syslog. События на узлах выявляются на основе анализа их состояния специальными правилами. Вы можете использовать встроенные правила анализа или создать собственные правила, с помощью которых будут анализироваться важные для вас параметры узлов.

Подробнее о работе с ViPNet StateWatcher см. в документе «Программный комплекс мониторинга защищенных сетей ViPNet StateWatcher. Сервер мониторинга. Руководство администратора».

Мониторинг по протоколу SNMP

С помощью SNMP-агента, входящего в состав ViPNet Coordinator KB, вы можете удаленно получать информацию о времени работы ViPNet Coordinator KB, активности сетевых интерфейсов и количестве пропущенных и блокированных IP-пакетов на них, а также уведомлять удаленную систему управления сетью (Network Management System, NMS) о наиболее важных событиях в работе ViPNet Coordinator KB (например, выходе из строя активного узла при работе ViPNet Coordinator KB в режиме кластера горячего резервирования).



Примечание. ViPNet Coordinator KB поддерживает протокол SNMP версий 1 и 2c. Параметры протокола `rocommunity` и `trapcommunity` по умолчанию имеют значение `public`.

При работе ViPNet Coordinator KB в режиме кластера горячего резервирования в используемую NMS передается информация только о работе активного узла кластера. Слежение за работой пассивного сервера кластера не производится.

Для получения информации о сетевых узлах ViPNet по протоколу SNMP необходимо выделить компьютер и установить на него ПО системы управления сетью, например WhatsUp Gold. Чтобы получение информации было возможно, импортируйте файл `VIPNET-MIB.txt`, в котором описаны используемые ПО ViPNet объекты SNMP, в NMS (подробнее см. в документации по используемой NMS).

В результате в систему управления локальной сетью будет приходить информация о состоянии ViPNet Coordinator KB. Для получения этой информации должна использоваться ветка:

```
.iso.org.dod.internet.private.enterprises.infotecs.vipnet (.1.3.6.1.4.1.10812.1)
```

Опрос ViPNet Coordinator KB по протоколу SNMP рекомендуется выполнять с защищенных узлов ViPNet. Соответствующие сетевые фильтры по умолчанию включены в список фильтров защищенной сети. Если опрос будет осуществляться с открытого узла, то в список локальных фильтров открытой сети необходимо добавить аналогичные разрешающие сетевые фильтры, указав в них IP-адрес этого узла.



Примечание. Чтобы SNMP-запросы обрабатывались корректно, необходимо отправлять их на IP-адрес ViPNet Coordinator KB, который доступен с опрашивающего узла по кратчайшему маршруту.

ViPNet Coordinator KB поддерживает базу управляющей информации (MIB) для ПО ViPNet и некоторые стандартные базы управляющей информации (см. Поддерживаемые базы управляющей информации SNMP на стр. 253). В случае, если на узле, выполняющем мониторинг ViPNet Coordinator KB, отсутствуют необходимые файлы базы MIB, их необходимо загрузить.

По умолчанию SNMP-агент выключен и не запускается автоматически при загрузке ViPNet Coordinator KB. При необходимости вы можете изменить параметры запуска SNMP-агента:

- Чтобы включить автоматический запуск SNMP-агента, выполните:

```
hostname# inet snmp mode {on | off}
```

- Чтобы запустить или остановить SNMP-агент, выполните:

```
hostname# inet snmp {start | stop}
```

Чтобы использовать оповещения (SNMP Traps), выполните настройку следующих параметров:

- Настройте IP-адрес удаленного узла, на который будут отправляться асинхронные оповещения о различных событиях при работе ПО ViPNet:

```
hostname# inet snmp trapsink <IP-адрес>
```



Примечание. Выключить использование оповещений SNMP Traps вы можете с помощью команды:

```
hostname# inet snmp trapsink null
```

- Чтобы в системе управления локальной сетью при приеме оповещений SNMP Traps проводилась авторизация, задайте пароль:

```
hostname# inet snmp community trap
```

- При необходимости задайте пароль доступа к SNMP-параметрам вашего ViPNet Coordinator KB:

```
hostname# inet snmp community ro
```



Примечание. Пароли должны содержать от 6 до 18 символов. В паролях вы можете использовать символы латинского алфавита, цифры, а также следующие специальные символы: «.», «*», «/», «-», «:», «_», «=», «@», «&», «(», «)», «,».

Просмотреть текущие значения паролей вы можете с помощью команды:

```
hostname# inet show snmp community
```

Информация о работе SNMP-агента записывается в системный журнал. Чтобы настроить уровень детализации записываемой информации, выполните:

```
hostname# inet snmp debug-level <уровень детализации>
```

В качестве уровня детализации вы можете задать следующие значения:

- `info` — записывается только информация, касающаяся инициализации SNMP-агента.
- `debug` — записывается служебная информация, используемая при отладке.
- `error` (по умолчанию) — записываются ошибки, после которых SNMP-агент может продолжать работу. Этот уровень задан по умолчанию.
- `critical` — записываются критические ошибки, после которых SNMP-агент не может продолжить работу.
- `off` — запись выключена.

Чтобы получить информацию о текущем состоянии SNMP-агента, IP-адресе удаленного узла, на который отправляются оповещения (SNMP Traps), а также о заданном уровне детализации, выполните:

```
hostname> inet show snmp
```

Описание SNMP-параметров для ПО ViPNet Coordinator KB

ViPNet Coordinator KB использует следующие общие параметры мониторинга ViPNet, для которых возможно только чтение данных:

- .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 KB (обнуляется каждый раз при перезапуске управляющей службы).
- .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.
- .1.3.6.1.4.1.10812.1.2.2 — последовательность объектов, описывающих сетевые интерфейсы.
- .1.3.6.1.4.1.10812.1.2.2.1 — объекты, описывающие сетевые интерфейсы:
  - .1.3.6.1.4.1.10812.1.2.2.1.1 — номер сетевого интерфейса;
  - .1.3.6.1.4.1.10812.1.2.2.1.2 — системное имя сетевого 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 — число пропущенных входящих нешифрованных пакетов;
  - .1.3.6.1.4.1.10812.1.2.2.1.5 — число блокированных входящих нешифрованных пакетов;
  - .1.3.6.1.4.1.10812.1.2.2.1.6 — число пропущенных исходящих нешифрованных пакетов;
  - .1.3.6.1.4.1.10812.1.2.2.1.7 — число блокированных исходящих нешифрованных пакетов;
  - .1.3.6.1.4.1.10812.1.2.2.1.8 — число пропущенных входящих шифрованных пакетов;
  - .1.3.6.1.4.1.10812.1.2.2.1.9 — число блокированных входящих шифрованных пакетов;
  - .1.3.6.1.4.1.10812.1.2.2.1.10 — число пропущенных исходящих шифрованных пакетов;
  - .1.3.6.1.4.1.10812.1.2.2.1.11 — число блокированных исходящих шифрованных пакетов.
- .1.3.6.1.4.1.10812.1.7 — последовательность объектов, описывающих состояние служб `iplircfg`, `failoverd`, `mftpd`, `algd`.
- .1.3.6.1.4.1.10812.1.7.1 — объекты, описывающие состояние служб ViPNet Coordinator KB:
  - .1.3.6.1.4.1.10812.1.7.1.1 — индекс службы;
  - .1.3.6.1.4.1.10812.1.7.1.2 — имя службы:
    - `Alg` — служба `algd`,

        - `Failover` — служба системы защиты от сбоев `failoverd`,
        - `Iplir` — управляющая служба `iplircfg`,
        - `Mftp` — служба транспортного сервера `mftpd`;
    - .1.3.6.1.4.1.10812.1.7.1.3 — состояние службы:
        - `unknown(0)` — состояние неизвестно,
        - `shutdown(1)` — работа службы завершена,
        - `running(2)` — служба работает;
    - .1.3.6.1.4.1.10812.1.7.1.4 — время работы службы.
- .1.3.6.1.4.1.10812.4.3.1 — версия агента мониторинга.
- .1.3.6.1.4.1.10812.4.3.2 — уникальный идентификатор установки агента мониторинга.
- .1.3.6.1.4.1.10812.4.4 — таблица плагинов агента мониторинга.
- .1.3.6.1.4.1.10812.4.5 — таблица функциональных возможностей агента мониторинга.
- .1.3.6.1.4.1.10812.4.20.5 — уникальный идентификатор устройства.
- .1.3.6.1.4.1.2021.13.16.2.1 — таблица температурных датчиков сетевого узла:
    - .1.3.6.1.4.1.2021.13.16.2.1.1.A — индекс температурного датчика;
    - .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 (град. C).
- .1.3.6.1.4.1.2021.13.16.3.1 — таблица датчиков скорости вращения вентиляторов:
    - .3.6.1.4.1.2021.13.16.3.1.1.B — индекс датчика скорости вращения вентилятора;
    - .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 (об/мин). Если вентилятор не подключен, параметр равен 0.

Дополнительные параметры мониторинга, специфичные для ViPNet Coordinator KB:

- .1.3.6.1.4.1.10812.5.4.1 — статус пароля администраторов (принимает значение 1 если срок действия истекает).
- .1.3.6.1.4.1.10812.5.4.2 — статус ключевой информации ДСДР (принимает значение 1 если срок действия истекает):
    - `dsdrIsOk(0)` — текущая серия ДСДР валидна.
    - `dsdrExpireSoon(1)` — срок действия серии ДСДР истекает менее чем через 6 месяцев.
    - `dsdrAreExpired(2)` — срок действия серии ДСДР истек.
    - `dsdrNotStarted(3)` — текущая серия ДСДР не введена в эксплуатацию, будет введена в соответствии с настройками ЦУСа.
    - `dsdrWereNotDeployed(4)` — не проведена первичная инициализация ключей ДСДР.

    - `forceAutochange(5)` — срок действия текущей серии ДСДР истекает менее чем через 6 месяцев. Автоматическое смена произойдет в соответствии с настройками ЦУСа.
    - `autochangeExpected(6)` — срок действия текущей серии ДСДР истек. Автоматическое смена произойдет в соответствии с настройками ЦУСа.
    - `dsdrCheckError(7)` — проверка состояния ключей ДСДР завершилась с ошибкой.
- .1.3.6.1.4.1.10812.12.1.1 — версия ДНСД.
- .1.3.6.1.4.1.10812.12.2.1.1.4 — время работы службы control_pack, сек.

ViPNet Coordinator KB использует следующие оповещения SNMP (SNMP Traps):

- .1.3.6.1.4.1.10812.1.0.1 — запуск управляющей службы `iplircfg`.
- .1.3.6.1.4.1.10812.1.0.2 — завершение работы управляющей службы `iplircfg`.
- .1.3.6.1.4.1.10812.1.0.3 — запуск службы `mftpd`.
- .1.3.6.1.4.1.10812.1.0.4 — завершение службы `mftpd`.
- .1.3.6.1.4.1.10812.1.0.7 — запуск службы `failoverd` (только в активном режиме).
- .1.3.6.1.4.1.10812.1.0.8 — завершение работы службы `failoverd` (только в активном режиме).
- .1.3.6.1.4.1.10812.1.0.9 — переключение службы `failoverd` из пассивного режима в активный.
- .1.3.6.1.4.1.10812.12.0.3 — состояние датчика вскрытия корпуса.
- .1.3.6.1.4.1.10812.12.0.4 — снижение уровня заряда элемента питания ДНСД.
- .1.3.6.1.4.1.10812.12.0.5 — критический уровень заряда элемента питания ДНСД.
- .1.3.6.1.4.1.10812.12.0.6 — выход уровня напряжения питания ДНСД за пределы допустимых значений.
- .1.3.6.1.4.1.10812.12.0.7 — нажатие кнопки экстренного стирания ключей.
- .1.3.6.1.4.1.10812.12.0.8 — инициация перезагрузки ViPNet Coordinator KB модулем ДНСД.
- .1.3.6.1.4.1.10812.12.0.9 — ошибка прохождения диагностического теста.
- .1.3.6.1.4.1.10812.12.0.10 — ошибка в модуле ДНСД.
- .1.3.6.1.4.1.10812.12.0.11 — ошибка регламентного контроля.



Примечание. Как правило, NMS позволяет настроить ответное действие при возникновении каждого из этих оповещений (например, отправка письма). Подробнее см. в документации по используемой системе управления локальной сетью.

Поддерживаемые базы управляющей информации SNMP

ViPNet Coordinator KB частично поддерживает следующие стандартные базы управляющей информации (MIB):

- MIB-II System (.1.3.6.1.2.1.1) — база управляющей информации для системных параметров. Включает в себя объекты, предназначенные для получения информации о различных системных параметрах ViPNet Coordinator KB (например, имя сетевого узла, название ОС, название аппаратной платформы, количество и параметры сетевых интерфейсов). Подробнее см. в RFC 1213.
- IF-MIB (.1.3.6.1.2.1.2) — база управляющей информации для сетевых интерфейсов ViPNet Coordinator KB. Включает в себя объекты, предназначенные для получения информации о различных параметрах сетевых интерфейсов ViPNet Coordinator KB (например, тип сетевого интерфейса, физический адрес, скорость). Подробнее см. в RFC 2863.
- IP-MIB (.1.3.6.1.2.1.4) — база управляющей информации для протокола IP. Включает в себя объекты, предназначенные для получения информации о различных характеристиках работы ViPNet Coordinator KB по протоколу IP (например, параметры сетевых интерфейсов, статистика для IP-трафика, таблица IP-адресов). Подробнее см. в RFC 4293.
- TCP-MIB (.1.3.6.1.2.1.6) — база управляющей информации для протокола TCP. Включает в себя объекты, предназначенные для получения информации о различных характеристиках работы ViPNet Coordinator KB по протоколу TCP (например, максимальный размер сегмента, количество активных соединений). Подробнее см. в RFC 4022.
- UDP-MIB (.1.3.6.1.2.1.7) — база управляющей информации для протокола UDP. Включает в себя объекты, предназначенные для получения информации о различных характеристиках работы ViPNet Coordinator KB по протоколу UDP (например, общее количество полученных и отправленных датаграмм). Подробнее см. в RFC 4113.
- HOST-RESOURCES-MIB (.1.3.6.1.2.1.25) — база управляющей информации, которая включает в себя объекты, предназначенные для получения информации о системных параметрах ViPNet Coordinator KB. Подробнее см. RFC 2790. На данный момент поддерживаются следующие системные параметры:
- HOST-RESOURCES-MIB (.1.3.6.1.2.1.25.1.1.0) hrSystemUptime — длительность работы ViPNet Coordinator KB с момента последнего запуска.
- UCD-SNMP-MIB (.1.3.6.1.4.1.2021) — база управляющей информации для протокола SNMP. Включает в себя объекты, предназначенные для получения информации о состоянии компьютера (например, состояние оперативной памяти и жесткого диска). Подробнее см. на веб-ресурсе.
- SNMP-FRAMEWORK-MIB (.1.3.6.1.6.3.10) — включает в себя параметры для получения информации о динамически загружаемых модулях SNMP в ViPNet Coordinator KB. Подробнее см. на веб-ресурсе.

- 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. Подробнее см. на веб-ресурсе.

A

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 KB могут различаться для разных групп OID в описывающих их таблицах. При настройке мониторинга по протоколу SNMP рекомендуется вместо индексов использовать названия (логические имена) сетевых интерфейсов.

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 11. SNMP-параметры. MIB-II System

OID	Описание
.1.3.6.1.2.1.1.1.0	Информация о сетевом узле (имя узла, информация об ОС).
.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 `inet snmp system contact`. Если информация не задана, параметр отсутствует.
.1.3.6.1.2.1.1.5.0	Имя, назначенное сетевому узлу. Задается командой `inet snmp system name`. Если имя не задано, параметр равен `hostname`.
.1.3.6.1.2.1.1.6.0	Место расположения сетевого узла. Задается командой `inet snmp system location`. 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 sysUpTime 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 связка 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 (bbnSpfIgp) — 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 proxy.

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 KB могут различаться для разных групп OID в описывающих их таблицах. При настройке мониторинга по протоколу SNMP рекомендуется вместо индексов использовать названия (логические имена) сетевых интерфейсов.

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 12. 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 (.1.3.6.1.2.1.2.2.1):	
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
.1.3.6.1.2.1.31.1.1.1.17.A	Наличие физического коннектора у интерфейса: • 1 (true) — есть коннектор (для ethernet), • 2 (false) — нет коннектора (для localhost).
.1.3.6.1.2.1.31.1.1.1.18.A	Имя интерфейса, заданное администратором. Для координатора всегда равно 0.
.1.3.6.1.2.1.31.1.1.1.19.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 13. 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, отклонённых по причине неизвестного или неподдерживаемого протокола.
1.3.6.1.2.1.4.8.0	Количество отклоненных пактов 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 сборок 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.
1.3.6.1.2.1.4.20.1.5.E	Максимальный размер входящего фрагментированного IP-пакета, который сетевой узел может собрать на этом интерфейсе.

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-MAC.
.1.3.6.1.2.1.4.22.1.2.A.E	MAC-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.

OID	Описание
.1.3.6.1.2.1.4.23.0	Количество отклоненных маршрутов.
.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.
.1.3.6.1.2.1.4.31.3.1.	Таблица, содержащая статистику (счетчики) IP-трафика различных версий на сетевом узле для каждого интерфейса. Аналогична таблице .1.3.6.1.2.1.4.31.1.1, но содержит счетчики для каждого интерфейса.

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 ближайшему узлу, • 3 (broadcast) — широковещательный.
.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 .1.3.6.1.2.1.2.2.1.1 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 (probeprobe) — 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.

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 14. SNMP-параметры. TCP-MIB

OID	Описание
.1.3.6.1.2.1.6.1.0	Алгоритм определения таймаута для повторной передачи неопознанных октетов.
.1.3.6.1.2.1.6.2.0	Минимальное допустимое значение таймаута для повторной передачи (в миллисекундах).
.1.3.6.1.2.1.6.3.0	Максимальное допустимое значение таймаута для повторной передачи (в миллисекундах).
.1.3.6.1.2.1.6.4.0	Максимальное количество TCP-соединений. При динамическом определении, параметр равен -1.
.1.3.6.1.2.1.6.5.0	Количество прямых переходов TCP-соединений из состояния CLOSED в состояние SYN-SENT.

OID	Описание
.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 + 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-соединении, 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,

OID	Описание
	• 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.
.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 15. SNMP-параметры. UDP-MIB

OID	Описание
.1.3.6.1.2.1.7.1.0	Количество входящих UDP-датаграмм.

OID	Описание
.1.3.6.1.2.1.7.2.0	Количество входящих UDP-датаграмм, для которых нет открытого порта.
.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 16. 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 исполняемых 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.
Таблица устройств сетевого узла (.1.3.6.1.2.1.25.3.2.1):	
B — индекс устройства	
.1.3.6.1.2.1.25.3.2.1.1.B	Идентификатор, присвоенный сетевым узлом (B).
.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).

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 17. 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 (swap) 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).
.1.3.6.1.4.1.2021.4.11.0	Общий объем доступной памяти (RAM + swap) (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 кэшированной памяти.
.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)	
B — 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).
.1.3.6.1.4.1.2021.9.1.10.B	Процент использования inodes в разделе.

OID	Описание
.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 2ТБайт, 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 2ТБайт, 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 2ТБайт, 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 2ТБайт, 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 2ТБайт, 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 2ТБайт, 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).
.1.3.6.1.4.1.2021.10.1.6.C	Загрузка системы в процентах (в виде числа с плавающей точкой).

OID	Описание
.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).
.1.3.6.1.4.1.2021.11.58.0	Количество блоков, полученное блочным устройством (диском).

OID	Описание
.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.

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 18. SNMP-параметры. SNMP-FRAMEWORK-MIB

OID	Описание
.1.3.6.1.6.3.10.2.1.1.0	Идентификатор SNMP-engine.

OID	Описание
.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).
.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 19. 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 KB (сбрасывается каждый раз при перезапуске службы `iplircfg`).
.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 блокированных 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 блокированных исходящих незашифрованных пакетов.
.1.3.6.1.4.1.10812.1.2.2.1.8	Количество пропущенных входящих зашифрованных пакетов.

OID	Описание
.1.3.6.1.4.1.10812.1.2.2.1.9	Количество блокированных входящих зашифрованных пакетов.
.1.3.6.1.4.1.10812.1.2.2.1.10	Количество пропущенных исходящих зашифрованных пакетов.
.1.3.6.1.4.1.10812.1.2.2.1.11	Количество блокированных исходящих зашифрованных пакетов.
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 служб: • 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	Таблица объектов, описывающих состояние службы `failoverd`, при включенном режиме кластера горячего резервирования.

SNMP-агент ViPNet Coordinator KB может отправлять оповещения:

Таблица 20. SNMP-оповещения. VIPNET-MIB

OID	Описание
.1.3.6.1.4.1.10812.1.0.1	Запуск службы `iplircfg`.
.1.3.6.1.4.1.10812.1.0.2	Остановка службы `iplircfg`.
.1.3.6.1.4.1.10812.1.0.3	Запуск службы `mftpd`.
.1.3.6.1.4.1.10812.1.0.4	Остановка службы `mftpd`.
1.3.6.1.4.1.10812.1.0.5	Запуск службы `alg`.
1.3.6.1.4.1.10812.1.0.6	Остановка службы `alg`.
.1.3.6.1.4.1.10812.1.0.7	Запуск службы `failoverd`.
.1.3.6.1.4.1.10812.1.0.8	Остановка службы `failoverd`.
.1.3.6.1.4.1.10812.1.0.9	Переключение службы `failoverd` из пассивного режима в активный.
.1.3.6.1.4.1.10812.1.0.10	Запуск службы `webgui`.
.1.3.6.1.4.1.10812.1.0.11	Остановка службы `webgui`.

VIPNETDNSD-MIB

VIPNETDNSD-MIB (.1.3.6.1.4.1.10812.12) включает в себя объекты, предназначенные для получения информации о параметрах ДНСД. Для получения этой информации используется ветвь: `.iso.org.dod.internet.private.enterprises.infotecs.vipnetdnsd` (.1.3.6.1.4.1.10812.12.1).

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 21. SNMP-параметры. VIPNETDNSD-MIB

OID	Описание
.1.3.6.1.4.1.10812.12.1.1.0	Версия прошивки ДНСД
.1.3.6.1.4.1.10812.12.1.3.0	Статус напряжения батареи
.1.3.6.1.4.1.10812.12.2.1.1.2	Наименование службы
.1.3.6.1.4.1.10812.12.2.1.1.3	Статус службы
.1.3.6.1.4.1.10812.12.2.1.1.4	Время работы службы

SNMP-агент ViPNet Coordinator KB может отправлять оповещения:

Таблица 22. SNMP-оповещения. VIPNETDNSD-MIB

OID	Описание
.1.3.6.1.4.1.10812.12.0.1	Запуск службы
.1.3.6.1.4.1.10812.12.0.2	Остановка службы
.1.3.6.1.4.1.10812.12.0.3	Срабатывание датчика вскрытия
.1.3.6.1.4.1.10812.12.0.4	Низкий заряд батареи (ниже 3,3 В)
.1.3.6.1.4.1.10812.12.0.5	Критический заряд батареи (ниже 3,0 В) **Внимание!** Разряженная батарея может стать причиной неработоспособности ViPNet Coordinator KB. Обязательно контролируйте уровень заряда и своевременно меняйте батарею
.1.3.6.1.4.1.10812.12.0.6	Выход контролируемых номиналов питания за пределы допустимого диапазона
.1.3.6.1.4.1.10812.12.0.7	Срабатывание кнопки экстренного стирания
.1.3.6.1.4.1.10812.12.0.8	Выдана команда на перезагрузку хоста
.1.3.6.1.4.1.10812.12.0.9	Ошибка диагностического контроля
.1.3.6.1.4.1.10812.12.0.10	Ошибка ДНСД
.1.3.6.1.4.1.10812.12.0.11	Ошибка периодического контроля

VIPNETKB-MIB

VIPNETDNSD-MIB (.1.3.6.1.4.1.10812.12) 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.vipnetdnsd` (.1.3.6.1.4.1.10812.12).

ПО ViPNet использует следующие объекты, для которых возможно только чтение данных:

Таблица 23. SNMP-параметры. VIPNETKB-MIB

OID	Описание
.1.3.6.1.4.1.10812.5.4.1	Статус пароля администраторов. Принимает значение 1 если срок действия пароля истекает.
.1.3.6.1.4.1.10812.5.4.2	Статус серии ДСДР: • 0 (`dsdrIsOk`) — текущая серия ДСДР действительна • 1 (`dsdrExpireSoon`) — срок действия серии ДСДР истекает менее чем через 6 месяцев • 2 (`dsdrAreExpired`) — срок действия серии ДСДР истек • 3 (`dsdrNotStarted`) — текущая серия ДСДР не введена в эксплуатацию, будет введена в соответствии с настройками ЦУСа • 4 (`dsdrWereNotDeployed`) — не проведена первичная инициализация ключей ДСДР • 5 (`forceAutochange`) — срок действия текущей серии ДСДР истекает менее чем через 6 месяцев. Автоматическая смена произойдет в соответствии с настройками ЦУСа • 6 (`autochangeExpected`) — срок действия текущей серии ДСДР истек. Автоматическая смена произойдет в соответствии с настройками ЦУСа • 7 (`dsdrCheckError`) — проверка состояния ключей ДСДР завершилась с ошибкой

SNMP-агент ViPNet Coordinator KB может отправлять оповещения:

Таблица 24. SNMP-оповещения. VIPNETKB-MIB

OID	Описание
.1.3.6.1.4.1.10812.5.0.1	Запуск службы `control_pack`
.1.3.6.1.4.1.10812.5.0.2	Остановка службы `control_pack`

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 25. 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.
.1.3.6.1.4.1.10812.4.20.5	Идентификатор устройства.
.1.3.6.1.4.1.10812.4.20.9	Идентификатор устройства в виде XXXXXXX-XXXXXXX-XXXXXXX-XXXXXXX-XXXXXXX, где каждая группа XXXXXXX сопоставляется с группе оборудования: **1** Системная информация **2** Информация об основной плате **3** Информация о хранилищах **4** Информация об оперативной памяти **5** Информация о сетевых устройствах

B

Сетевые фильтры по умолчанию

Сетевые фильтры, в том числе фильтры системы защиты от сбоев, создаваемые в ViPNet Coordinator KB по умолчанию, перечислены в таблицах ниже.

Таблица 26. Фильтры защищенной сети (`vpn`)

Название	Источник	Назначение	Протокол	Действие
В режиме кластера (нередактируемый)				
ViPNet Service	@any	@any	udp: from 2060 to 2060	pass
Общие сетевые фильтры				
Block not original udp port	@local	@any	udp: from 0-2045, from 2047-65535, to 2046	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, from 68 to 67	pass
Allow DHCP-Relay service	@any	@any	udp: from 67 to 67	pass
Allow ViPNet DBViewer	@any	@any	tcp: to 2047	pass
Allow ViPNet StateWatcher	@any	@any	tcp: to 5100, 10092	pass
Allow ViPNet MFTP	@any	@any	tcp: to 5000-5003	pass

Название	Источник	Назначение	Протокол	Действие
Allow ViPNet WebGUI	@any	@local	tcp: to 8080	pass
Allow ICMP Ping	@any	@any	icmp8	pass
Allow DNS	@any	@any	tcp/udp: to 53	pass
Allow NTP	@any	@any	udp: to 123	pass
Allow UPS service	@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 сетевой фильтр (нередактируемый). Данный фильтр имеет самый низкий приоритет и действует для трафика, не попадающего под действие всех остальных сетевых фильтров.				
Block All Traffic	@any	@any	@any	drop

Таблица 27. Локальные фильтры открытой сети (`local`)

Название	Источник	Назначение	Протокол	Действие
В одиночном режиме системы защиты от сбоев, в активном и пассивном режимах кластера (нередактируемые)				
ViPNet Service CommonIn	@any	@local	tcp/udp: to 2046, 2047, 10096, 5100, 10092	drop
ViPNet Service CommonOut	@local	@any	tcp/udp: from 2046	drop
В активном режиме кластера (нередактируемые)				
Failover test IP	@local	<список значений параметров `testip` из `failover.ini`>	icmp	pass
Failover Channel	@local	iface <значение параметра `device` из `failover.ini`>	@any	pass
Failover Channel	<значение параметра `activeip` второго сервера из `failover.ini`>	iface <значение параметра `device` из `failover.ini`>	@any	pass
В пассивном режиме кластера (нередактируемые)				

Название	Источник	Назначение	Протокол	Действие
Failover Channel	@local	iface <значение параметра `device` из `failover.ini`>	@any	pass
Failover Channel	<значение параметра `activeip` второго сервера 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
Общие сетевые фильтры				
Allow DHCP service	@any	@any	udp: from 67 to 68, from 68 to 67	pass
Allow DHCP-Relay service	@any	@any	udp: from 67 to 67	pass
Allow ICMP Ping	@local	@any	icmp8	pass
Allow DNS	@local	@any	tcp/udp: to 53	pass
Allow NTP	@local	@any	udp: to 123	pass
Блокирующий фильтр (нередактируемый)				
Block All Traffic	@any	@any	@any	drop

Таблица 28. Транзитные фильтры открытой сети (`forward`)

Название	Источник	Назначение	Протокол	Действие
В пассивном режиме кластера				
Block all forward traffic	@any	@any	@any	drop
Блокирующий фильтр (нередактируемый)				
Block All Traffic	@any	@any	@any	drop

Таблица 29. Фильтры туннелируемых узлов (`tunnel`)

Название	Источник	Назначение	Протокол	Действие
В пассивном режиме кластера (нередактируемый)				

Название	Источник	Назначение	Протокол	Действие
Block all tunnel traffic	@any	@any	@any	drop
Общий сетевой фильтр				
To all tunnel nodes	@any	@tunneledip	@any	pass
From all tunnel nodes	@tunneledip	@any	@any	pass
Блокирующий фильтр (нередактируемый)				
Block All Traffic	@any	@any	@any	drop

C

Пользовательские группы протоколов по умолчанию

По умолчанию в ViPNet Coordinator KB настроены пользовательские группы протоколов, перечисленные в таблице ниже.

Таблица 30. Пользовательские группы протоколов, настроенные 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
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

Имя группы протоколов	Состав группы протоколов
LDAP	TCP:to 389, UDP:to 389
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
UPnP	TCP:to 1900, TCP:to 2869, UDP:to 1900, UDP:to 2869
MFTP	TCP:to 5000-5003
StateWatcher	TCP:to 2047, TCP:to 5100, TCP:to 10092
ViPNetBase	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

D

События в журнале регистрации IP-пакетов

Типы событий, регистрируемых в журнале IP-пакетов ViPNet Coordinator KB, можно поделить на следующие группы и подгруппы:

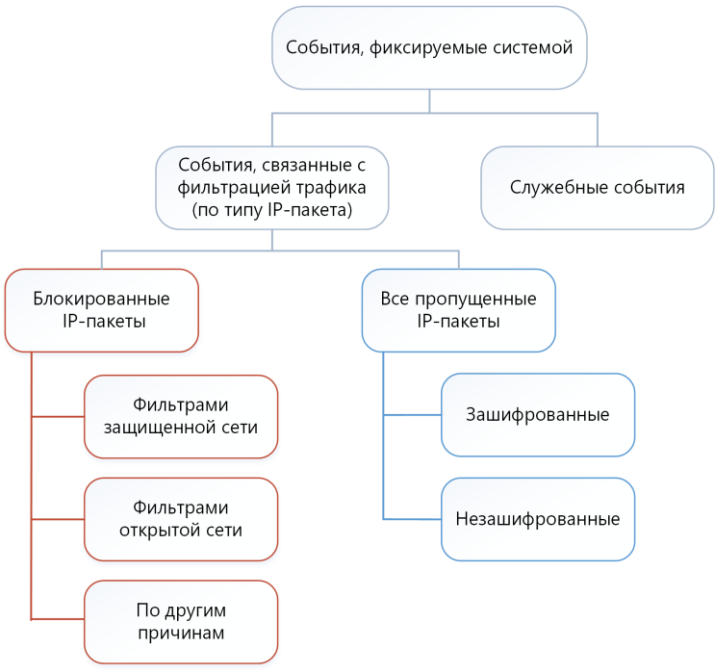


Рисунок 44. Классификация событий в журнале IP-пакетов

Описание типов событий каждой подгруппы приведено в таблицах.

*Таблица 31. События подгруппы **Все IP-пакеты/Блокированные IP-пакеты/IP-пакеты, блокированные сетевыми фильтрами защищенной сети***

Номер события	Описание события
1	Не найден ключ для передачи IP-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 IP-пакета, то есть время его отправки отличается от времени его получения на величину, большую указанной в соответствующей настройке
5	Входящий IP-пакет отправлен сетевым узлом с версией драйвера ViPNet, не совместимой с версией драйвера ViPNet на сетевом узле получателя
7	Метод шифрования, код которого указан во входящем пакете, не поддерживается
8	Недопустимые параметры в расшифрованном пакете
9	IP-пакет блокирован главным фильтром защищенной сети
10	Неизвестен идентификатор сетевого узла отправителя пакета
13	Превышено максимальное время пребывания IP-пакета в сети
14	Неверный адрес получателя пакета
15	Превышено допустимое количество одновременно обрабатываемых фрагментированных IP-пакетов
16	Исчерпана лицензия на количество туннелируемых узлов
17	Неверный адрес получателя
18	Неизвестный адрес получателя
19	Подмена узла отправителя
26	Узел с данного координатора недоступен
70	Транзитный IP-пакет заблокирован фильтром защищенной сети

*Таблица 32. События подгруппы **Все IP-пакеты/Блокированные IP-пакеты/IP-пакеты, блокированные сетевыми фильтрами открытой сети***

Номер события	Описание события
20	Заблокирован открытый пакет
21	Идентификатор отправителя пакета неизвестен
22	От защищенного узла получен открытый пакет

Номер события	Описание события
23	От защищенного узла получен открытый широковещательный пакет
24	На порт, используемый одной из служб, получен открытый пакет
30	Локальный пакет заблокирован фильтром открытой сети
31	Транзитный пакет заблокирован фильтром открытой сети
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 33. 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

Номер события	Описание события
97	Заблокирован пакет с установленным флагом DF (Don't Fragment), передача которого без фрагментации невозможна
99	Заблокирован фрагментированный IP-пакет
101	Транзитный IP-пакет не может быть маршрутизирован
102	Прикладной пакет не обработан, так как не загружен модуль обработки на прикладном уровне
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 сетевых адресов (в пуле нет свободных портов)
111	Не найден ключ обмена
112	Неверное значение имитозащитной вставки в незашифрованном пакете версии 3.0
113	Неизвестный идентификатор сетевого узла отправителя
115	Не удалось найти маршрут для IP-пакета в таблице маршрутизации
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 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

Номер события	Описание события
134	Ошибка обработки прикладных протоколов: ошибка в таблице соответствия между прикладными сервисами и сетевыми протоколами, портами
135	Невозможно правильно разобрать пакет DNS-запроса
137	Шифрование исходящего IP-пакета с использованием данного алгоритма запрещено (тесты алгоритма не прошли)
138	Запрос на публичный DNS-сервер заблокирован: запрашивается имя корпоративного ресурса
139	Запрос на защищенный DNS-сервер заблокирован: DNS-сервер не поддерживает разрешение имен из данной доменной зоны
147	Пакет неизвестного протокола заблокирован параметром `block-other-protocols`

*Таблица 34. 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 широковещательный 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 35. 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 широковещательный 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 для незарегистрированного узла ViPNet

*Таблица 36. 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 сетевых узлов, которые работают через межсетевой экран с динамической трансляцией сетевых адресов
48	Сетевой узел доступен по широковещательному адресу
49	Изменился IP-адрес собственного сетевого узла
110	Новый IP-адрес сетевого узла зарегистрирован на DNS-сервере

E

События системного журнала, связанные с аутентификацией и настройкой оборудования

Ниже приведены протоколируемые события ViPNet Coordinator KB, связанные с аутентификацией и настройкой оборудования, и соответствующие им записи в системном журнале.

Таблица 37. Протоколируемые события операционной системы Linux

Событие	Пример записей в журнале
Запуск операционной системы	`vmunix: Linux version 4.4.163`

Завершение работы операционной системы	`vmunix: [] drviplir: ViPNet IpLir driver unloaded`
	`vmunix: [] itcscrpt: ViPNet Crypto Driver unloaded`
	`vmunix: [] itcswd: ViPNet WatchDog driver unloaded`
	`vmunix: [] itcskrniface: Kernel interface driver is unloaded`
	`acpid: exiting`
	`syslogd: exiting on signal 15`

Таблица 38. События, протоколируемые при работе ViPNet Coordinator KB

Событие	Пример записей в журнале
Завершение работы операционной системы по команде	`rvpn_shell[1380]: <HALT> Command 'machine halt'`
Перезагрузка операционной системы по команде	`rvpn_shell[1356]: <REBOOT> Command: machine reboot`
Выход из командного интерпретатора	`login[20446]: pam_unix(login:session): session closed for user user` `rvpn_shell[2190]: <EXIT> Command 'exit'`
Успешный переход в режим администратора	`rvpn_shell[1334]: <ENABLE> Command 'enable' – Administrator has logged on`
Не удалось перейти в режим администратора	`rvpn_shell[2190]: <ENABLE> Command 'enable' - Failed to log on as administrator.`
Переход из режима администратора в режим пользователя	`rvpn_shell[2190]: <EXIT> Command 'exit'`
Вход в системную командную оболочку	`rvpn_shell[2190]: <I_SHELL> You have exited to the Linux system shell`
Выход из системной командной оболочки	Запись о событии не заносится в журнал
Принудительное завершение сессии командного интерпретатора	`rvpn_shell[1324]: <A_KICK> Command: 'admin kick'`

Успешное изменение конфигурационного файла `iplir.conf`	`rvpn_shell: <I_CFG> Command: iplir config - starting`
	`rvpn_shell: <I_CFG> Command: iplir config. Config file edited successfully.`
	Отображение изменения:
	`rvpn_shell: 15655c15655`
	`rvpn_shell: < visibility= auto`
	`rvpn_shell: ---`
	`rvpn_shell: > visibility= real`
	`rvpn_shell: <I_CFG> Command: iplir config - iplir.conf has been edited successfully`

Событие	Пример записей в журнале

Не удалось изменить конфигурационный файл `iplir.conf`	`rvpn_shell: <I_CFG> Command: iplir config - starting`
	`rvpn_shell: <I_CFG> Command: iplir config. Incorrect config file.`
	или
	`rvpn_shell: <I_CFG> Command: iplir config - starting`
	`rvpn_shell: <I_CFG> Command: iplir config Cannot edit config while iplir is running`

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: 'abn_000a.dst' has been successfully unpacked`
Локальное обновление ПО	`rvpn_shell[1664]: Command: admin upgrade, starting upgrade script`
Задание IP-адреса для сетевого интерфейса	`rvpn_shell[1949]: <IFCONF> Command: inet ifconfig eth2 10.0.0.1 netmask 255.255.255.0 Old ifconfig value: BROADCAST MULTICAST MTU:1500 Metric:1`
Сброс настроек сетевого интерфейса	`rvpn_shell[1949]: Interface eth2 is reset`
Включение сетевого интерфейса	`rvpn_shell[1949]: <IFCONF> The command 'inet ifconfig eth2 up' executed`
Выключение сетевого интерфейса	`rvpn_shell[1949]: <IFCONF> The command 'inet ifconfig eth2 down' executed`
Установка параметров скорости сетевого интерфейса	`rvpn_shell[1949]: <IFCONF_SPD> Command: inet ifconfig eth2 speed 10 duplex full`
Добавление статического маршрута	`rvpn_shell[1949]: <RT_ADD> Command: inet route add 123.45.67.89 netmask 255.255.255.0 gateway 192.168.32.50. Static route has been added successfully`
Удаление статического маршрута	`rvpn_shell[1949]: <RT_DEL> Command: inet route delete 123.45.67.89. Static route has been deleted successfully`
Создания виртуального интерфейса VLAN	`rvpn_shell[1949]: <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[1949]: <I_VLAN_DELETE> Command: 'inet ifconfig eth2 vlan delete 2'. The vlan interface eth2.2 has been deleted`
Сохранение копии конфигурации VPN	`rvpn_shell[1949]: <CFG_SV> Command: admin config save test_conf`
Загрузка копии конфигурации VPN	`rvpn_shell[1949]: <CFG_LD> Command: admin config load test_conf`

Событие	Пример записей в журнале
Экспорт журнала регистрации IP-пакетов на USB-носитель	`rvpn_shell[1949]: <ADM_EXP> Command: 'admin export packetdb usb'`
Экспорт справочников, ключей и настроек	`rvpn_shell[1949]: <ADM_EXP> Command: 'admin export binary-encrypted'`

Изменение сетевых фильтров и правил трансляции сетевых адресов	Отображение изменения:
	`iplircfg[21590]: User rules dataname has been changed.`
	`iplircfg[21590]: <Rules>`
	`iplircfg[21590]: + <Rule IsActive="true" xsi:type="vn:LocalRule">`
	`iplircfg[21590]: + <RuleId>100001</RuleId>`
	`iplircfg[21590]: </Rule>`

Включение вывода сообщений о событиях	`rvpn_shell[1949]: <DBG_ON> Command 'debug on -device tty1, facility daemon, level debug'`
Выключение вывода сообщений о событиях	`rvpn_shell[1949]: <DBG_OFF> Command 'debug off -device tty1'`

Таблица 39. События, связанные с запуском или остановкой служб ViPNet Coordinator KB

Событие	Пример записей в журнале
Запуск службы системы защиты от сбоев `failoverd`	`failoverd[1479]: Starting failover in single/cluster mode` `rvpn_shell[1399]: <F_STR> Command: failover start`
Завершение службы системы защиты от сбоев `failoverd`	`failoverd[1217]: Caught quit signal, exiting` `rvpn_shell[1399]: <F_STP> Command: failover stop`
Запуск управляющей службы `iplircfg`	`iplircfg[1448]: IPLIR daemon is running` `rvpn_shell[1399]: <I_STR> Command: iplir start`
Завершение работы управляющей службы `iplircfg`	`iplircfg[1363]: Exiting daemon` `rvpn_shell[1399]: <I_STP> Command: iplir stop`
Запуск транспортного модуля MFTP	`rvpn_shell[1399]: <M_STR> Command: mftp start` `mftpd[1463]: Init MFTP manager ... OK`
Завершение работы транспортного модуля MFTP	`mftpd[1229]: Exiting MFTP manager` `rvpn_shell[1399]: <M_STP> Command: mftp stop`
Перезапуск службы `algd`, отвечающего за обработку прикладных протоколов	`rvpn_shell[1399]: <ALG_RST> Command: 'alg restart'.`

Таблица 40. События, связанные с работой системы защиты от сбоев

Событие	Пример записей в журнале
Смена режима работы узла кластера из пассивного в активный	`failoverd[1284]: switching to active mode`

Перезагрузка узла кластера из-за сбоя на сетевом интерфейсе	`failoverd[1258]: failure detected on interface eth1`
	`failoverd[1258]: one or more subsystems failed, rebooting`
	`failoverd[1258]: Rebooted due to network error on eth1 at <дата и время>`

Перезагрузка узла кластера из-за конфликта режимов	`failoverd[1411]: Rebooted due conflict mode at <дата и время>`
Перезагрузка узла кластера из-за обнаружения службой `failoverd` ошибки в работе контролируемой службы	`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 <дата и время>`

Таблица 41. События, связанные с обменом служебной информацией между сетевыми узлами ViPNet

Событие	Пример записей в журнале
Получение сообщения о доступности сетевого узла ViPNet (сообщения рассылаются при включении узла, а затем периодически)	`iplircfg[13787]: [11-16 16:40:44] ProcessMonChannel: [src/ipserver.cpp]: got message: type 1 from ID 1C2000A (msg name '85FTKIK3.IAY')`
Получение сообщения о доступности сетевого узла ViPNet (сообщения рассылаются при первом обращении к серверу IP-адресов сети ViPNet)	`iplircfg[13787]: [11-16 16:40:39] ProcessMonChannel: [src/ipserver.cpp]: got message: type 3 from ID 1C2000A (msg name '773PYNG3.2HH')`
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 сетевого 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 сетевым узлом ViPNet (запрос)	`iplircfg[13787]: [11-16 16:40:44] ProcessMonChannel: [src/ipserver.cpp]: got message: type 5 from ID 1C2000A (msg name '85FTKIK3.IAY')`

Событие	Пример записей в журнале
Подтверждение получения сообщения (кроме уведомлений об изменении адреса доступа удаленного сетевого узла ViPNet)	`iplircfg[13787]: [11-16 16:40:44] ProcessMonChannel: [src/ipserver.cpp]: got message: type 7 from ID 1C2000A (msg name '85FTKIK3.IAY')`
Уведомление об изменении адреса доступа удаленного сетевого узла ViPNet	`iplircfg[13787]: [11-16 16:40:44] ProcessMonChannel: [src/ipserver.cpp]: got message: type 50 from ID 1C2000A (msg name '85FTKIK3.IAY')`
Подтверждение получения уведомления об изменении адреса доступа удаленного сетевого узла ViPNet	`iplircfg[13787]: [11-16 16:40:44] ProcessMonChannel: [src/ipserver.cpp]: got message: type 70 from ID 1C2000A (msg name '85FTKIK3.IAY')`

Изменение параметров адреса доступа удаленного сетевого 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 42. 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.`

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.`

Событие	Пример записей в журнале

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] itcscrpt: Run CSP test.` `vmunix: [ 34.573920] itcscrpt: Driver selftest: TestLinealRecurrentGenerator failed`

Проверка при запуске и регламентная проверка шифраторов в криптодрайвере	`vmunix: [ 34.206950] itcscrpt: Run CSP test.`
	`vmunix: [ 34.206957] itcscrpt: Driver selftest: TestLowStreamImit failed`
	`vmunix: [ 34.206957] itcscrpt: Driver selftest: TestLowCloseStreamImit failed`
	`vmunix: [ 34.206958] itcscrpt: Driver selftest: TestLowStreamEnCryptCFB failed`
	`vmunix: [ 34.206959] itcscrpt: Driver selftest: TestLowStreamDeCryptCFB failed`

Событие	Пример записей в журнале

Периодическая проверка ДСЧ в криптодрайвере	`vmunix: [  652.427598] itcscrpt: Driver selftest: TestLowStreamImit failed`
	`vmunix: [  652.427600] itcscrpt: Driver selftest: TestLowCloseStreamImit failed`
	`vmunix: [  652.427601] itcscrpt: Driver selftest: TestLowStreamEnCryptCFB failed`
	`vmunix: [  652.427602] itcscrpt: Driver selftest: TestLowStreamDeCryptCFB failed`
	`vmunix: [  652.427606] itcscrpt: Driver selftest: There are tests with failures.`



Примечание. Записи об успешной работе ДСЧ или шифратора не протоколируются.

F

События журнала СКЗИ

Ниже приведены протоколируемые события, связанные с выполнением криптографических функций ViPNet Coordinator KB.

Таблица 43. События, связанные выполнением криптографических функций

Событие	Пример записей в журнале
Начальная инициализация ДСЧ	`Key event [random numbers generation]`

Ввод ключей контроля целостности	`Key event [RKKC generation]`
	`Key event [PKKC generation]`
	`Key event [TKKC generation]`

Выполнение диагностического контроля ДНСД	`Diagnose check on the technological key, status: ok...`

Запись эталонных значений контрольных сумм	`Key event [checksums with TKKC generation]`
	`Key event [checksums with RKKC generation]`
	`Key event [checksums with PKKC generation]`

Установка ключевого блокнота ДСДР, в том числе автоматическая смена и попытка ввода устаревших и недействительных серий	Фактом данного события считается генерация ключей регламентного и периодического ключей контроля целостности, так как они генерируются после установки ключевого блокнота ДСДР.
	`Key event [RKKC generation]`
	`Key event [PKKC generation]`

Ввод действующего персонального ключа пользователя	`Key event [predefined generation]`
Выполнения процедуры контроля целостности (см. Контроль целостности файлов ViPNet Coordinator KB на стр. 39)	`Key event [checksums with PKKC checking]`

Событие	Пример записей в журнале
Вызов функции экстренного стирания ключевой информации	`Sensor 0x3101 was triggered!`
Снижение заряда элемента питания ДНСД до критического значения	`Battery level is critical: 0!`



Термины и сокращения

DAD (Duplicate address detection)

Duplicate address detection (обнаружение дублирования адреса) — 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.

DHCP (Dynamic Host Configuration Protocol)

Сетевой протокол прикладного уровня, позволяющий компьютерам автоматически получать IP-адреса и другие параметры, необходимые для работы в сети TCP/IP. К таким параметрам относятся маска подсети, IP-адрес шлюза, IP-адреса DNS- и WINS-серверов.

DHCP-relay

Функция ретрансляции запросов и ответов от (к) внешнего DHCP-сервера в локальную сеть.

DHCP-сервер

Сервер, автоматически администрирующий IP-адреса DHCP-клиентов и выполняющий соответствующую настройку для сети.

DMZ (демилитаризованная зона)

Физическая или логическая подсеть, предоставляющая доступ к внешним корпоративным службам из большей сети, с которой нет отношений доверия, как правило, из Интернета. При этом серверы, отвечающие на запросы из внешней сети или направляющие туда запросы, находятся в этой подсети и ограничены в доступе к основным сегментам сети с помощью межсетевого экрана.

Прямых соединений между внутренней сетью и внешней нет: любые соединения возможны только с серверами в DMZ, которые обрабатывают запросы и формируют свои, возвращая ответ получателю уже от своего имени.

DNS-сервер

Сервер, содержащий часть базы данных DNS, используемой для доступа к именам сетевых узлов в интернет-домене. Например, ns.domain.net. Как правило, информация о домене хранится на двух DNS-серверах, называемых «Primary DNS» и «Secondary DNS» (дублирование повышает отказоустойчивость системы).

Также DNS-сервер называют сервером доменных имен, сервером имен DNS.

GRE (Generic Routing Encapsulation)

Протокол туннелирования сетевых пакетов, разработанный компанией Cisco Systems. Подробнее см. RFC 2784.

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.

NTP-сервер

Сервер точного времени, который необходим для синхронизации времени компьютеров, рабочих станций, серверов и прочих сетевых устройств. Этот сервер играет роль посредника между эталоном времени и сетью. Он получает время от эталона по специальному каналу (интерфейсу) и выдает его для любого узла сети, обеспечивая тем самым синхронизацию устройств.

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.

RIB (Routing Information Base)

Таблица маршрутизации, которая содержит все статические маршруты, созданные администратором, и все динамические маршруты, полученные по протоколам DHCP и OSPF.

ViPNet Policy Manager

Программа для централизованного управления политиками безопасности узлов защищенной сети ViPNet.

ViPNet Registration Point

Программное обеспечение, предназначенное для регистрации пользователей ViPNet и хранения их регистрационных данных, а также для выдачи сертификатов подписи и дистрибутивов ключей, создаваемых в программе ViPNet Удостоверяющий и ключевой центр по соответствующим запросам.

ViPNet Удостоверяющий и ключевой центр (УКЦ)

Программа, входящая в состав программного обеспечения ViPNet Administrator. Предназначена для формирования и обновления ключей сетевых узлов ViPNet, а также для управления сертификатами и списками аннулированных сертификатов.

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 осуществляется 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 и обладающее правом доступа к программе ViPNet Центр управления сетью и (или) ViPNet Удостоверяющий и ключевой центр.

Адреса видимости

IP-адреса, виртуальные или реальные, по которым данный узел видит остальные узлы сети ViPNet и по которым приложения отправляют свой трафик.

Адреса доступа

IP-адреса, по которым узел доступен в сети (например, адреса межсетевого экрана, за которым он находится).

Базовый виртуальный адрес

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, то 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.

Датчик несанкционированного доступа (ДНСД)

Электротехническое устройство, встраиваемое в ViPNet Coordinator KB, несанкционированный доступ внутрь корпуса которого требуется контролировать. Обработка события несанкционированного доступа может обрабатываться независимо как самим датчиком, так промежуточным или самим контролируемым устройством с информированием администратора безопасности контролируемого устройства.

Дистрибутив ключей

Файл с расширением `*.dst`, создаваемый в программе ViPNet Удостоверяющий и ключевой центр для каждого пользователя сетевого узла ViPNet. Содержит справочники, ключи и файл лицензии, необходимые для обеспечения первичного запуска и последующей работы программы ViPNet на сетевом узле. Для обеспечения 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.

ДСЧ (датчик случайных чисел)

Программный генератор случайных чисел, представляющий собой генератор гаммы по алгоритму ГОСТ 28147-89.

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 KB интерфейсам можно назначить следующие классы: `access`, `trunk`.

По умолчанию сетевому интерфейсу назначен класс `access`. Если требуется, чтобы сетевой интерфейс обрабатывал трафик из нескольких VLAN, ему необходимо назначить класс `trunk`.

Кластер горячего резервирования

Кластер горячего резервирования состоит из двух взаимосвязанных ViPNet Coordinator KB, 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 KB на активном ViPNet Coordinator KB, пассивный ViPNet Coordinator KB переключается в активный режим для выполнения функций. При этом сбойный ViPNet Coordinator KB перезагружается и становится пассивным.

Кластер горячего резервирования

Кластер горячего резервирования состоит из двух взаимосвязанных ПАК ViPNet Coordinator KB, один из которых (активный) выполняет функции координатора сети ViPNet, а другой (пассивный) находится в режиме ожидания.

В случае сбоев, критичных для работоспособности ViPNet Coordinator KB, выполняющего функции координатора сети ViPNet, второй ПАК, находившийся в режиме ожидания берет на себя выполнение функций координатора сети ViPNet (становится активным). При этом сбойный ПАК перезагружается и становится в режим ожидания (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.

Ключевой блокнот ДСДР

Ключевой блокнот ДСДР (диск ДСДР) предназначен для средств криптографической защиты данных, не содержащих сведений, составляющих государственную тайну. Ключевой блокнот ДСДР содержит ключевую информацию, которая используется для шифрования конфиденциальной информации.

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 KB 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.

Мастер-ключ

Ключ, который администратор сети ViPNet использует для формирования симметричных ключей пользователей и узлов. В сети ViPNet формируется три вида мастер-ключей:

- мастер-ключ ключей обмена;
- мастер-ключ ключей защиты ключей обмена;
- мастер-ключ персональных ключей пользователей.

Мастер-ключ формируется с помощью датчика случайных чисел. Он хранится в программе ViPNet Удостоверяющий и ключевой центр в полной секретности, поскольку компрометация мастер-ключа приводит к компрометации всех ключей, сформированных на его основе.

Межсетевая информация

Информация о доверенной сети или своей сети, предназначенная для организации или изменения межсетевого взаимодействия. В состав межсетевой информации входят связи между сетевыми объектами, параметры сетевых узлов ViPNet и служебная информация (сертификаты издателей, списки аннулированных сертификатов).

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 осуществляет преобразование внутренних IP-адресов 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.

Область маршрутизации

Одна или несколько IP-сетей, в которых осуществляется 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 ViPNet Coordinator KB

Пароль для включения режима администратора ViPNet Coordinator KB. Создается администратором сети ViPNet в программе ViPNet Удостоверяющий и ключевой центр.

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.

PIN-код

Код доступа к подключенному токену, на котором записан персональный ключ пользователя.

Публичный IP-адрес

IP-адрес, который может применяться в интернете.

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 узел ViPNet

Узел, на котором установлено программное обеспечение ViPNet, зарегистрированный в программе ViPNet Центр управления сетью.

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 (идентификатор).

Справочники и ключи

Справочники, ключи узла и ключи пользователя.

Таблица маршрутизации

Таблица, согласно которой происходит процесс выбора пути для передачи данных в сети.

Токен

Компактное устройство аутентификации, предназначенное для обеспечения информационной безопасности пользователя.

Транспортный конверт

Зашифрованная информация служб или приложений, доставляемая на защищенные узлы ViPNet транспортным сервером MFTP.

Транспортный сервер MFTP

Компонент программного обеспечения ViPNet, предназначенный для обмена информацией в сети ViPNet.

Транспортный сервер MFTP

Компонент координатора, обеспечивающий маршрутизацию транспортных конвертов MFTP между узлами сети ViPNet.

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.

Частный IP-адрес

Для сетей на базе протокола IP, не требующих непосредственного подключения к интернету, выделено три диапазона IP-адресов: 10.0.0.0–10.255.255.255; 172.16.0.0–172.31.255.255; 192.168.0.0–192.168.255.255, которые никогда не используются в интернете. Чтобы выйти в интернет с адресом из такого диапазона, необходимо использовать межсетевой экран с функцией NAT или технологию прокси.

Любая организация может использовать любые наборы адресов из этих диапазонов для узлов своей локальной сети.

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.