ViPNet Coordinator HW 4

Сценарии работы

infotecs

ФРКЕ.00130-03 90 03

ОАО «ИнфоТеКС»

127287, г. Москва, Старый Петровско-Разумовский проезд, дом 1/23, строение 1

Тел: (495) 737-61-96 (горячая линия), 737-61-92, факс 737-72-78

Сайт компании «ИнфоТеКС»: (http://www.infotecs.ru)

Электронный адрес службы поддержки: hotline@infotecs.ru

Содержание

Введение

О документе

Для кого предназначен документ

Данный документ предназначен для администраторов, осуществляющих настройку и эксплуатацию программно-аппаратного комплекса ViPNet Coordinator HW. В документе описаны практические сценарии использования ViPNet Coordinator HW, которые требуют комплексного применения различных команд и базовых схем настройки ViPNet Coordinator HW.

Соглашения документа

Ниже перечислены соглашения, принятые в этом документе для выделения информации.

Таблица 1. Обозначения, используемые в примечаниях

Обозначение	Описание
	**Внимание!** Указывает на обязательное для исполнения или следования действие или информацию.
	**Примечание.** Указывает на необязательное, но желательное для исполнения или следования действие или информацию.
	**Совет.** Содержит дополнительную информацию общего характера.

Таблица 2. Обозначения, используемые для выделения информации в тексте

Обозначение	Описание
Название	Название элемента интерфейса. Например, заголовок окна, название поля, кнопки или клавиши.
Клавиша+Клавиша	Сочетание клавиш. Чтобы использовать сочетание клавиш, следует нажать первую клавишу и, не отпуская ее, нажать вторую клавишу.
Меню > Подменю > Команда	Иерархическая последовательность элементов. Например, пункты меню или разделы на панели навигации.
`Код`	Имя файла, путь, фрагмент текстового файла (кода) или команда, выполняемая из командной строки.

При описании команд в данном документе используются следующие условные обозначения:

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

  ```
  hostname# команда
  ```

- Команды, которые могут быть выполнены в режиме и пользователя, и администратора, содержат приглашение с символом «>». Например:

  ```
  hostname> команда
  ```

- Параметры, которые должны быть заданы пользователем, 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}
  ```

Связанные документы

В таблице ниже перечислены документы, входящие в комплект документации ViPNet Coordinator HW, помимо данного документа.

Таблица 3. Связанные документы

Документ	Содержание
«ViPNet Coordinator HW. Общее описание»	Описание общей информации по ViPNet Coordinator HW, а также существующих исполнений и характеристик аппаратных конфигураций
«ViPNet Coordinator HW. Подготовка к работе»	Описание подготовки ViPNet Coordinator HW к использованию, развертывания виртуального образа ViPNet Coordinator HW, работы со справочниками и ключами узла, обновления ПО, резервного копирования и восстановления настроек
«ViPNet Coordinator HW. Настройка с помощью командного интерпретатора»	Описание основных сценариев настройки ViPNet Coordinator HW с помощью командного интерпретатора, работы с журналами и мониторинга ViPNet Coordinator HW
«ViPNet Coordinator HW. Настройка с помощью веб-интерфейса»	Описание основных сценариев настройки ViPNet Coordinator HW с помощью веб-интерфейса
«ViPNet Coordinator HW. Справочное руководство по командному интерпретатору»	Описание команд ViPNet Coordinator HW
«ViPNet Coordinator HW. Справочное руководство по конфигурационным файлам»	Описание конфигурационных файлов управляющего демона и системы защиты от сбоев
«ViPNet Coordinator HW. Лицензионные соглашения на компоненты сторонних производителей»	Лицензионные соглашения на компоненты сторонних производителей, которые использовались при разработке ПО для ViPNet Coordinator HW

О программно-аппаратном комплексе ViPNet Coordinator HW

Программно-аппаратный комплекс ViPNet Coordinator HW представляет собой интегрированное решение на базе специализированной аппаратной платформы и программного обеспечения ViPNet, которое функционирует под управлением адаптированной ОС GNU/Linux.

ViPNet Coordinator HW выступает в роли VPN-сервера и предназначен для использования в IP-сетях, защита которых организуется с применением комплекса программных продуктов ViPNet.

ViPNet Coordinator HW в сети ViPNet реализует функции координатора (см. глоссарий, стр. 71), а также ряд дополнительных функций.

Описание всех функций ViPNet Coordinator HW см. в документе «ViPNet Coordinator HW. Общее описание».

Обратная связь

Дополнительная информация

Сведения о продуктах и решениях ViPNet, распространенные вопросы и другая полезная информация собраны на сайте ОАО «ИнфоТеКС»:

- Веб-портал документации ViPNet http://docs.infotecs.ru.
- Описание продуктов ViPNet http://www.infotecs.ru/products/line/.
- Информация о решениях ViPNet http://www.infotecs.ru/solutions/.
- Сборник часто задаваемых вопросов (FAQ) http://www.infotecs.ru/support/faq/.
- Форум пользователей продуктов ViPNet http://www.infotecs.ru/forum.
- Законодательная база в сфере защиты информации http://www.infotecs.ru/laws/.

Контактная информация

С вопросами по использованию продуктов ViPNet, пожеланиями или предложениями свяжитесь со специалистами ОАО «ИнфоТеКС». Для решения возникающих проблем обратитесь в службу технической поддержки.

- Техническая поддержка для пользователей продуктов ViPNet: hotline@infotecs.ru.
- Форма запроса в службу технической поддержки http://www.infotecs.ru/support/request/.
- Регистрация продуктов и консультации по телефону для клиентов, имеющих расширенный уровень технического сопровождения:

  8 (495) 737-6196,

  8 (800) 250-0260 — бесплатный звонок из любого региона России (кроме Москвы).

Распространение информации об уязвимостях продуктов ОАО «ИнфоТеКС» регулируется политикой ответственного разглашения http://infotecs.ru/products/disclosure.php. Если вы обнаружили уязвимости в продуктах компании, сообщите о них по адресу security-notifications@infotecs.ru.

1

Использование ViPNet Coordinator HW в качестве межсетевого экрана

Организация доступа пользователей к Интернету

С помощью ViPNet Coordinator HW можно организовать доступ пользователей, имеющих частные адреса (см. глоссарий, стр. 73), 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 (см. глоссарий, стр. 72), которая заключается в преобразовании частных адресов локальной сети в один публичный адрес.

На рисунке ниже представлена схема организации доступа пользователей локальной сети к Интернету с помощью ViPNet Coordinator HW.

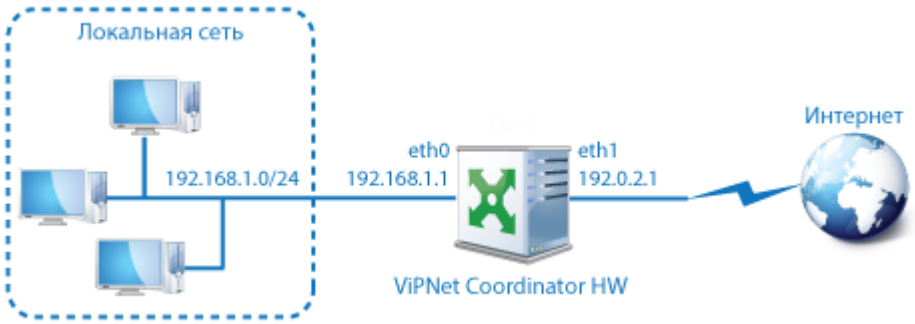

Рисунок 1. Организация доступа пользователей к Интернету с помощью ViPNet Coordinator HW

На приведенной схеме локальная сеть имеет адресное пространство 192.168.1.0/24. Один интерфейс ViPNet Coordinator HW (eth0) подключен к локальной сети и имеет частный адрес 192.168.1.1, другой интерфейс (eth1) подключен к Интернету и имеет публичный адрес 192.0.2.1.

Чтобы пользователи могли подключаться к Интернету, на ViPNet Coordinator HW необходимо задать правило трансляции IP-адресов и сетевой фильтр транзитных IP-пакетов. Для этого в командном интерпретаторе выполните следующие действия:

1. Перейдите в режим администратора с помощью команды `enable`.
2. Создайте правило трансляции частных адресов отправителей в адрес внешнего интерфейса ViPNet Coordinator HW:

   ```
   hostname# firewall nat add src 192.168.1.0/24 dst @InternetIP change src 192.0.2.1
   ```

3. 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 узлы (серверы), которые взаимодействуют с внешними системами или предоставляют различные сервисы внешним пользователям (почтовые, веб-серверы и так далее). В случае атаки на такие серверы под угрозой оказываются остальные компьютеры сети. Чтобы защитить локальную сеть, ее разбивают на сегменты, размещая серверы в демилитаризованной зоне (см. глоссарий, стр. 68). Эту зону отделяют от остальной сети межсетевым экраном (см. глоссарий, стр. 71), который контролирует трафик между сегментами. При этом сервисы, предоставляемые серверами, доступны как внешним пользователям, так и пользователям локальной сети. В то же время локальная сеть недоступна внешним пользователям.

В качестве межсетевого экрана, отделяющего демилитаризованную зону от внутреннего сегмента сети, можно использовать ViPNet Coordinator HW. На рисунке ниже приведена схема сети с веб-сервером, размещенным в демилитаризованной зоне, и ViPNet Coordinator HW с тремя интерфейсами, установленным на границе локальной сети.

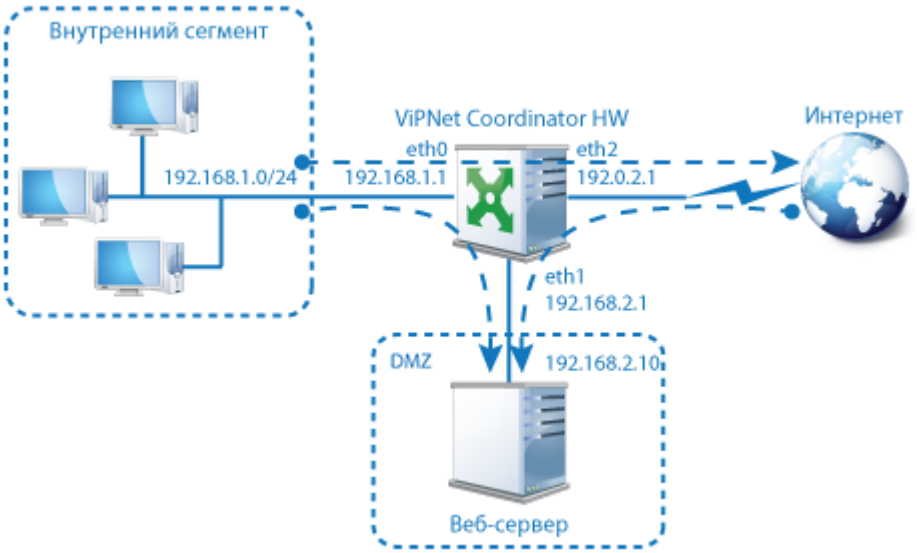

Рисунок 2. Использование ViPNet Coordinator HW для размещения общедоступного сервера в DMZ

Два интерфейса ViPNet Coordinator HW (eth0 и eth1) имеют частные IP-адреса (см. глоссарий, стр. 73) и подключены к сегментам локальной сети, интерфейс eth2 имеет публичный IP-адрес и подключен к Интернету. Пусть заданы следующие сетевые настройки:

- компьютеры во внутреннем сегменте имеют адреса 192.168.1.0/24;
- веб-сервер имеет адрес 192.168.2.10;

- интерфейсы eth0, eth1, eth2 имеют адреса 192.168.1.1, 192.168.2.1, 192.0.2.1 соответственно.

Предполагается, что в приведенной схеме допустимы инициативные соединения по следующим направлениям:

- из внутреннего сегмента в Интернет;
- из внутреннего сегмента к веб-серверу, находящемуся в демилитаризованной зоне (DMZ);
- из Интернета к веб-серверу.

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

1. Перейдите в режим администратора с помощью команды `enable`.
2. Создайте следующие 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 фильтром, настроенным по умолчанию.
3. Создайте следующие правила трансляции 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.

Обеспечение доступа удаленных клиентов к серверу по технологии PPTP VPN

ViPNet Coordinator HW позволяет обеспечить доступ удаленных клиентов по технологии PPTP VPN (см. глоссарий, стр. 69) к серверу, установленному в локальной сети.

Пусть в локальной сети в качестве межсетевого экрана используется ViPNet Coordinator HW, за которым установлен PPTP-сервер. Требуется, чтобы с этим сервером могли взаимодействовать PPTP-клиенты, находящиеся вне сети. На рисунке ниже приведена схема взаимодействия PPTP-клиентов с PPTP-сервером через ViPNet Coordinator HW.

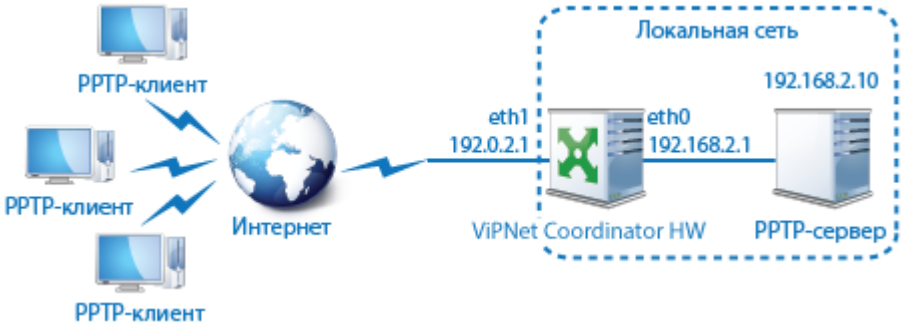

Рисунок 3. Схема взаимодействия PPTP-клиентов с PPTP-сервером через ViPNet Coordinator HW

На приведенной схеме один интерфейс ViPNet Coordinator HW (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 (см. глоссарий, стр. 68). Когда PPTP-клиент обращается к PPTP-серверу, клиент инициирует TCP-соединение с портом назначения 1723. После этого PPTP-сервер инициирует соединение по IP-протоколу GRE, который инкапсулирует пакеты протокола PPTP в IP-пакеты. Через GRE-соединение происходит пересылка данных, а TCP-соединение используется как управляющий канал для поддержки GRE-соединения.

Таким образом, чтобы удаленные PPTP-клиенты могли взаимодействовать с PPTP-сервером, на ViPNet Coordinator HW необходимо задать правила трансляции адресов для протоколов TCP и GRE и транзитные фильтры открытой сети. Для этого выполните следующие действия:

1. Выполните команду `enable` для перехода в режим администратора. В ответ на приглашение введите пароль администратора.
2. Добавьте следующие 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
```

3 Добавьте следующие 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.

2

Использование VPN-функций ViPNet Coordinator HW

Организация защищенного взаимодействия открытых узлов (туннелирование)

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 «Настройка туннелируемых адресов» в документе «ViPNet Coordinator HW Настройка с помощью командного интерпретатора».

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

Рассмотрим типовой случай взаимодействия двух офисов через Интернет. Пусть 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 программного обеспечения 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 HW, к которому подключаются компьютеры.

Рисунок 4: Организация туннелей с использованием ViPNet Coordinator HW

Каждый из ViPNet Coordinator HW 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 (см. глоссарий, стр. 73) и подключен к локальной сети офиса (внутренний интерфейс). Пусть заданы следующие сетевые настройки:

- ViPNet Coordinator HW 1 имеет частный адрес 192.168.1.1 и идентификатор 0x00010101 в сети ViPNet;

- компьютеры 1, 2, 3, подключенные к ViPNet Coordinator HW 1, имеют адреса с 192.168.1.2 по 192.168.1.4;
- ViPNet Coordinator HW 2 имеет частный адрес 192.168.2.1 и идентификатор 0x00010201 в сети ViPNet;
- сервер, подключенный к ViPNet Coordinator HW 2, имеет адрес 192.168.2.2.

Чтобы настроить работу туннелей, необходимо выполнить следующие действия:

1. На компьютерах 1, 2, 3 в качестве шлюза по умолчанию задать частный адрес ViPNet Coordinator HW 1, на сервере — частный адрес ViPNet Coordinator HW 2.
2. На каждом ViPNet Coordinator HW задать адреса компьютеров, которые туннелирует он сам и противоположный ViPNet Coordinator HW (см. ниже).
3. Убедиться, что в настройках каждого ViPNet Coordinator HW разрешен трафик для заданных туннелируемых узлов. По умолчанию на ViPNet Coordinator HW разрешен трафик для всех туннелируемых узлов, но эта настройка могла быть изменена. Подробнее см. в разделе Ограничение доступа туннелируемых узлов (на стр. 21).

Адреса туннелируемых узлов можно задать двумя способами:

- Централизованно в программе ViPNet Центр управления сетью (ЦУС) (см. глоссарий, стр. 69). В этом случае для каждого ViPNet Coordinator HW задаются только адреса узлов, которые туннелирует он сам. После рассылки обновлений справочников и ключей заданные адреса появятся в настройках обоих ViPNet Coordinator HW. Этот способ является предпочтительным.
- Локально на ViPNet Coordinator HW путем редактирования файла конфигурации `iplir.conf` (подробнее см. в документе «ViPNet Coordinator HW. Справочное руководство по конфигурационным файлам»).

Для локальной настройки туннелей на ViPNet Coordinator HW 1 выполните следующие действия:

1. Выполните команду `enable` для перехода в режим администратора. В ответ на приглашение введите пароль администратора.
2. Завершите работу управляющего демона с помощью команды:

   ```
   hostname# iplir stop
   ```
3. Для редактирования файла конфигурации `iplir.conf` выполните команду:

   ```
   hostname# iplir config
   ```
4. 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. Вместо этой строки вы можете записать только первую ее часть:

   ```
   tunnel= 192.168.1.2-192.168.1.4
   ```

   В этом случае строка будет дополнена дублирующимся IP-адресом после запуска управляющего демона. Это справедливо для всех строк, начинающихся с `tunnel=`.
5. В секцию `[id]`, описывающую ViPNet Coordinator HW 2 (содержащую параметр `id= 0x00010201`), добавьте параметр:

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

6. Нажмите сочетание клавиш **Ctrl+O**, чтобы сохранить файл конфигурации, затем нажмите клавишу **Enter**.
7. Нажмите сочетание клавиш **Ctrl+X**, чтобы закрыть файл.
8. Запустите управляющий демон с помощью команды:

```
hostname# iplir start
```

На ViPNet Coordinator HW 2 выполните аналогичные действия:

1. Выполните команду `enable` для перехода в режим администратора. В ответ на приглашение введите пароль администратора.
2. Завершите работу управляющего демона с помощью команды:

```
hostname# iplir stop
```

3. Для редактирования файла конфигурации `iplir.conf` выполните команду:

```
hostname# iplir config
```

4. В собственную секцию `[id]` (содержащую параметр `id= 0x00010201`) добавьте параметр:

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

5. В секцию `[id]`, описывающую ViPNet Coordinator HW 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
```

6. Нажмите сочетание клавиш **Ctrl+O**, чтобы сохранить файл конфигурации, затем нажмите клавишу **Enter**.
7. Нажмите сочетание клавиш **Ctrl+X**, чтобы закрыть файл.
8. Запустите управляющий демон с помощью команды:

```
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 HW 1 и 2 пакеты будут передаваться в зашифрованном виде.

Ограничение доступа туннелируемых узлов

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 «Настройка туннелируемых адресов» в документе «ViPNet Coordinator HW Настройка с помощью командного интерпретатора».

В приведенном выше примере доступ к серверу разрешен всем открытым компьютерам, туннелируемым ViPNet Coordinator HW 1, так как по умолчанию на ViPNet Coordinator HW разрешен трафик для всех туннелируемых узлов.

При необходимости можно запретить доступ к серверу каким-либо компьютерам, не исключая их из списка туннелируемых узлов. Для этого на ViPNet Coordinator HW 1 нужно удалить сетевые фильтры туннелируемых узлов, заданные по умолчанию, и явно указать, каким компьютерам доступ разрешен.

Пусть требуется запретить доступ к серверу компьютеру 1 и разрешить компьютерам 2 и 3. Для этого с помощью командного интерпретатора на ViPNet Coordinator HW 1 выполните следующие действия:

1. Перейдите в режим администратора с помощью команды `enable`.
2. Для просмотра фильтров туннелируемых узлов, заданных на ViPNet Coordinator HW 1, выполните команду:

   ```
   hostname# firewall tunnel show
   ```

   По умолчанию на ViPNet Coordinator HW 1 заданы фильтры, разрешающие трафик для всех туннелируемых узлов.
3. Удалите эти фильтры с помощью команды:

   ```
   hostname# firewall tunnel delete <номер фильтра>
   ```
4. Создайте фильтры туннелируемых узлов с помощью команд:

   ```
   hostname# firewall tunnel add src 192.168.1.3-192.168.1.4 dst 192.168.2.2 pass
   hostname# firewall tunnel add src 192.168.2.2 dst 192.168.1.3-192.168.1.4 pass
   ```

   Эти фильтры будут разрешать трафик в обоих направлениях только между туннелируемыми компьютерами 2, 3 и ViPNet Coordinator HW 2, туннелирующим сервер. Для компьютера 1 защищенное взаимодействие с сервером будет недоступно.

В рассмотренном примере взаимодействие компьютеров с сервером разрешено по всем протоколам и портам. Чтобы ограничить это взаимодействие конкретными протоколами и портами, нужно создать соответствующие сетевые фильтры.

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

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

  ```
  hostname# firewall tunnel add src 192.168.1.3-192.168.1.4 dst 0x00010201 tcp dport 80 pass
  hostname# firewall tunnel add src 0x00010201 dst 192.168.1.3-192.168.1.4 tcp sport 80 pass
  ```
- На ViPNet Coordinator HW 2 удалите фильтр по умолчанию и создайте фильтры, разрешающие TCP-трафик в обоих направлениях между туннелируемым веб-сервером и ViPNet Coordinator HW 1:

  ```
  hostname# firewall tunnel add src 192.168.2.2 dst 0x00010101 tcp sport 80 pass
  hostname# firewall tunnel add src 0x00010101 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 «Настройка туннелируемых адресов» в документе «ViPNet Coordinator HW. Настройка с помощью командного интерпретатора».

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

Рисунок 5. 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 программы ViPNet Центр управления сетью (ЦУС) (см. глоссарий, стр. 69). В этом случае задайте для ViPNet Coordinator HW 2 в качестве IP-адреса туннелируемого соединения IP-адрес сервера и разошлите справочники на ViPNet Coordinator HW 2 и все связанные с ним узлы вашей сети ViPNet (подробнее см. документ «ViPNet Центр управления сетью. Руководство администратора», раздел «Настройка туннелирования»).

  Мы рекомендуем использовать этот способ.

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

Чтобы организовать туннель локально, выполните следующие действия:

1 На ViPNet Coordinator HW 2 задайте адрес туннелируемого сервера. Для этого выполните следующие действия:

1.1 Выполните команду `enable` для перехода в режим администратора. В ответ на приглашение введите пароль администратора.

1.2 Завершите работу управляющего демона с помощью команды:

```
hostname# iplir stop
```

1.3 Для редактирования файла конфигурации `iplir.conf` выполните команду:

1.4 `hostname# iplir config`

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

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

1.6 Нажмите сочетание клавиш **Ctrl+O**, чтобы сохранить файл конфигурации, затем нажмите клавишу **Enter**.

1.7 Нажмите сочетание клавиш **Ctrl+X**, чтобы закрыть файл.

1.8 Запустите управляющий демон с помощью команды:

```
hostname# iplir start
```

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

2.1 Выполните команду `enable` для перехода в режим администратора. В ответ на приглашение введите пароль администратора.

2.2 Для просмотра фильтров туннелируемых узлов, заданных на ViPNet Coordinator HW 2, выполните команду:

```
hostname# firewall tunnel show
```

2.3 Удалите фильтры туннелируемых узлов с помощью команды:

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

2.4 Добавьте следующие фильтры туннелируемых узлов:

```
hostname# firewall tunnel add src 192.168.2.2 dst 0x00010102-0x00010104 pass
hostname# firewall tunnel add src 0x00010102-0x00010104 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 HW 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»).

4 После произведенных настроек компьютеры 1, 2, 3 смогут обращаться к серверу по адресу 192.168.2.2. При этом на участке между ViPNet Coordinator HW 1 и 2 пакеты будут передаваться в зашифрованном виде. Если требуется ограничить доступ к серверу со стороны некоторых узлов, например запретить доступ к серверу компьютеру 1, то на ViPNet Coordinator HW 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.

Организация защищенного взаимодействия между двумя удаленными офисами

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

Рассмотрим пример взаимодействия двух офисов через Интернет (в общем случае офисов может быть больше). В одном из офисов компьютеры объединены в локальную сеть с частными IP-адресами (см. глоссарий, стр. 73) 192.168.1.0/24, в другом — в локальную сеть с частными IP-адресами 192.168.2.0/24. На границе каждой сети установлен шлюз с публичным IP-адресом.

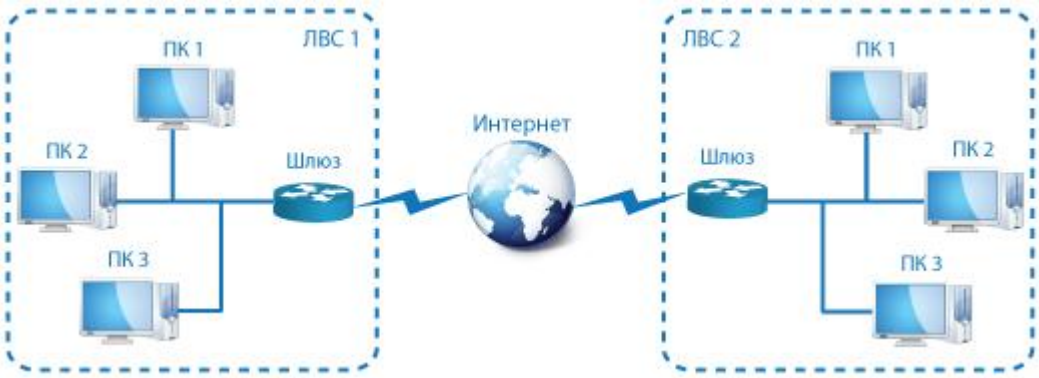

Рисунок 6. Схема сетей двух удаленных офисов

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

Для решения этой задачи выполните следующие действия:

1. В каждой локальной сети установите за шлюзом ViPNet Coordinator HW (или компьютером, на котором развернут виртуальный образ ViPNet Coordinator HW) и создайте для него связи с компьютерами сети. Каждый ViPNet Coordinator HW будет выполнять в своей сети роль координатора (см. глоссарий, стр. 71).
2. Для каждого ViPNet Coordinator HW задайте публичный IP-адрес. Это можно сделать централизованно в программе ViPNet Центр управления сетью (ЦУС) (см. глоссарий, стр. 69) либо локально на ViPNet Coordinator HW с помощью команды:

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

3 В каждой локальной сети установите программу ViPNet Client for Windows на компьютеры, которые должны быть защищены от несанкционированного доступа. Эти компьютеры будут выполнять в сети ViPNet роль клиентов (см. глоссарий, стр. 71).

   При создании сетевых узлов ViPNet зарегистрируйте клиентов каждой сети за соответствующим координатором (подробнее см. в документе «ViPNet Центр управления сетью. Руководство администратора»).

4 Для компьютера, который остается открытым (на него не устанавливается программа ViPNet Client for Windows), организуйте туннель. Для этого на каждом ViPNet Coordinator HW задайте адрес открытого компьютера в качестве туннелируемого адреса (см. «Организация туннелей между защищенными и открытыми узлами» на стр. 23).

В результате две локальные сети будут объединены в единую защищенную сеть ViPNet (см. глоссарий, стр. 72), состоящую из двух сегментов.

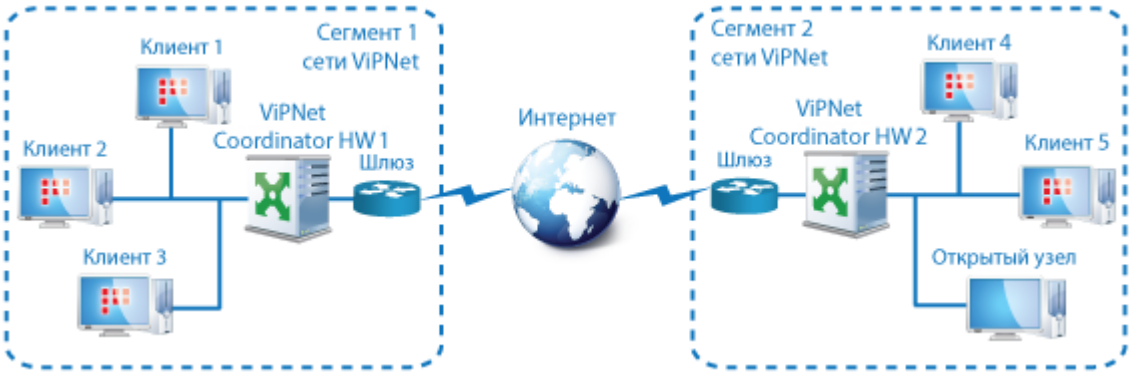

Рисунок 7. Схема защищенной виртуальной сети

Использование ViPNet Coordinator HW в качестве сервера открытого Интернета

ViPNet Coordinator HW может выполнять функцию сервера открытого Интернета в технологии «Открытый Интернет» (см. глоссарий, стр. 71). Эта технология позволяет организовать безопасное подключение к Интернету отдельных узлов сети без их физического отключения от локальной сети и тем самым решить следующие задачи:

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

Внимание! Технология «Открытый Интернет» доступна только в сетях, работающих под управлением программы ViPNet Центр управления сетью (ЦУС) (см. глоссарий, стр. 69).

Технология «Открытый Интернет» основана на создании в локальной сети виртуального контура компьютеров, трафик которых при работе в сети Интернет будет полностью изолирован от остальной локальной сети. Для обеспечения возможности использования данной технологии выполните следующие действия:

1. Установите на границе сети узел ViPNet Coordinator HW, для которого в программе ViPNet Центр управления сетью была включена функция открытого Интернета (подробнее см. в документе «ViPNet Центр управления сетью. Руководство администратора»).
2. Установите прокси-сервер (см. глоссарий, стр. 72) прикладного уровня (например, Squid) на какой-либо компьютер, выделенный для этой цели.
3. Разместите компьютер с прокси-сервером в демилитаризованной зоне (см. глоссарий, стр. 68) за отдельным интерфейсом ViPNet Coordinator HW и добавьте этот компьютер в список узлов, туннелируемых ViPNet Coordinator HW (см. «Организация туннелей между защищенными и открытыми узлами» на стр. 23).
4. Выполните на ViPNet Coordinator HW настройки, приведенные в данном разделе ниже.
5. Установите на компьютеры пользователей программу ViPNet Client for Windows — эти компьютеры будут выполнять в сети ViPNet роль клиентов (см. глоссарий, стр. 71).

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

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

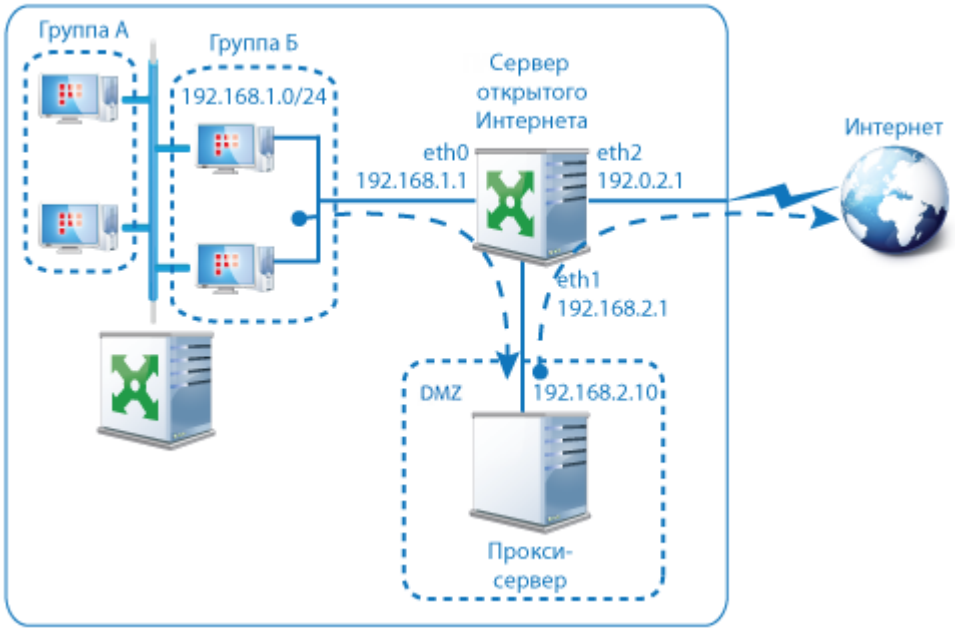

Рисунок 8. Использование ViPNet Coordinator HW в качестве сервера открытого Интернета

Защита сети обеспечивается путем разделения времени работы клиентов группы Б в сети Интернет и в локальной сети. Клиенты, связанные с сервером открытого Интернета, могут работать только в одном из следующих режимов:

- Работа в Интернете. В этом режиме блокируется любое взаимодействие как с локальной сетью, так и с другими сетевыми узлами, кроме сервера открытого Интернета.
- Работа в локальной сети. В этом режиме блокируется любое взаимодействие с сервером открытого Интернета, доступ в Интернет невозможен.

В качестве сервера открытого Интернета ViPNet Coordinator HW выполняет следующие функции:

- организует доступ к ресурсам Интернета через прокси-сервер прикладного уровня;
- запрещает доступ к ресурсам Интернета для клиентов группы А;
- организует защищенный туннель между собой и клиентом группы Б на время его работы с ресурсами Интернета, без возможности доступа к этому туннелю со стороны всех остальных пользователей локальной сети;
- выполняет функции межсетевого экрана, который запрещает доступ в локальную сеть из Интернета.

Для настройки ViPNet Coordinator HW, используемого в качестве сервера открытого Интернета, выполните следующие действия (предполагается, что сетевые настройки соответствуют приведенным на рисунке выше):

1 Выполните команду `enable` для перехода в режим администратора. В ответ на приглашение введите пароль администратора.

2 Добавьте правило трансляции адреса прокси-сервера в адрес внешнего интерфейса ViPNet Coordinator HW (eth2):

```
hostname# firewall nat add src 192.168.2.10 dst @InternetIP change src 192.0.2.1
```

3 Добавьте транзитный фильтр открытой сети, разрешающий соединения прокси-сервера с любыми IP-адресами в Интернете:

```
hostname# firewall forward add src 192.168.2.10 dst @InternetIP pass
```

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

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

5 Добавьте фильтр туннелируемых узлов, разрешающий соединения клиентов с прокси-сервером:

```
hostname# firewall tunnel add src 192.168.1.0/24 dst 192.168.2.10 pass
```

Внимание! Необходимо обратить внимание на настройку маршрутизации на ViPNet Coordinator HW. Рекомендуется в качестве шлюза по умолчанию указывать шлюз, который сообщит ваш интернет-провайдер. В случае если потребуется обеспечить маршрутизацию между сегментами локальной сети, используйте статические маршруты.

После произведенных настроек ViPNet Coordinator HW будет выполнять функции сервера открытого Интернета.

Использование альтернативных каналов доступа к координатору

Рассмотрим пример использования двух альтернативных каналов для взаимодействия координаторов ViPNet Coordinator HW. Допустим, координатор ViPNet Coordinator HW 1 находится в центральном офисе организации, а координатор ViPNet Coordinator HW 2 находится в филиале. Координаторы устанавливают соединение друг с другом через Интернет и обеспечивают связь между ViPNet-клиентами и туннелируемыми узлами, находящимися в двух офисах. Чтобы избежать риска потери соединения, например, из-за неполадок в оборудовании Интернет-провайдера, каждый координатор имеет один основной канал подключения к Интернету и один резервный. Таким образом, каждый координатор имеет по два публичных IP-адреса доступа (см. глоссарий, стр. 70).

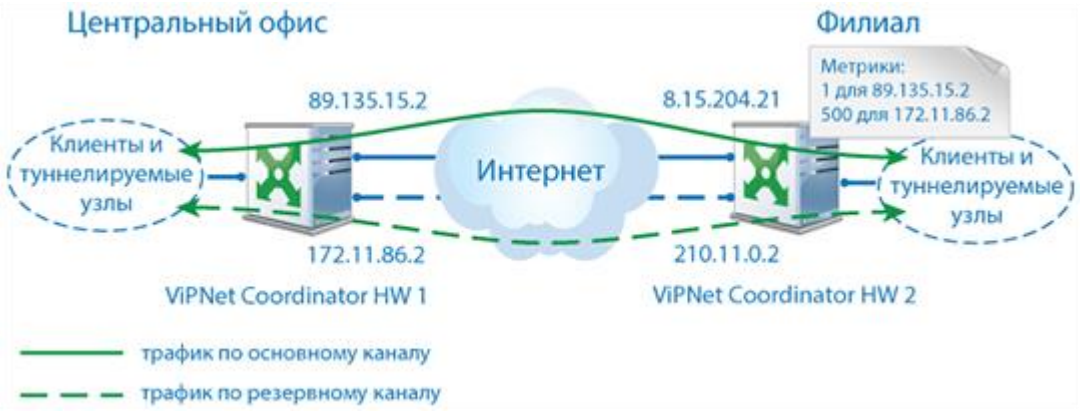

Рисунок 9. Схема взаимодействия двух ViPNet Coordinator HW по двум каналам

Чтобы обеспечить бесперебойное соединение между двумя офисами организации, в первую очередь для передачи трафика между координаторами ViPNet Coordinator HW должен использоваться основной канал, а при отсутствии связи по основному каналу должно происходить автоматическое переключение на резервный канал. Для этого необходимо на каждом координаторе ViPNet Coordinator HW выполнить настройку маршрутизации, а также как минимум на одном из координаторов задать приоритет адресов доступа к другому координатору.

Приоритет адресов доступа задается с помощью метрик. Метрика определяет задержку (в миллисекундах) опроса IP-адресов доступа при проверке доступности адреса. Соединение устанавливается по тому IP-адресу, доступность которого будет определена быстрее в результате опроса. Поэтому рекомендуется для резервного адреса доступа задавать значение метрики, не менее чем на 100 миллисекунд превышающее значение метрики для основного адреса доступа.

Настройка приоритета каналов доступа к ViPNet Coordinator HW

Чтобы настроить приоритет использования каналов передачи трафика между координаторами, выполните следующие действия:

1 Убедитесь, что на обоих ViPNet Coordinator HW сетевые интерфейсы, которые будут использоваться для организации основного и резервного каналов, имеют доступ к Интернету. Также убедитесь, что на каждом ViPNet Coordinator HW задан маршрут по умолчанию.

2 Убедитесь, что администратор вашей сети в программе ViPNet Центр управления сетью (ЦУС) (см. глоссарий, стр. 69) указал IP-адреса каждого ViPNet Coordinator HW, и эта информация передана на координаторы ViPNet.

   Для этого на каждом ViPNet Coordinator HW просмотрите файл конфигурации защищенной сети `iplir.conf` с помощью команды `iplir show config`. Убедитесь, что в секции `[id]` другого координатора указаны его IP-адреса. Например, на ViPNet Coordinator HW 2 в секции `[id]`, описывающей ViPNet Coordinator HW 1, должны быть следующие параметры (см. Рисунок 9 на стр. 31):

   ```
   ip= 89.135.15.2, 11.0.0.8
   ip= 172.11.86.2, 11.1.0.8
   ```

   В этом примере `11.0.0.8` и `11.1.0.8` — виртуальные адреса ViPNet Coordinator HW 1.

3 На ViPNet Coordinator HW 1 настройте статические маршруты для реальных IP-адресов ViPNet Coordinator HW 2 следующим образом:

   - задайте для адреса 8.15.204.21 маршрут через сетевой интерфейс с адресом 89.135.15.2 (основной канал);
   - задайте для адреса 210.11.0.2 маршрут через сетевой интерфейс с адресом 172.11.86.2 (резервный канал).

Внимание! Маршруты следует настраивать для реальных IP-адресов координатора, даже если для него задана видимость по виртуальным адресам (см. глоссарий, стр. 70).

   Например, если на основном канале шлюз Интернет-провайдера имеет IP-адрес 89.135.15.1, чтобы настроить маршрут для основного канала, выполните команду:

   ```
   hostname# inet route add 8.15.204.21 next-hop 89.135.15.1
   ```

4 На ViPNet Coordinator HW 2 настройте аналогичные статические маршруты для реальных IP-адресов ViPNet Coordinator HW 1:

   - задайте для адреса 89.135.15.2 маршрут через сетевой интерфейс с адресом 8.15.204.21 (основной канал);
   - задайте для адреса 172.11.86.2 маршрут через сетевой интерфейс с адресом 210.11.0.2 (резервный канал).

5 На ViPNet Coordinator HW 2 задайте приоритет адресов доступа к ViPNet Coordinator HW 1 с помощью метрик:

5.1 Завершите работу управляющего демона с помощью команды:

```
hostname# iplir stop
```

5.2 Откройте файл `iplir.conf` для редактирования с помощью команды:

```
hostname# iplir config
```

Если для ViPNet Coordinator HW 1 в программе ViPNet Центр управления сетью были указаны IP-адреса, в секции `[id]` этого координатора будут присутствовать следующие параметры:

```
accessiplist = 89.135.15.2, auto, 0.0.0.0, 0, addrdoc
accessiplist = 172.11.86.2, auto, 0.0.0.0, 0, addrdoc
```

5.3 Чтобы задать метрики для IP-адресов ViPNet Coordinator HW 1, в каждом параметре `accessiplist` вместо слова `auto` укажите значение метрики в миллисекундах. Например:

```
accessiplist = 89.135.15.2, 1, 0.0.0.0, 0, addrdoc
accessiplist = 172.11.86.2, 500, 0.0.0.0, 0, addrdoc
```

В примере для основного адреса доступа указана метрика 1, а для резервного адреса — 500.

5.4 Сохраните изменения в файле `iplir.conf` и запустите управляющий демон с помощью команды:

```
hostname# iplir start
```

Примечание. На ViPNet Coordinator HW 1 нет необходимости задавать метрики для адресов доступа к ViPNet Coordinator HW 2. Если вы зададите метрики на обоих координаторах, их значения должны быть согласованы с точки зрения приоритета каналов.

После запуска управляющего демона IP-адрес 89.135.15.2 будет определен как наиболее приоритетный адрес доступа к ViPNet Coordinator HW 1, обмен трафиком между ViPNet Coordinator HW и расположенными за ними ViPNet-клиентами и туннелируемыми узлами будет осуществляться по основному каналу. Если по какой-то причине связь по основному каналу прервется, ViPNet Coordinator HW 2 начнет отправлять IP-пакеты на ViPNet Coordinator HW 1 по адресу 172.11.86.2, ответные пакеты будут приходить на адрес 210.11.0.2 (то есть будет использоваться резервный канал). После того как связь по основному каналу восстановится, ViPNet Coordinator HW 2 переключится на основной канал.

Внимание! В случае недоступности одного из каналов для переключения на второй канал может потребоваться до 15 минут. Чтобы уменьшить время переключения, вы можете задать период опроса координаторов в параметре `checkconnection_interval` в секции `[id]` файла `iplir.conf` (подробнее см. в документе «ViPNet Coordinator HW. Справочное руководство по конфигурационным файлам»). При этом необходимо учитывать, что сокращение периода опроса приведет к увеличению количества служебного трафика между координаторами, в результате чего может снизиться производительность координаторов.

Настройка альтернативных каналов доступа после обновления до версии 4.x

Если при работе с ViPNet Coordinator HW версии 4.0 и ниже у вас было настроено использование фиксированных альтернативных каналов для каких-либо сетевых узлов (в секции `[channels]`), после обновления программного обеспечения выполните аналогичные настройки с помощью множественных адресов доступа и их метрик, так как настройки фиксированных альтернативных каналов не конвертируются при обновлении.

При этом учтите следующие особенности:

- Группы узлов больше не создаются, то есть параметр `group` в секциях `[id]` не используется.
- Внешний адрес узла, который использовался для связи через фиксированный альтернативный канал (параметр `channelfirewallip`), теперь указывается в секции `[id]` этого узла в качестве одного из адресов доступа к узлу (параметр `accessiplist`).
- Приоритет использования того или иного адреса доступа к узлу задается с помощью метрики в параметре `accessiplist`.
- Если раньше для узлов могли назначаться различные порты доступа при использовании фиксированных альтернативных каналов связи (параметр `channelport` в секциях `[id]`), то теперь для всех каналов связи с узлом используется один порт (параметр `port` в секциях `[id]`).

Примечание. Пример настройки альтернативных каналов доступа см. в документе «ViPNet Coordinator HW. Сценарии работы».

Подробную информацию о параметрах конфигурационных файлов ViPNet Coordinator HW см. в документе «ViPNet Coordinator HW. Справочное руководство по конфигурационным файлам».

3

Схемы организации кластера горячего резервирования

Назначение и принципы работы системы защиты от сбоев

Система защиты от сбоев предназначена для контроля работоспособности программы ViPNet Coordinator HW и создания отказоустойчивого решения на базе узлов ViPNet Coordinator HW. Данная система может работать в одиночном режиме или в режиме кластера горячего резервирования. Систему защиты от сбоев можно реализовать на базе ViPNet Coordinator HW, выполняющего любые заявленные функции, в том числе обработку трафика из нескольких VLAN.

Настройка системы защиты от сбоев выполняется путем редактирования конфигурационного файла `failover.ini`. Подробнее о параметрах, содержащихся в этом файле см. в документе «ViPNet Coordinator HW. Справочное руководство по конфигурационным файлам».

Типовая схема организации кластера

Пример типовой схемы организации кластера горячего резервирования (см. глоссарий, стр. 70) приведен ниже. Интерфейсы eth0 используются для организации резервного канала.

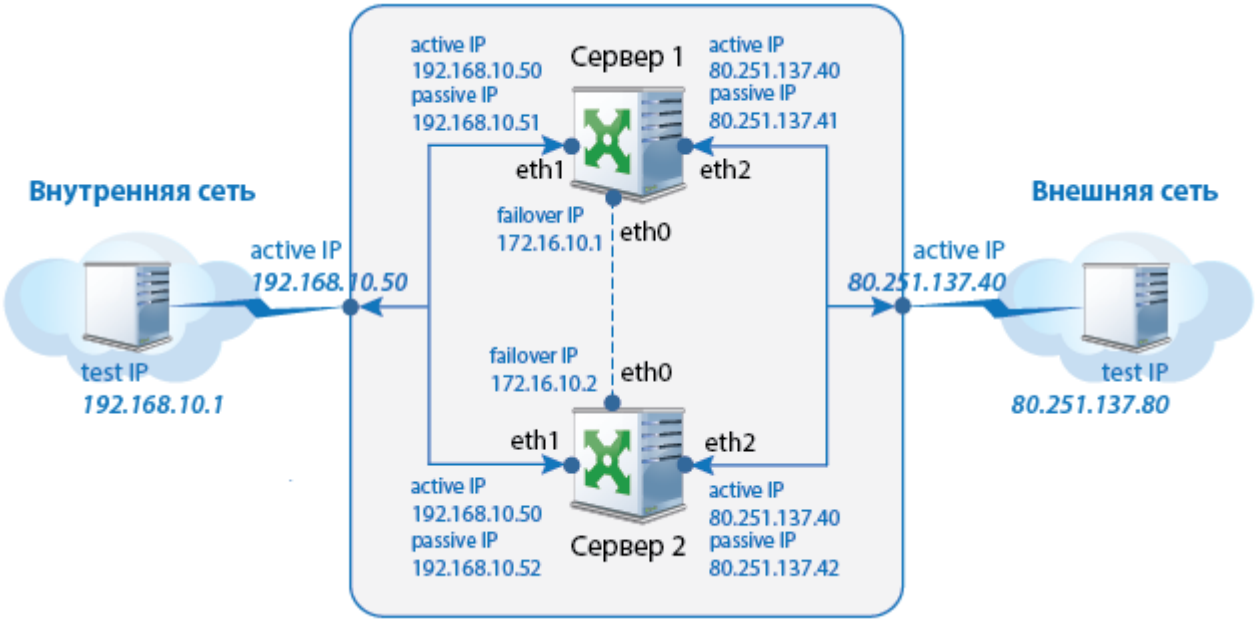

Рисунок 10. Типовая схема организации кластера горячего резервирования

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 «ViPNet Coordinator HW. Справочное руководство по конфигурационным файлам».

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.
- При настройке параметров в секциях `[channel]` IP-адреса, указанные в параметрах `activeip` и `passiveip` для внутреннего (eth1) и внешнего (eth2) интерфейсов, должны находиться в одной подсети. При этом маску подсети указывать необязательно.
- Значения параметров `device`, `activeip` и `testip` должны быть одинаковыми на обоих серверах кластера, а параметры `passiveip` должны различаться. Таким образом, в типовой схеме в каждой из сетей, к которым подключены контролируемые интерфейсы кластера, должны быть выделены три IP-адреса: один для `activeip` и два для `passiveip`. IP-адрес, указанный в параметре `passiveip`, должен совпадать с адресом, который установлен для данного интерфейса в системе (командой `inet ifconfig <интерфейс>`).

- Для интерфейсов, подключенных к одинаковым сетям, параметры `ident` должны совпадать на обоих серверах кластера — именно по этим параметрам система защиты от сбоев определяет интерфейсы, которые выполняют одинаковые функции на серверах кластера.

Внимание! Настоятельно рекомендуем назначать одинаковые имена интерфейсам, выполняющим одинаковые функции на серверах кластера. В следующих версиях ViPNet Coordinator HW параметр `ident` не будет использоваться, и отличающиеся имена интерфейсов приведут к нарушению работоспособности сети ViPNet.

Таблица ниже содержит настройки параметров системы защиты от сбоев для типовой схемы организации кластера.

Таблица 4. Настройки параметров системы защиты от сбоев для типовой схемы организации кластера

Настройки на первом сервере	Настройки на втором сервере
[channel]	**[channel]**
`device = eth1`	`device = eth1`
`activeip = 192.168.10.50`	`activeip = 192.168.10.50`
`passiveip = 192.168.10.51`	`passiveip = 192.168.10.52`
`testip = 192.168.10.1`	`testip = 192.168.10.1`
`ident = if-1`	`ident = if-1`
`checkonlyidle = yes`	`checkonlyidle = yes`
[channel]	**[channel]**
`device = eth2`	`device = eth2`
`activeip = 80.251.137.40`	`activeip = 80.251.137.40`
`passiveip = 80.251.137.41`	`passiveip = 80.251.137.42`
`testip = 80.251.137.80`	`testip = 80.251.137.80`
`ident = if-2`	`ident = if-2`
`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`

Настройки на первом сервере	Настройки на втором сервере
`activeip = 172.16.10.2` (соответствует failover IP второго сервера)	`activeip = 172.16.10.1` (соответствует failover IP первого сервера)

Алгоритм работы активного сервера в такой схеме следующий. Каждые `checktime` секунд проверяется работоспособность каждого из указанных в конфигурационном файле интерфейсов. Если параметр `checkonlyidle` установлен в значение `yes`, то анализируется весь сетевой трафик, проходящий через интерфейс. Если в течение `checktime` через интерфейс не прошло ни одного пакета, через этот интерфейс посылаются эхо-запросы на узел с адресом `testip`, и в течение `timeout` ожидаются ответы на эти запросы. Если ни одного ответа не приходит, то счетчик сбоев этого интерфейса увеличивается на единицу, и посылаются новые эхо-запросы. Если хотя бы один ответ на эхо-запрос приходит, то счетчик сбоев интерфейса обнуляется. В случае обнуления счетчиков сбоев всех интерфейсов происходит возврат к началу цикла проверки (ожидание в течение `checktime` и так далее). При достижении счетчиком сбоев какого-либо интерфейса значения `channelretries` интерфейс считается неработоспособным, и система перезагружается.

Примечание. Чтобы эхо-запросы на узел с 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 интерфейса, равно:

```
checktime + (timeout * channelretries)
```

Алгоритм работы пассивного сервера в такой схеме следующий. Каждые `checktime` секунд из системной ARP-таблицы удаляются записи для `activeip`. Затем со всех интерфейсов посылаются UDP-запросы на узел с адресом `activeip`, точнее система сначала посылает ARP-запрос и только в случае получения ответа на него посылает UDP-запрос. По истечении `timeout` проверяется наличие ARP-записи для каждого `activeip` в системной ARP-таблице и делается вывод о работоспособности соответствующего интерфейса активного сервера. Если ни от одного интерфейса не был получен ответ, счетчик сбоев (один для всех интерфейсов) увеличивается. Если хотя бы от одного интерфейса был получен ответ, счетчик сбоев обнуляется. В случае достижения счетчиком сбоев значения `activeretries`, производится переход сервера в активный режим.

Таким образом, максимальное время, проходящее с момента начала перезагрузки активного сервера до обнаружения пассивным сервером 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 сервера система переводит его интерфейсы в нерабочее состояние не сразу, а через некоторое время, после завершения работы других подсистем (обычно около 30 секунд).

Поэтому, например, если проверяются два интерфейса и только на одном из них произошел сбой, то адрес второго интерфейса будет доступен еще некоторое время, и в течение этого времени пассивный сервер будет получать от него ответы.

Внимание! Если в кластере используется функция L2OverIP (см. «Защита соединения между удаленными сегментами сети на канальном уровне модели OSI» на стр. 59) и требуется проверять состояние его рабочего интерфейса, то в качестве тестового IP-адреса для этого интерфейса необходимо задать адрес из локального сегмента, а также выключить блокирование одноадресных Ethernet-кадров с неизвестным MAC-адресом получателя командой `iplir set l2overip unsolicited-frames broadcast`. Настоятельно не рекомендуется указывать для рабочего интерфейса L2OverIP тестовый IP-адрес в удаленном сегменте.

Схема организации кластера в условиях ограничений по выделению IP-адресов

При отсутствии возможности выделения трех IP-адресов в одной сети для реализации типовой схемы организации кластера горячего резервирования (например, при использовании публичных IP-адресов или в условиях ограниченного адресного пространства сети), можно организовать кластер по схеме, в которой требуется только один IP-адрес для активного сервера. Пример такой схемы приведен ниже.

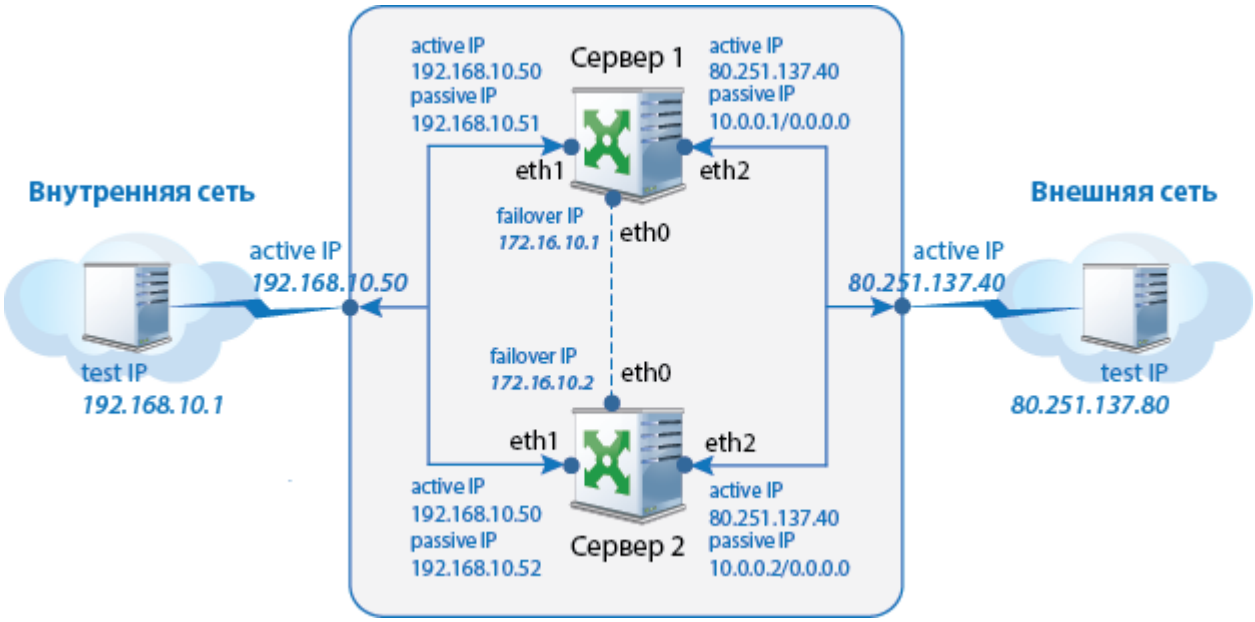

Рисунок 11. Схема организации кластера горячего резервирования в условиях ограничений по выделению IP-адресов

В отличие от типовой схемы в этом случае для внешних интерфейсов (на схеме eth2) вместо трех публичных IP-адресов выделен только один, а адреса пассивных интерфейсов выбраны из диапазона частной сети. Общий принцип работы состоит в том, чтобы использовать на пассивном сервере адреса из другой подсети (например, частной). Чтобы при этом пассивный сервер мог контролировать работоспособность активного путем посылки ARP-запросов, на соответствующих интерфейсах пассивного сервера необходимо установить маску подсети 0.0.0.0 и широковещательный адрес 255.255.255.255.

Примечание. Если на интерфейсе, для которого вы хотите установить маску 0.0.0.0, уже был задан IP-адрес, нужно предварительно очистить настройки адреса интерфейса с помощью команды:

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

Затем вы можете задать нужные IP-адрес и маску с помощью команды:

```
hostname# ifconfig <имя интерфейса> address <IP-адрес> netmask 0.0.0.0
```

Таким образом, при настройке схемы, использующей только один публичный IP-адрес, важно явно задать маски подсети в параметрах `activeip` и `passiveip` файла `failover.ini`. Причем для `activeip` необходимо использовать реальную маску подсети, а для `passiveip` — нулевую маску. При настройке маршрутизации на адреса интерфейсов активного сервера через интерфейсы с маской 0.0.0.0, пассивный сервер всегда будет сначала запрашивать MAC-адреса этих интерфейсов, а затем отправлять по этим адресам пакеты (независимо от того, к каким подсетям принадлежат адреса активного и пассивного серверов).

Внимание! Маршрутизацию через интерфейсы с маской 0.0.0.0 нужно настроить только на адреса интерфейсов активного сервера. В противном случае эти настройки могут нарушить работу других интерфейсов.

Настраивать маски рекомендуется только в файле `failover.ini`, использование других методов крайне нежелательно и может привести к неработоспособности кластера горячего резервирования.

Для настройки маршрутизации предназначен специальный сценарий (shell-скрипт), входящий в комплект поставки ViPNet Coordinator HW. Этот скрипт задает маршруты на сервере при работе в активном и пассивном режимах. Вызов данного скрипта необходимо прописать в параметрах `afterifconf` и `beforeifconf` секции `[network]` в файле `failover.ini` (см. таблицу ниже). В результате демон failoverd будет вызывать shell-скрипт в двух случаях:

- после конфигурирования сетевых интерфейсов — на пассивном сервере этот скрипт будет устанавливать в системе маршруты на IP-адреса активного сервера, при переходе сервера в активный режим работы этот скрипт будет устанавливать статические маршруты (в том числе маршруты по умолчанию).
- до конфигурирования сетевых интерфейсов — на активном сервере этот скрипт будет удалять из системы маршруты, установленные при работе в пассивном режиме.

Необходимая служебная информация передается скрипту через набор переменных окружения.

Таблица 5. Настройки параметров системы защиты от сбоев в условиях ограничений по выделению IP-адресов

Настройки на первом сервере	Настройки на втором сервере
[channel]	**[channel]**
`device = eth1`	`device = eth1`
`activeip = 192.168.10.50`	`activeip = 192.168.10.50`
`passiveip = 192.168.10.51`	`passiveip = 192.168.10.52`
`testip = 192.168.10.1`	`testip = 192.168.10.1`
`ident = if-1`	`ident = if-1`
`checkonlyidle = yes`	`checkonlyidle = yes`
[channel]	**[channel]**
`device = eth2`	`device = eth2`
`activeip = 80.251.137.40/24`	`activeip = 80.251.137.40/24`
`passiveip = 10.0.0.1/0.0.0.0`	`passiveip = 10.0.0.2/0.0.0.0`
`testip = 80.251.137.80`	`testip = 80.251.137.80`
`ident = if-2`	`ident = if-2`
`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`
`afterifconf = /sbin/change_route.sh after`	`afterifconf = /sbin/change_route.sh after`
`beforeifconf = /sbin/change_route.sh before`	`beforeifconf = /sbin/change_route.sh before`
[sendconfig]	**[sendconfig]**
`device = eth0`	`device = eth0`
`activeip = 172.16.10.2` (соответствует failover IP второго сервера)	`activeip = 172.16.10.1` (соответствует failover IP первого сервера)

4

Использование сервисных функций ViPNet Coordinator HW

Организация обработки трафика из нескольких VLAN

Вы можете использовать ViPNet Coordinator HW для обработки трафика в разветвленной сети, состоящей из нескольких независимых виртуальных локальных сетей VLAN (см. глоссарий, стр. 69). Это возможно благодаря поддержке виртуальных интерфейсов и тегированию (маркировке) трафика в соответствии со стандартом IEEE 802.1 Q.

Для организации обработки трафика из нескольких VLAN выполните следующее:

1. Подключите один из сетевых интерфейсов ViPNet Coordinator HW (или компьютера, на котором развернут виртуальный образ ViPNet Coordinator HW) к коммутатору, объединяющему виртуальные сети.
2. Установите для порта коммутатора, к которому подключен интерфейс ViPNet Coordinator HW, режим trunk.
3. Разделите интерфейс ViPNet Coordinator HW, подключенный к коммутатору, на виртуальные интерфейсы по числу VLAN и настройте их аналогично приведенному ниже примеру.

Подробное описание перечисленных действий приведено ниже.

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

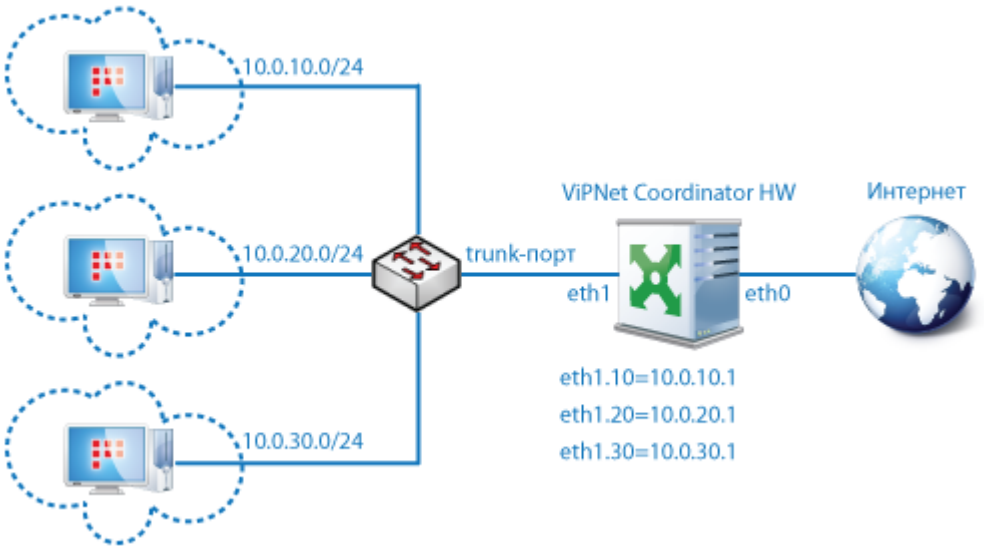

Рисунок 12. Схема использования ViPNet Coordinator HW в сети с VLAN

Чтобы обеспечить работоспособность этой схемы, выполните следующие действия (предполагается, что сетевые настройки соответствуют приведенным на рисунке):

1. Выполните команду `enable` для перехода в режим администратора. В ответ на приглашение введите пароль администратора.

2 Завершите работу управляющего демона с помощью команды:

```
hostname# iplir stop
```

3 Измените класс интерфейса eth1 с помощью команды:

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

4 Добавьте виртуальные интерфейсы eth1.10, eth1.20, eth1.30, которые будут соответствовать виртуальным сетям с номерами 10, 20 и 30, с помощью следующих команд:

```
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 с помощью следующих команд:

```
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 Для редактирования файла конфигурации `iplir.conf` выполните команду:

```
hostname# iplir config
```

7 Добавьте секции `[adapter]`, описывающие виртуальные интерфейсы eth1.10, eth1.20, eth1.30. В каждой секции укажите следующие параметры (на примере интерфейса eth1.10):

```
name= eth1.10
type= internal
allowtraffic= on
```

8 В секции `[adapter]` с описанием интерфейса `eth1` (то есть в секции, содержащей строку «`name= eth1`») измените значение параметра `allowtraffic` на `off`:

```
[adapter]
name= eth1
allowtraffic= off
```

9 Нажмите сочетание клавиш **Ctrl+O**, чтобы сохранить файл конфигурации, затем нажмите клавишу **Enter**.

10 Нажмите сочетание клавиш **Ctrl+X**, чтобы закрыть файл.

11 Включите интерфейс eth1 с помощью команды:

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

При этом автоматически будут включены виртуальные интерфейсы eth1.10, eth1.20, eth1.30.

12 Запустите управляющий демон с помощью команды:

```
hostname# iplir start
```

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

Организация работы клиентов с локальным или удаленным DHCP-сервером

ViPNet Coordinator HW может выполнять функцию DHCP-сервера (см. глоссарий, стр. 68) или функцию агента DHCP-relay. Эти возможности позволяют организовать работу клиентов с DHCP-сервером с помощью ViPNet Coordinator HW двумя способами:

- Использовать ViPNet Coordinator HW в качестве DHCP-сервера для узлов своей сети (см. «Организация работы клиентов с локальным DHCP-сервером» на стр. 47).
- Использовать ViPNet Coordinator HW в качестве посредника между узлами своей сети и удаленным DHCP-сервером (см. «Организация работы клиентов с удаленным DHCP-сервером» на стр. 48).

Организация работы клиентов с локальным DHCP-сервером

В локальной сети небольшой организации, имеющей один офис, можно развернуть собственный DHCP-сервер, используя для этой цели ViPNet Coordinator HW. Для этого выполните следующие действия:

1. Подключите один из интерфейсов ViPNet Coordinator HW (или компьютера, на котором развернут виртуальный образ ViPNet Coordinator HW) к локальной сети. Через этот интерфейс ViPNet Coordinator HW будет взаимодействовать с клиентами при выделении им IP-адресов.
2. Установите на этом интерфейсе IP-адрес, принадлежащий адресному пространству локальной сети.
3. Выполните на ViPNet Coordinator HW приведенные ниже настройки.

Чтобы обеспечить функционирование ViPNet Coordinator HW в качестве DHCP-сервера для клиентов, выполните следующие действия:

1. Выполните команду `enable` для перехода в режим администратора. В ответ на приглашение введите пароль администратора.
2. Задайте интерфейс, на котором будет работать DHCP-сервер, с помощью команды:

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

   В качестве параметра укажите имя интерфейса, подключенного к локальной сети.
3. Задайте диапазон IP-адресов, выделяемых клиентам, с помощью команды:

   ```
   hostname# inet dhcp server range <начальный IP-адрес> <конечный IP-адрес>
   ```

IP-адрес интерфейса ViPNet Coordinator HW, подключенного к локальной сети, не должен входить в этот диапазон.

4 Задайте срок (время аренды), на который клиентам предоставляются IP-адреса, с помощью команды:

```
hostname# inet dhcp server lease <время аренды>
```

5 Задайте IP-адрес шлюза по умолчанию, если его необходимо передавать клиентам, с помощью команды:

```
hostname# inet dhcp server router <IP-адрес>
```

6 Проверьте текущие параметры DHCP-сервера с помощью команды:

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

При необходимости откорректируйте параметры с помощью приведенных выше команд.

7 Включите интерфейс, на котором будет работать DHCP-сервер, с помощью команды:

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

8 Включите автоматический запуск DHCP-сервера, чтобы не запускать его вручную при каждой загрузке ViPNet Coordinator HW. Для этого выполните команду:

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

9 Запустите DHCP-сервер, если это необходимо сделать в текущем сеансе работы, с помощью команды:

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

При следующей загрузке ViPNet Coordinator HW DHCP-сервер будет запущен автоматически (если автоматический запуск включен).

Внимание! Вместе с IP-адресом DHCP-сервер, функционирующий на ViPNet Coordinator HW, всегда предоставляет клиентам свой адрес в качестве адресов DNS-сервера и NTP-сервера. Поэтому клиенты будут осуществлять запросы на разрешение имен и синхронизацию времени через соответствующие сервисы, запущенные на ViPNet Coordinator HW. Если на ViPNet Coordinator HW эти сервисы не запущены, то клиенты не смогут работать с DNS-именами и синхронизировать свое время.

Организация работы клиентов с удаленным DHCP-сервером

Пусть у организации есть несколько офисов, при этом в центральном офисе установлен DHCP-сервер. Клиентам, находящимся в филиале, требуется динамически выделять IP-адреса, однако устанавливать в филиале свой DHCP-сервер нецелесообразно. В этом случае с помощью ViPNet Coordinator HW можно организовать взаимодействие клиентов филиала с DHCP-сервером в центральном офисе, используя ViPNet Coordinator HW в качестве агента DHCP-relay. ViPNet Coordinator HW будет принимать от клиентов филиала DHCP-запросы и передавать их DHCP-

серверу. Ответы, полученные от DHCP-сервера, ViPNet Coordinator HW будет перенаправлять клиентам филиала.

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 ViPNet, то между ViPNet Coordinator HW, выступающим в качестве агента DHCP-relay, и координатором, туннелирующим DHCP-сервер, необходимо настроить видимость по реальным IP-адресам (см.документ «ViPNet Coordinator HW. Настройка с помощью командного интерпретатора», раздел «Настройка параметров видимости узлов»). Если между этими координаторами настроена видимость по виртуальным адресам, ответы от DHCP-сервера будут отправляться на реальный адрес ViPNet Coordinator HW, выступающего в качестве агента DHCP-relay, и не будут доставлены.

В общем случае ViPNet Coordinator HW может обслуживать в качестве агента DHCP-relay несколько подсетей, подключенных к разным интерфейсам. 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. Для организации взаимодействия подсетей с DHCP-сервером посредством ViPNet Coordinator HW выполните следующие действия (см. схему на рисунке ниже):

1. Подключите один из интерфейсов ViPNet Coordinator HW (или компьютера, на котором развернут виртуальный образ ViPNet Coordinator HW) к подсети 10.0.0.0/24 (на схеме eth0), второй интерфейс — к подсети 10.0.1.0/24 (на схеме eth2).

   На каждом из этих интерфейсов должен быть задан статический адрес, принадлежащий адресному пространству соответствующей подсети. На схеме интерфейс eth0 имеет адрес 10.0.0.1, интерфейс eth2 — адрес 10.0.1.1.

2. Подключите третий интерфейс ViPNet Coordinator HW (или компьютера, на котором развернут виртуальный образ ViPNet Coordinator HW) к внешней сети (на схеме eth1).
3. Задайте на DHCP-сервере для каждой подсети диапазон выделяемых адресов и маршрут в сторону ViPNet Coordinator HW.
4. Выполните на ViPNet Coordinator HW приведенные ниже настройки.

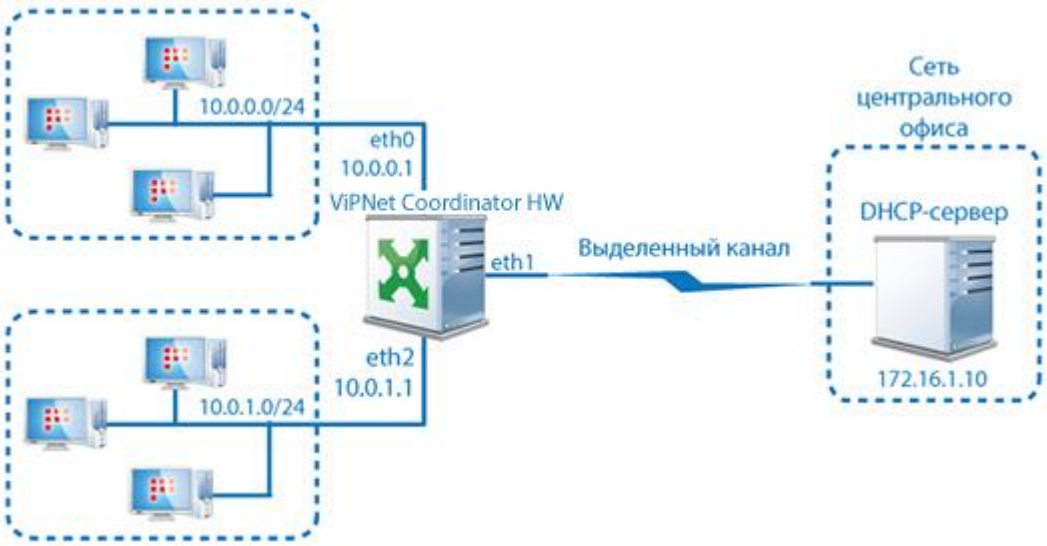

Рисунок 13. Схема использования ViPNet Coordinator HW в качестве агента DHCP-relay

Чтобы обеспечить работоспособность этой схемы, выполните следующие действия:

1. Выполните команду `enable` для перехода в режим администратора. В ответ на приглашение введите пароль администратора.
2. Задайте интерфейсы eth0 и eth2 в качестве интерфейсов, принимающих от клиентов DHCP-запросы, с помощью следующих команд:

   ```
   hostname# inet dhcp relay add listen-interface eth0
   hostname# inet dhcp relay add listen-interface eth2
   ```
3. Задайте параметры для связи ViPNet Coordinator HW с внешним DHCP-сервером с помощью команды:

   ```
   hostname# inet dhcp relay external-interface eth1 server 172.16.1.10
   ```
4. Проверьте текущие параметры службы DHCP-relay с помощью команды:

   ```
   hostname# inet show dhcp relay
   ```

   При необходимости откорректируйте параметры с помощью приведенных выше команд. Для удаления интерфейсов из списка принимающих запросы используйте команду:

   ```
   hostname# inet dhcp relay delete listen-interface <интерфейс>
   ```
5. Включите автоматический запуск на ViPNet Coordinator HW службы DHCP-relay, чтобы не запускать ее вручную при каждой загрузке ViPNet Coordinator HW. Для этого выполните команду:

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

   По умолчанию автоматический запуск службы DHCP-relay выключен.
6. Запустите службу DHCP-relay, если это необходимо сделать в текущем сеансе работы, с помощью команды:

   ```
   hostname# inet dhcp relay start
   ```

   При следующей загрузке ViPNet Coordinator HW служба будет запущена автоматически (если автоматический запуск включен).

После произведенных настроек клиенты филиала смогут получать IP-адреса от DHCP-сервера, установленного в центральном офисе.

Организация работы клиентов удаленных офисов с DNS- и NTP-серверами, расположенными в центральном офисе

Пусть у организации есть несколько офисов, при этом в центральном офисе установлены корпоративные DNS- (см. глоссарий, стр. 68) и NTP-серверы (см. глоссарий, стр. 69). Требуется, чтобы к этим серверам могли обращаться клиенты, находящиеся в филиале. Чтобы решить эту задачу, можно использовать ViPNet Coordinator HW. Для этого выполните следующие действия:

1. Установите в филиале ViPNet Coordinator HW.
2. В сетевых настройках клиентов задайте IP-адрес ViPNet Coordinator HW в качестве DNS- и NTP-серверов. Таким образом, клиенты филиала будут направлять запросы на разрешение DNS-имен и на синхронизацию времени непосредственно на ViPNet Coordinator HW.
3. Для перенаправления запросов клиентов на корпоративные серверы выполните на ViPNet Coordinator HW следующие настройки:

   3.1 Выполните команду `enable` для перехода в режим администратора. В ответ на приглашение введите пароль администратора.

   3.2 Включите автоматический запуск DNS-сервера, чтобы не запускать его вручную при каждой загрузке ViPNet Coordinator HW. Для этого выполните команду:

   ```
   hostname# inet dns mode on
   ```

   Если при установке справочников и ключей (см. глоссарий, стр. 72) вы уже включили автоматический запуск DNS-сервера при загрузке (см. документ «ViPNet Coordinator HW. Подготовка к работе», раздел «Настройка DNS-сервера»), эту команду можно не выполнять.

   3.3 По умолчанию DNS-запросы к серверу разрешены для любых узлов. Если необходимо явно указать IP-адреса узлов и подсетей, узлам которых разрешены запросы к DNS-серверу, задайте их с помощью команды:

   ```
   hostname# inet dns clients add {<IP-адрес> | <IP-адрес/длина маски>}
   ```

   Для проверки текущего списка IP-адресов, которым разрешены DNS-запросы к серверу, используйте команду:

   ```
   hostname# inet dns clients list
   ```

   Для удаления IP-адресов из списка используйте команду:

   ```
   hostname# inet dns clients delete {<IP-адрес> | <IP-адрес/длина маски>}
   ```

3.4 Добавьте IP-адрес корпоративного DNS-сервера, на который ViPNet Coordinator HW будет перенаправлять DNS-запросы, с помощью команды:

```
hostname# inet dns forwarders add <IP-адрес>
```

По умолчанию DNS-запросы перенаправляются на корневые DNS-серверы из домена `root-servers.net`.

Для проверки текущего списка DNS-серверов пересылки используйте команду:

```
hostname# inet dns forwarders list
```

Для удаления DNS-серверов из списка используйте команду:

```
hostname# inet dns forwarders delete <IP-адрес>
```

3.5 Проверьте текущие параметры DNS-сервера с помощью команды:

```
hostname# inet show dns
```

При необходимости откорректируйте параметры с помощью соответствующих команд.

3.6 Запустите DNS-сервер, если это необходимо сделать в текущем сеансе работы, с помощью команды:

```
hostname# inet dns start
```

При следующей загрузке ViPNet Coordinator HW DNS-сервер будет запущен автоматически (если автоматический запуск включен).

3.7 Включите автоматический запуск NTP-сервера, чтобы не запускать его вручную при каждой загрузке ViPNet Coordinator HW. Для этого выполните команду:

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

Если при установке справочников и ключей (см. глоссарий, стр. 72) вы уже включили автоматический запуск NTP-сервера при загрузке (см. документ «ViPNet Coordinator HW. Подготовка к работе», раздел «Настройка NTP-сервера»), эту команду можно не выполнять.

3.8 Добавьте IP-адрес корпоративного NTP-сервера, который необходимо использовать для синхронизации времени, с помощью команды:

```
hostname# inet ntp add <IP-адрес>
```

По умолчанию для синхронизации времени используются публичные NTP-серверы из кластера `pool.ntp.org`.

Для проверки текущего списка NTP-серверов используйте команду:

```
hostname# inet ntp list
```

Для удаления NTP-серверов из списка используйте команду:

```
hostname# inet ntp delete <IP-адрес>
```

3.9 Проверьте текущие параметры NTP-сервера с помощью команды:

```
hostname# inet show ntp
```

При необходимости откорректируйте параметры с помощью соответствующих команд.

3.10 Запустите NTP-сервер, если это необходимо сделать в текущем сеансе работы, с помощью команды:

```
hostname# inet ntp start
```

При следующей загрузке ViPNet Coordinator HW NTP-сервер будет запущен автоматически (если автоматический запуск включен).

После произведенных настроек клиенты филиала смогут обращаться к корпоративным DNS- и NTP-серверам с соответствующими запросами.

Организация агрегированного канала между ViPNet Coordinator HW и коммутатором

Если требуется повысить надежность ваших каналов передачи данных, вы можете объединить несколько физических сетевых интерфейсов ViPNet Coordinator HW в один логический — агрегированный интерфейс (см. глоссарий, стр. 70). При этом соответствующие каналы связи объединяются на канальном уровне сетевой модели OSI.

Агрегированные каналы целесообразно использовать в отказоустойчивых сетях, например в сетях центров обработки данных.

Рассмотрим пример организации агрегированного канала для взаимодействия координатора ViPNet Coordinator HW, имеющего четыре физических интерфейса Ethernet, и коммутатора производства компании Cisco.

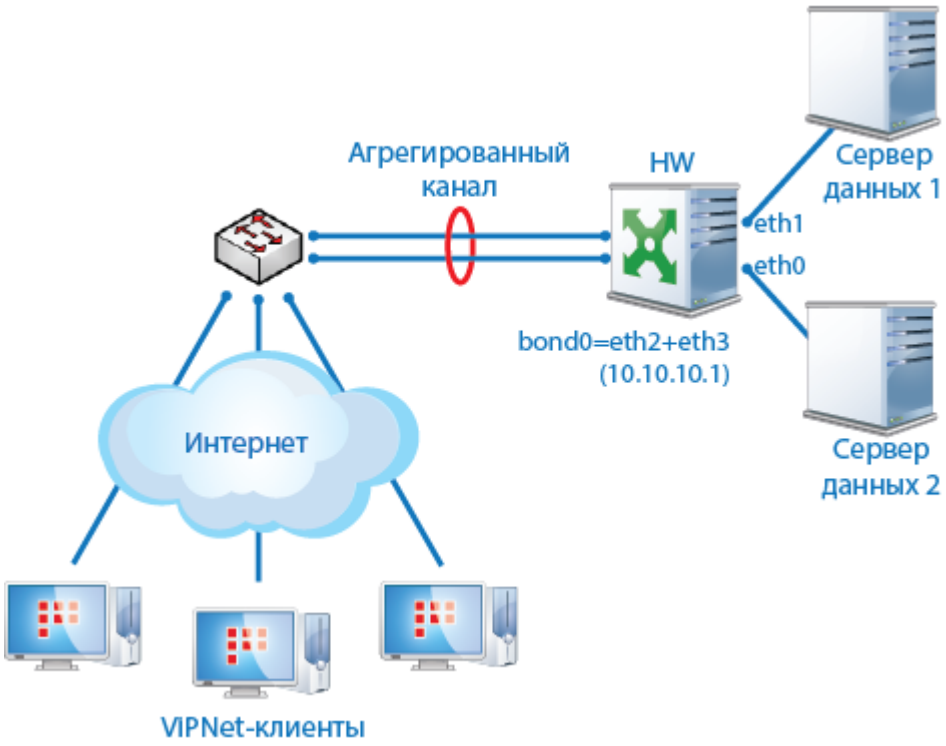

Рисунок 14. Агрегированный канал между ViPNet Coordinator HW и коммутатором

К интерфейсам `eth0` и `eth1` ViPNet Coordinator HW подключены два сервера данных, доступ к которым необходимо обеспечить для ViPNet-клиентов. ViPNet-клиенты подключены через коммутатор. Для обеспечения отказоустойчивости вы можете объединить на ViPNet Coordinator HW интерфейсы `eth2` и `eth3` в агрегированный интерфейс `bond0` и организовать агрегированный канал связи с коммутатором.

Так как при передаче пакетов от ViPNet-клиентов через коммутатор MAC-адрес, IP-адрес и порт сетевого узла отправителя не подменяется, помимо режимов работы агрегированных каналов `balance-rr`, `active-backup` и `broadcast` 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. документ «ViPNet Coordinator HW. Настройка с помощью командного интерпретатора», раздел «Режимы работы агрегированного интерфейса».

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 MAC-адреса, IP-адреса, а также порты TCP или UDP отправителей и получателей.

Чтобы настроить описанный агрегированный канал связи, на ViPNet Coordinator HW выполните следующие действия:

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

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

2. Задайте режим работы создаваемого агрегированного интерфейса `803.2ad` с помощью команды:

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

3. Задайте алгоритм вычисления хэш-функции с помощью команды:

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

4. Отредактируйте файл конфигурации `iplir.conf`. Для этого выполните следующие действия:

   4.1 Остановите управляющий демон с помощью команды:

   ```
   hostname> iplir stop
   ```

   4.2 Откройте файл `iplir.conf` с помощью команды:

   ```
   hostname# iplir config
   ```

   4.3 Добавьте секцию `[adapter]` следующего содержания:

   ```
   [adapter]
   name= bond0
   allowtraffic= on
   type= internal
   ```

   4.4 Сохраните файл конфигурации с помощью сочетания клавиш **Ctrl+O**. Затем нажмите клавишу **Enter**.

   4.5 Закройте файл с помощью сочетания клавиш **Ctrl+X**.

   4.6 Запустите управляющий демон с помощью команды:

   ```
   hostname> iplir start
   ```

5 Задайте для созданного агрегированного интерфейса IP-адрес с помощью следующей команды:

```
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 с помощью команд:

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

8 Включите агрегированный сетевой интерфейс с помощью команды:

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

Чтобы настроить подключение к агрегированному каналу на коммутаторе Cisco (например, Cisco Catalyst 2960-S), выполните следующие действия:

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):

```
interface Port-channel2
   switchport access vlan 116
   swithcport mode access
end
```

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
   swithcport mode access
   channel-group 2 mode active
!
interface GigabitEthernet0/2
   switchport access vlan 116
   swithcport mode access
   channel-group 2 mode active
!
```

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

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

В результате координатор будет доступен по заданному IP-адресу, а трафик между коммутатором и координатором будет идти по агрегированному каналу, состоящему из двух подчиненный 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.

Защита соединения между удаленными сегментами сети на канальном уровне модели OSI

В ViPNet Coordinator HW реализована поддержка технологии L2OverIP (см. глоссарий, стр. 69), 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 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 (центров обработки данных), локальные сети которых объединены высокоскоростным каналом связи и представляют собой единый домен коллизий (см. глоссарий, стр. 70).

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.

Исполнения ViPNet Coordinator HW50 N1, N2, N3, а также ViPNet Coordinator HW100 X1, X8 не поддерживают функцию L2OverIP.

Для работы функции L2OverIP в исполнении ViPNet Coordinator HW-VA необходимо в настройках среды виртуализации включить неразборчивый режим (Promiscuous Mode) для интерфейса, к которому привязан адаптер виртуальной машины. Подробнее см. руководство администратора платформы виртуализации, которую вы используете.

Технология L2OverIP предполагает взаимодействие между узлами нескольких удаленных сегментов сети через ViPNet Coordinator HW, 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. Каждый ViPNet Coordinator HW 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 HW по защищенному каналу. ViPNet Coordinator HW, получивший IP-пакеты специального формата, извлекает из них исходные кадры и передает получателям в своем сегменте.

С помощью функции L2OverIP можно объединить несколько сегментов сети, в том числе сегменты, состоящие из виртуальных локальных сетей (VLAN).

Подробнее о технологии L2OverIP см. в документе «ViPNet Coordinator HW. Настройка с помощью командного интерпретатора», раздел «Реализация технологии L2OverIP».

Настройка функции L2OverIP при отсутствии VLAN

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

Каждый сегмент сети подключается к одному из интерфейсов ViPNet Coordinator HW, установленному на границе сегмента. На рисунке ниже приведена схема подключения, на которой два ViPNet Coordinator HW условно обозначены как HW-1 и HW-2. По такой же схеме можно объединить больше двух сегментов сети (но не более 31).

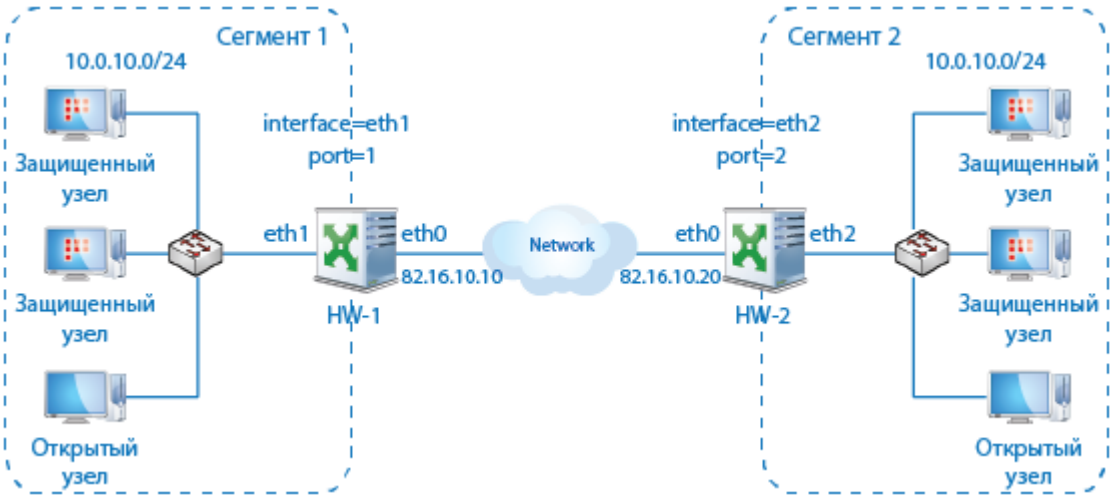

Рисунок 15. Защита соединения удаленных сегментов сети на канальном уровне модели OSI

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

1. Перейдите в режим администратора с помощью команды `enable`. В ответ на приглашение введите пароль администратора.
2. Укажите сетевой интерфейс, сегмент сети которого будет объединяться с другим сегменом с помощью функции L2OverIP:
   - На HW-1 выполните команду:

     ```
     hostname# iplir set l2overip interface eth1
     ```
   - На HW-2 выполните команду:

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

   Если сегментов больше двух, аналогичным образом укажите рабочий интерфейс на остальных ViPNet Coordinator HW.
3. Задайте параметры локального сегмента сети, указав уникальный номер порта и адрес внешнего интерфейса:
   - На HW-1 выполните команду:

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

- На HW-2 выполните команду:

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

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

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

   - На HW-1 выполните команду:

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

   - На HW-2 выполните команду:

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

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

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

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

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

   Если адрес видимости другого сегмента изменился, выполните команду:

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

6. Добавьте сетевой фильтр защищенной сети, разрешающий любые соединения по протоколу 97:

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

7. Включите функцию L2OverIP:

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

8. Убедитесь в корректности выполненных настроек функции L2OverIP с помощью команды:

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

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

Настройка функции L2OverIP в случае использования VLAN

С помощью функции L2OverIP можно объединять сегменты сети, состоящие из нескольких виртуальных локальных сетей (VLAN (см. глоссарий, стр. 69)).

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

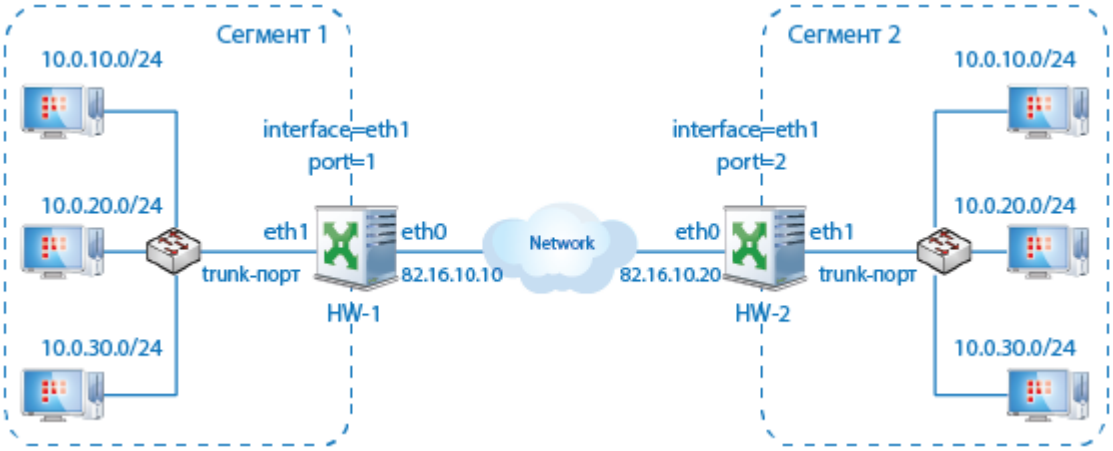

Рисунок 16. Схема объединения сегментов с одинаковым числом VLAN

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

1. Перейдите в режим администратора с помощью команды `enable`. В ответ на приглашение введите пароль администратора.
2. Разделите интерфейс eth1, подключенный к коммутатору, на виртуальные интерфейсы для VLAN с номерами 10, 20, 30 и выполните необходимые настройки (см. «Организация обработки трафика из нескольких VLAN» на стр. 45).
3. Укажите сетевой интерфейс, сегмент сети которого будет объединяться с другим удаленным сегменом с помощью функции L2OverIP:

   ```
   hostname# iplir set l2overip interface eth1
   ```
4. Укажите параметры локального сегмента сети:
   - На HW-1 выполните команду:

     ```
     hostname# iplir set l2overip local-port 1 82.16.10.10
     ```
   - На HW-2 выполните команду:

     ```
     hostname# iplir set l2overip local-port 2 82.16.10.20
     ```
5. Укажите параметры удаленного сегмента сети:
   - На HW-1 выполните команду:

     ```
     hostname# iplir set l2overip remote-port 2 82.16.10.20
     ```
   - На HW-2 выполните команду:

     ```
     hostname# iplir set l2overip remote-port 1 82.16.10.10
     ```
6. Добавьте сетевой фильтр защищенной сети, разрешающий любые соединения по протоколу 97:

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

7 Включите функцию L2OverIP:

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

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

Если в приведенном примере требуется объединить только одну виртуальную сеть одного сегмента с той же виртуальной сетью другого сегмента, то в качестве рабочего интерфейса для функции L2OverIP вместо физического интерфейса eth1 укажите виртуальный интерфейс для соответствующей VLAN (например, eth1.10).

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

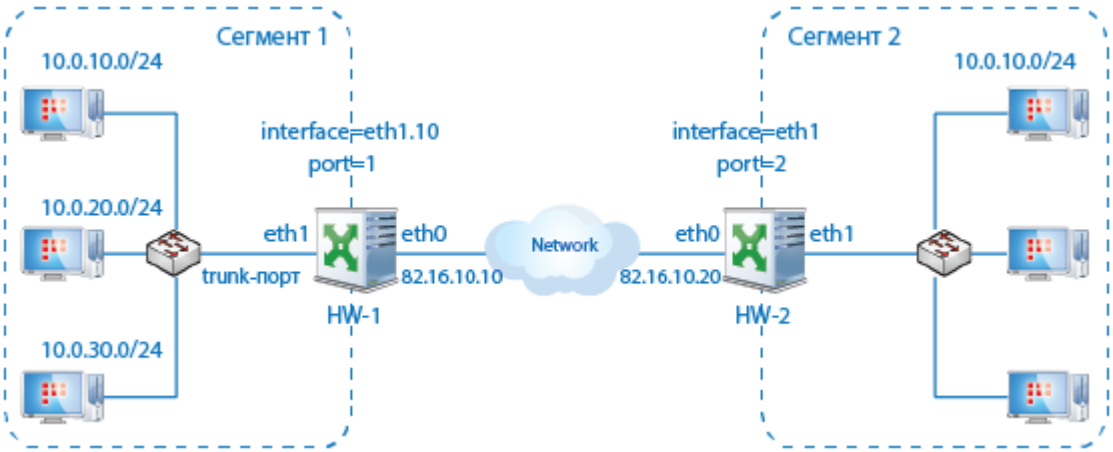

Рисунок 17. Схема объединения одной VLAN с сегментом без VLAN

В этом случае измените настройку следующим образом:

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

  ```
  hostname# iplir set l2overip interface eth1.10
  ```

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

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

Настройка функции L2OverIP для обеспечения работоспособности протоколов динамической маршрутизации

Технология L2OverIP также может использоваться для объединения удаленных сегментов сети с целью организации между ними динамической маршрутизации (см. глоссарий, стр. 71) по протоколу OSPF (см. глоссарий, стр. 69).

Для динамической маршрутизации по протоколу OSPF должно выполняться следующее условие: работа по протоколу OSPF должна поддерживаться на всем пути следования IP-пакетов. В сетях общего пользования, в частности, в сети Интернет, протокол OSPF не поддерживается. В связи с этим динамическая маршрутизация между несколькими сегментами сети, объединенными Интернетом, не сможет осуществляться. Решить данную проблему можно путем объединения сегментов сети с помощью технологии L2OverIP. В результате маршрутизация IP-пакетов будет производиться как бы в одном сегменте сети, а не в двух, разделенных сетью Интернет.

Общие сведения по протоколу OSPF см. в документе «ViPNet Coordinator HW. Настройка с помощью командного интерпретатора».

Например, пусть требуется объединить два сегмента сети, в каждом из которых создана разветвленная структура узлов с несколькими маршрутизаторами. Каждый сегмент сети подключается к одному из интерфейсов ViPNet Coordinator HW, установленному на границе сегмента. На всех маршрутизаторах обоих сегментов поддерживается работа по протоколу OSPF.

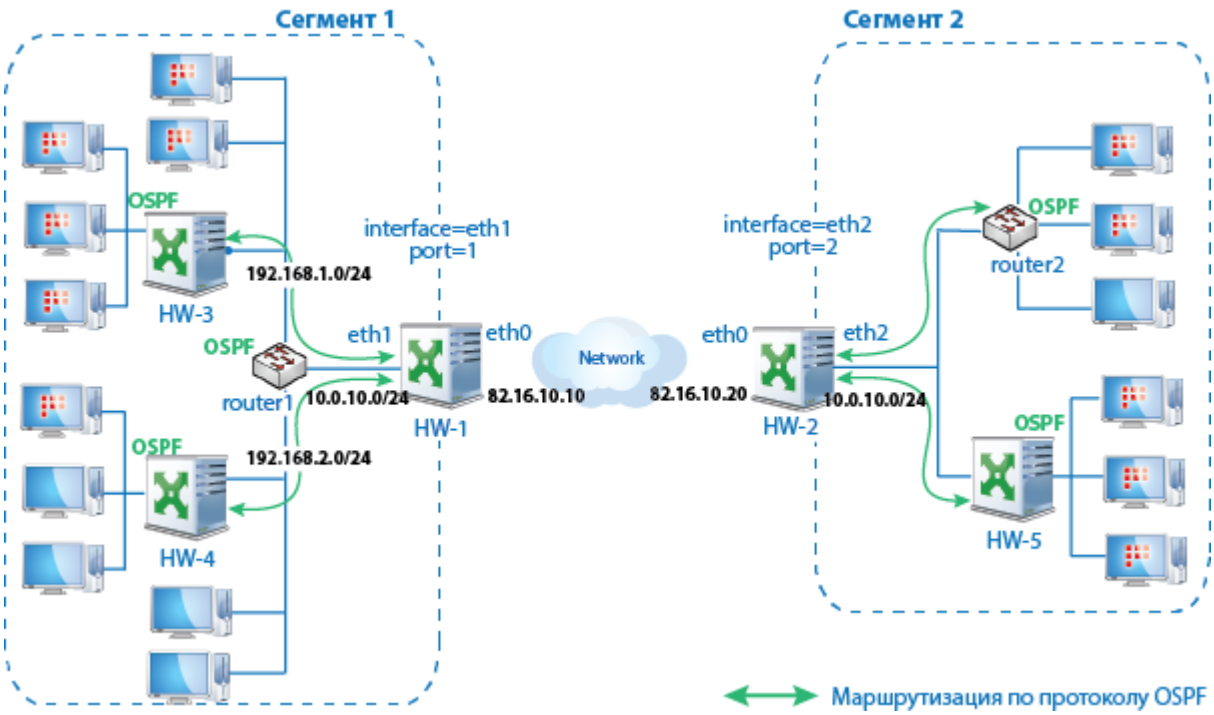

Рисунок 18. Схема работы L2OverIP и динамической маршрутизации по протоколу OSPF

Чтобы организовать маршрутизацию по протоколу OSPF в рассмотренном примере, требуется выполнить следующие действия:

1 Настроить параметры L2OverIP на координаторах HW-1 и HW-2 (см. «Настройка параметров L2OverIP» на стр. 65).

2 Настроить работу по протоколу OSPF на маршрутизаторах router1 и router2 и координаторах HW-3, HW-4, HW-5 (см. «Настройка протокола OSPF» на стр. 66).

Примечание. На HW-1 и HW-2, установленных на границах сегментов, настройка протокола OSPF не требуется, поскольку в данном сценарии эти координаторы не участвуют в маршрутизации IP-трафика.

Настройка параметров L2OverIP

Для настройки параметров L2OverIP выполните следующие действия на координаторах HW-1 и HW-2:

1 Перейдите в режим администратора с помощью команды `enable`. В ответ на приглашение введите пароль администратора.

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

- На HW-1 выполните команду:

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

- На HW-2 выполните команду:

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

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

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

- На HW-1 выполните команду:

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

- На HW-2 выполните команду:

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

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

- На HW-1 выполните команду:

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

- На HW-2 выполните команду:

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

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

5 На каждом координаторе добавьте сетевой фильтр защищенной сети, разрешающий любые соединения по протоколу 97:

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

6 На каждом координаторе включите функцию L2OverIP:

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

7 На каждом координаторе убедитесь в корректности выполненных настроек функции L2OverIP с помощью команды:

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

Настройка протокола OSPF

Для настройки работы по протоколу OSPF выполните следующие действия на координаторах HW-3, HW-4, HW-5:

1 На каждом координаторе включите использование протокола OSPF:

```
hostname# inet ospf mode on
```

2 Укажите на координаторах сеть, в которой осуществляется обмен информацией по протоколу OSPF:

- На HW-3 выполните команду:

  ```
  hostname# inet ospf network add 192.168.1.0 netmask 255.255.255.0 area 1
  ```

- На HW-4 выполните команду:

  ```
  hostname# inet ospf network add 192.168.2.0 netmask 255.255.255.0 area 1
  ```

- На HW-5 выполните команду:

  ```
  hostname# inet ospf network add 10.0.10.0 netmask 255.255.255.0 area 1
  ```

Примечание. В случае, рассмотренном на примере, все координаторы, работающие по протоколу OSPF, принадлежат одной области маршрутизации (см. глоссарий, стр. 71).

---

3 На каждом координаторе создайте следующие сетевые фильтры открытой сети:

- Фильтры, разрешающие входящий однонаправленный IP-трафик (unicast) и многоадресный IP-трафик (multicast) по протоколу OSPF от соседних OSPF-маршрутизаторов, с которыми взаимодействует координатор.

  Если политика безопасности вашей организации не запрещает прохождение служебного IP-трафика по протоколу OSPF (IP:89) от любого узла, то создайте следующие фильтры:

  ```
  hostname# firewall local add 1 rule "Rule 1" src @any dst @local service @OSPF
  pass
  ```

  ```
  hostname# firewall local add 1 rule "Rule 1" src @any dst @multicast service
  @OSPF pass
  ```

Если создание фильтров, пропускающих служебный IP-трафик по протоколу OSPF от любых узлов запрещено, создайте аналогичные фильтры для каждого OSPF-маршрутизатора, который является соседним для координатора.

Таблица 6. Фильтры для входящего OSPF-трафика на конкретном координаторе

Имя координатора	Название	Источник	Назначение	Протокол	Действие
HW-3	Rule 1	<IP-адрес router1>	@local	IP:89	pass
	Rule 2	<IP-адрес router1>	@multicast	IP:89	pass
HW-4	Rule 1	<IP-адрес router1>	@local	IP:89	pass
	Rule 2	<IP-адрес router1>	@multicast	IP:89	pass
HW-5	Rule 1	<IP-адрес router1>	@local	IP:89	pass
	Rule 2	<IP-адрес router1>	@multicast	IP:89	pass
	Rule 3	<IP-адрес router2>	@local	IP:89	pass
	Rule 4	<IP-адрес router2>	@multicast	IP:89	pass

- Фильтр, разрешающий исходящий IP-трафик с координатора по протоколу OSPF на любой IP-адрес назначения:

```
hostname# firewall local add 1 rule "Rule 1" src @local dst @any service @OSPF pass
```

Примечание. На маршрутизаторах router1 и router2 настройка протокола OSPF должна быть выполнена по тому сценарию, который они поддерживают.

В результате сегменты сети будут объединены по технологии L2OverIP и между ними будет организована динамическая маршрутизация по протоколу OSPF.

Внимание! При использовании протокола OSPF в случае сбоя на одном из маршрутов для переключения на альтернативный маршрут может потребоваться до 15 минут. Чтобы уменьшить время переключения, рекомендуется в файле `iplir.conf` в секциях `[id]`, соответствующих связанным координаторам ViPNet, задать более короткий период опроса, изменив значение параметра `checkconnection_interval` (подробнее см. в документе «ViPNet Coordinator HW. Справочное руководство по конфигурационным файлам»). Например, можно задать для параметра значение 30 секунд При этом необходимо учитывать, что сокращение периода опроса приведет к увеличению количества служебного трафика между координаторами, в результате чего может снизиться производительность координаторов.

A

Глоссарий

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 компьютеров в интернет-домене. Например, ns.domain.net. Как правило, информация о домене хранится на двух DNS-серверах, называемых «Primary DNS» и «Secondary DNS» (дублирование делается для повышения отказоустойчивости системы).

Также DNS-сервер называют сервером доменных имен, сервером имен DNS.

GRE

Протокол туннелирования сетевых пакетов, разработанный компанией Cisco Systems. Подробнее см. RFC 2784 https://tools.ietf.org/html/rfc2784.

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 смогут взаимодействовать друг с другом так, как будто они находятся в одном сегменте с прямой видимостью по MAC-адресам. 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.

PPTP

Туннельный протокол типа точка-точка, позволяющий компьютеру устанавливать защищённое соединение с сервером за счёт создания специального туннеля в стандартной незащищённой сети. Подробнее см. RFC 2637 (http://www.ietf.org/rfc/rfc2637.txt).

ViPNet Центр управления сетью (ЦУС)

ViPNet Центр управления сетью — это 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 УКЦ;
- задание полномочий пользователей сетевых узлов ViPNet.

VLAN

Виртуальная локальная компьютерная сеть, представляет собой группу 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.

Агрегированный сетевой интерфейс

Логический сетевой интерфейс, образованный из нескольких физических интерфейсов Ethernet, объединенных на канальном уровне сетевой модели OSI.

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).

Внешняя сеть

Сеть, отделенная от внутренней сети межсетевым экраном.

Внутренняя сеть

Локальная сеть, где находятся рассматриваемые узлы, которая отделена от внешней сети межсетевым экраном.

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.

Кластер горячего резервирования

Кластер горячего резервирования состоит из двух взаимосвязанных серверов ViPNet Coordinator HW, один из которых (активный) выполняет функции координатора сети ViPNet, а другой сервер (пассивный) находится в режиме ожидания. В случае сбоев, критичных для работоспособности ПО ViPNet на активном сервере, пассивный сервер переключается в активный режим для выполнения функций сбойного сервера. При этом сбойный сервер перезагружается и становится пассивным.

Клиент (ViPNet-клиент)

Сетевой узел ViPNet, который является начальной или конечной точкой передачи данных. В отличие от координатора клиент не выполняет функции маршрутизации трафика и служебной информации.

Координатор (ViPNet-координатор)

Сетевой узел, представляющий собой компьютер с установленным программным обеспечением координатора (ViPNet Coordinator) или специальный программно-аппаратный комплекс. В рамках сети ViPNet координатор выполняет серверные функции, а также маршрутизацию трафика и служебной информации.

Маршрутизация

Процесс выбора пути для передачи информации в сети.

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 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.

Область маршрутизации

Одна или несколько IP-сетей, в которых осуществляется обмен информацией по определенному протоколу, в частности, по протоколу OSPF (см. глоссарий, стр. 69).

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 32-разрядное число, которое записывается так же как и IP-адрес — в десятично-точечном формате (в виде четырех однобайтовых чисел, разделенных точками).

Открытый Интернет

Технология, реализованная в программном обеспечении ViPNet. При подключении к Интернету узлы локальной сети изолируются от сети ViPNet, а при работе в сети ViPNet — от Интернета, что обеспечивает защиту от возможных сетевых атак извне без физического отключения компьютеров от локальной сети.

Открытый узел

Узел, с которым обмен информацией происходит в незашифрованном виде.

Прозрачный режим работы прокси-сервера

Режим работы, при котором не требуется выполнять настройку программного обеспечения на рабочих местах пользователей, подключающихся к Интернету через прокси-сервер.

Прокси-сервер

Программа, транслирующая соединения по некоторым протоколам из внутренней сети во внешнюю и выступающая при этом как посредник между клиентами и сервером.

Сервер IP-адресов

Функциональность координатора, обеспечивающая регистрацию, рассылку и предоставление информации о состоянии защищенных узлов.

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 сетевых узлов ViPNet.

Сеть ViPNet имеет свою адресацию, позволяющую наладить обмен информацией между ее узлами. Каждая сеть ViPNet имеет свой уникальный номер (идентификатор).

Справочники и ключи

Справочники, ключи узла и ключи пользователя.

Трансляция сетевых адресов (NAT)

Технология, позволяющая преобразовывать IP-адреса и порты, использующиеся в одной сети, в адреса и порты, использующиеся в другой.

Транспортный сервер

Функциональность координатора, обеспечивающая маршрутизацию транспортных конвертов между узлами сети ViPNet.

Туннелирование

Технология, позволяющая защитить соединения между узлами локальных сетей, которые обмениваются информацией через Интернет или другие публичные сети, путем инкапсуляции и шифрования трафика этих узлов не самими узлами, а координаторами, которые установлены на границе их локальных сетей. При этом установка программного обеспечения ViPNet на эти узлы необязательна, то есть туннелируемые узлы могут быть как защищенными, так и открытыми.

Узел сети ViPNet

Сетевой узел, на котором установлено программное обеспечение ViPNet с функцией шифрования трафика на сетевом уровне.

Частный 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.

Указатель

Н

О

П

С

Т

Ч