



ViPNet Coordinator IG 5

Сценарии работы

Версия продукта: 5.2.0

ViPNet Coordinator IG10

ViPNet Coordinator IG100

ViPNet Coordinator IG1000

© АО «ИнфоТеКС», 2025

ФРКЕ.467759.003ИС5

Версия продукта 5.2.0

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

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

ViPNet® является зарегистрированным товарным знаком АО «ИнфоТеКС».

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

Все условия эксплуатации и ограничения на использование ViPNet Coordinator IG приведены в Правилах пользования, Лицензионном соглашении с конечным пользователем либо в другой эксплуатационной документации, сопровождающей ViPNet Coordinator IG.

АО «ИнфоТеКС»

125167, г. Москва, вн. тер. г. муниципальный округ Хорошевский, ул. Викторенко, д. 9, стр. 1, помещ. 47

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

Сайт: infotecs.ru

Служба поддержки: hotline@infotecs.ru

Содержание

Введение.....	5
О документе	6
Соглашения документа	7
Обратная связь	8
Глава 1. ViPNet Coordinator IG как межсетевой экран сегмента промышленной сети	9
Защита сегмента промышленной сети.....	10
Защита сегмента с устройствами Modbus.....	10
Защита сегмента с устройствами МЭК 60870-5-104	12
Доступ из защищённого сегмента.....	14
Размещение общедоступного сервера в демилитаризованной зоне	16
Настройка политик безопасности для разных режимов работы	18
Глава 2. Построение защищённого канала связи	20
Защищённое взаимодействие сегментов промышленной сети	21
Удалённый доступ клиента к защищённому сегменту промышленной сети	24
Глава 3. Использование встроенного шлюза Modbus	26
Взаимодействие устройств Modbus RTU с сервером сбора данных	27
Защищённое взаимодействие устройств Modbus RTU с удалёнными сетевыми узлами	29
Защищённый удалённый доступ клиента к устройствам Modbus RTU.....	32
Защищённое удалённое взаимодействие устройств Modbus RTU.....	34
Доверенное взаимодействие между двумя последовательными сетями	38
Доверенное взаимодействие между несколькими последовательными сетями	41
Глава 4. Использование технологии PoE	44
Защита канала связи между сервером и видеокамерами с PoE	45
Глава 5. Использование интерфейса GPIO	48
Подключение датчика вскрытия внешнего шкафа	49
Подключение внешнего устройства для перехода в специальный режим	51
Настройка сигнализации для администратора информационной безопасности.....	53
Глава 6. Использование функции MultiWAN для резервирования каналов связи	55
Резервирование каналов доступа в интернет.....	56
Создание резервного канала Ethernet или Wi-Fi.....	56

Создание резервного канала 3G/4G	58
Создание двух резервных каналов	61
Мониторинг состояния каналов связи	65
Настройка мониторинга по SNMP	66
Настройка мониторинга по Syslog	67
Глава 7. Повышение отказоустойчивости ViPNet Coordinator IG	69
Организация кластера	70
Типовая схема организации кластера	70
Кластер с беспроводными интерфейсами	74
Кластер с использованием шлюза Modbus и интерфейса GPIO	78
Кластер с использованием DHCP-сервера и DHCP-relay	81
Кластер с резервированием канала доступа в интернет	82
Организация агрегированного канала	87
Агрегированный канал между ViPNet Coordinator IG и коммутатором	87
Агрегированный канал между двумя ViPNet Coordinator IG	90
Глава 8. Использование сервисных функций ViPNet Coordinator IG	92
Организация обработки трафика из нескольких VLAN	93
Организация работы клиентов с локальным или удалённым DHCP-сервером	95
Организация работы клиентов с локальным DHCP-сервером	95
Организация работы клиентов с удалённым DHCP-сервером	96
Использование DHCP-сервера и DHCP-relay в разных сетях	98
Использование запасного DHCP-сервера	101
Использование сторонних DHCP-серверов	102
Одновременное использование DHCP-сервера и DHCP-relay	104
Организация работы клиентов удалённых офисов с DNS- и NTP-серверами, расположенными в центральном офисе	106
Защита соединения между удалёнными сегментами сети на канальном уровне модели OSI	108
Настройка L2OverIP при отсутствии VLAN	108
Настройка L2OverIP в случае использования VLAN	110
Настройка L2OverIP для обеспечения работоспособности протоколов динамической маршрутизации	113
Настройка параметров L2OverIP	115
Настройка протокола OSPF	116
Настройка резервирования каналов с помощью BGP	118
Использование ViPNet Coordinator IG в качестве защищённого интернет-шлюза	120
Приложение А. Термины и сокращения	122



Введение

О документе	6
Соглашения документа	7
Обратная связь	8

О документе

В документе описаны практические сценарии использования программно-аппаратного комплекса ViPNet Coordinator IG 5 (далее — ViPNet Coordinator IG), которые требуют комплексного применения различных команд и базовых схем настройки.

Соглашения документа

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



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

Обозначения при описании команд в документе:

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

Обратная связь

Контактная информация

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

Дополнительная информация на сайте ИнфоТеКС

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

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



1

ViPNet Coordinator IG как межсетевой экран сегмента промышленной сети

Защита сегмента промышленной сети	10
Размещение общедоступного сервера в демилитаризованной зоне	16
Настройка политик безопасности для разных режимов работы	18

Защита сегмента промышленной сети

С помощью ViPNet Coordinator IG вы можете защитить сегмент промышленной сети:

- ограничить взаимодействие сегмента с другими узлами сети;
- настроить фильтрацию трафика на прикладном уровне для промышленных протоколов Modbus и МЭК 60870-5-104.

Защита сегмента с устройствами Modbus

Исходные данные

Устройства, работающие по протоколу Modbus, выделены в отдельный сегмент промышленной сети с помощью ViPNet Coordinator IG (промышленное исполнение). Для подключения этих устройств к сети Ethernet используется [Modbus-преобразователь](#), который преобразует интерфейс Ethernet в интерфейс RS-485 и обратно.

На территории промышленного объекта расположен диспетчерский пункт, в котором размещён сервер для управления Modbus-устройствами.

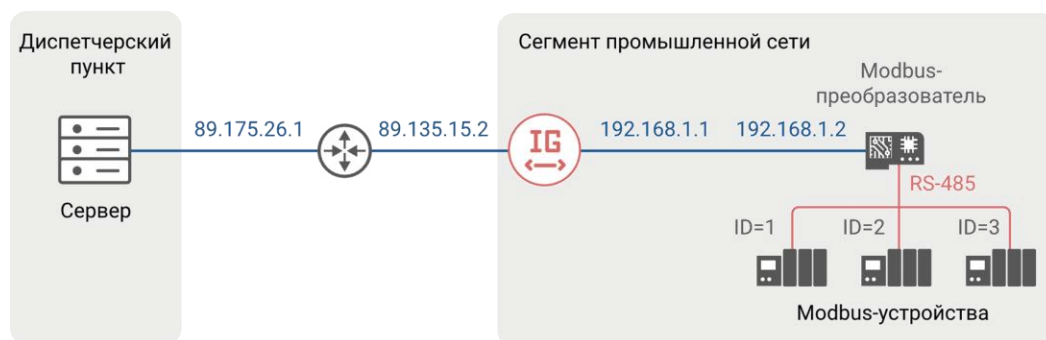


Рисунок 1. Защита сегмента промышленной сети с устройствами, работающими по протоколу Modbus

Задача

- Обеспечить доступ серверу диспетчерского пункта к Modbus-устройствам сегмента промышленной сети. Другие узлы сети не должны иметь доступа к сегменту промышленной сети.
- Запретить прохождение любых команд протокола Modbus, кроме разрешённых:
 - Для устройства с ID=1:
 - Команды с кодами функций 1–4 и чтением регистров с номерами 1–50.

- Команды с кодами функций 5, 6, 15 и записью в регистры с номерами 1–30.
- Для устройства с ID=2:
 - Команды с кодами функций 1–4 и чтением регистров с номерами 1–50.
 - Команды с кодами функций 5, 6, 15 и записью в регистры с номерами 3–30.
- Для устройства с ID=3:
 - Команды с кодами функций 5, 6, 15 и записью в регистры с номерами 1–30.
 - Команды с кодом функции 23 с чтением регистров с номерами 1–28 и записью в регистры с номерами 10–30.

Настройка

- 1 Создайте правило трансляции внешнего адреса ViPNet Coordinator IG во внутренний адрес Modbus-преобразователя при обращении сервера диспетчерского пункта к Modbus-устройствам:

```
hostname# firewall nat add src 89.175.26.1 dst 89.135.15.2 tcp dport 502 change dst 192.168.1.2:502
```

- 2 Создайте правило трансляции внутреннего адреса Modbus-преобразователя во внешний адрес ViPNet Coordinator IG при обращении к серверу диспетчерского пункта:

```
hostname# firewall nat add src 192.168.1.2 dst 89.175.26.1 change src 89.135.15.2
```

- 3 Создайте правила фильтрации протокола Modbus, пропускающие разрешённые команды:

- 3.1 Создайте набор правил с произвольным именем (например, RuleSet_01):

```
hostname# ipf modbus add rules RuleSet_01
```

- 3.2 В составе набора создайте правило транспортного уровня, разрешающее трафик от сервера диспетчерского пункта к Modbus-преобразователю:

```
hostname# ipf modbus rules RuleSet_01 fw add table forward client 89.175.26.1 server 192.168.1.2 tcp port 502
```

- 3.3 В составе набора создайте следующие правила прикладного уровня:

- правило Read_functions_pass, пропускающее все команды, которые запрашивают чтение регистров 1–50 от устройств с идентификаторами 1 и 2:

```
hostname# ipf modbus rules RuleSet_01 rule add 1 name Read_functions_pass id 1,2 fc @any r 1-50 pass
```

- правило Write_functions_drop, блокирующее команды с кодами функций 5, 6, 15, которые запрашивают запись регистров 1–2 от устройства с идентификатором 2:

```
hostname# ipf modbus rules RuleSet_01 rule add 2 name Write_functions_drop id 2 fc 5,6,15 w 1-2 drop
```

- правило Write_functions_pass, пропускающее команды с кодами функций 5, 6, 15, которые запрашивают запись регистров 1–30 от устройств с идентификаторами 1–3:

```
hostname# ipf modbus rules RuleSet_01 rule add 3 name Write_functions_pass id 1-3 fc 5,6,15 w 1-30 pass
```

- правило `Read_write_functions_pass`, пропускающее команду с кодом функции 23, которая запрашивает чтение регистров 1–28 и запись регистров 10–30 от устройства с идентификатором 3:

```
hostname# ipf modbus rules RuleSet_01 rule add 4 name Read_write_functions_pass
id 3 fc 23 r 1-28 w 10-30 pass
```



Примечание. Правила применяются в порядке их приоритета, который задается номером правила. Чем меньше номер, тем выше приоритет. Номер задается после лексемы `add`. Подробнее см. в документе «Настройка с помощью командного интерпретатора», в разделе «Основные принципы фильтрации промышленных протоколов».

Команды протокола Modbus, не соответствующие ни одному из разрешающих правил прикладного уровня, будут блокироваться правилом по умолчанию.

4 Включите фильтрацию протокола Modbus:

```
hostname# ipf modbus mode on
```

Остальной трафик будет блокироваться сетевым фильтром по умолчанию.

Защита сегмента с устройствами МЭК 60870-5-104

Исходные данные

Сегмент промышленной сети включает в себя датчики и счётчики. На границе сегмента установлен ViPNet Coordinator IG, выполняющий роль межсетевого экрана.

На территории промышленного объекта расположен диспетчерский пункт, в котором размещён сервер для сбора показателей датчиков и счётчиков.

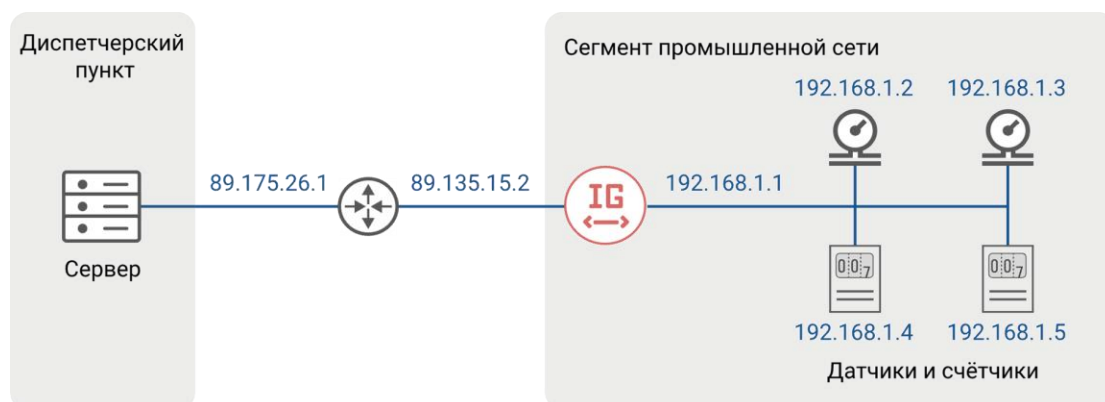


Рисунок 2. Защита сегмента с устройствами, работающими по протоколу МЭК 60870-5-104

Задача

- Обеспечить доступ серверу сбора данных к датчикам и счётчикам защищённого сегмента промышленной сети. Другие узлы сети не должны иметь доступа к сегменту промышленной сети.
- Разрешить серверу сбора данных взаимодействовать по протоколу МЭК 60870-5-104 со счётчиками для получения телесигнализации с адресами 18688–18690, которая передается кадрами с идентификатором 30 (одноэлементная информация с меткой времени СР56Время 2а). Счётчики имеют адреса ASDU 1 и 2. При передаче должен использоваться следующий формат кадра МЭК 60870-5-104:
 - Размер поля, содержащего общий адрес ASDU: 2 байта;
 - Размер поля, содержащего адрес объекта информации: 3 байта;
 - Размер поля, содержащего причину передачи ASDU: 1 байт;
 - Максимальный размер кадра: 253 байта.

Настройка

- 1 Создайте правила трансляции публичного адреса ViPNet Coordinator IG в частные адреса приборов при обращении сервера сбора данных (для доступа к каждому прибору будет использоваться отдельный порт из диапазона 10000–10003):

```
hostname# firewall nat add src 89.175.26.1 dst 89.135.15.2 tcp dport 10000 change dst 192.168.1.2:2404
```

```
hostname# firewall nat add src 89.175.26.1 dst 89.135.15.2 tcp dport 10001 change dst 192.168.1.3:2404
```

```
hostname# firewall nat add src 89.175.26.1 dst 89.135.15.2 tcp dport 10002 change dst 192.168.1.4:2404
```

```
hostname# firewall nat add src 89.175.26.1 dst 89.135.15.2 tcp dport 10003 change dst 192.168.1.5:2404
```

- 2 Создайте транзитный фильтр открытой сети, разрешающий передачу информации от сервера сбора данных к датчикам по порту 2404:

```
hostname# firewall forward add src 89.175.26.1 dst 192.168.1.2-192.168.1.3 tcp dport 2404 pass
```

- 3 Создайте правила фильтрации, разрешающие трафик от сервера сбора данных к счетчикам по протоколу МЭК 60870-5-104:

- 3.1 Создайте набор правил с произвольным именем (например, RuleSet_01):

```
hostname# ipf iec104 add rules RuleSet_01
```

- 3.2 В составе набора создайте правило транспортного уровня, разрешающее трафик от сервера диспетчерского пункта к счётчикам:

```
hostname# ipf iec104 rules RuleSet_01 fw add table forward client 89.175.26.1 server 192.168.1.4-192.168.1.5 tcp port 2404,10000-10003
```

- 3.3 Используемый формат кадра соответствует настройкам по умолчанию, кроме размера поля, содержащего причину передачи ASDU. Задайте размер поля, равный 1 байту:

```
hostname# ipf iec104 rules RuleSet_01 iecformat set cot 1
```

- 3.4 В составе набора создайте правило прикладного уровня, пропускающее сигналы с адресами 18688–18690 и идентификатором типа 30 до счётчиков с адресами ASDU 1 и 2:

```
hostname# ipf iec104 rules RuleSet_01 rule add 1 asdu 1-2 ioa 18688-18690 iot 30 pass
```

Команды протокола МЭК 60870-5-104, не соответствующие ни одному из разрешающих правил прикладного уровня, будут блокироваться правилом по умолчанию.

- 4 Включите фильтрацию протокола МЭК 60870-5-104:

```
hostname# ipf iec104 mode on
```

Остальной трафик будет блокироваться сетевым фильтром по умолчанию.

Доступ из защищённого сегмента

Исходные данные

Сегмент промышленной сети включает в себя датчики и счётчики. На границе сегмента установлен ViPNet Coordinator IG (промышленное исполнение), выполняющий роль межсетевого экрана. Датчики и счётчики подключаются к ViPNet Coordinator IG по технологии Wi-Fi.

На территории промышленного объекта расположен диспетчерский пункт, в котором размещён веб-сервер для сбора показаний с датчиков и счётчиков.

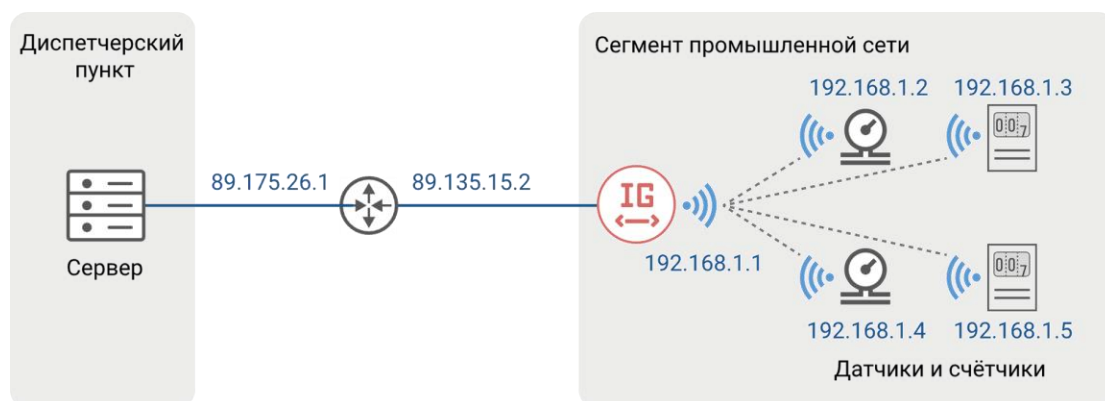


Рисунок 3. Организация доступа датчиков к внешнему веб-серверу с помощью ViPNet Coordinator IG

Задача

Обеспечить доступ датчикам и счётчикам защищённого сегмента промышленной сети к веб-серверу. Остальной трафик должен блокироваться.

Настройка

- 1 Создайте правило трансляции внутренних адресов датчиков во внешний адрес ViPNet Coordinator IG при обращении к веб-серверу:

```
hostname# firewall nat add src 192.168.1.2-192.168.1.5 dst 89.175.26.1 change src  
89.135.15.2
```

- 2 Создайте транзитный фильтр, разрешающий передачу трафика от датчиков к веб-серверу по протоколам HTTP и HTTPS:

```
hostname# firewall forward add src 192.168.1.2-192.168.1.5 dst 89.175.26.1 tcp dport  
80 tcp dport 443 pass
```

Остальной трафик будет блокироваться сетевым фильтром по умолчанию.

Размещение общедоступного сервера в демилитаризованной зоне

Наиболее уязвимыми для сетевых атак являются узлы (серверы), которые взаимодействуют с внешними системами или предоставляют различные сервисы внешним устройствам и пользователям, например, сервер обновлений (репозиторий ПО). В случае атаки на такие серверы под угрозой оказываются остальные узлы и устройства сети. Чтобы защитить сеть, её разбивают на сегменты, размещая серверы в демилитаризованной зоне. Эту зону отделяют от остальной сети межсетевым экраном, который контролирует трафик между сегментами. При этом сервисы, предоставляемые серверами, доступны как внешним устройствам и пользователям, так и находящимся во внутреннем сегменте сети. В то же время внутренний сегмент сети недоступен пользователям и устройствам извне.

Исходные данные

ViPNet Coordinator IG установлен между сервером обновлений, внутренним сегментом промышленной сети и интернетом. Сетевые интерфейсы ViPNet Coordinator IG, подключённые к серверу обновлений и внутреннему сегменту, имеют частные IP-адреса, а интерфейс, подключённый к интернету, имеет публичный IP-адрес.

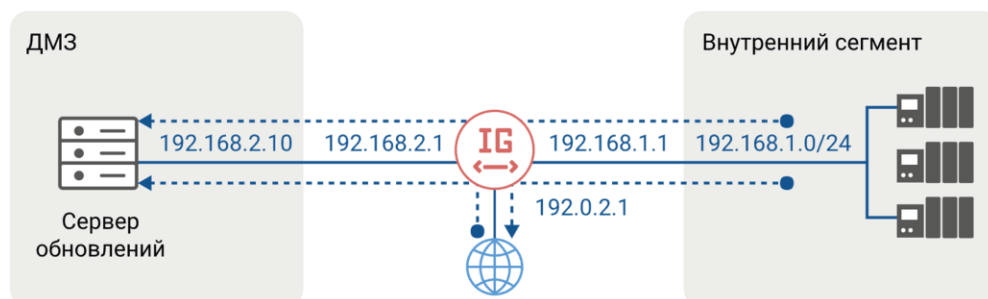


Рисунок 4. Использование ViPNet Coordinator IG для размещения общедоступного сервера в демилитаризованной зоне (ДМЗ)

Задача

- Обеспечить прохождение IP-пакетов:
 - из внутреннего сегмента в интернет;
 - из внутреннего сегмента к серверу обновлений;
 - из интернета к серверу обновлений.
- Блокировать остальной трафик.

Настройка

1 Создайте транзитные фильтры открытой сети:

- чтобы разрешить соединения из внутреннего сегмента к любым узлам назначения:

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

- чтобы разрешить соединения с сервером обновлений по протоколу HTTP с любых IP-адресов:

```
hostname# firewall forward add src @any dst 192.168.2.10 tcp dport 80 pass
```

Остальной транзитный трафик будет блокироваться сетевым фильтром по умолчанию.

2 Создайте правила трансляции адресов:

- для доступа в интернет из внутреннего сегмента:

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

- для доступа к серверу обновлений из интернета:

```
hostname# firewall nat add src @InternetIP dst 192.0.2.1 tcp dport 80 change dst 192.168.2.10
```

В результате узлы внутреннего сегмента промышленной сети и внешние узлы смогут обращаться к серверу обновлений. Узлы сегмента промышленной сети будут иметь доступ в интернет, при этом они будут защищены от атак из интернета.

Настройка политик безопасности для разных режимов работы

Если ViPNet Coordinator IG работает как межсетевой экран типа «Д», вы можете переключать режимы его работы. Предусмотрено три режима:

- **Штатный.** В этом режиме работают все функции ViPNet Coordinator IG. Недоступны: обновление ПО, справочников и ключей, создание и загрузка копий конфигураций VPN. В штатный режим ViPNet Coordinator IG переходит после установки справочников и ключей.
- **Регламентного обслуживания.** Этот режим предназначен для обновления ПО, справочников и ключей, создания и загрузки копий конфигураций VPN, а также регламентного обслуживания защищаемых узлов.
- **Специальный.** Этот режим предназначен для чрезвычайных ситуаций в работе системы. Сетевые фильтры по умолчанию в этом режиме настроены пропускать любой сетевой трафик. Обновление ПО, справочников и ключей, создание и загрузка копий конфигураций VPN в этом режиме недоступны.

С процедурами обновления ПО, справочников и ключей, а также с порядком создания и загрузки копий конфигураций VPN вы можете ознакомиться в документе «Настройка с помощью командного интерпретатора».

Каждому из режимов соответствует свой набор сетевых фильтров (см. документ «Настройка с помощью командного интерпретатора», приложение «Сетевые фильтры по умолчанию»). Все ваши действия с пользовательскими сетевыми фильтрами (создание, изменение, удаление) применяются только для текущего режима работы меж сетевого экрана.

Рассмотрим пример, когда требуется настроить сетевые фильтры ViPNet Coordinator IG таким образом, чтобы в штатном режиме работы ViPNet Coordinator IG пропускал только Modbus-трафик, а в режиме регламентного обслуживания — Modbus-трафик и трафик протокола HTTPS (для настройки и удалённого обновления ПО оборудования и узлов АСУ ТП). Предположим, что сетевые фильтры для Modbus-трафика уже настроены, тогда чтобы настроить фильтрацию для HTTPS-трафика:

- 1 Если межсетевой экран ViPNet Coordinator IG работает в режиме, отличном от штатного, переведите его в штатный режим:

```
hostname# machine mode set normal
```
- 2 Создайте сетевой фильтр, запрещающий транзитный трафик по протоколу HTTPS:

```
hostname# firewall forward add src @any dst @any service @HTTPS drop
```
- 3 Переведите межсетевой экран ViPNet Coordinator IG в режим регламентного обслуживания:

```
hostname# machine mode set service
```
- 4 Создайте сетевой фильтр, разрешающий транзитный трафик по протоколу HTTPS:

```
hostname# firewall forward add src @any dst @any service @HTTPS pass
```

5 Переведите межсетевой экран ViPNet Coordinator IG обратно в штатный режим:

```
hostname# machine mode set normal
```

После этого, для удалённой настройки или обновления ПО на оборудовании, защищаемом ViPNet Coordinator IG, достаточно перевести межсетевой экран ViPNet Coordinator IG в режим регламентного обслуживания, а по завершении работ — обратно в штатный режим.

2

Построение защищённого канала связи

Защищённое взаимодействие сегментов промышленной сети	21
Удалённый доступ клиента к защищённому сегменту промышленной сети	24

Защищённое взаимодействие сегментов промышленной сети

Исходные данные

Оборудование на территории промышленного объекта взаимодействует с АРМ инженера и SCADA-сервером, расположенными в диспетчерском пункте. Обмен информацией осуществляется через открытые каналы связи.

На границе сети диспетчерского пункта установлено серверное исполнение ViPNet Coordinator IG (IG-1), а на границе промышленного объекта — промышленное исполнение ViPNet Coordinator IG (IG-2), которые входят в одну сеть ViPNet.

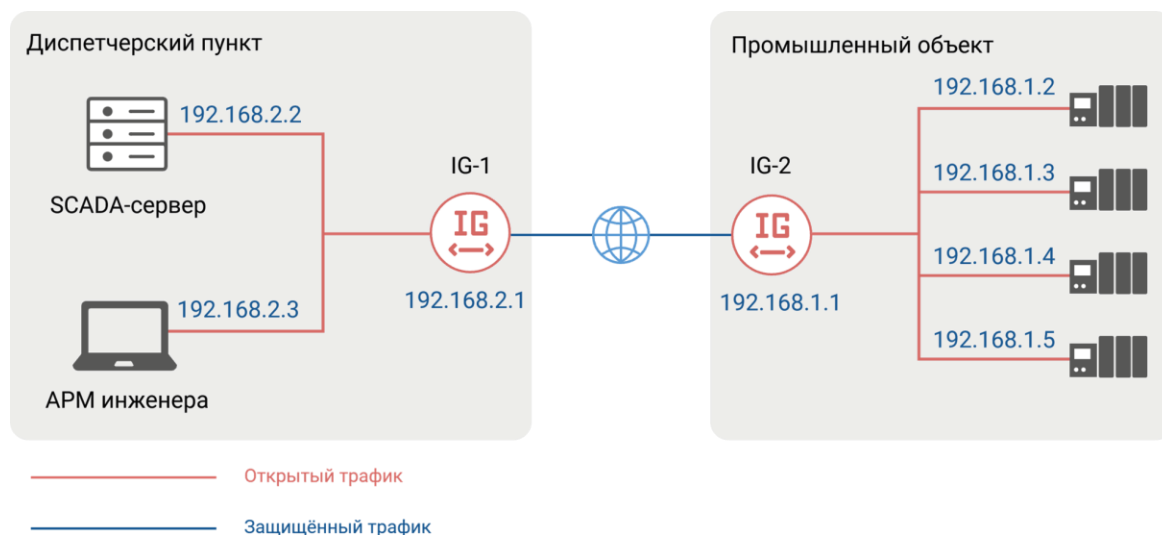


Рисунок 5. Организация защищенного туннеля к сегменту промышленной сети с помощью ViPNet Coordinator IG

Задача

Защитить канал передачи данных между диспетчерским пунктом и промышленным объектом с помощью технологии туннелирования.

Настройка

- 1 На компьютере инженера и SCADA-сервере в качестве шлюза по умолчанию задайте частный адрес IG-1, на оборудовании промышленного объекта — частный адрес IG-2.
- 2 На обоих координаторах задайте адреса узлов, которые туннелируют они сами и противоположный координатор. Адреса **туннелируемых узлов** можно задать двумя способами:
 - Централизованно в ViPNet Prime. В этом случае для каждого координатора задаются только адреса узлов, которые туннелирует он сам. После рассылки обновлений

справочников и ключей заданные адреса появятся в настройках обоих координаторов. Этот способ является предпочтительным.

Примечание. Если ViPNet Coordinator IG работает как межсетевой экран типа «Д»:



- Перед рассылкой справочников и ключей переключите его в режим регламентного обслуживания: `hostname# machine mode set service`
- После обновления справочников и ключей переключите его обратно в штатный режим: `hostname# machine mode set normal`

-
- Локально на ViPNet Coordinator IG путём редактирования файла конфигурации `iplir.conf`.

Для локального задания туннелируемых узлов:

2.1 На IG-1:

2.1.1 Завершите работу службы `iplircfg`:

```
hostname# iplir stop
```

2.1.2 Откройте файл `iplir.conf` для редактирования:

```
hostname# iplir config
```

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

```
tunnel= 192.168.2.2-192.168.2.3 to 192.168.2.2-192.168.2.3
```

2.1.4 В секцию `[id]`, описывающую IG-2, добавьте параметр:

```
tunnel= 192.168.1.2-192.168.1.5 to 192.168.1.2-192.168.1.5
```

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

2.1.6 Запустите службу `iplircfg`:

```
hostname# iplir start
```

2.2 На IG-2:

2.2.1 Завершите работу службы `iplircfg`:

```
hostname# iplir stop
```

2.2.2 Откройте файл `iplir.conf` для редактирования:

```
hostname# iplir config
```

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

```
tunnel= 192.168.1.2-192.168.1.5 to 192.168.1.2-192.168.1.5
```

2.2.4 В секцию `[id]`, описывающую IG-1, добавьте параметр:

```
tunnel= 192.168.2.2-192.168.2.3 to 192.168.2.2-192.168.2.3
```

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

2.2.6 Запустите службу `iplircfg`:

```
hostname# iplir start
```

- 3 В настройках межсетевого экрана координаторов для всех туннелируемых узлов задан разрешающий фильтр по умолчанию. Из соображений безопасности мы рекомендуем удалить этот фильтр и создать разрешающие фильтры для конкретных узлов.

3.1 Для просмотра фильтров туннелируемых узлов:

```
hostname# firewall tunnel show
```

3.2 Удалите фильтры туннелируемых узлов на обоих координаторах:

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

3.3 Добавьте фильтры туннелируемых узлов на обоих координаторах:

```
hostname# firewall tunnel add src 192.168.1.2-192.168.1.5 dst  
192.168.2.2-192.168.2.3 pass
```

```
hostname# firewall tunnel add src 192.168.2.2-192.168.2.3 dst  
192.168.1.2-192.168.1.5 pass
```

Эти фильтры разрешают трафик в обоих направлениях между узлами диспетчерского пункта и оборудованием промышленного объекта.



Примечание. Если оборудование работает по протоколу Modbus TCP или МЭК 60870-5-104, вы можете настроить фильтрацию трафика на прикладном уровне. Подробнее см. [Защита сегмента промышленной сети](#).

Если при этом необходимо запретить трафик, не относящийся к промышленному протоколу, удалите фильтры туннелируемых узлов, добавленные выше.

В результате на участке между IG-1 и IG-2 трафик будет передаваться в зашифрованном виде.

Удалённый доступ клиента к защищённому сегменту промышленной сети

Исходные данные

АРМ инженера сторонней обслуживающей компании удалённо подключается к оборудованию промышленного объекта по открытым каналам связи. Для защиты этого соединения на границе сегмента промышленной сети установлен ViPNet Coordinator IG, а на АРМ инженера — ПО ViPNet Client, которые входят в одну сеть ViPNet.

АРМ инженера в сети ViPNet имеет идентификатор узла 0x00010201. При этом из соображений безопасности этот клиент имеет связь в сети ViPNet только с ViPNet Coordinator IG.



Рисунок 6. Организация доступа удалённого клиента к защищённому сегменту промышленной сети с помощью ViPNet Coordinator IG

Задача

Защитить канал передачи данных между АРМ инженера и оборудованием промышленного объекта с помощью технологии туннелирования.

Настройка

- 1 На ViPNet Coordinator IG задайте адреса туннелируемых узлов одним из способов:
 - Централизованно в ViPNet Prime. В этом случае для ViPNet Coordinator IG задаются адреса узлов, которые он туннелирует. После рассылки обновлений справочников и ключей

заданные адреса появятся в настройках координаторов и клиентов. Этот способ является предпочтительным.

Примечание. Если ViPNet Coordinator IG работает как межсетевой экран типа «Д»:



- Перед рассылкой справочников и ключей переключите его в режим регламентного обслуживания: `hostname# machine mode set service`
 - После обновления справочников и ключей переключите его обратно в штатный режим: `hostname# machine mode set normal`
-

- Локально на ViPNet Coordinator IG и ViPNet Client.

Для локального задания туннелируемых узлов:

- 1.1 Завершите работу службы `iplircfg`:

```
hostname# iplir stop
```

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

```
hostname# iplir config
```

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

```
tunnel= 192.168.1.2-192.168.1.5 to 192.168.1.2-192.168.1.5
```

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

- 1.5 Запустите службу `iplircfg`:

```
hostname# iplir start
```

- 2 В настройках ViPNet Coordinator IG для всех туннелируемых узлов задан разрешающий фильтр по умолчанию. Из соображений безопасности рекомендуется удалить данный фильтр и создать разрешающие фильтры для конкретных узлов. Для этого выполните действия:

- 2.1 Для просмотра фильтров туннелируемых узлов:

```
hostname# firewall tunnel show
```

- 2.2 Удалите фильтры туннелируемых узлов:

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

- 2.3 Добавьте следующие фильтры туннелируемых узлов:

```
hostname# firewall tunnel add src 192.168.1.2-192.168.1.5 dst 0x00010201 pass
```

```
hostname# firewall tunnel add src 0x00010201 dst 192.168.1.2-192.168.1.5 pass
```

Эти фильтры разрешают трафик в обоих направлениях между туннелируемыми узлами и защищённым АРМ инженера.

- 3 Если адреса туннелируемых узлов заданы локально на ViPNet Coordinator IG, то на АРМ инженера в ViPNet Client задайте в качестве IP-адресов для туннелирования адреса 192.168.1.2–192.168.1.5 для узла ViPNet Coordinator IG (подробнее см. в документации на ViPNet Client).

В результате между ViPNet Coordinator IG и АРМ инженера трафик будет передаваться в зашифрованном виде.

3

Использование встроенного шлюза Modbus

Взаимодействие устройств Modbus RTU с сервером сбора данных	27
Защищённое взаимодействие устройств Modbus RTU с удалёнными сетевыми узлами	29
Защищённый удалённый доступ клиента к устройствам Modbus RTU	32
Защищённое удалённое взаимодействие устройств Modbus RTU	34
Доверенное взаимодействие между двумя последовательными сетями	38
Доверенное взаимодействие между несколькими последовательными сетями	41

Взаимодействие устройств Modbus RTU с сервером сбора данных

Исходные данные

В сегменте промышленной сети находятся устройства, работающие по протоколу Modbus RTU и использующие интерфейс подключения RS-485. На границе сегмента установлен ViPNet Coordinator IG (промышленное исполнение).

На территории промышленного объекта расположен диспетчерский пункт, в котором размещён сервер для управления Modbus-устройствами.

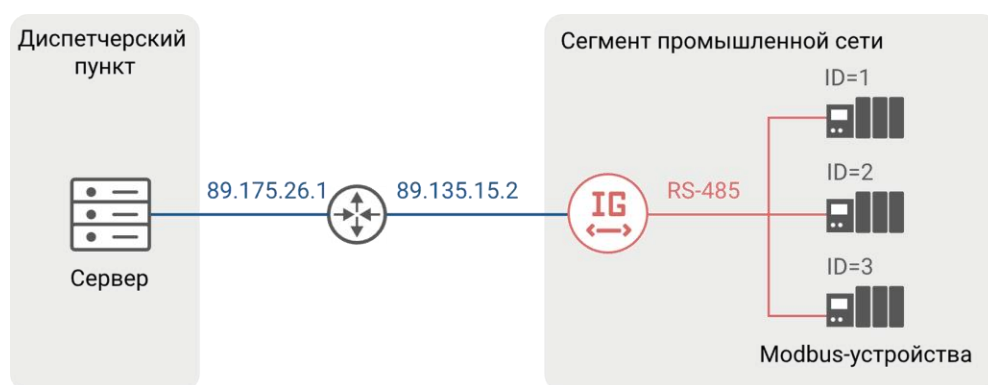


Рисунок 7. Организация взаимодействия Modbus-устройств и диспетчерского пункта с помощью ViPNet Coordinator IG

Задача

- Настроить взаимодействие между Modbus-устройствами и сервером диспетчерского пункта с помощью встроенного [шлюза Modbus](#).
- Запретить прохождение любого трафика в сегмент промышленной сети, кроме Modbus-трафика.

Настройка

- 1 Настройте шлюз Modbus на последовательном интерфейсе RS-485 с произвольным именем (например, GW-485) и TCP-портом 502:

```
hostname# service mbgw add name GW-485 serial rs485 port 502
```

По умолчанию заданы следующие настройки шлюза Modbus:

- Режим работы: TCP to RTU;
- Время ожидания запроса: 1500 мс;
- Время ожидания ответа: 1500 мс;
- Скорость TTY: 115200 бод;

- Контроль чётности: even;
- Задержка до отправки: 100 мс;
- Задержка после отправки: 0 мс.

2 Чтобы шлюз Modbus автоматически запускался при загрузке ViPNet Coordinator IG:

```
hostname# service mbgw name GW-485 mode on
```

3 Запустите шлюз Modbus:

```
hostname# service mbgw name GW-485 start
```

4 Разрешите прохождение трафика, относящегося к протоколу Modbus:

4.1 Создайте набор правил фильтрации с произвольным именем (например, RuleSet_01):

```
hostname# ipf modbus add rules RuleSet_01
```

4.2 В составе набора создайте правило транспортного уровня, разрешающее трафик от сервера диспетчерского пункта до шлюза Modbus:

```
hostname# ipf modbus rules RuleSet_01 fw add table local client 89.175.26.1 server 89.135.15.2 tcp port 502
```

4.3 В составе набора создайте правило прикладного уровня (например, All_pass), пропускающее любой Modbus-трафик:

```
hostname# ipf modbus rules RuleSet_01 rule add 1 name All_pass id @any fc @any pass
```

4.4 Включите фильтрацию протокола Modbus:

```
hostname# ipf modbus mode on
```



Примечание. Если необходимо настроить фильтрацию Modbus-трафика по адресам устройств, кодам функций, регистрам чтения и записи, см. [Защита сегмента промышленной сети](#).

Остальной трафик будет блокироваться сетевым фильтром по умолчанию.

Защищённое взаимодействие устройств Modbus RTU с удалёнными сетевыми узлами

Исходные данные

На территории промышленного объекта находятся устройства, работающие по протоколу Modbus RTU и использующие интерфейс подключения RS-485. В диспетчерском пункте расположены сервер и АРМ инженера.

На границе сети диспетчерского пункта установлено серверное исполнение ViPNet Coordinator IG (IG-1), а на границе сети промышленного объекта — промышленное исполнение ViPNet Coordinator IG (IG-2), которые входят в одну сеть ViPNet. Координатор IG-1 в защищённой сети имеет идентификатор узла 0x00010201.

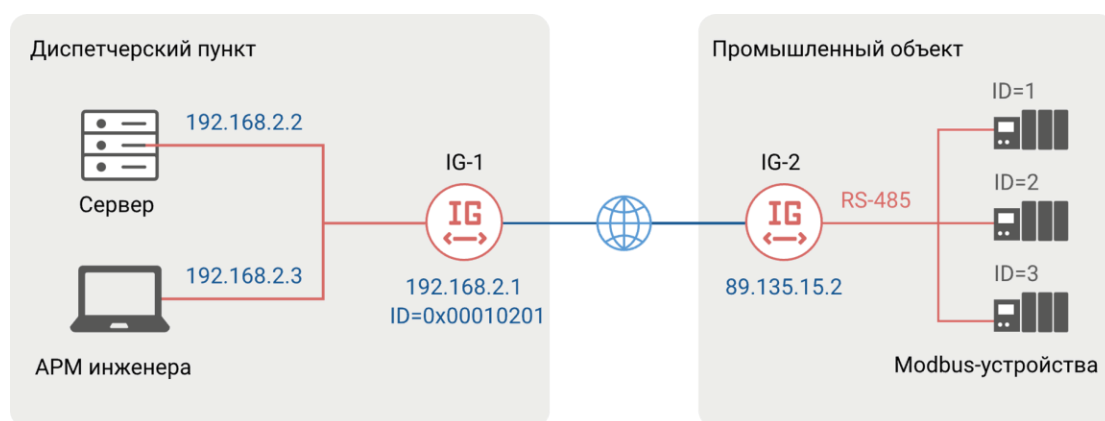


Рисунок 8. Организация защищённого взаимодействия Modbus-устройств и диспетчерского пункта с помощью ViPNet Coordinator IG

Задача

- Настроить взаимодействие между Modbus-устройствами промышленного объекта и диспетчерским пунктом с помощью встроенного шлюза Modbus.
- Защитить канал передачи данных между диспетчерским пунктом и промышленным объектом с помощью технологии туннелирования.
- Запретить прохождение любого трафика к Modbus-устройствам, кроме трафика, относящегося к протоколу Modbus.

Настройка

- 1 На IG-2 настройте шлюз Modbus:

- 1.1 Настройте шлюз Modbus на последовательном интерфейсе RS-485 с произвольным именем (например, GW-485) и TCP-портом 502:

```
hostname# service mbgw add name GW-485 serial rs485 port 502
```

По умолчанию заданы следующие настройки шлюза Modbus:

- Режим работы: TCP to RTU;
- Время ожидания запроса: 1500 мс;
- Время ожидания ответа: 1500 мс;
- Скорость TTY: 115200 бод;
- Контроль чётности: even;
- Задержка до отправки: 100 мс;
- Задержка после отправки: 0 мс.

- 1.1 Чтобы шлюз Modbus автоматически запускался при загрузке ViPNet Coordinator IG:

```
hostname# service mbgw name GW-485 mode on
```

- 1.2 Запустите шлюз Modbus:

```
hostname# service mbgw name GW-485 start
```

- 2 На IG-2 разрешите прохождение трафика, относящегося к протоколу Modbus:

- 2.1 Создайте набор правил фильтрации с произвольным именем (например, RuleSet_01):

```
hostname# ipf modbus add rules RuleSet_01
```

- 2.2 В составе набора создайте правило транспортного уровня, разрешающее защищённый трафик от IG-1 диспетчерского пункта до шлюза Modbus:

```
hostname# ipf modbus rules RuleSet_01 fw add table vpn client 0x00010201 server @local tcp port 502
```

- 2.3 В составе набора создайте правило прикладного уровня (например, All_pass), пропускающее любой Modbus-трафик:

```
hostname# ipf modbus rules RuleSet_01 rule add 1 name All_pass id @any fc @any pass
```

- 2.4 Включите фильтрацию протокола Modbus:

```
hostname# ipf modbus mode on
```



Примечание. Если необходимо настроить фильтрацию Modbus-трафика по адресам устройств, кодам функций, регистрам чтения и записи, см. [Защита сегмента промышленной сети](#).

Остальной трафик будет блокироваться сетевым фильтром по умолчанию.

- 3 В ViPNet Prime настройте туннелирование сервера и АРМ инженера координатором IG-1:

- 3.1 Добавьте IP-адрес сервера и IP-адрес АРМ инженера в список адресов туннелируемых узлов IG-1.

- 3.2 Выполните рассылку обновлений справочников и ключей.

- 4 В настройках IG-1 для всех туннелируемых узлов задан разрешающий фильтр по умолчанию. Из соображений безопасности рекомендуется удалить данный фильтр и создать разрешающие фильтры для конкретных узлов:

4.1 Для просмотра фильтров туннелируемых узлов:

```
hostname# firewall tunnel show
```

4.2 Удалите фильтры туннелируемых узлов:

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

4.3 Добавьте следующие фильтры туннелируемых узлов:

```
hostname# firewall tunnel add src 192.168.2.2-192.168.2.3 dst 89.135.15.2 pass
```

```
hostname# firewall tunnel add src 89.135.15.2 dst 192.168.2.2-192.168.2.3 pass
```

Защищённый удалённый доступ клиента к устройствам Modbus RTU

Исходные данные

На территории промышленного объекта находятся устройства, работающие по протоколу Modbus RTU и использующие интерфейс подключения RS-485.

Для организации защищённого доступа с АРМ инженера сторонней обслуживающей компании к Modbus-устройствам на границе сети этого объекта установлен ViPNet Coordinator IG (промышленное исполнение), а на АРМ инженера — ПО ViPNet Client, которые входят в одну сеть ViPNet.

АРМ инженера в сети ViPNet имеет идентификатор узла 0x00010201. При этом из соображений безопасности этот клиент имеет связь в сети ViPNet только с ViPNet Coordinator IG.

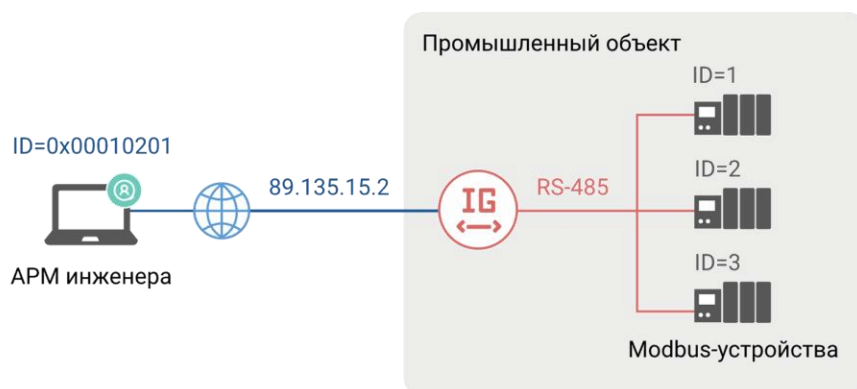


Рисунок 9. Организация защищенного взаимодействия Modbus-устройств и АРМ инженера с помощью ViPNet Coordinator IG

Задача

- Настроить взаимодействие между АРМ инженера и Modbus-устройствами промышленного объекта с помощью встроенного шлюза Modbus.
- Защитить канал передачи данных между АРМ инженера и Modbus-устройствами промышленного объекта.
- Запретить прохождение любого трафика к Modbus-устройствам, кроме трафика, относящегося к протоколу Modbus.

Настройка

- 1 Настройте шлюз Modbus на последовательном интерфейсе RS-485 с произвольным именем (например, GW-485) и TCP-портом 502:

```
hostname# service mbgw add name GW-485 serial rs485 port 502
```

По умолчанию заданы следующие настройки шлюза Modbus:

- Режим работы: TCP to RTU;
- Время ожидания запроса: 1500 мс;
- Время ожидания ответа: 1500 мс;
- Скорость ТТУ: 115200 бод;
- Контроль чётности: even;
- Задержка до отправки: 100 мс;
- Задержка после отправки: 0 мс.

2 Чтобы шлюз Modbus автоматически запускался при загрузке ViPNet Coordinator IG:

```
hostname# service mbgw name GW-485 mode on
```

3 Запустите шлюз Modbus:

```
hostname# service mbgw name GW-485 start
```

4 Разрешите прохождение трафика, относящегося к протоколу Modbus:

4.1 Создайте набор правил фильтрации с произвольным именем (например, RuleSet_01):

```
hostname# ipf modbus add rules RuleSet_01
```

4.2 В составе набора создайте правило транспортного уровня, разрешающее защищённый трафик от АРМ инженера до шлюза Modbus:

```
hostname# ipf modbus rules RuleSet_01 fw add table vpn client 0x00010201 server  
@local tcp port 502
```

4.3 В составе набора создайте правило прикладного уровня (например, All_pass), пропускающее любой Modbus-трафик:

```
hostname# ipf modbus rules RuleSet_01 rule add 1 name All_pass id @any fc @any pass
```

4.4 Включите фильтрацию протокола Modbus:

```
hostname# ipf modbus mode on
```



Примечание. Если необходимо настроить фильтрацию Modbus-трафика по адресам устройств, кодам функций, регистрам чтения и записи, см. [Защита сегмента промышленной сети](#).

Остальной трафик будет блокироваться сетевым фильтром по умолчанию.

Защищённое удалённое взаимодействие устройств Modbus RTU

Исходные данные

Промышленная сеть разделена на два территориально удалённых сегмента. В каждом сегменте размещены устройства, работающие по протоколу Modbus и использующие интерфейс подключения RS-485. При этом в одном из сегментов установлен промышленный сервер, который является Modbus RTU Master. Обмен информацией между оборудованием в двух сегментах промышленной сети осуществляется в открытом виде по последовательным линиям связи.

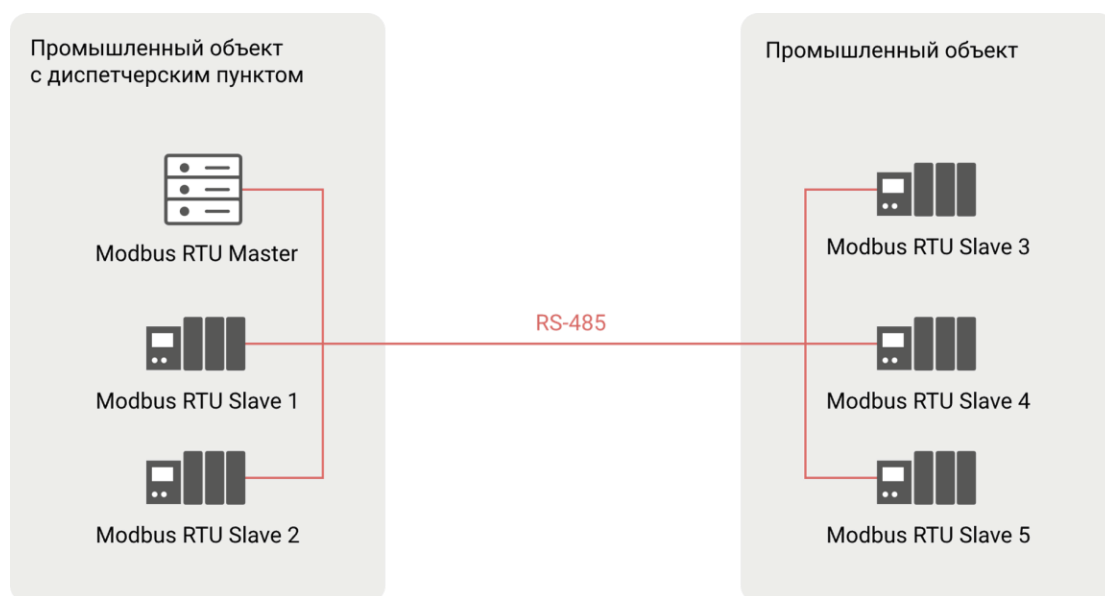


Рисунок 10. Связь между сегментами промышленной сети до модернизации

Задача

Защитить участок промышленной сети между сегментами.

Порядок действий

- 1 Установите на границе каждого сегмента ViPNet Coordinator IG (промышленное исполнение).
- 2 Убедитесь, что оба ViPNet Coordinator IG связаны между собой в сети ViPNet. Вы можете сделать это в ViPNet Prime.
- 3 Настройте шлюз Modbus и фильтрацию трафика (см. ниже).

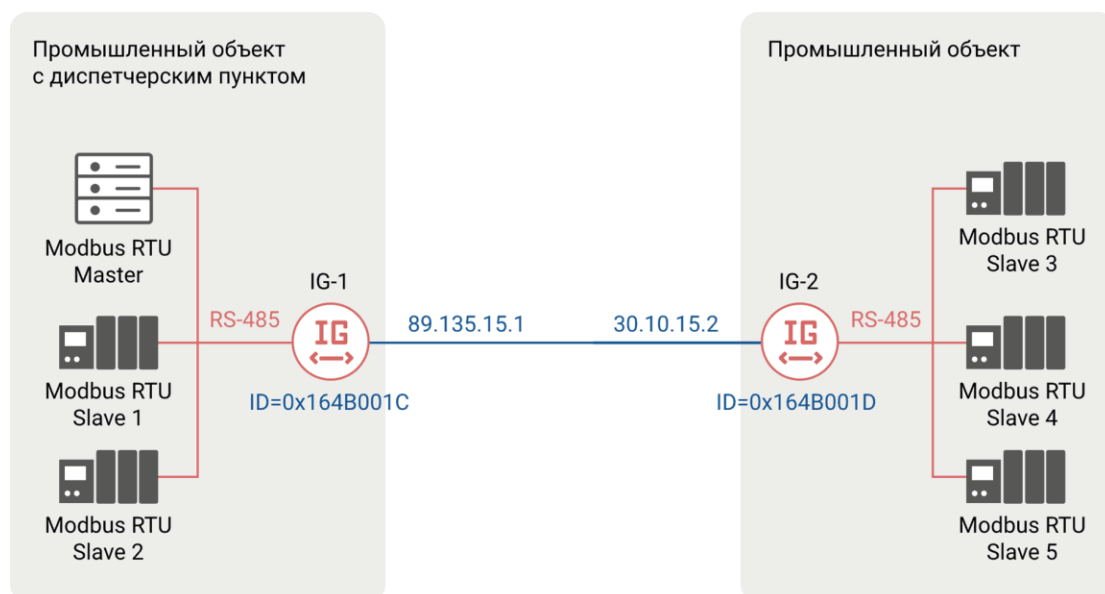


Рисунок 11. Организация защищенного взаимодействия между сегментами промышленной сети с помощью ViPNet Coordinator IG

Настройка

1 На координаторе IG-1:

- 1.1 Настройте шлюз Modbus на последовательном интерфейсе RS-485 с произвольным именем (например, GW-485) и TCP-портом 502:

```
hostname# service mbgw add name GW-485 serial rs485 port 502
```

По умолчанию заданы следующие настройки шлюза Modbus:

- Режим работы: TCP to RTU;
- Время ожидания запроса: 1500 мс;
- Время ожидания ответа: 1500 мс;
- Скорость ТТУ: 115200 бод;
- Контроль чётности: even;
- Задержка до отправки: 100 мс;
- Задержка после отправки: 0 мс.

- 1.1 Измените режим работы шлюза Modbus на RTU to TCP:

```
hostname# service mbgw name GW-485 type rtu-to-tcp
```

- 1.2 Настройте маршрутизацию Modbus-устройств 3, 4 и 5 из сегмента сети удалённого промышленного объекта через внешний интерфейс IG-2:

```
hostname# service mbgw name GW-485 routing add 3-5 ip 30.10.15.2
```

- 1.3 Чтобы шлюз Modbus автоматически запускался при загрузке ViPNet Coordinator IG:

```
hostname# service mbgw name GW-485 mode on
```

- 1.4 Запустите шлюз Modbus:

```
hostname# service mbgw name GW-485 start
```

- 1.5 Создайте набор правил фильтрации с произвольным именем (например, RuleSet_01):

```
hostname# ipf modbus add rules RuleSet_01
```

- 1.6 В составе набора создайте правило транспортного уровня, разрешающее исходящий Modbus-трафик в адрес координатора IG-2, так как инициатором соединения является сервер Modbus RTU Master:

```
hostname# ipf modbus rules RuleSet_01 fw add table vpn client @local server  
0x164B001D tcp port 502
```

- 1.7 В составе набора создайте правило прикладного уровня (например, All_pass), пропускающее любой Modbus-трафик:

```
hostname# ipf modbus rules RuleSet_01 rule add 1 name All_pass id @any fc @any pass
```

- 1.8 Включите фильтрацию протокола Modbus:

```
hostname# ipf modbus mode on
```



Примечание. Если необходимо настроить фильтрацию Modbus-трафика по адресам устройств, кодам функций, регистрам чтения и записи, см. [Защита сегмента промышленной сети](#).

Остальной трафик будет блокироваться сетевым фильтром по умолчанию.

2 На координаторе IG-2:

- 2.1 Настройте шлюз Modbus на последовательном интерфейсе RS-485 с произвольным именем (например, GW-485) и TCP-портом 502:

```
hostname# service mbgw add name GW-485 serial rs485 port 502
```

По умолчанию уже задан режим работы TCP to RTU.

- 2.2 Чтобы шлюз Modbus автоматически запускался при загрузке ViPNet Coordinator IG:

```
hostname# service mbgw name GW-485 mode on
```

- 2.3 Запустите шлюз Modbus:

```
hostname# service mbgw name GW-485 start
```

- 2.4 Создайте набор правил фильтрации с произвольным именем (например, RuleSet_01):

```
hostname# ipf modbus add rules RuleSet_01
```

- 2.5 В составе набора создайте правило транспортного уровня, разрешающее входящий Modbus-трафик из сегмента сети с сервером Modbus RTU Master:

```
hostname# ipf modbus rules RuleSet_01 fw add table vpn client 0x164B001C server  
@local tcp port 502
```

- 2.6 В составе набора создайте правило прикладного уровня (например, All_pass), пропускающее любой Modbus-трафик:

```
hostname# ipf modbus rules RuleSet_01 rule add 1 name All_pass id @any fc @any pass
```

- 2.7 Включите фильтрацию протокола Modbus:

```
hostname# ipf modbus mode on
```



Примечание. Если необходимо настроить фильтрацию Modbus-трафика по адресам устройств, кодам функций, регистрам чтения и записи, см. [Защита сегмента промышленной сети](#).

Остальной трафик будет блокироваться сетевым фильтром по умолчанию.

В результате на участке между координаторами IG-1 и IG-2 трафик будет передаваться в зашифрованном виде.

Доверенное взаимодействие между двумя последовательными сетями

Исходные данные

Промышленная сеть разделена на два сегмента. В сегменте 1 размещён диспетчерский пункт, представляющий из себя SCADA-сервер. В сегменте 2 — устройства, работающие по протоколу Modbus RTU и использующие интерфейс подключения RS-485. SCADA-сервер является Modbus RTU Master и использует интерфейс подключения RS-232. Между сегментами установлен ViPNet Coordinator IG (промышленное исполнение).

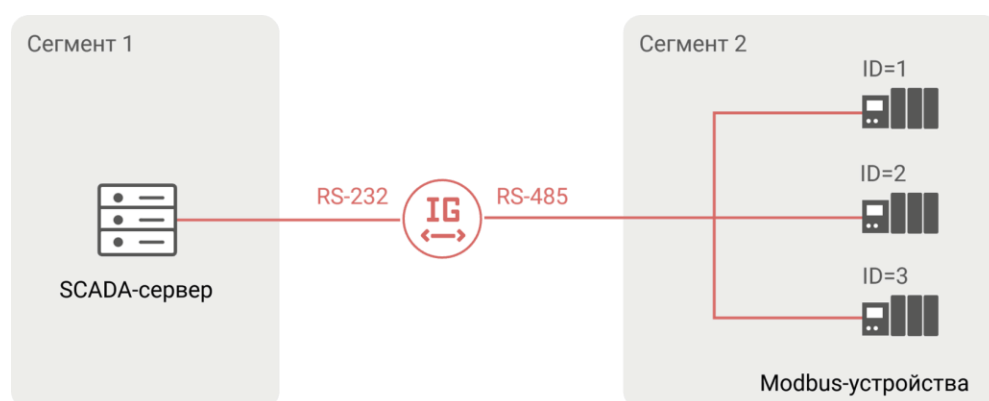


Рисунок 12. Организация доверенного взаимодействия между двумя последовательными сетями

Задача

- Разрешить серверу диспетчерского пункта чтение регистров со всех Modbus-устройств сегмента 2.
- Разрешить серверу диспетчерского пункта передавать команды с кодами функций 5, 6, 15 на устройства с ID=2 и ID=3.
- Запретить серверу диспетчерского пункта передавать иные команды, которые не разрешены.

Настройка

- 1 Настройте шлюз Modbus на последовательном интерфейсе RS-232 с произвольным именем и TCP-портом, например GW-232 и порт 502:

```
hostname# service mbgw add name GW-232 serial rs232 port 502
```

По умолчанию заданы следующие настройки шлюза Modbus:

- Режим работы: TCP to RTU;

- Время ожидания запроса: 1500 мс;
- Время ожидания ответа: 1500 мс;
- Скорость ТТУ: 115200 бод;
- Контроль чётности: even.

2 Измените режим работы шлюза Modbus на RTU to TCP:

```
hostname# service mbgw name GW-232 type rtu-to-tcp
```

3 Настройте шлюз Modbus на последовательном интерфейсе RS-485 с произвольным именем и TCP-портом, например GW-485 и порт 503:

```
hostname# service mbgw add name GW-485 serial rs485 port 503
```

4 Выберите любой сетевой интерфейс Ethernet, например eth0. При этом не важно, используется он или нет. Убедитесь, что он включён и на нём задан статический IP-адрес:

```
hostname# inet show interface eth0
eth0  flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.1  netmask 255.255.255.0 broadcast 192.168.0.255
...
```

5 Настройте маршрутизацию к Modbus-устройствам 1, 2 и 3 от сервера диспетчерского пункта через сетевой интерфейс eth0:

```
hostname# service mbgw name GW-232 routing add 1-3 ip 192.168.1.1 port 503
```

В параметре port указан порт шлюза GW-485.

6 Включите автозапуск и запустите оба шлюза Modbus:

```
hostname# service mbgw name GW-232 mode on
hostname# service mbgw name GW-232 start
hostname# service mbgw name GW-485 mode on
hostname# service mbgw name GW-485 start
```

7 Разрешите передачу команд от сервера диспетчерского пункта к Modbus-устройствам в соответствии с поставленной задачей:

7.1 Создайте набор правил фильтрации с произвольным именем, например RuleSet_01:

```
hostname# ipf modbus add rules RuleSet_01
```

7.2 В составе набора создайте правило транспортного уровня, указав в качестве клиента и сервера IP-адрес интерфейса eth0, а также порт шлюза GW-485:

```
hostname# ipf modbus rules RuleSet_01 fw add table local client 192.168.1.1 server
192.168.1.1 tcp port 503
```

7.3 В составе набора создайте правило прикладного уровня, например ID_read, разрешающее серверу диспетчерского пункта выполнять чтение регистров всех Modbus-устройств сегмента 2:

```
hostname# ipf modbus rules RuleSet_01 rule add 1 name ID_read id 1-3 fc 1-4,23 r
@any pass
```

- 7.4 В составе набора создайте правило прикладного уровня, например `ID_write`, разрешающее серверу диспетчерского пункта передавать команды с кодами функций 5, 6 и 15 на Modbus-устройства с ID=2 и ID=3:

```
hostname# ipf modbus rules RuleSet_01 rule add 2 name ID_write id 2,3 fc 5,6,15  
pass
```

- 7.5 Включите фильтрацию протокола Modbus:

```
hostname# ipf modbus mode on
```

Остальной Modbus-трафик будет блокироваться правилом фильтрации по умолчанию.

Доверенное взаимодействие между несколькими последовательными сетями

Исходные данные

Промышленная сеть разделена на несколько сегментов. В сегменте 1 размещён диспетчерский пункт, представляющий из себя SCADA-сервер. В других сегментах — устройства, работающие по протоколу Modbus RTU и использующие интерфейс подключения RS-232. SCADA-сервер является Modbus RTU Master и использует интерфейс RS-485. Между сегментом 1 и каждым Modbus-устройством из других сегментов установлен ViPNet Coordinator IG (промышленное исполнение).

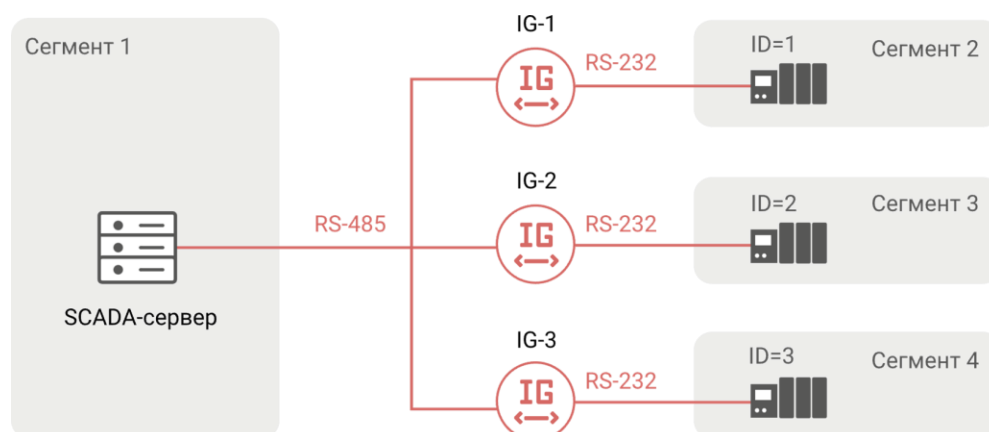


Рисунок 13. Организация доверенного взаимодействия между несколькими последовательными сетями

Задача

- Разрешить серверу диспетчерского пункта чтение регистров со всех Modbus-устройств.
- Разрешить серверу диспетчерского пункта передавать команды с кодами функций 5, 6, 15 на устройства с ID=2 и ID=3.
- Запретить серверу диспетчерского пункта передавать иные команды, которые не разрешены.

Настройка

На каждом ViPNet Coordinator IG:

- 1 Настройте шлюз Modbus на последовательном интерфейсе RS-485 с произвольным именем и TCP-портом, например GW-485 и порт 502:

```
hostname# service mbgw add name GW-485 serial rs485 port 502
```

По умолчанию заданы следующие настройки шлюза Modbus:

- Режим работы: TCP to RTU;
- Время ожидания запроса: 1500 мс;
- Время ожидания ответа: 1500 мс;
- Скорость ТТУ: 115200 бод;
- Контроль чётности: even;
- Задержка до отправки: 100 мс;
- Задержка после отправки: 0 мс.

2 Измените режим работы шлюза Modbus на RTU to TCP:

```
hostname# service mbgw name GW-485 type rtu-to-tcp
```

3 Настройте шлюз Modbus на последовательном интерфейсе RS-232 с произвольным именем и TCP-портом, например GW-232 и порт 503:

```
hostname# service mbgw add name GW-232 serial rs232 port 503
```

4 Выберите любой сетевой интерфейс Ethernet, например eth0. При этом не важно, используется он или нет. Убедитесь, что он включён и на нём задан статический IP-адрес:

```
hostname# inet show interface eth0

eth0  flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.1  netmask 255.255.255.0 broadcast 192.168.0.255
...
```

5 Настройте маршрутизацию к Modbus-устройствам 1, 2 и 3 от сервера диспетчерского пункта через сетевой интерфейс eth0:

- На IG-1:

```
hostname# service mbgw name GW-485 routing add 1 ip 192.168.1.1 port 503
```

- На IG-2:

```
hostname# service mbgw name GW-485 routing add 2 ip 192.168.1.1 port 503
```

- На IG-3:

```
hostname# service mbgw name GW-485 routing add 3 ip 192.168.1.1 port 503
```

В параметре port указан порт шлюза GW-232.

6 Включите автозапуск и запустите оба шлюза Modbus:

```
hostname# service mbgw name GW-485 mode on
hostname# service mbgw name GW-485 start
hostname# service mbgw name GW-232 mode on
hostname# service mbgw name GW-232 start
```

7 Разрешите передачу команд от сервера диспетчерского пункта к Modbus-устройствам в соответствии с поставленной задачей:

7.1 Создайте набор правил фильтрации с произвольным именем, например RuleSet_01:

```
hostname# ipf modbus add rules RuleSet_01
```

- 7.2 В составе набора создайте правило транспортного уровня, указав в качестве клиента и сервера IP-адрес интерфейса `eth0`, а также порт шлюза `GW-232`:

```
hostname# ipf modbus rules RuleSet_01 fw add table local client 192.168.1.1 server 192.168.1.1 tcp port 503
```

- 7.3 В составе набора создайте правило прикладного уровня, например `ID_read`, разрешающее серверу диспетчерского пункта выполнять чтение регистров всех Modbus-устройств:

- На IG-1:

```
hostname# ipf modbus rules RuleSet_01 rule add 1 name ID_read id 1 fc 1-23 r @any pass
```

- На IG-2:

```
hostname# ipf modbus rules RuleSet_01 rule add 1 name ID_read id 2 fc 1-23 r @any pass
```

- На IG-3:

```
hostname# ipf modbus rules RuleSet_01 rule add 1 name ID_read id 3 fc 1-23 r @any pass
```

- 7.4 В составе набора создайте правило прикладного уровня, например `ID_write`, разрешающее серверу диспетчерского пункта передавать команды с кодами функций 5, 6 и 15 на Modbus-устройства с ID=2 и ID=3:

- На IG-2:

```
hostname# ipf modbus rules RuleSet_01 rule add 2 name ID_write id 2 fc 5,6,15 pass
```

- На IG-3:

```
hostname# ipf modbus rules RuleSet_01 rule add 2 name ID_write id 3 fc 5,6,15 pass
```

- 7.5 Включите фильтрацию протокола Modbus:

```
hostname# ipf modbus mode on
```

Остальной Modbus-трафик будет блокироваться правилом фильтрации по умолчанию.



4

Использование технологии PoE

Защита канала связи между сервером и видеокамерами с PoE

45

Защита канала связи между сервером и видеокамерами с PoE

Исходные данные

На территории промышленного объекта установлены две цифровые видеокamеры. К месту их установки невозможно провести сеть электропитания, поэтому камеры получают питание от коммутатора по технологии Power over Ethernet (PoE). Коммутатор установлен в другом помещении и подключен к сети электропитания.

Таблица 1. Характеристики видеокamер

Видеокамера	Класс мощности	Максимальная мощность потребления, Вт
Камера 1	2	6,5
Камера 2	3	13

В диспетчерском пункте расположен сервер видеонаблюдения, принимающий видеопоток от камер.

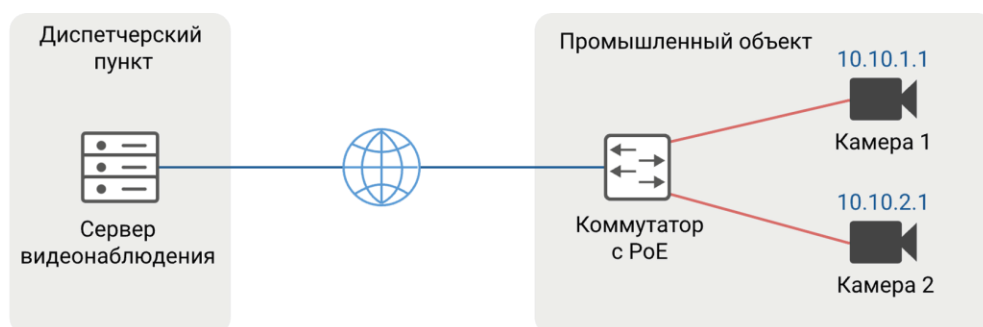


Рисунок 14. Схема системы видеонаблюдения

Задача

- Защитить канал передачи данных между камерами и сервером видеонаблюдения.
- Обеспечить быстрое восстановление работоспособности камеры 2 от сбоев.

Порядок действий

- 1 Установите исполнение ViPNet Coordinator IG с поддержкой PoE (далее — IG-2) на место коммутатора с PoE:
 - 1.1 Подключите Ethernet-кабель, который связывал коммутатор с интернетом, к порту **WAN** на корпусе ViPNet Coordinator IG.
 - 1.2 Подключите Ethernet-кабели от камер к портам **LAN** на корпусе ViPNet Coordinator IG.

1.3 Заземлите и подключите питание к ViPNet Coordinator IG.

2 Установите любой координатор (далее — IG-1) на границе сети диспетчерского пункта.

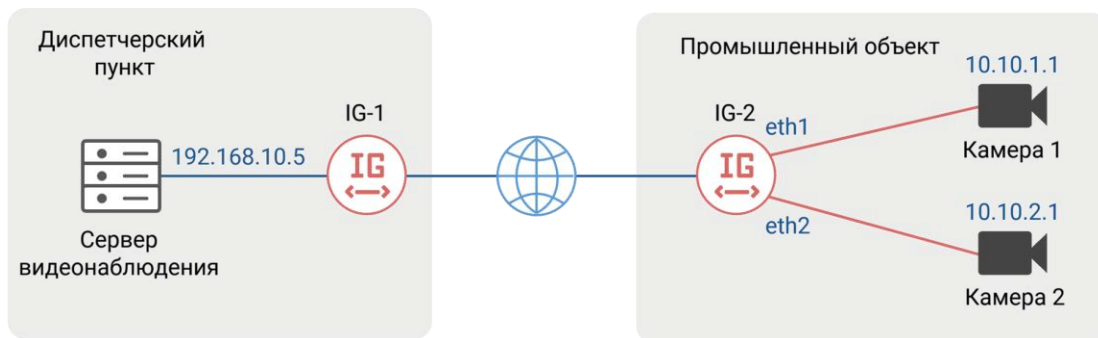


Рисунок 15. Защита канала связи между сервером и видеокameraми

Настройка

1 Настройте подачу питания на камеры, подключённые к IG-2:

- 1.1 По умолчанию на ViPNet Coordinator IG установлено ограничение на подачу питания — 7 Вт на каждый порт LAN. Чтобы обеспечить питанием камеру 2 с потребляемой мощностью 13 Вт, измените ограничение для eth2 до 14 Вт:

```
hostname# poe interface eth2 limit 14
```

- 1.2 По умолчанию питание подаётся сначала на eth1, а потом на eth2. Чтобы в первую очередь обеспечить питанием камеру 2, поменяйте порядок подачи питания на интерфейсы:

```
hostname# poe interface eth2 priority 1
```

```
hostname# poe interface eth1 priority 2
```



Примечание. Если ViPNet Coordinator IG будет получать питание от коммутатора по технологии PoE, он сможет выделить не более 15 Вт суммарно на оба интерфейса. Поэтому, если суммарная потребляемая мощность двух камер больше 15 Вт, питание будет подано только на одну камеру в соответствии с настроенными приоритетами.

- 1.3 В случае перезагрузки ViPNet Coordinator IG питание на камеры будет подано только после полной загрузки ПО ViPNet (несколько минут). Чтобы быстрее запустить камеру 2, настройте подачу питания на интерфейс eth2 до загрузки ПО:

```
hostname# poe interface eth2 faststart on
```

- 1.4 Включите подачу питания на интерфейсы eth2 и eth1:

```
hostname# poe interface eth2 power on
```

```
hostname# poe interface eth1 power on
```

2 Убедитесь, что камеры запустились.

3 Чтобы контролировать работоспособность камеры 2, настройте периодическую проверку её доступности:

- 3.1 Задайте IP-адрес, по которому ViPNet Coordinator IG будет проверять доступность камеры:

```
hostname# poe interface eth2 check-ip address 10.10.2.1
```

3.2 Включите проверку доступности камеры 2:

```
hostname# poe interface eth2 check-ip mode on
```

Теперь информация о доступности камеры 2 будет записываться в системный журнал ViPNet Coordinator IG.

- 4 Настройте автоматическую перезагрузку камеры 2 в случае отсутствия ответа с её IP-адреса. Для этого настройте прерывание подачи питания на камеру и возобновление питания через одну секунду:

```
hostname# poe interface eth2 check-ip action add restart 1
```

- 5 Убедитесь, что настройки PoE выполнены верно:

```
hostname# poe show
```

Описание вывода команды см. в документе «Справочник команд и конфигурационных файлов».

- 6 В ViPNet Prime настройте туннелирование камер и сервера видеонаблюдения:

6.1 Добавьте IP-адрес сервера 192.168.10.5 в список туннелируемых узлов IG-1.

6.2 Добавьте IP-адреса камер 10.10.1.1 и 10.10.2.1 в список туннелируемых узлов IG-2.

Примечание. Если ViPNet Coordinator IG работает как межсетевой экран типа «Д»:



- Перед настройкой туннелирования переключите ViPNet Coordinator IG в режим регламентного обслуживания: `hostname# machine mode set service`
 - После обновления справочников и ключей переключите его обратно в штатный режим: `hostname# machine mode set normal`
-

- 7 В настройках координаторов для всех туннелируемых узлов задан разрешающий фильтр по умолчанию. Из соображений безопасности рекомендуется удалить данный фильтр и создать разрешающие фильтры только для конкретных туннелируемых узлов. Для этого:

- 7.1 Просмотрите номера фильтров туннелируемых узлов:

```
hostname# firewall tunnel show
```

- 7.2 Удалите разрешающие фильтры туннелируемых узлов по одному с помощью команды:

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

- 7.3 Добавьте фильтр, разрешающий трафик от камер к серверу видеонаблюдения:

```
hostname# firewall tunnel add src 10.10.1.1,10.10.2.1 dst 192.168.10.5 pass
```



5

Использование интерфейса GPIO

Подключение датчика вскрытия внешнего шкафа	49
Подключение внешнего устройства для перехода в специальный режим	51
Настройка сигнализации для администратора информационной безопасности	53

Подключение датчика вскрытия внешнего шкафа

Для организации технического контроля за местом установки ViPNet Coordinator IG вы можете установить ViPNet Coordinator IG в специальный внешний шкаф с датчиком контроля вскрытия. Вы можете подключить датчик вскрытия шкафа, где установлен ViPNet Coordinator IG, к интерфейсу GPIO на ViPNet Coordinator IG.

Подключите датчик вскрытия к входному разъёму интерфейса GPIO на корпусе ViPNet Coordinator IG, как показано на рисунке ниже. Параметры разъёма GPIO (дискретного входа «IN») на корпусе ViPNet Coordinator IG:

- уровень логического 0 — от +5 В до +30 В (для исполнения ViPNet Coordinator IG100 I4 — от +2 В до +30 В);
- уровень логической 1 — от 0 В до +1 В;
- входной ток — не более 4 мА.

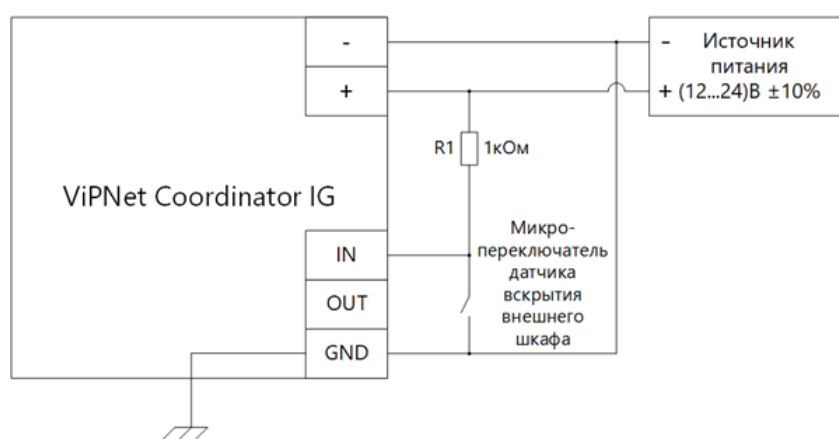


Рисунок 16. Подключение датчика вскрытия внешнего шкафа

В работе интерфейса GPIO ViPNet Coordinator IG предусмотрена реакция на сигнал, полученный от датчика вскрытия внешнего шкафа, в который смонтирован ViPNet Coordinator IG.

Настройка

- 1 Остановите работу интерфейса GPIO:

```
hostname# service piomonitor stop
```

- 2 Задайте активный уровень сигнала 1 (соответствующий состоянию «шкаф вскрыт») для входного контакта GPIO:

```
hostname# service piomonitor set In activelevel 1
```

При закрытом шкафу, на входном контакте GPIO будет сигнал уровня 0.

- 3 Настройте обработку сигнала от датчика вскрытия внешнего шкафа для входного контакта GPIO:

```
hostname# service piomonitor set In tamper
```

- 4 Чтобы ViPNet Coordinator IG сигнализировал о зарегистрированном событии вскрытия шкафа на подключённое к выходному разъёму GPIO устройство сигнализации, задайте активный уровень сигнала 1 для отправки на выходной контакт GPIO:

```
hostname# service piomonitor set Out activelevel 1
hostname# service piomonitor set Out event on tamper
```

- 5 Чтобы информация о вскрытии внешнего шкафа передавалась во внешнюю систему для последующей обработки, настройте отправку сообщений системного журнала:

```
hostname# machine set loghost 192.168.10.10
```

где 192.168.10.10 — IP-адрес внешней системы (можно указать доменное имя).

- 6 Включите автозапуск и запустите работу интерфейса GPIO:

```
hostname# service piomonitor mode on
hostname# service piomonitor start
```

В результате выполненных действий ViPNet Coordinator IG будет фиксировать события вскрытия внешнего шкафа в системный журнал, отправлять их во внешнюю систему для последующей обработки, и сигнализировать на подключённые к выходному контакту устройства сигнализации.

События имеют вид:

Месяц Дата Часы:минуты:секунды год (гггг) Имя_узла piomonitord[номер_процесса]: Сообщение

Например: Nov 12 17:20:24 2020 ig100-347e000d piomonitor[2491]: Input pin In tamper flag set to false

Подробнее о настройке сигнализации для администратора информационной безопасности см. в разделе [Настройка сигнализации для администратора информационной безопасности](#).

Подключение внешнего устройства для перехода в специальный режим

Если ViPNet Coordinator IG работает как межсетевой экран типа «Д», вы можете настроить его перевод в специальный режим по сигналу от внешнего устройства. На промышленных объектах, где установлена АСУ ТП, как правило имеются меры, применяемые в случае возникновения аварийной ситуации. Одной из таких мер является обеспечение работы систем аварийной безопасности на объекте. При необходимости должностные лица, отвечающие за безопасность на объекте, могут быстро переключить фильтры межсетевого экрана на ViPNet Coordinator IG с помощью внешнего устройства. В зависимости от типа подключаемого внешнего устройства, «кнопка» или «переключатель», вы можете переключать режим одним из двух способов:

- С помощью устройства типа «кнопка» вы можете переключать ViPNet Coordinator IG только в специальный режим.
- С помощью устройства типа «переключатель» вы можете переключать ViPNet Coordinator IG как в специальный режим, так и обратно в режим, из которого произошло переключение в специальный режим.



Примечание. Переключить ViPNet Coordinator IG в штатный режим или режим регламентного обслуживания вы можете с помощью веб-интерфейса, командного интерпретатора или политики, полученной из ViPNet Prime Policy Management.

Подробнее о режимах работы ViPNet Coordinator IG см. в документе «Подготовка к работе», раздел «Режимы работы».

Рассмотрим пример такой настройки ViPNet Coordinator IG, к которому подключено внешнее устройство типа «кнопка»:

- 1 Остановите работу интерфейса GPIO:

```
hostname# service piomonitor stop
```

- 2 Задайте активный уровень сигнала 0 (соответствующий состоянию «кнопка не нажата») для входного контакта GPIO:

```
hostname# service piomonitor set In activelevel 0
```

- 3 Настройте обработку сигнала от внешней кнопки для перевода межсетевого экрана ViPNet Coordinator IG в специальный режим:

```
hostname# service piomonitor set In emergency button
```

- 4 Задайте активный уровень сигнала 1 (соответствующий состоянию «ПАК работает в специальном режиме») для отправки на выходной контакт GPIO:

```
hostname# service piomonitor set Out activelevel 1
```

5 Включите автозапуск и запустите работу интерфейса GPIO с помощью команд:

```
hostname# service piomonitor mode on  
hostname# service piomonitor start
```

В результате ViPNet Coordinator IG будет переключать межсетевой экран в специальный режим по нажатию внешней кнопки и записывать сообщение об этом событии в системный журнал.

При переходе в специальный режим ViPNet Coordinator IG применит набор сетевых фильтров, настроенных для специального режима. Подробнее о настройке сетевых фильтров см. в документе «Настройка с помощью командного интерпретатора», раздел «Настройка сетевых фильтров».

Подробнее о настройке сигнализации для администратора информационной безопасности см. в разделе [Настройка сигнализации для администратора информационной безопасности](#).

Настройка сигнализации для администратора информационной безопасности

ViPNet Coordinator IG может уведомлять администратора информационной безопасности о регистрируемых событиях с помощью внешнего устройства сигнализации (индикатора, сирены и т.д.), которое подключено к выходному разъёму интерфейса GPIO и размещено на рабочем месте администратора информационной безопасности. Например, если ViPNet Coordinator IG установлен во внешний шкаф, как описано в разделе [Подключение датчика вскрытия внешнего шкафа](#), или к ViPNet Coordinator IG подключено [внешнее устройство для перевода ViPNet Coordinator IG в специальный режим](#).

Вы можете настроить ViPNet Coordinator IG для отправки сигналов о следующих событиях:

- вскрытие внешнего шкафа;
- переход ViPNet Coordinator IG в один из режимов работы (при работе ViPNet Coordinator IG как межсетевого экрана типа «Д»);
- переключение кластера.

В случае наступления контролируемого события ViPNet Coordinator IG посредством интерфейса GPIO направляет сигнал на световые оповещатели, динамики или сирены, которые установлены в непосредственной близости от администратора информационной безопасности. В результате администратор безопасности может своевременно предпринять действия по устранению возникших угроз в соответствии с регламентом организации.

Настройка

1 Остановите работу интерфейса GPIO:

```
hostname# service piomonitor stop
```

2 Задайте активный уровень сигнала 1 (соответствующий состоянию «событие наступило») для выходного контакта GPIO:

```
hostname# service piomonitor set Out activelevel 1
```

3 Настройте отправку на выходной контакт GPIO исходящего сигнала, соответствующего одному из событий (вы можете назначить несколько событий для одного контакта):

- Чтобы отправлять на выходной контакт сигнал о переводе межсетевого экрана ViPNet Coordinator IG в специальный режим:

```
hostname# service piomonitor set Out event on emergency
```

Выполнение этой команды возможно только в случае, если ViPNet Coordinator IG работает как межсетевой экран типа «Д».

- Чтобы отправлять на выходной контакт GPIO сигнал о переводе межсетевого экрана ViPNet Coordinator IG в режим регламентного обслуживания:

```
hostname# service piomonitor set Out event on service
```

Выполнение этой команды возможно только в случае, если ViPNet Coordinator IG работает как межсетевой экран типа «Д».

- Чтобы отправлять на выходной контакт GPIO сигнал о переводе межсетевого экрана ViPNet Coordinator IG в режим ограниченного функционирования:

```
hostname# service piomonitor set Out event on restricted
```

- Чтобы отправлять на выходной контакт GPIO сигнал о вскрытии внешнего шкафа:

```
hostname# service piomonitor set Out event on tamper
```

- Чтобы отправлять на выходной контакт GPIO сигнал о переключении ViPNet Coordinator IG в активный режим при работе в кластере:

```
hostname# service piomonitor set Out event on cluster
```

4 Включите автозапуск и запустите работу интерфейса GPIO:

```
hostname# service piomonitor mode on
```

```
hostname# service piomonitor start
```

6

Использование функции MultiWAN для резервирования каналов СВЯЗИ

Резервирование каналов доступа в интернет	56
Мониторинг состояния каналов связи	65

Резервирование каналов доступа в интернет

С помощью ViPNet Coordinator IG вы можете обеспечить бесперебойную передачу данных между промышленным объектом и удаленным диспетчерским пунктом при использовании каналов связи нескольких интернет-провайдеров. Для этого ViPNet Coordinator IG поддерживает функцию резервирования каналов связи (MultiWAN). С помощью MultiWAN вы можете настроить автоматическое перенаправление трафика с одного канала на другой при потере соединения со шлюзом одного из интернет-провайдеров.

Создание резервного канала Ethernet или Wi-Fi

Исходные данные

Промышленный объект подключён к интернету через шлюзы двух интернет-провайдеров:

- основной канал — Ethernet (шлюз GW-1);
- резервный канал — Ethernet или Wi-Fi (шлюз GW-2).

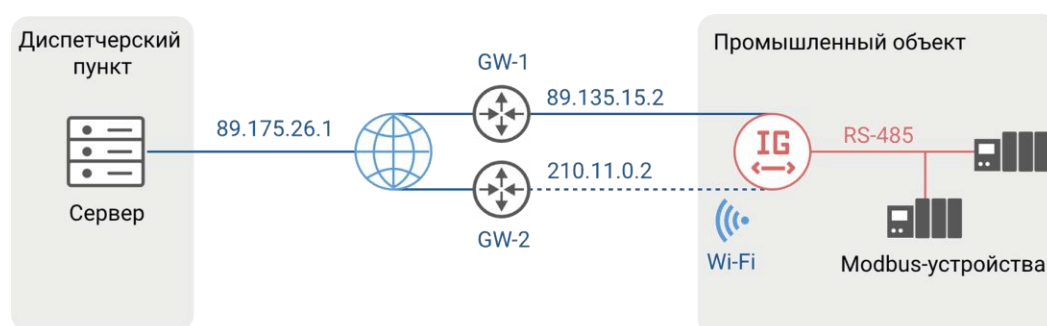


Рисунок 17. Схема резервирования канала доступа в интернет

Задача

Настроить автоматическое перенаправление трафика с одного канала на другой при потере соединения со шлюзом одного из интернет-провайдеров.

Настройка

- 1 Если вы используете DHCP для интерфейса основного или резервного канала, отключите получение маршрутов по умолчанию и перезагрузите интерфейс:

```
hostname# inet ifconfig <интерфейс> dhcp route off
```

```
hostname# inet ifconfig <интерфейс> down
```

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

2 Удалите маршруты по умолчанию из таблицы по умолчанию:

```
hostname# inet route delete default
```

3 Добавьте маршруты по умолчанию для шлюзов провайдеров GW-1 и GW-2:

```
hostname# inet route add default next-hop 89.135.15.2 table 1024 name GW-1
hostname# inet route add default next-hop 210.11.0.2 table 1025 name GW-2
```

4 Создайте политики маршрутизации, по которым трафик будет маршрутизироваться через шлюз GW-2 в случае недоступности шлюза GW-1:

4.1 Добавьте более приоритетные правила для таблицы по умолчанию:

```
hostname# inet policy rule add route-via-GW-1-policy 1024 table default
hostname# inet policy rule add route-via-GW-2-policy 1024 table default
```

Это необходимо для маршрутизации ответного трафика.

4.2 Добавьте менее приоритетные правила для пользовательских таблиц:

```
hostname# inet policy rule add route-via-GW-1-policy 1025 table name GW-1
hostname# inet policy rule add route-via-GW-2-policy 1025 table name GW-2
```



Примечание. Приоритет правил маршрутизации должен быть в диапазоне от 1024 до 2048. Чем меньше значение, тем выше приоритет. Общие правила маршрутизации, применяемые для всего трафика, должны иметь приоритет ниже, чем у правил, применяемых для конкретных диапазонов IP-адресов.

5 Запустите службу проверки состояния шлюзов (Dead Gateway Detection, DGD):

```
hostname# inet dgd mode on
```

6 Настройте проверку доступности шлюзов GW-1 и GW-2:

```
hostname# inet dgd next-hop add GW-1 address 89.135.15.2 check icmp
hostname# inet dgd next-hop add GW-2 address 210.11.0.2 check icmp
```



Примечание. Если вы настраиваете резервирование канала доступа в интернет в кластере, то для проверки доступности шлюзов вместо команды:

```
inet dgd next-hop add <имя шлюза> address <адрес шлюза> check icmp
```

используйте команду:

```
inet dgd next-hop add <имя шлюза> address <адрес шлюза> test-address
<тестовый адрес> check icmp
```

где в качестве <тестовый адрес> используйте IP-адрес любого стабильного узла, расположенного за шлюзом и не являющегося узлом ViPNet. А IP-адрес шлюза используйте в параметре `testip` при настройке кластера.

7 Создайте правила службы DGD, по которым действует политика маршрутизации

route-via-GW-1-policy, если доступен шлюз GW-1:

```
hostname# inet dgd rule add GW-1-up match-next-hop GW-1 up
hostname# inet dgd rule add GW-1-up action-priority 10 service route command policy
active route-via-GW-1-policy
```

- 8 Создайте правила службы DGD, по которым действует политика маршрутизации route-via-GW-2-policy, если доступен шлюз GW-2, а шлюз GW-1 недоступен:

```
hostname# inet dgd rule add GW-1-down match-next-hop GW-1 down
hostname# inet dgd rule add GW-1-down match-next-hop GW-2 up
hostname# inet dgd rule add GW-1-down action-priority 10 service route command policy
active route-via-GW-2-policy
```

- 9 Создайте правила службы DGD, по которым действует политика маршрутизации по умолчанию и создается запись в системном журнале, если недоступны шлюзы обоих провайдеров:

```
hostname# inet dgd rule add all-down match-next-hop GW-1 down
hostname# inet dgd rule add all-down match-next-hop GW-2 down
hostname# inet dgd rule add all-down action-priority 20 service log command
All-Routers-Down
hostname# inet dgd rule add all-down action-priority 10 service route command policy
active default
```

- 10 Перезапустите службу DGD для применения новых правил:

```
hostname# inet dgd mode off
hostname# inet dgd mode on
```

Создание резервного канала 3G/4G

Исходные данные

Промышленный объект подключён к интернету через шлюзы двух интернет-провайдеров:

- основной канал — Ethernet (шлюз GW-1);
- резервный канал — 3G/4G (шлюз GW-2).

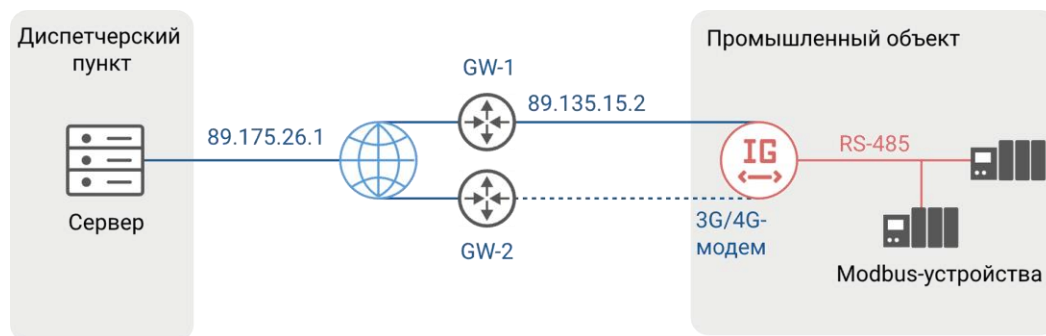


Рисунок 18. Схема резервирования канала доступа в интернет с использованием модема

Задача

Настроить автоматическое перенаправление трафика с одного канала на другой при потере соединения со шлюзом одного из интернет-провайдеров.

Настройка

- 1 Если вы используете DHCP для Ethernet-интерфейса, отключите получение маршрутов по умолчанию и перезагрузите интерфейс:

```
hostname# inet ifconfig <интерфейс> dhcp route off
hostname# inet ifconfig <интерфейс> down
hostname# inet ifconfig <интерфейс> up
```

- 2 Отключите модем:

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

- 3 Отключите получение маршрутов на модемном интерфейсе:

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

- 4 Задайте IP-адрес шлюза мобильного оператора, например, 10.64.64.64:

```
hostname# inet usb-modem ip-gw 10.64.64.64
```



Примечание. В качестве шлюза мобильного оператора, вы можете задать любой неиспользуемый IP-адрес, например, 10.255.255.254. Но в этом случае трафик до сетевых узлов с IP-адресами 10.*.*.* будет маршрутизироваться через модем.

- 5 Включите модем:

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

- 6 Удалите маршруты по умолчанию из таблицы по умолчанию:

```
hostname# inet route delete default
```

- 7 Добавьте маршрут по умолчанию для шлюзов GW-1 и GW-2:

```
hostname# inet route add default next-hop 89.135.15.2 table 1024 name GW-1
hostname# inet route add default next-hop 10.64.64.64 table 1025 name GW-2
```

- 8 Создайте политики маршрутизации, по которым трафик будет маршрутизироваться через шлюз GW-2 в случае недоступности шлюза GW-1:

- 8.1 Добавьте более приоритетные правила для таблицы по умолчанию:

```
hostname# inet policy rule add route-via-GW-2-policy 1024 table default
hostname# inet policy rule add route-via-GW-1-policy 1024 table default
```

Это необходимо для маршрутизации ответного трафика.

- 8.2 Добавьте менее приоритетные правила для пользовательских таблиц:

```
hostname# inet policy rule add route-via-GW-2-policy 1025 table name GW-2
hostname# inet policy rule add route-via-GW-1-policy 1025 table name GW-1
```



Примечание. Приоритет правил маршрутизации должен быть в диапазоне от 1024 до 2048. Чем меньше значение, тем выше приоритет. Общие правила маршрутизации, применяемые для всего трафика, должны иметь приоритет ниже, чем у правил, применяемых для конкретных диапазонов IP-адресов.

9 Запустите службу проверки состояния шлюзов (DGD):

```
hostname# inet dgd mode on
```

10 Настройте проверку доступности шлюза GW-1:

```
hostname# inet dgd next-hop add GW-1 address 89.135.15.2 check icmp
```

Примечание. Если вы настраиваете резервирование канала доступа в интернет в кластере, то для проверки доступности шлюзов вместо команды:

```
inet dgd next-hop add <имя шлюза> address <адрес шлюза> check icmp
```



используйте команду:

```
inet dgd next-hop add <имя шлюза> address <адрес шлюза> test-address  
<тестовый адрес> check icmp
```

где в качестве <тестовый адрес> используйте IP-адрес любого стабильного узла, расположенного за шлюзом и не являющегося узлом ViPNet. А IP-адрес шлюза используйте в параметре testip при настройке кластера.

11 Настройте проверку доступности шлюза GW-2:

```
hostname# inet dgd next-hop add GW-2 address 10.64.64.64 test-address 77.88.8.8 check  
icmp
```



Примечание. Для проверки работоспособности шлюза мобильного оператора мы рекомендуем в качестве тестового IP-адреса выбирать открытый стабильный узел во внешней сети. Поэтому в параметре test-address задан IP-адрес DNS-сервера Yandex (77.88.8.8) как одного из наиболее стабильных сетевых узлов в интернете.

12 Создайте правила службы DGD, по которым действует политика маршрутизации route-via-GW-1-policy, если доступен шлюз GW-1:

```
hostname# inet dgd rule add GW-1-up match-next-hop GW-1 up
```

```
hostname# inet dgd rule add GW-1-up action-priority 10 service route command policy  
active route-via-GW-1-policy
```

13 Создайте правила службы DGD, по которым действует политика маршрутизации route-via-GW-2-policy, если доступен шлюз GW-2, а шлюз GW-1 недоступен:

```
hostname# inet dgd rule add GW-1-down match-next-hop GW-1 down
```

```
hostname# inet dgd rule add GW-1-down match-next-hop GW-2 up
```

```
hostname# inet dgd rule add GW-1-down action-priority 10 service route command policy  
active route-via-GW-2-policy
```

14 Создайте правила службы DGD, по которым действует политика маршрутизации по умолчанию и создается запись в системном журнале, если недоступны шлюзы обоих провайдеров:

```
hostname# inet dgd rule add all-down match-next-hop GW-1 down
```

```
hostname# inet dgd rule add all-down match-next-hop GW-2 down
```

```
hostname# inet dgd rule add all-down action-priority 20 service log command  
All-Routers-Down
```

```
hostname# inet dgd rule add all-down action-priority 10 service route command policy  
active default
```

15 Перезапустите службу DGD для применения новых правил:

```
hostname# inet dgd mode off
hostname# inet dgd mode on
```

Создание двух резервных каналов

Исходные данные

Промышленный объект подключён к интернету через шлюзы трёх интернет-провайдеров:

- основной канал — Ethernet (шлюз GW-1);
- первый резервный канал — Ethernet (шлюз GW-2);
- второй резервный канал — 3G/4G (шлюз GW-3).

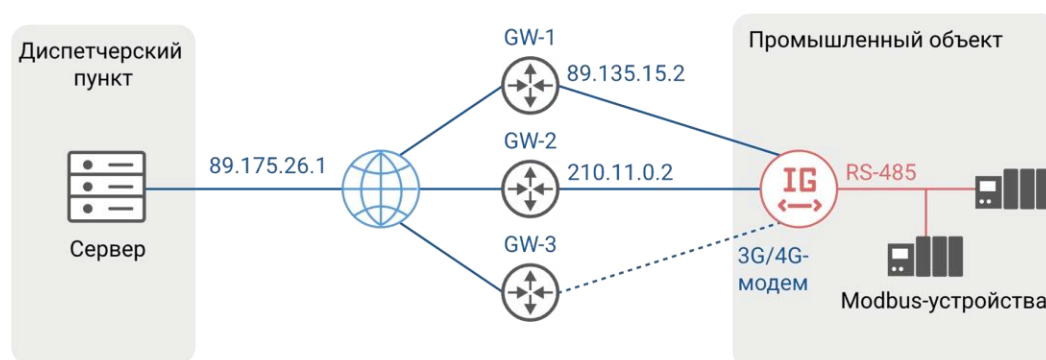


Рисунок 19. Схема резервирования канала доступа в интернет с двумя резервными каналами

Задача

Настроить автоматическое перенаправление трафика с одного канала на другой при потере соединения со шлюзом одного из интернет-провайдеров.

Настройка

- 1 Если вы используете DHCP для Ethernet-интерфейса основного или резервного канала, отключите получение маршрутов по умолчанию и перезагрузите интерфейс:

```
hostname# inet ifconfig <интерфейс> dhcp route off
hostname# inet ifconfig <интерфейс> down
hostname# inet ifconfig <интерфейс> up
```

- 2 Отключите модем:

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

- 3 Отключите получение маршрутов на модемном интерфейсе:

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

- 4 Задайте IP-адрес шлюза мобильного оператора, например, 10.64.64.64:

```
hostname# inet usb-modem ip-gw 10.64.64.64
```



Примечание. В качестве шлюза мобильного оператора, вы можете задать любой неиспользуемый IP-адрес, например, 10.255.255.254. Но в этом случае трафик до сетевых узлов с IP-адресами 10.*.* будет маршрутизироваться через модем.

5 Включите модем:

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

6 Удалите маршруты по умолчанию из таблицы по умолчанию:

```
hostname# inet route delete default
```

7 Добавьте маршруты по умолчанию для шлюзов GW-1, GW-2 и GW-3:

```
hostname# inet route add default next-hop 89.135.15.2 table 1024 name GW-1
```

```
hostname# inet route add default next-hop 210.11.0.2 table 1025 name GW-2
```

```
hostname# inet route add default next-hop 10.64.64.64 table 1026 name GW-3
```

8 Создайте политики маршрутизации:

8.1 Добавьте более приоритетные правила для таблицы по умолчанию:

```
hostname# inet policy rule add route-via-GW-1-policy 1024 table default
```

```
hostname# inet policy rule add route-via-GW-2-policy 1024 table default
```

```
hostname# inet policy rule add route-via-GW-3-policy 1024 table default
```

Это необходимо для маршрутизации ответного трафика.

8.2 Добавьте менее приоритетные правила для пользовательских таблиц:

```
hostname# inet policy rule add route-via-GW-1-policy 1025 table name GW-1
```

```
hostname# inet policy rule add route-via-GW-2-policy 1025 table name GW-2
```

```
hostname# inet policy rule add route-via-GW-3-policy 1025 table name GW-3
```



Примечание. Приоритет правил маршрутизации должен быть в диапазоне от 1024 до 2048. Чем меньше значение, тем выше приоритет. Общие правила маршрутизации, применяемые для всего трафика, должны иметь приоритет ниже, чем у правил, применяемых для конкретных диапазонов IP-адресов.

9 Запустите службу проверки состояния шлюзов (DGD):

```
hostname# inet dgd mode on
```

10 Настройте проверку доступности шлюзов GW-1 и GW-2:

```
hostname# inet dgd next-hop add GW-1 address 89.135.15.2 check icmp
```

```
hostname# inet dgd next-hop add GW-2 address 210.11.0.2 check icmp
```

Примечание. Если вы настраиваете резервирование канала доступа в интернет в кластере, то для проверки доступности шлюзов вместо команды:

```
inet dgd next-hop add <имя шлюза> address <адрес шлюза> check icmp
```

используйте команду:



```
inet dgd next-hop add <имя шлюза> address <адрес шлюза> test-address  
<тестовый адрес> check icmp
```

где в качестве <тестовый адрес> используйте IP-адрес любого стабильного узла, расположенного за шлюзом и не являющегося узлом ViPNet. А IP-адрес шлюза используйте в параметре `testip` при настройке кластера.

11 Настройте проверку доступности шлюза GW-3:

```
hostname# inet dgd next-hop add GW-3 address 10.64.64.64 test-address 77.88.8.8 check  
icmp
```

Примечание. Для проверки работоспособности шлюза мобильного оператора мы рекомендуем в качестве тестового IP-адреса выбирать стабильный узел во внешней сети. Поэтому в параметре `test-address` задан IP-адрес DNS-сервера Yandex (77.88.8.8) как одного из наиболее стабильных сетевых узлов в интернете.

12 Создайте правила службы DGD:

12.1 Правила, по которым действует политика маршрутизации `route-via-GW-1-policy`, если доступен шлюз GW-1:

```
hostname# inet dgd rule add GW-1-Up match-next-hop GW-1 up  
  
hostname# inet dgd rule add GW-1-Up action-priority 10 service route command policy  
active route-via-GW-1-policy
```

12.2 Правила, по которым действует политика маршрутизации `route-via-GW-2-policy`, если недоступен шлюз GW-1, а доступен GW-2:

```
hostname# inet dgd rule add GW-1-down match-next-hop GW-1 down  
  
hostname# inet dgd rule add GW-1-down match-next-hop GW-2 up  
  
hostname# inet dgd rule add GW-1-down action-priority 10 service route command  
policy active route-via-GW-2-policy
```

12.3 Правила, по которым действует политика маршрутизации `route-via-GW-3-policy`, если недоступны шлюзы GW-1 и GW-2, но доступен GW-3:

```
hostname# inet dgd rule add GW-1-2-down match-next-hop GW-1 down  
  
hostname# inet dgd rule add GW-1-2-down match-next-hop GW-2 down  
  
hostname# inet dgd rule add GW-1-2-down match-next-hop GW-3 up  
  
hostname# inet dgd rule add GW-1-2-down action-priority 10 service route command  
policy active route-via-GW-3-policy
```

12.4 Правила, по которым действует политика маршрутизации по умолчанию и создается запись в системном журнале, если недоступны шлюзы всех провайдеров:

```
hostname# inet dgd rule add all-down match-next-hop GW-1 down  
  
hostname# inet dgd rule add all-down match-next-hop GW-2 down  
  
hostname# inet dgd rule add all-down match-next-hop GW-3 down
```

```
hostname# inet dgd rule add all-down action-priority 10 service route command policy  
active default
```

```
hostname# inet dgd rule add all-down action-priority 20 service log command  
All-Routers-Down
```

13 Перезапустите службу DGD для применения новых правил:

```
hostname# inet dgd mode off
```

```
hostname# inet dgd mode on
```

Мониторинг состояния каналов связи

После настройки резервирования каналов доступа в интернет вы можете осуществлять удалённый мониторинг состояния этих каналов. Например, вы можете в режиме реального времени получать информацию о доступности интернет-шлюзов и смене политик маршрутизации. Для этого в сети промышленного объекта размещается сервер мониторинга, собирающий данную информацию.

Вы можете настроить мониторинг одним из способов:

- по протоколу SNMP;
- по протоколу Syslog.

Исходные данные

Промышленный объект подключён к интернету через шлюзы двух интернет-провайдеров:

- основной канал — Ethernet (шлюз GW-1);
- резервный канал — Ethernet или Wi-Fi (шлюз GW-2).

Настроено автоматическое перенаправление трафика с одного канала на другой при потере соединения со шлюзом одного из интернет-провайдеров.

В сети промышленного объекта выделен сетевой узел с Linux для использования в качестве сервера мониторинга.

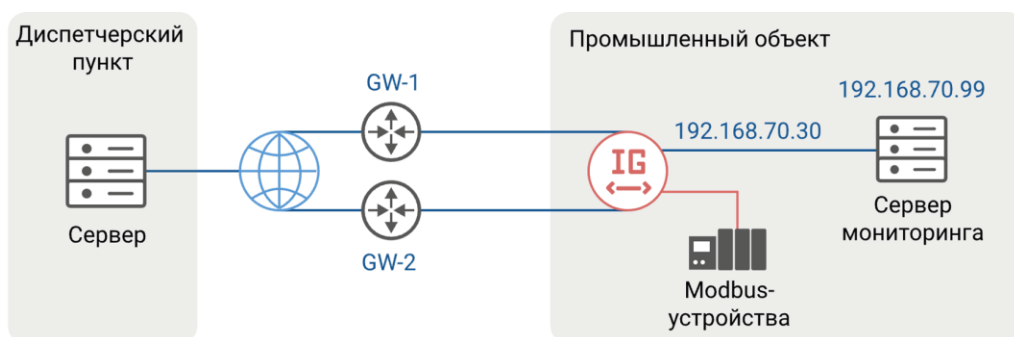


Рисунок 20. Схема резервирования канала доступа в интернет с удалённым мониторингом состояния каналов связи

Задача

Настроить передачу информации о доступности интернет-шлюзов и смене политик маршрутизации на сервер мониторинга.

Настройка мониторинга по SNMP

- 1 Установите на сервер мониторинга программу, предназначенную для сбора информации по протоколу SNMP (SNMP-менеджер), например, Zabbix.
- 2 Настройте SNMP-агент ViPNet Coordinator IG на отправку уведомлений по протоколу SNMPv3:



Примечание. ViPNet Coordinator IG поддерживает одновременную работу по протоколам SNMPv1, SNMPv2c и SNMPv3. Порядок настройки SNMPv1 и SNMPv2c см. в документе «Настройка с помощью командного интерпретатора» в разделе «Настройка SNMP-агента».

- 2.1 Добавьте нового пользователя для получения уведомлений, например User1:

```
hostname# inet snmp user add User1
```

и задайте его пароль.

- 2.2 При необходимости шифрования уведомлений задайте ключ шифрования для пользователя:

```
hostname# inet snmp user set User1 key
```

- 2.3 Разрешите отправку уведомлений для пользователя:

```
hostname# inet snmp user set User1 trapsess on
```

- 2.4 Настройте отправку уведомлений типа INFORM на сервер мониторинга:

```
hostname# inet snmp user set User1 trapsess add 192.168.70.99 port 162 inform
```

- 2.5 Включите отправку уведомлений по протоколу SNMPv3:

```
hostname# inet snmp v3 traps on
```

- 2.6 Включите автозапуск SNMP-агента:

```
hostname# inet snmp autostart on
```

- 2.7 Проверьте настройки SNMP-агента:

```
hostname# inet snmp show
```

- 2.8 Включите SNMP-агент:

```
hostname# inet snmp start
```

- 3 Настройте SNMP-менеджер на сервере мониторинга на получение уведомлений от имени пользователя User1.

SNMP-параметры, содержащие информацию о доступности интернет-шлюзов и смене политик маршрутизации, описаны в базе управляющей информации VIPNETMULTIWAN-MIB (см. документ «Настройка с помощью командного интерпретатора», приложение «Базы управляющей информации (MIB) для протокола SNMP»).

Настройка мониторинга по Syslog

Информация о доступности интернет-шлюзов и смене политик маршрутизации автоматически вносится в системный журнал ViPNet Coordinator IG.



Примечание. Примеры записей журнала, связанные с доступностью каналов связи в режиме работы MultiWAN, см. в документе «Настройка с помощью командного интерпретатора», в приложении «События системного журнала и журнала аудита».

Для мониторинга состояния каналов связи с помощью системного журнала необходимо:

- 1 На сервере мониторинга настроить ПО, которое будет принимать события от ViPNet Coordinator IG по протоколу Syslog и анализировать их. При необходимости вы можете настроить оповещение администратора о смене состояния интернет-шлюза или переключении политик маршрутизации.
- 2 На ViPNet Coordinator IG задать IP-адрес сервера мониторинга в качестве места хранения системного журнала.

Чтобы на сервере мониторинга установить и настроить программу Syslog-ng для сбора сообщений о событиях по протоколу Syslog:

- 1 В зависимости от используемого дистрибутива Linux выполните одну из команд:
 - Для Debian или Ubuntu Linux:
`apt install syslog-ng`
 - Для Fedora:
`dnf install syslog-ng`
 - Для CentOS или RedHat:
`yum install syslog-ng`
 - Для Suse Linux:
`zipper install syslog-ng`
 - Для другого дистрибутива Linux см. документацию по его настройке.
- 2 Для настройки Syslog-ng откройте файл `/etc/syslog-ng/syslog-ng.conf` и в конце файла вставьте строки:

```
# IP-адрес сервера мониторинга и порт приема сообщений от ViPNet Coordinator IG
source s_ig-1 { udp(ip(192.168.70.99) port(514)); };

# путь и имя файла системного журнала для записи сообщений от ViPNet Coordinator IG
destination d_ig-1 { file("/var/log/remote/ig-1.log" owner(root) group(root)
perm(0777) dir_perm(0777) create_dirs(yes)); };

# фильтр, разрешающий сбор сообщений только от ViPNet Coordinator IG
filter f_ig-1 { host(192.168.70.30); };

# правило, включающее сбор сообщений на основе предыдущих правил
log { source(s_ig-1); filter(f_ig-1); destination(d_ig-1); };
```

3 Включите автозапуск службы сбора сообщений:

```
systemctl enable syslog-ng
```

4 Включите службу сбора сообщений:

```
systemctl start syslog-ng
```

Чтобы на ViPNet Coordinator IG задать IP-адрес сервера мониторинга в качестве места хранения системного журнала, выполните команду:

```
hostname# machine set loghost 192.168.70.99
```

Внимание!



- Если для удалённого протоколирования будет использоваться открытый узел, на ViPNet Coordinator IG создайте фильтр открытой сети, разрешающий исходящий трафик по протоколу UDP на 514-й порт этого открытого узла: `hostname# firewall local add src @local dst 192.168.70.99 udp dport 514 pass`
- При удалённом протоколировании ViPNet Coordinator IG не хранит копий системного журнала. Поэтому в случае потери журнала на удалённом узле вы не сможете восстановить его с помощью ViPNet Coordinator IG.

В результате системный журнал ViPNet Coordinator IG будет записываться на сервер мониторинга в файл с именем `/var/log/remote/ig-1.log`.

На сервере мониторинга вы можете использовать любую программу, которая позволяет анализировать файл журнала и рассылать уведомления о событиях администратору, например Zabbix.



7

Повышение отказоустойчивости ViPNet Coordinator IG

Организация кластера	70
Организация агрегированного канала	87

Организация кластера

Чтобы ViPNet Coordinator IG продолжал выполнять свои функции даже в случае отказа аппаратного или программного обеспечения, вы можете объединить два ViPNet Coordinator IG с одинаковой аппаратной платформой в **кластер**, состоящий из активного и пассивного узлов. В случае нарушения работоспособности активного узла кластера, пассивный узел переключается в активный режим и выполняет функции сбойного узла, а сбойный узел переходит в пассивный режим.

Подробное описание работы кластера, а также порядок его развертывания см. в документе «Настройка с помощью командного интерпретатора», глава «Настройка системы защиты от сбоев». Ниже приведены примеры настроек различных схем организации кластера.

Типовая схема организации кластера

Исходные данные

Оба ViPNet Coordinator IG в кластере имеют одинаковую аппаратную платформу, версию ПО, справочники и ключи. На всех резервируемых интерфейсах установлен класс `access`. Узлы кластера подключены к одним и тем же сетям. При этом для подключения к каждой сети используются одинаковые интерфейсы на разных узлах кластера. Для синхронизации настроек узлы кластера соединены напрямую.

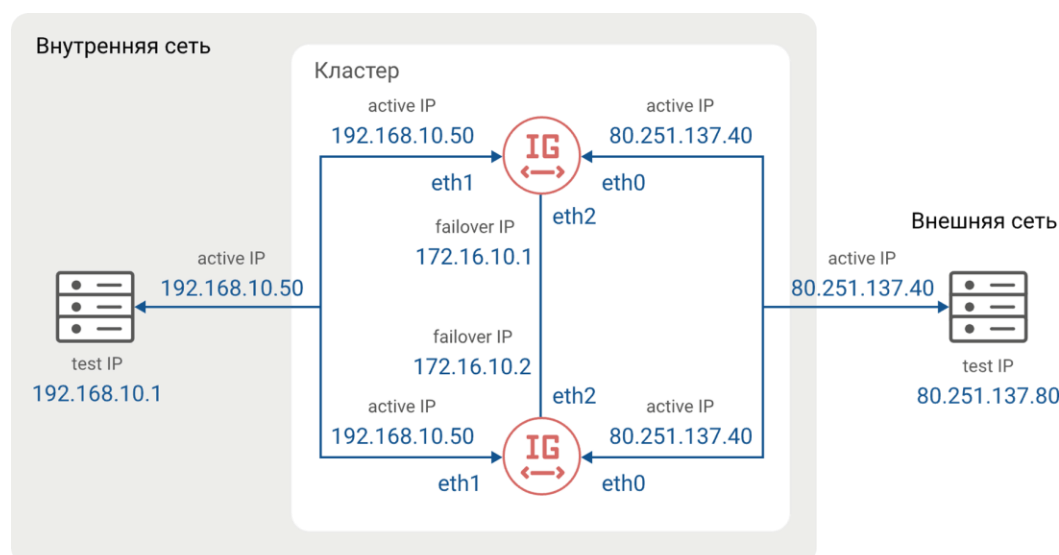


Рисунок 21: Типовая схема организации кластера

Задача

Настроить кластер.

Настройка

На каждом ViPNet Coordinator IG выполните действия (настройку можно выполнять параллельно на обоих ViPNet Coordinator IG до запуска системы защиты от сбоев):

- 1 Завершите работу системы защиты от сбоев:

```
hostname# failover stop
```

- 2 Откройте файл `failover.ini` для редактирования:

```
hostname# failover config edit
```

- 3 Задайте параметры файла `failover.ini` согласно таблице:

Таблица 2. Настройки конфигурационного файла `failover.ini` для типовой схемы организации кластера

Настройки на первом узле	Настройки на втором узле
[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
virtualmacprefix = 39	virtualmacprefix = 39
[channel]	[channel]
device = eth0	device = eth0
activeip = 80.251.137.40/24	activeip = 80.251.137.40/24
testip = 80.251.137.80	testip = 80.251.137.80
testdevice = eth0	testdevice = eth0
ident = external	ident = external
checkonlyidle = yes	checkonlyidle = yes
usevirtualmac = yes	usevirtualmac = yes
[channel]	[channel]
device = eth1	device = eth1
activeip = 192.168.10.50/24	activeip = 192.168.10.50/24
testip = 192.168.10.1	testip = 192.168.10.1
testdevice = eth1	testdevice = eth1
ident = internal	ident = internal
checkonlyidle = yes	checkonlyidle = yes
[sendconfig]	[sendconfig]

Настройки на первом узле	Настройки на втором узле
<code>activeip = 172.16.10.2</code> (соответствует failover IP второго узла)	<code>activeip = 172.16.10.1</code> (соответствует failover IP первого узла)
<code>device = eth2</code>	<code>device = eth2</code>
[misc]	[misc]
<code>synconnections = yes</code>	<code>synconnections = yes</code>

В параметре `activeip` задан IP-адрес, который будет присваиваться сетевому интерфейсу, когда узел кластера будет переходить в активный режим.



Примечание. Если сетевой интерфейс получает IP-адрес по DHCP, укажите ключевое слово `dynamic` вместо IP-адреса: `activeip = dynamic`.

IP-адрес, заданный в параметре `testip`, используется для определения работоспособности интерфейса. Мы рекомендуем задавать IP-адрес ближайшего маршрутизатора.

В параметре `testdevice` задан проводной сетевой интерфейс, физическое состояние которого будет контролироваться системой защиты от сбоев. При потере соединения узлы кластера поменяются местами. Параметр позволяет быстрее переключить узлы кластера при физической потере соединения.

Подробнее о параметрах конфигурационного файла `failover.ini` см. в документе «Справочник команд и конфигурационных файлов».

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

5 Переведите систему защиты от сбоев в режим кластера:

```
hostname# failover config mode cluster
```

6 Убедитесь, что настройки коммутационного оборудования, к которому подключён кластер, соответствуют требованиям раздела «Работа кластера с коммутационным оборудованием» в документе «Настройка с помощью командного интерпретатора».

При использовании виртуальных MAC-адресов (`usevirtualmac = yes`) убедитесь, что настройки `port-security` коммутационного оборудования не блокируют смену MAC-адресов сетевых интерфейсов кластера.

7 Проверьте связь с каждым IP-адресом, заданным в параметрах `testip`:

```
hostname# inet ping 80.251.137.80
```

```
hostname# inet ping 192.168.10.1
```

8 Запустите систему защиты от сбоев:

```
hostname# failover start
```

ViPNet Coordinator IG, на котором система защиты от сбоев будет запущена первой, перейдет в режим активного узла кластера.

Алгоритм работы кластера

Активный узел кластера контролирует наличие физического соединения на сетевых интерфейсах, указанных в параметре `testdevice`. В случае потери соединения на одном из таких интерфейсов узлы кластера переключаются в течение двух секунд.

Независимо от использования параметра `testdevice` активный узел проверяет работоспособность своих интерфейсов по следующему алгоритму:

- 1 Если параметр `checkonlyidle` установлен в значение `yes`, то анализируется сетевой трафик, проходящий через каждый интерфейс, указанный в конфигурационном файле.
- 2 Если в течение `checktime` через интерфейс не прошло ни одного IP-пакета, через этот интерфейс посылаются эхо-запросы на узел с IP-адресом `testip`.



Примечание. Чтобы эхо-запросы на узел с IP-адресом `testip` посылались каждые `checktime` секунд без проверки прохождения трафика, установите параметр `checkonlyidle` в значение `no`.

- 3 Если в течение `timeout` ни одного ответа не пришло, то счётчик сбоев сетевого интерфейса увеличивается на единицу, и посылаются новые эхо-запросы. Если на эхо-запрос пришёл хотя бы один ответ, счётчик сбоев сбрасывается. Если сброшены счётчики всех интерфейсов, происходит возврат к началу цикла проверки (ожидание в течение `checktime` и так далее). При достижении счётчиком сбоев любого интерфейса значения `channelretries` интерфейс считается неработоспособным, и происходит переключение узлов кластера.

Максимальное время, необходимое системе защиты от сбоев для определения неработоспособности интерфейса, вычисляется по формуле:

$$\text{checktime} + (\text{timeout} * \text{channelretries})$$

Одновременно пассивный узел проверяет работоспособность активного узла:

- 1 Каждые `checktime` секунд посылаются ARP-запросы в режиме DAD (Duplicate address detection) на узел с адресом `activeip`.
- 2 По истечении `timeout` проверяется наличие положительного ответа для каждого `activeip` в системной ARP-таблице.
- 3 Если ни от одного интерфейса не получен ответ, счётчик сбоев (один для всех интерфейсов) увеличивается. Если ответ получен хотя бы от одного интерфейса, счётчик сбоев сбрасывается.
- 4 При достижении счётчиком сбоев значения `activeretries` узел переходит в активный режим.

Максимальное время, проходящее с момента сбоя активного узла до обнаружения пассивным узлом этого факта, вычисляется по формуле:

$$\text{checktime} + (\text{timeout} * \text{activeretries})$$

Пассивный узел также следит за состоянием интерфейса синхронизации. Это позволяет обнаружить потерю питания на активном узле и быстрее переключить кластер:

- 1 При потере сигнала на интерфейсе, заданном в параметре `device` секции `[sendconfig]`, на все резервируемые интерфейсы активного узла посылаются ARP-запросы в режиме DAD.

- По истечении `timeout` проверяется наличие положительного ответа для каждого `activeip` в системной ARP-таблице.

Если ни от одного интерфейса не получен ответ, пассивный узел переходит в активный режим.

Кластер с беспроводными интерфейсами

Исходные данные

Эта схема отличается от [типовой схемы организации кластера](#) использованием беспроводных интерфейсов (Wi-Fi или 3G/4G). При этом хотя бы один из резервируемых интерфейсов должен быть проводным.

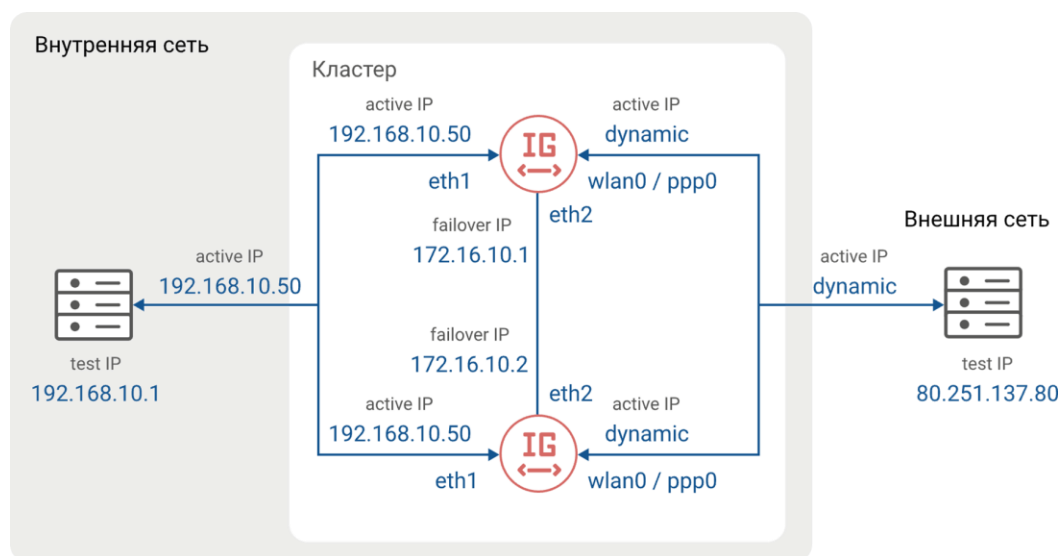


Рисунок 22. Схема организации кластера с беспроводными интерфейсами



Примечание. При использовании модема каждый узел кластера может иметь свои настройки для подключения к интернету.

Задача

Настроить кластер.

Настройка

На каждом ViPNet Coordinator IG выполните действия (настройку можно выполнять параллельно на обоих ViPNet Coordinator IG до запуска системы защиты от сбоев):

- Завершите работу системы защиты от сбоев:
`hostname# failover stop`
- Откройте файл `failover.ini` для редактирования:

```
hostname# failover config edit
```

- 3 Задайте параметры файла `failover.ini` согласно одной из следующих таблиц в зависимости от используемого беспроводного интерфейса (Wi-Fi или 3G/4G):

Таблица 3. Настройки для организации кластера с использованием Wi-Fi

Настройки на первом узле	Настройки на втором узле
[network]	[network]
checktime = 10	checktime = 10
timeout = 2	timeout = 2
activeretries = 3	activeretries = 3
channelretries = 20	channelretries = 20
synctime = 5	synctime = 5
fastdown = yes	fastdown = yes
[channel]	[channel]
device = eth1	device = eth1
activeip = 192.168.10.50/24	activeip = 192.168.10.50/24
testip = 192.168.10.1	testip = 192.168.10.1
testdevice = eth1	testdevice = eth1
ident = internal	ident = internal
checkonlyidle = yes	checkonlyidle = yes
[channel]	[channel]
device = wlan0	device = wlan0
activeip = dynamic	activeip = dynamic
testip = 80.251.137.80	testip = 80.251.137.80
ident = external	ident = external
checkonlyidle = yes	checkonlyidle = yes
[sendconfig]	[sendconfig]
activeip = 172.16.10.2 (соответствует failover IP второго узла)	activeip = 172.16.10.1 (соответствует failover IP первого узла)
device = eth2	device = eth2

Таблица 4. Настройки для организации кластера с использованием модема

Настройки на первом узле	Настройки на втором узле
[network]	[network]
checktime = 10	checktime = 10
timeout = 2	timeout = 2

Настройки на первом узле	Настройки на втором узле
<code>activeretries = 3</code>	<code>activeretries = 3</code>
<code>channelretries = 20</code>	<code>channelretries = 20</code>
<code>synctime = 5</code>	<code>synctime = 5</code>
<code>fastdown = yes</code>	<code>fastdown = yes</code>
[channel]	[channel]
<code>device = eth1</code>	<code>device = eth1</code>
<code>activeip = 192.168.10.50/24</code>	<code>activeip = 192.168.10.50/24</code>
<code>testip = 192.168.10.1</code>	<code>testip = 192.168.10.1</code>
<code>testdevice = eth1</code>	<code>testdevice = eth1</code>
<code>ident = internal</code>	<code>ident = internal</code>
<code>checkonlyidle = yes</code>	<code>checkonlyidle = yes</code>
[channel]	[channel]
<code>device = ppp0</code>	<code>device = ppp0</code>
<code>activeip = dynamic</code>	<code>activeip = dynamic</code>
<code>testip = 77.88.8.8</code>	<code>testip = 77.88.8.8</code>
<code>ident = external</code>	<code>ident = external</code>
<code>checkonlyidle = yes</code>	<code>checkonlyidle = yes</code>
[sendconfig]	[sendconfig]
<code>activeip = 172.16.10.2 (соответствует failover IP второго узла)</code>	<code>activeip = 172.16.10.1 (соответствует failover IP первого узла)</code>
<code>device = eth2</code>	<code>device = eth2</code>

Количество неудачных попыток опроса IP-адресов, указанных в параметрах `testip`, после которых интерфейс считается неработоспособным, увеличено до 20 в параметре `channelretries`. Это снижает риск переключения активного узла кластера при кратковременных разрывах связи, которые возможны в сетях 3G/4G или Wi-Fi.

Для проверки работоспособности интерфейса активного узла кластера мы рекомендуем выбирать стабильный узел в локальной сети, например, маршрутизатор. Но для модема мы рекомендуем выбирать стабильный узел во внешней сети. Поэтому в качестве значения параметра `testip` указан IP-адрес DNS-сервера Yandex (77.88.8.8) как одного из наиболее стабильных сетевых узлов в интернете.

В параметре `testdevice` задан проводной сетевой интерфейс, физическое состояние которого будет контролироваться системой защиты от сбоев. При потере соединения узлы кластера поменяются местами. Параметр позволяет быстрее переключить узлы кластера при физической потере соединения.

Подробнее о параметрах конфигурационного файла `failover.ini` см. в документе «Справочник команд и конфигурационных файлов».

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

5 Переведите систему защиты от сбоев в режим кластера:

```
hostname# failover config mode cluster
```

6 Убедитесь, что настройки коммутационного оборудования, к которому подключён кластер, соответствуют требованиям раздела «Работа кластера с коммутационным оборудованием» в документе «Настройка с помощью командного интерпретатора».

7 Проверьте связь с каждым IP-адресом, заданным в параметрах `testip`:

```
hostname# inet ping <IP-адрес>
```

8 Запустите систему защиты от сбоев:

```
hostname# failover start
```

ViPNet Coordinator IG, на котором система защиты от сбоев будет запущена первой, перейдет в режим активного узла кластера.

Алгоритм работы кластера

Активный узел кластера контролирует наличие физического соединения на интерфейсах `eth1`, указанных в параметре `testdevice`. В случае потери соединения на одном из таких интерфейсов узлы кластера переключаются в течение двух секунд.

Независимо от использования параметра `testdevice` активный узел проверяет работоспособность своих интерфейсов по следующему алгоритму:

- 1 Если параметр `checkonlyidle` установлен в значение `yes`, то анализируется сетевой трафик, проходящий через каждый интерфейс, указанный в конфигурационном файле.
- 2 Если в течение `checktime` через интерфейс не прошло ни одного IP-пакета, через этот интерфейс посылаются эхо-запросы на узел с IP-адресом `testip`.



Примечание. Чтобы эхо-запросы на узел с IP-адресом `testip` посылались каждые `checktime` секунд без проверки прохождения трафика, установите параметр `checkonlyidle` в значение `no`.

- 3 Если в течение `timeout` ни одного ответа не пришло, то счётчик сбоев сетевого интерфейса увеличивается на единицу, и посылаются новые эхо-запросы. Если на эхо-запрос пришёл хотя бы один ответ, счётчик сбоев сбрасывается. Если сброшены счётчики всех интерфейсов, происходит возврат к началу цикла проверки (ожидание в течение `checktime` и так далее). При достижении счётчиком сбоев любого интерфейса значения `channelretries` интерфейс считается неработоспособным, и происходит переключение узлов кластера.

Максимальное время, необходимое системе защиты от сбоев для определения неработоспособности интерфейса, вычисляется по формуле:

$$\text{checktime} + (\text{timeout} * \text{channelretries})$$

Одновременно пассивный узел проверяет работоспособность активного узла:

- 1 Каждые `checktime` секунд посылаются ARP-запросы в режиме DAD (Duplicate address detection) на узел с адресом `activeip`.
- 2 По истечении `timeout` проверяется наличие положительного ответа для каждого `activeip` в системной ARP-таблице.
- 3 Если ни от одного интерфейса не получен ответ, счётчик сбоев (один для всех интерфейсов) увеличивается. Если ответ получен хотя бы от одного интерфейса, счётчик сбоев сбрасывается.
- 4 При достижении счётчиком сбоев значения `activeretries` узел переходит в активный режим.

Максимальное время, проходящее с момента сбоя активного узла до обнаружения пассивным узлом этого факта, вычисляется по формуле:

$$\text{checktime} + (\text{timeout} * \text{activeretries})$$

Пассивный узел также следит за состоянием интерфейса синхронизации. Это позволяет обнаружить потерю питания на активном узле и быстрее переключить кластер:

- 1 При потере сигнала на интерфейсе, заданном в параметре `device` секции `[sendconfig]`, на все резервируемые интерфейсы активного узла посылаются ARP-запросы в режиме DAD.
- 2 По истечении `timeout` проверяется наличие положительного ответа для каждого `activeip` в системной ARP-таблице.
- 3 Если ни от одного интерфейса не получен ответ, пассивный узел переходит в активный режим.

Кластер с использованием шлюза Modbus и интерфейса GPIO

Если вы используете Modbus-устройства, оборудованные интерфейсами RS-232 и подключённые к ViPNet Coordinator IG, то со стороны Modbus-устройств может быть задействована только одна линия связи. Чтобы зарезервировать передачу данных от Modbus-устройств через ViPNet Coordinator IG, необходимо организовать кластер с использованием реле и интерфейса GPIO (при подключении к кластеру устройств с RS-485 использование реле не требуется).

Исходные данные

Из двух ViPNet Coordinator IG организован кластер (см. [Типовая схема организации кластера](#)).

Трафик от Modbus-устройств с интерфейсом RS-232 поступает на реле, которое подключено к обоим узлам кластера из ViPNet Coordinator IG. Один из узлов кластера подключён к реле по интерфейсу GPIO.

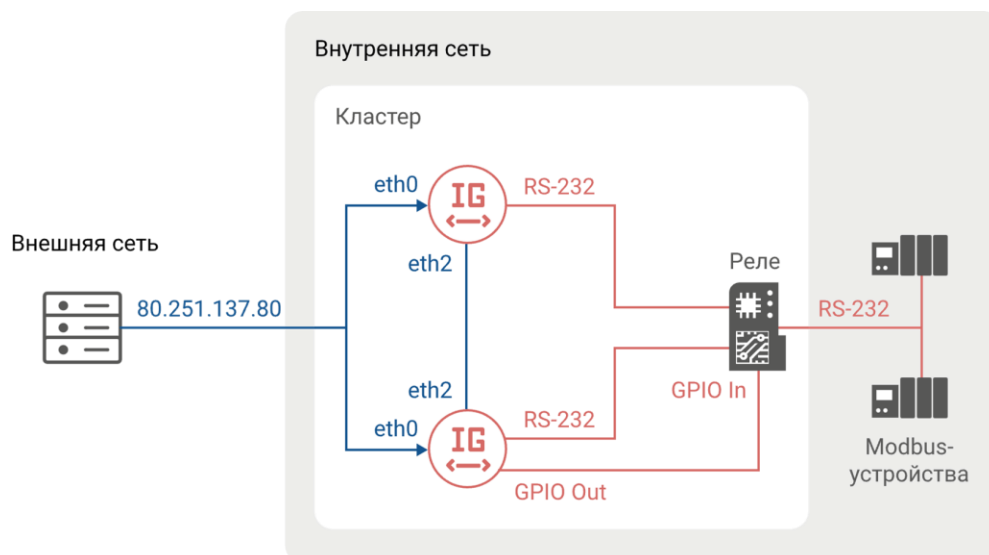


Рисунок 23. Схема подключения последовательной сети к кластеру

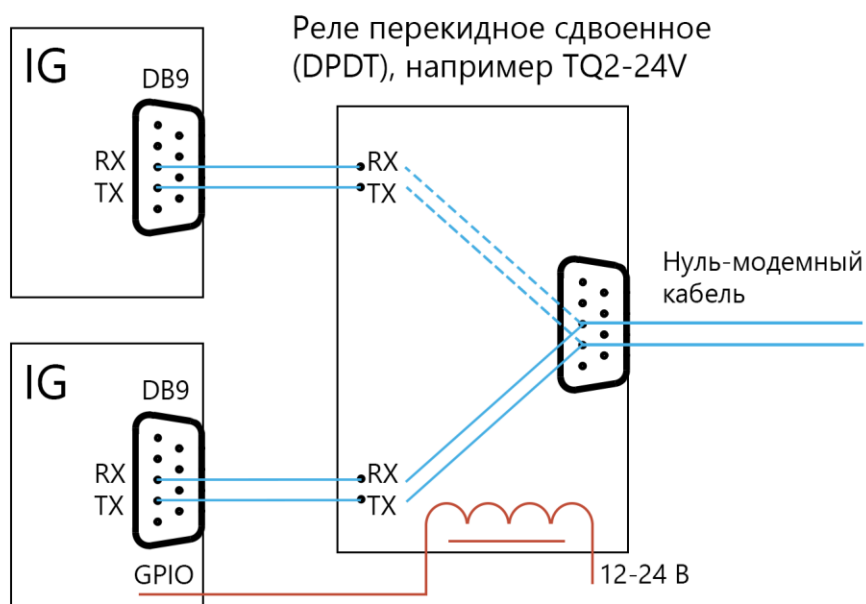


Рисунок 24. Устройство реле и схема его подключения

Задача

Настроить работу кластера таким образом, чтобы при переключении узлов кластера реле переключало движение трафика по линии RS-232 на активный в данный момент узел кластера. Для этого:

- ViPNet Coordinator IG, подключённый к реле по интерфейсу GPIO, должен передавать сигнал на реле, только когда он является активным узлом кластера;
- реле должно замыкать контакты на одном из двух разъемов в зависимости от наличия или отсутствия сигнала на интерфейсе GPIO: есть сигнал — замыкать один разъем, нет сигнала — замыкать другой разъем.

Настройка

На активном узле кластера:

- 1 Создайте шлюз Modbus на последовательном интерфейсе RS-232 с произвольным именем (например, GW-232) и TCP-портом 502:

```
hostname# service mbgw add name GW-232 serial rs232 port 502
```

По умолчанию заданы следующие настройки шлюза Modbus:

- Режим работы: TCP to RTU;
- Время ожидания запроса: 1500 мс;
- Время ожидания ответа: 1500 мс;
- Скорость TTY: 115200 бод;
- Контроль чётности: even.

- 2 Остановите работу интерфейса GPIO:

```
hostname# service piomonitor stop
```

- 3 Задайте активный уровень сигнала 1 для выходного контакта GPIO:

```
hostname# service piomonitor set Out activelevel 1
```

- 4 Задайте отправку на выходной контакт GPIO исходящего сигнала, соответствующего текущему режиму работы ViPNet Coordinator IG в кластере:

```
hostname# service piomonitor set Out cluster
```

- 5 Включите автозапуск и запустите работу шлюза Modbus:

```
hostname# service mbgw name GW-232 mode on
```

```
hostname# service mbgw name GW-232 start
```

- 6 Включите автозапуск и запустите работу интерфейса GPIO:

```
hostname# service piomonitor mode on
```

```
hostname# service piomonitor start
```

- 7 Дождитесь синхронизации узлов кластера, по умолчанию — 1 минуту (период синхронизации задается в параметре `sendtime` секции `[sendconfig]` файла `failover.ini`).

В результате вы получите отказоустойчивое решение для передачи Modbus-трафика во внешнюю сеть.



Внимание! При переключении узлов кластера трафик по линии RS-232 может прерываться до одной минуты.

Кластер с использованием DHCP-сервера и DHCP-relay

Чтобы организовать кластер с DHCP-сервером и DHCP-relay на интерфейсах кластера:

- 1 Организуйте кластер из двух ViPNet Coordinator IG (см. документ «Настройка с помощью командного интерпретатора», раздел «Развёртывание кластера»).

IP-адрес каждого резервируемого интерфейса, на котором будет настроен DHCP-сервер или DHCP-relay, должен совпадать с IP-адресом, установленным в параметре `activeip` секции `[channel]` файла `failover.ini` для соответствующего интерфейса кластера.

- 2 На активном узле кластера настройте параметры DHCP-сервера и DHCP-relay (см. документ «Настройка с помощью командного интерпретатора», разделы «Настройка DHCP-сервера» и «Настройка DHCP-relay»). Примеры настройки DHCP-сервера и DHCP-relay вы можете посмотреть в разделах [Организация работы клиентов с локальным DHCP-сервером](#) и [Организация работы клиентов с удалённым DHCP-сервером](#) (см. [Организация работы клиентов с удалённым DHCP-сервером](#)).

- 3 Убедитесь, что параметры DHCP-сервера и DHCP-relay успешно скопировались на пассивный узел кластера. Для этого на пассивном узле кластера выполните команды:

```
hostname> inet show dhcp server
hostname> inet show dhcp relay [<номер копии>]
```

- 4 Запустите службы DHCP-сервера и DHCP-relay на активном узле кластера:

```
hostname# inet dhcp server start
hostname# inet dhcp relay [<номер копии>] start
```

- 5 Включите автоматический запуск DHCP-сервера и DHCP-relay при загрузке ViPNet Coordinator IG с помощью команд:

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

- 6 Переключите режим работы пассивного узла кластера в активный. Для этого перезагрузите активный узел кластера:

```
hostname# machine reboot
```

- 7 Убедитесь, что на узле кластера, который стал активным, службы DHCP-сервера и DHCP-relay запущены и имеют верные параметры. Для этого выполните команды:

```
hostname> inet show dhcp server
hostname> inet show dhcp relay [<номер копии>]
```

Кластер с резервированием канала доступа в интернет

Чтобы обеспечить непрерывную работу ViPNet Coordinator IG в сочетании с бесперебойным доступом в интернет, вы можете организовать кластер с функцией MultiWAN (см. [Резервирование каналов доступа в интернет](#)). Один из ViPNet Coordinator IG в кластере будет продолжать выполнять свои функции в случае отказа другого, а функция MultiWAN обеспечит бесперебойный доступ в интернет при использовании шлюзов нескольких интернет-провайдеров.

Алгоритм работы кластера с функцией MultiWAN:

- Активный узел кластера направляет трафик через интернет-шлюз, который задан в качестве основного (основной шлюз).
- Если соединение через основной шлюз недоступно, трафик перенаправляется на резервный шлюз.
- При восстановлении доступа через основной шлюз, трафик перенаправляется на него.
- Если недоступны оба шлюза или нарушена работоспособность активного узла кластера, выполняется переключение узлов кластера.
- Когда пассивный узел кластера становится активным, выполняется попытка подключения к интернету через основной шлюз. Если соединение недоступно, выполняется попытка подключения через резервный шлюз.
- При использовании модема каждый узел кластера может подключаться к шлюзу своего мобильного оператора.

Исходные данные

Оба ViPNet Coordinator IG в кластере имеют одинаковую аппаратную платформу, версию ПО, справочники и ключи. Узлы кластера подключены к одним и тем же сетям. При этом для подключения к каждой сети используются одинаковые интерфейсы на разных узлах кластера. Для синхронизации настроек узлы кластера соединены напрямую.

Каждый ViPNet Coordinator IG подключён к интернету через шлюзы двух интернет-провайдеров:

- основной канал — Ethernet (через шлюз GW-1);
- резервный канал — 3G/4G (через шлюз GW-2).

Таблица 5. Настройки для кластера с резервным каналом доступа в интернет через мобильную сеть

Настройки на первом узле	Настройки на втором узле
[network]	[network]
checktime = 10	checktime = 10
timeout = 2	timeout = 2
activeretries = 3	activeretries = 3
channelretries = 20	channelretries = 20
synctime = 5	synctime = 5
fastdown = yes	fastdown = yes
[channel]	[channel]
device = eth0	device = eth0
activeip = 89.135.15.28/24	activeip = 89.135.15.28/24
testip = 89.135.15.2	testip = 89.135.15.2
testdevice = eth0	testdevice = eth0
ident = external	ident = external
checkonlyidle = yes	checkonlyidle = yes
channelgroup = 1	channelgroup = 1
[channel]	[channel]
device = eth1	device = eth1
activeip = 192.168.10.50/24	activeip = 192.168.10.50/24
testip = 192.168.10.1	testip = 192.168.10.1
testdevice = eth1	testdevice = eth1
ident = internal	ident = internal
checkonlyidle = yes	checkonlyidle = yes
[channel]	[channel]
device = ppp0	device = ppp0
activeip = dynamic	activeip = dynamic
testip = 77.88.8.8	testip = 77.88.8.8
ident = mobile	ident = mobile
checkonlyidle = yes	checkonlyidle = yes
channelgroup = 1	channelgroup = 1
[sendconfig]	[sendconfig]
activeip = 172.16.10.2 (IP-адрес интерфейса синхронизации второго узла)	activeip = 172.16.10.1 (IP-адрес интерфейса синхронизации первого узла)

Настройки на первом узле

```
device = eth2
```

Настройки на втором узле

```
device = eth2
```

В параметре `activeip` задан IP-адрес, который будет присваиваться сетевому интерфейсу, когда узел кластера будет переходить в активный режим.

IP-адрес, заданный в параметре `testip`, используется для определения работоспособности интерфейса. Мы рекомендуем задавать IP-адрес ближайшего маршрутизатора. При этом тестовый IP-адрес, заданный при настройке резервирования канала доступа в интернет, должен принадлежать узлу, который расположен дальше этого маршрутизатора (на пути IP-пакета от ViPNet Coordinator IG к этому узлу должен встретиться маршрутизатор, IP-адрес которого задан в `testip`).



Примечание. Для модема в качестве параметра `testip` и в качестве тестового адреса при настройке резервирования канала доступа в интернет мы рекомендуем выбирать стабильный узел во внешней сети. Поэтому в данном примере указан IP-адрес DNS-сервера Yandex (77.88.8.8) как одного из наиболее стабильных сетевых узлов в интернете.

Не используйте в качестве `testip` адрес шлюза мобильного оператора, заданный командой `inet usb-modem ip-gw`.

В параметре `testdevice` задан проводной сетевой интерфейс, физическое состояние которого будет контролироваться системой защиты от сбоев. При потере соединения узлы кластера поменяются местами.



Внимание! Одинаковые значения параметра `channelgroup` для интерфейсов `eth0` и `ppp0` указывают на то, что эти интерфейсы используются для резервирования канала доступа в интернет с помощью функции MultiWAN. Система защиты от сбоев считает любой из интерфейсов, входящих в группу `channelgroup`, работоспособным, пока работает другой интерфейс из этой группы. Это необходимо для того, чтобы узлы кластера не переключались при недоступности одного канала доступа в интернет, если доступен другой (основной или резервный).

Количество неудачных попыток опроса IP-адресов, указанных в параметрах `testip`, после которых интерфейс считается неработоспособным, увеличено до 20 в параметре `channelretries`. Это снижает риск переключения активного узла кластера при кратковременных разрывах связи, которые возможны в сетях 3G/4G или Wi-Fi.

Описание всех параметров конфигурационного файла `failover.ini` см. в документе «Справочник команд и конфигурационных файлов» в разделе «Файл `failover.ini`».

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

3.5 Переведите систему защиты от сбоев в режим кластера:

```
hostname# failover config mode cluster
```

4 Запустите систему защиты от сбоев:

4.1 Перед запуском системы защиты от сбоев убедитесь, что настройки коммутационного оборудования, к которому подключён кластер, соответствуют требованиям раздела «Работа кластера с коммутационным оборудованием» в документе «Настройка с помощью командного интерпретатора».

4.2 Проверьте связь с каждым IP-адресом, заданным в параметрах `testip` секции `[channel]` файла `failover.ini`:

```
hostname# inet ping <IP-адрес>
```

4.3 Запустите систему защиты от сбоев на ViPNet Coordinator IG, на котором настроено резервирование канала доступа в интернет:

```
hostname# failover start active
```

ViPNet Coordinator IG будет запущен как активный узел кластера.

4.4 Запустите систему защиты от сбоев на другом ViPNet Coordinator IG:

```
hostname# failover start passive
```

Этот ViPNet Coordinator IG будет запущен как пассивный узел кластера.

Настройки резервирования канала доступа в интернет будут переданы с активного узла кластера на пассивный во время следующей синхронизации (период синхронизации задаётся в параметре `sendtime` секции `[sendconfig]` файла `failover.ini`).

Организация агрегированного канала

Если требуется повысить надёжность ваших каналов передачи данных, вы можете объединить несколько физических сетевых интерфейсов ViPNet Coordinator IG в один логический — агрегированный интерфейс. При этом соответствующие каналы связи объединяются на канальном уровне сетевой модели OSI. Надёжность обеспечивается за счет перенаправления трафика с одного сетевого интерфейса на другой в случае отказа первого. Пропускная способность агрегированного канала не возрастает при объединении сетевых интерфейсов.

Агрегированные каналы целесообразно использовать в сетях, где особенно важно реализовать повышенную отказоустойчивость, например между удалёнными сегментами промышленной сети.

Агрегированный канал между ViPNet Coordinator IG и коммутатором

Исходные данные

ViPNet-клиенты взаимодействуют с сервером обработки данных через ViPNet Coordinator IG. Подключение клиентов к ViPNet Coordinator IG выполняется через коммутатор.

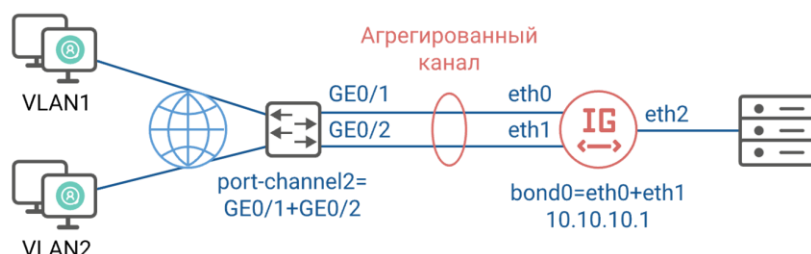


Рисунок 26. Агрегированный канал между ViPNet Coordinator IG и коммутатором

Задача

Обеспечить отказоустойчивость канала связи между коммутатором и ViPNet Coordinator IG с помощью агрегированного канала.

Настройка

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 «Настройка с помощью командного интерпретатора», раздел «Режимы работы агрегированного интерфейса».

Используем режим 802.3ad. В этом режиме для определения подчинённого 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 подчинённый физический интерфейс. Хэш-функция вычисляется по алгоритму layer3+4, поэтому учитываются IP-адреса и порты TCP или UDP отправителей и получателей.

Чтобы настроить описанный агрегированный канал связи:

- 1 Установите для физических интерфейсов eth0 и eth1 класс slave:

```
hostname# inet ifconfig eth0 class slave
hostname# inet ifconfig eth1 class slave
```

- 2 Задайте режим работы создаваемого агрегированного интерфейса 802.3ad:

```
hostname# inet bonding add 0 mode 802.3ad slaves eth0 eth1
```

- 3 Задайте алгоритм вычисления хэш-функции:

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

- 4 Добавьте агрегированный интерфейс в конфигурацию ViPNet Coordinator IG:

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

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

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

- 5 Задайте для созданного агрегированного интерфейса IP-адрес:

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

- 6 Соедините интерфейсы eth0 и eth1 с интерфейсами коммутатора.

- 7 Последовательно включите все подчинённые физические интерфейсы:

```
hostname# inet ifconfig eth0 up
hostname# inet ifconfig eth1 up
```

- 8 Включите агрегированный сетевой интерфейс:

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

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

- 1 Предварительно объедините сетевые интерфейсы коммутатора, к которым подключены ViPNet-клиенты, а также интерфейсы, которые будут подключены к агрегированному каналу, в виртуальную сеть VLAN с определенным номером (в примере VLAN имеет номер 116).
- 2 Создайте логический интерфейс `port-channel` (в примере он имеет номер 2):



Примечание. Для настройки коммутатора:

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

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

- 3** Укажите два физических интерфейса (в примере имена этих интерфейсов — `GigabitEthernet0/1` и `GigabitEthernet0/2`), которые будут входить в состав логического интерфейса `port-channel2`. Эти физические интерфейсы будут использоваться для подключения к агрегированному каналу.

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

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

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

В результате координатор будет доступен по заданному IP-адресу, а трафик между коммутатором и координатором будет идти по агрегированному каналу, состоящему из двух подчинённых физических каналов. В случае отказа одного из подчинённых каналов оставшийся продолжает работу, и сессии с отказавшего канала перенаправляются на оставшийся.

Агрегированный канал между двумя ViPNet Coordinator IG

Исходные данные

Сетевые узлы в одном сегменте сети взаимодействуют с сервером в другом сегменте сети. На границах обоих сегментов установлены координаторы ViPNet Coordinator IG.

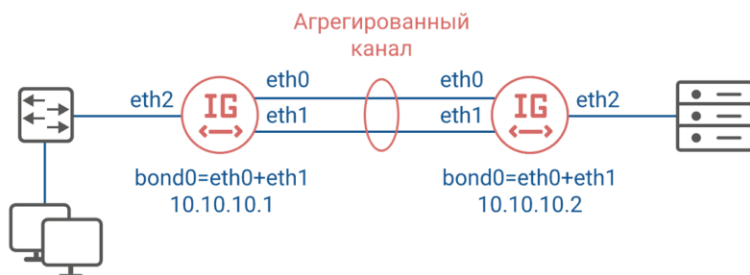


Рисунок 27. Агрегированный канал между двумя ViPNet Coordinator IG

Задача

Обеспечить отказоустойчивость канала связи между двумя ViPNet Coordinator IG для бесперебойного взаимодействия сетевых узлов с сервером.

Настройка

Так как при передаче пакетов через ViPNet Coordinator IG MAC-адрес, порт и IP-адрес сетевого узла отправителя подменяется, режимы работы агрегированных каналов, связанные с распределением нагрузки по MAC-адресам, IP-адресам и портам, использовать нельзя. Таким образом, можно использовать только следующие режимы работы агрегированных интерфейсов:

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

Подробнее о режимах работы агрегированных интерфейсов см. в документе «Настройка с помощью командного интерпретатора», раздел «Режимы работы агрегированного интерфейса».

Будем использовать режим `balance-rr`. В этом режиме исходящие пакеты, попадающие на агрегированный интерфейс, отправляются через подчинённые физические интерфейсы поочерёдно: первый пакет отправляется через один подчинённый интерфейс, второй пакет — через следующий подчинённый интерфейс и так далее. По агрегированному каналу связи будет отправляться как защищённый, так и открытый трафик.



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

Для настройки агрегированного канала:

1 На обоих ViPNet Coordinator IG:

1.1 Установите для физических интерфейсов eth0 и eth1 класс slave:

```
hostname# inet ifconfig eth0 class slave
hostname# inet ifconfig eth1 class slave
```

1.2 Задайте режим работы создаваемого агрегированного интерфейса balance-rr:

```
hostname# inet bonding add 0 mode balance-rr slaves eth0 eth1
```

1.3 Добавьте новый агрегированный интерфейс в конфигурацию ViPNet Coordinator IG:

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

2 Задайте для созданных агрегированных интерфейсов IP-адреса:

○ На первом ViPNet Coordinator IG:

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

○ На втором ViPNet Coordinator IG:

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

3 Соедините интерфейсы eth0 обоих ViPNet Coordinator IG.

4 Соедините интерфейсы eth1 обоих ViPNet Coordinator IG.

5 На обоих ViPNet Coordinator IG:

5.1 Последовательно включите все подчинённые физические интерфейсы:

```
hostname# inet ifconfig eth0 up
hostname# inet ifconfig eth1 up
```

5.2 Включите агрегированный сетевой интерфейс:

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

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

8

Использование сервисных функций ViPNet Coordinator IG

Организация обработки трафика из нескольких VLAN	93
Организация работы клиентов с локальным или удалённым DHCP-сервером	95
Организация работы клиентов удалённых офисов с DNS- и NTP-серверами, расположенными в центральном офисе	106
Защита соединения между удалёнными сегментами сети на канальном уровне модели OSI	108
Настройка резервирования каналов с помощью BGP	118
Использование ViPNet Coordinator IG в качестве защищённого интернет-шлюза	120

Организация обработки трафика из нескольких VLAN

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

Исходные данные

ViPNet Coordinator IG подключён к коммутатору, объединяющему виртуальные сети. Для порта коммутатора, к которому подключён ViPNet Coordinator IG, установлен режим `trunk`.

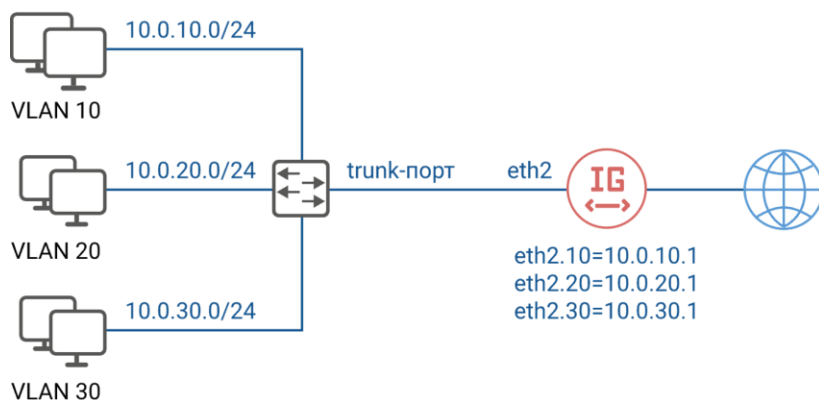


Рисунок 28. Схема использования ViPNet Coordinator IG в сети с VLAN

Задача

Настроить обработку трафика из VLAN 10, VLAN 20 и VLAN 30.

Настройка

- 1 Для сетевого интерфейса `eth2` укажите класс `trunk`:

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

- 2 Создайте виртуальные интерфейсы `eth2.10`, `eth2.20`, `eth2.30`, которые будут соответствовать виртуальным сетям с номерами 10, 20 и 30:

```
hostname# inet ifconfig eth2 vlan add 10
```

```
hostname# inet ifconfig eth2 vlan add 20
```

```
hostname# inet ifconfig eth2 vlan add 30
```

- 3 Задайте параметры созданных виртуальных интерфейсов:

```
hostname# inet ifconfig eth2.10 address 10.0.10.1 netmask 255.255.255.0
```

```
hostname# inet ifconfig eth2.20 address 10.0.20.1 netmask 255.255.255.0
hostname# inet ifconfig eth2.30 address 10.0.30.1 netmask 255.255.255.0
```

4 Добавьте новые виртуальные интерфейсы в файл конфигурации `iplir.conf`:

```
hostname# iplir adapter add eth2.10 traffic on
hostname# iplir adapter add eth2.20 traffic on
hostname# iplir adapter add eth2.30 traffic on
```

Вы можете не использовать параметр `traffic on`, тогда трафик не будет проходить через виртуальный интерфейс. Позже вы можете разрешить прохождение трафика через виртуальный интерфейс:

```
hostname# iplir adapter traffic <виртуальный интерфейс> on
```

5 Чтобы трафик проходил только через виртуальные интерфейсы, запретите прохождение трафика через физический интерфейс `eth2`:

```
hostname# iplir adapter traffic eth2 off
```

6 Включите интерфейс `eth2`:

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

Виртуальные интерфейсы `eth2.10`, `eth2.20`, `eth2.30` будут включены автоматически.

После произведенных настроек ViPNet Coordinator IG сможет обрабатывать трафик из трёх виртуальных сетей на соответствующих виртуальных интерфейсах.

Организация работы клиентов с локальным или удалённым DHCP-сервером

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

- Использовать ViPNet Coordinator IG в качестве DHCP-сервера для узлов своей сети (см. [Организация работы клиентов с локальным DHCP-сервером](#)).
- Использовать ViPNet Coordinator IG в качестве посредника между узлами своей сети и удалённым DHCP-сервером (см. [Организация работы клиентов с удалённым DHCP-сервером](#)).

Организация работы клиентов с локальным DHCP-сервером

В локальной сети небольшой организации, имеющей один офис, вы можете развернуть собственный DHCP-сервер с помощью ViPNet Coordinator IG:

- 1 Остановите DHCP-сервер, если он был запущен:

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

- 2 Подключите один из интерфейсов ViPNet Coordinator IG к локальной сети.
- 3 Установите на нем IP-адрес, принадлежащий адресному пространству локальной сети.
- 4 Включите интерфейс:

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

- 5 Задайте этот интерфейс для работы DHCP-сервера:

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

- 6 Задайте диапазон IP-адресов, выделяемых клиентам:

```
hostname# inet dhcp server add range <начальный IP-адрес> <конечный IP-адрес> interface <имя интерфейса>
```

Диапазон IP-адресов должен принадлежать сети интерфейса. При этом IP-адрес интерфейса не должен входить в этот диапазон.

- 7 Задайте срок (время аренды в секундах), на который клиентам предоставляются IP-адреса:

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

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

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

9 Задайте IP-адрес DNS-и NTP-сервера, если их необходимо передавать клиентам:

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

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



Примечание. Вы можете задать другие параметры, передаваемые DHCP-сервером своим клиентам. Подробнее см. в документе «Справочник команд и конфигурационных файлов».

10 Проверьте текущие параметры DHCP-сервера:

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

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

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

12 Запустите DHCP-сервер:

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



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

Организация работы клиентов с удалённым DHCP-сервером

Исходные данные

У организации есть несколько офисов. При этом только в центральном офисе установлен DHCP-сервер. В филиалах устанавливать свой DHCP-сервер нецелесообразно.

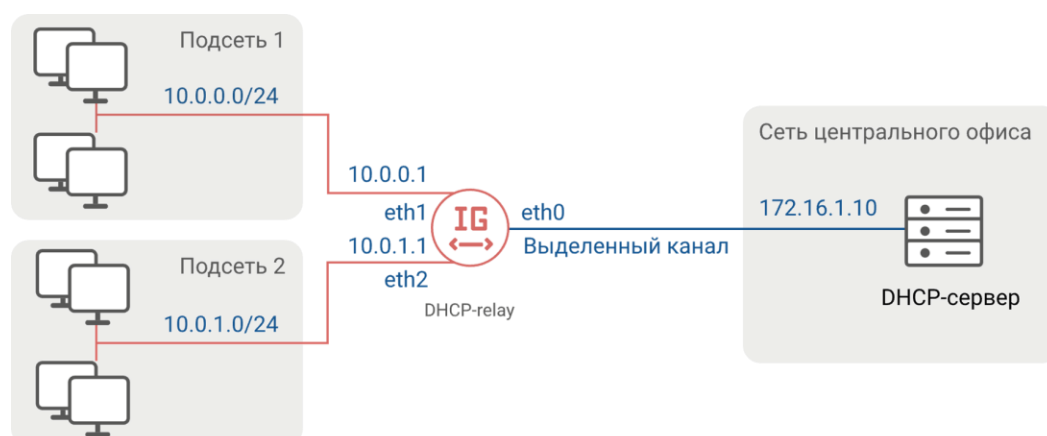


Рисунок 29: Схема использования ViPNet Coordinator IG в качестве агента DHCP-relay

Задача

Настроить перенаправление DHCP-запросов от клиентов филиала на DHCP-сервер центрального офиса.

Настройка

Взаимодействие клиентов филиала с DHCP-сервером в центральном офисе можно организовать, используя ViPNet Coordinator IG в качестве агента DHCP-relay.



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

- 1 Задайте на DHCP-сервере для каждой подсети диапазон выделяемых адресов и маршрут в сторону ViPNet Coordinator IG.
- 2 На ViPNet Coordinator IG:
 - 2.1 На каждом из интерфейсов задайте статический IP-адрес, принадлежащий адресному пространству соответствующей подсети:
 - eth1 — 10.0.0.1,
 - eth2 — 10.0.1.1.
 - 2.2 Задайте eth1 и eth2 в качестве интерфейсов, принимающих от клиентов DHCP-запросы:

```
hostname# inet dhcp relay add listen-interface eth1
hostname# inet dhcp relay add listen-interface eth2
```
 - 2.3 Задайте параметры для связи ViPNet Coordinator IG с внешним DHCP-сервером:

```
hostname# inet dhcp relay add external-interface eth0 server 172.16.1.10
```
 - 2.4 Проверьте текущие параметры DHCP-relay:

```
hostname# inet show dhcp relay
```
 - 2.5 Включите автоматический запуск DHCP-relay при загрузке ViPNet Coordinator IG:

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

По умолчанию автоматический запуск DHCP-relay выключен.
 - 2.6 Запустите DHCP-relay:

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

Использование DHCP-сервера и DHCP-relay в разных сетях

Исходные данные

Вычислительная сеть организации состоит из трёх удалённых сетей. Координатор одной из них (IG-1) выступает в роли DHCP-сервера, а координаторы других сетей (IG-2, IG-3) работают как агенты DHCP-relay.

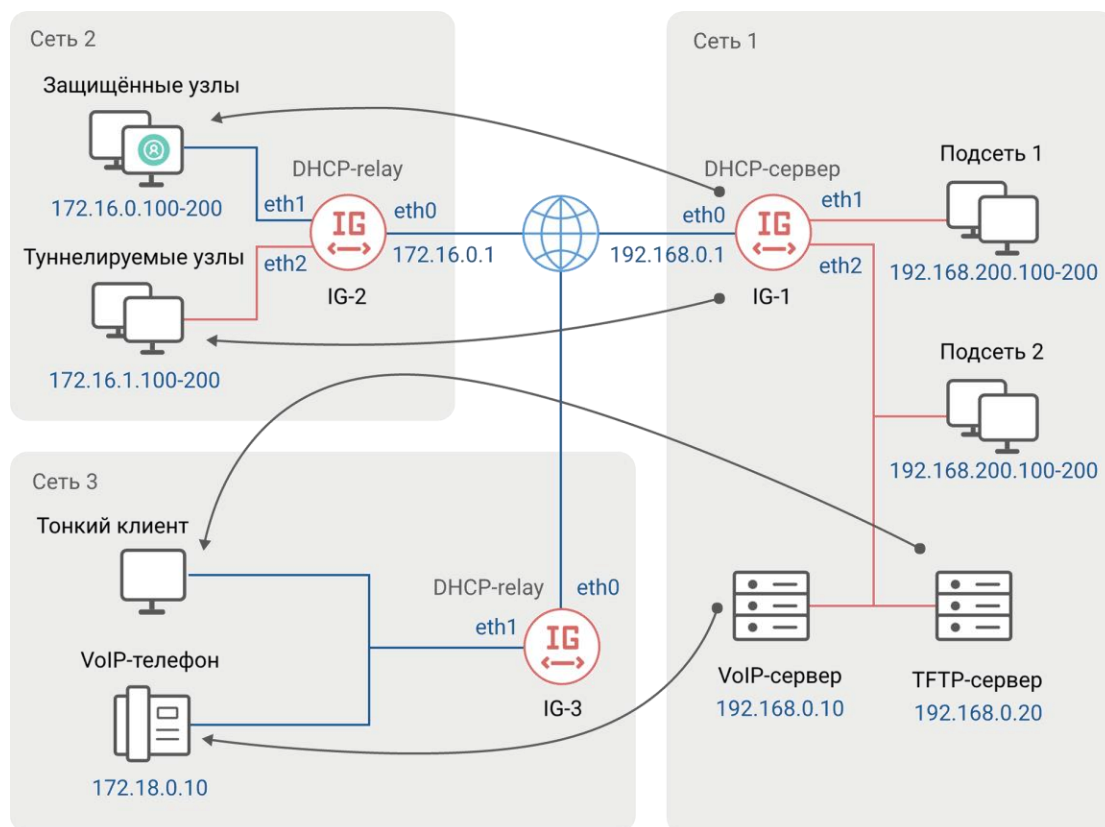


Рисунок 30. Схема DHCP-сервера и DHCP-relay в разных сетях

Связь между сетями осуществляется через интернет, поэтому для координатора, работающего в качестве DHCP-сервера (IG-1) настроена видимость по реальным IP-адресам координаторов, работающих в качестве агентов DHCP-relay (IG-2, IG-3).

В сети 1 DHCP-сервер, настроенный на координаторе IG-1, работает для следующих DHCP-клиентов:

- клиенты подсетей 1 и 2, находящиеся в его же сети (диапазоны 192.168.200.100-200, 192.168.201.100-200);
- VoIP-сервер, передающий данные для VoIP-телефона в сети 3;
- TFTP-сервер, передающий данные для тонкого клиента в сети 3;
- клиенты сети 2:

- защищённые узлы (диапазон 172.16.0.100-200);
- туннелируемые узлы (диапазон 172.16.1.100-200);
- клиенты сети 3:
 - VoIP-телефон Phone-1, получающий данные от VoIP-сервера в сети 1;
 - Тонкий клиент Thin-1, получающий данные от TFTP-сервера в сети 1.

Задача

Настроить DHCP-сервер и DHCP-relay на координаторах.

Настройка

1 На IG-1 настройте DHCP-сервер:

1.1 Задайте рабочий сетевой интерфейс DHCP-сервера для сетей 2 и 3:

```
hostname# inet dhcp server add interface eth0
```

1.2 Задайте рабочий сетевой интерфейс DHCP-сервера для подсети 1:

```
hostname# inet dhcp server add interface eth1
```

1.3 Задайте рабочий сетевой интерфейс DHCP-сервера для подсети 2:

```
hostname# inet dhcp server add interface eth2
```

1.4 Задайте диапазон IP-адресов, выделяемых DHCP-сервером для подсети 1:

```
hostname# inet dhcp server add range 192.168.200.100 192.168.200.200 interface eth1
```

1.5 Задайте диапазон IP-адресов, выделяемых DHCP-сервером для подсети 2:

```
hostname# inet dhcp server add range 192.168.201.100 192.168.201.200 interface eth2
```

1.6 Добавьте подсеть защищённых узлов сети 2 в качестве удалённой подсети для DHCP-сервера:

```
hostname# inet dhcp server add relay-interface eth0 remote subnet 172.16.0.0 mask 255.255.255.0
```

1.7 Задайте диапазон IP-адресов, выделяемых DHCP-сервером для защищённых узлов сети 2:

```
hostname# inet dhcp server add range 172.16.0.100 172.16.0.200 remote subnet 172.16.0.0 mask 255.255.255.0
```

1.8 Задайте IP-адрес шлюза по умолчанию для сети 2:

```
hostname# inet dhcp server add router 172.16.0.1 remote subnet 172.16.0.0 mask 255.255.255.0
```

1.9 Добавьте подсеть туннелируемых узлов сети 2 в качестве удалённой подсети для DHCP-сервера:

```
hostname# inet dhcp server add relay-interface eth0 remote subnet 172.16.1.0 mask 255.255.255.0
```

1.10 Задайте диапазон IP-адресов, выделяемых DHCP-сервером для туннелируемых узлов сети 2:

```
hostname# inet dhcp server add range 172.16.1.100 172.16.1.200 remote subnet  
172.16.1.0 mask 255.255.255.0
```

1.11 Задайте IP-адрес шлюза по умолчанию для туннелируемых узлов сети 2:

```
hostname# inet dhcp server add router 172.16.1.1 remote subnet 172.16.1.0 mask  
255.255.255.0
```

1.12 Добавьте узлы сети 3 в качестве удалённой подсети для DHCP-сервера:

```
hostname# inet dhcp server add relay-interface eth0 remote subnet 172.18.0.0 mask  
255.255.255.0
```

1.13 Задайте IP-адрес шлюза по умолчанию для сети 3:

```
hostname# inet dhcp server add router 172.18.0.1 remote subnet 172.18.0.0 mask  
255.255.255.0
```

1.14 Резервируйте IP-адрес 172.18.0.10 для туннелируемого узла Phone-1:

```
hostname# inet dhcp server add host Phone-1 hardware <MAC-адрес узла Phone-1>  
address 172.18.0.10 remote subnet 172.18.0.0 mask 255.255.255.0
```

1.15 Резервируйте IP-адрес 192.168.0.20 для TFTP-сервера:

```
hostname# inet dhcp server add host TFTP hardware <MAC-адрес TFTP-сервера> address  
192.168.0.20 interface eth2
```

1.16 Резервируйте IP-адрес 192.168.0.10 для VoIP-сервера:

```
hostname# inet dhcp server add host VOIP hardware <MAC-адрес VoIP-сервера> address  
192.168.0.10 interface eth2
```

1.17 Задайте для туннелируемого узла Phone-1 IP-адрес VoIP-сервера:

```
hostname# inet dhcp server add voip 192.168.0.10 host Phone-1
```

1.18 Задайте для тонкого клиента Thin-1 IP-адрес TFTP-сервера и путь к файлу с параметрами загрузки:

```
hostname# inet dhcp server add tftp 192.168.0.20 file <путь к файлу на TFTP-сервере>  
remote subnet 172.18.0.0 mask 255.255.255.0
```

1.19 Запустите DHCP-сервер:

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

1.20 Включите автоматический запуск DHCP-сервера при загрузке ViPNet Coordinator IG:

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

2 На IG-2 настройте DHCP-relay, принимающий данные от DHCP-сервера координатора IG-1:

2.1 Задайте сетевой интерфейс eth0 как внешний интерфейс службы DHCP-relay:

```
hostname# inet dhcp relay add external-interface eth0 server 192.168.0.1
```

2.2 Задайте сетевой интерфейс eth1 как слушающий интерфейс для защищённых узлов:

```
hostname# inet dhcp relay add listen-interface eth1
```

2.3 Задайте сетевой интерфейс eth2 как слушающий интерфейс для туннелируемых узлов:

```
hostname# inet dhcp relay add listen-interface eth2
```

2.4 Запустите службу DHCP-relay:

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

2.5 Включите автоматический запуск службы DHCP-relay при загрузке ViPNet Coordinator IG:

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

3 На IG-3 настройте DHCP-relay:

3.1 Задайте сетевой интерфейс для связи с внешним DHCP-сервером:

```
hostname# inet dhcp relay add external-interface eth0 server 192.168.0.1
```

3.2 Задайте сетевой интерфейс, принимающий запросы от клиентов к DHCP-серверу:

```
hostname# inet dhcp relay add listen-interface eth1
```

3.3 Запустите службу DHCP-relay:

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

3.4 Включите автоматический запуск службы DHCP-relay при загрузке ViPNet Coordinator IG:

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

Использование запасного DHCP-сервера

Исходные данные

Имеется две удалённых друг от друга сети (сеть 1 и сеть 2). ViPNet Coordinator IG в сети 2 (IG-2) туннелирует два сторонних DHCP-сервера, один из которых является запасным, а ViPNet Coordinator IG в сети 1 (IG-1) туннелирует открытые узлы, которым требуется динамически выделять IP-адреса.

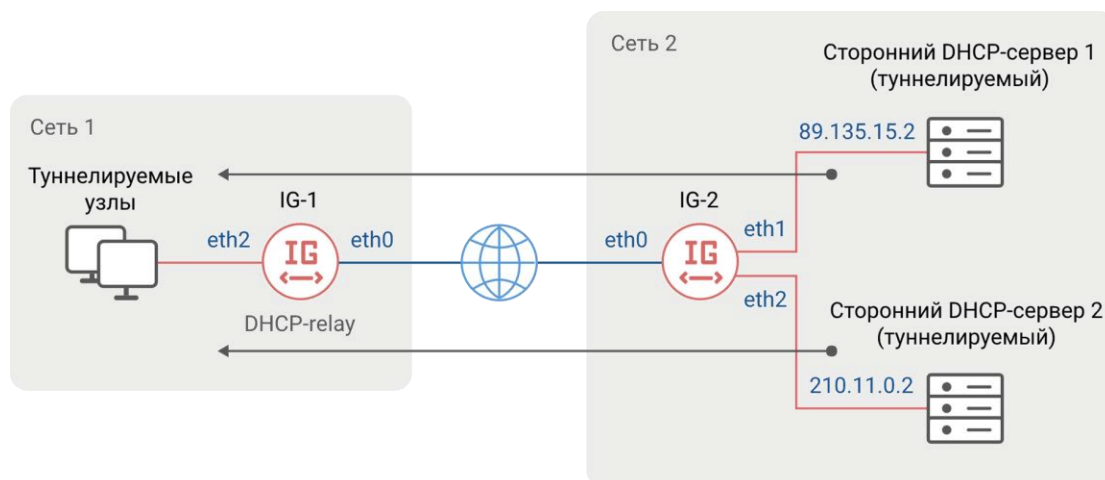


Рисунок 31. Использование DHCP-relay для запасного DHCP-сервера

Связь между сетями осуществляется через интернет, поэтому между их координаторами настроена видимость по реальным IP-адресам. Сторонние DHCP-серверы, подключённые к координатору IG-2, работают для туннелируемых узлов сети 1.

Задача

Настроить DHCP-relay на координаторе IG-1 для узлов сети 1.

Настройка

На координаторе IG-1:

- 1 Задайте сетевой интерфейс eth0 как внешний интерфейс службы DHCP-relay:

```
hostname# inet dhcp relay add external-interface eth0 server 89.135.15.2
```
- 2 Задайте сетевой интерфейс eth0 как интерфейс службы DHCP-relay, подключённый к запасному DHCP-серверу:

```
hostname# inet dhcp relay add backup-interface eth0 server 210.11.0.2
```
- 3 Задайте сетевой интерфейс eth2 как слушающий интерфейс для туннелируемых узлов:

```
hostname# inet dhcp relay add listen-interface eth2
```
- 4 Запустите службу DHCP-relay:

```
hostname# inet dhcp relay start
```
- 5 Включите автоматический запуск службы DHCP-relay при загрузке ViPNet Coordinator IG:

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

Использование сторонних DHCP-серверов

Исходные данные

Имеется две удалённых друг от друга сети (сеть 1 и сеть 2). ViPNet Coordinator IG в сети 2 (IG-2) туннелирует два сторонних DHCP-сервера, а ViPNet Coordinator IG в сети 1 (IG-1) туннелирует VoIP-телефоны в своей сети.

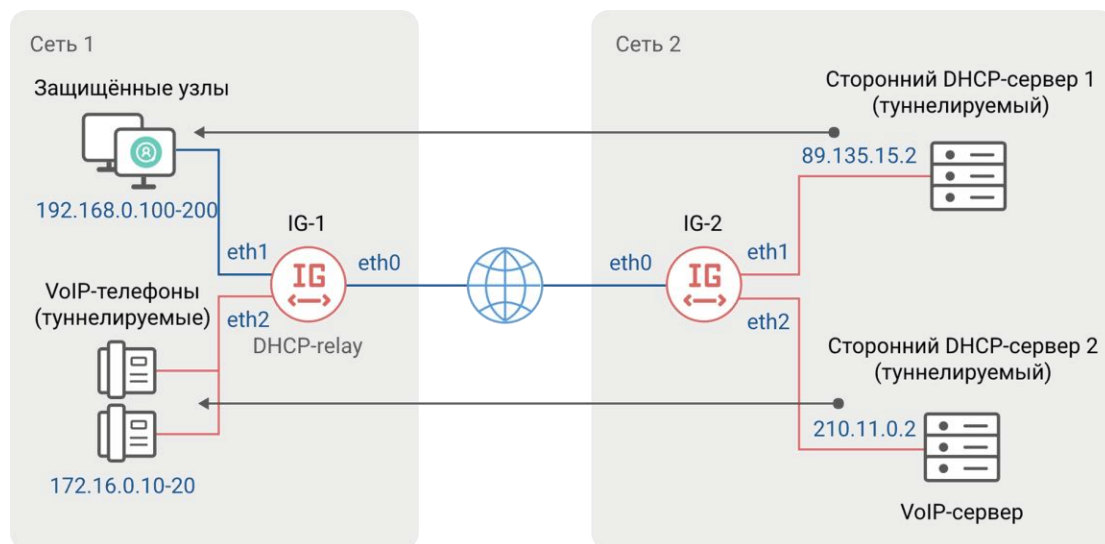


Рисунок 32. Использование сторонних DHCP-серверов

Связь между сетями осуществляется через интернет, поэтому для координатора, туннелирующего DHCP-серверы (IG-2), настроена видимость по реальным IP-адресам координатора, работающего в качестве агента DHCP-relay (IG-1).

В сети 2 сторонние DHCP-серверы работают для следующих DHCP-клиентов сети 1:

- DHCP-сервер 1 — для защищённых узлов (диапазон 192.168.0.100-200);
- DHCP-сервер 2 — для VoIP-телефонов, подключённых к координатору IG-2 с помощью туннелирования (диапазон 172.16.0.10-20).

DHCP-сервер 2 также выступает в роли VoIP-сервера для VoIP-телефонов в сети 1.

Задача

Настроить DHCP-relay на координаторе IG-1 для узлов сети 1.

Настройка

На координаторе IG-1:

1 Настройте службу DHCP-relay для защищённых узлов:

1.1 Задайте сетевой интерфейс `eth0` как внешний интерфейс службы DHCP-relay для защищённых узлов:

```
hostname# inet dhcp relay 1 add external-interface eth0 server 89.135.15.2
```

1.2 Задайте сетевой интерфейс `eth1` как слушающий интерфейс для защищённых узлов:

```
hostname# inet dhcp relay 1 add listen-interface eth1
```

2 Настройте службу DHCP-relay для туннелируемых узлов:

2.1 Задайте сетевой интерфейс `eth0` как внешний интерфейс службы DHCP-relay для туннелируемых узлов:

```
hostname# inet dhcp relay 2 add external-interface eth0 server 210.11.0.2
```

2.2 Задайте сетевой интерфейс `eth2` как слушающий интерфейс для туннелируемых узлов:

```
hostname# inet dhcp relay 2 add listen-interface eth2
```

3 Запустите службу DHCP-relay (обе копии):

```
hostname# inet dhcp relay 1 start
```

```
hostname# inet dhcp relay 2 start
```

4 Включите автоматический запуск службы DHCP-relay при загрузке ViPNet Coordinator IG:

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

Одновременное использование DHCP-сервера и DHCP-relay

Исходные данные

Имеется две удалённых друг от друга сети (сеть 1 и сеть 2). ViPNet Coordinator IG в сети 2 (IG-2) туннелирует сторонний DHCP-сервер, а ViPNet Coordinator IG в сети 1 (IG-1) туннелирует открытые узлы, которым требуется динамически выделять IP-адреса.

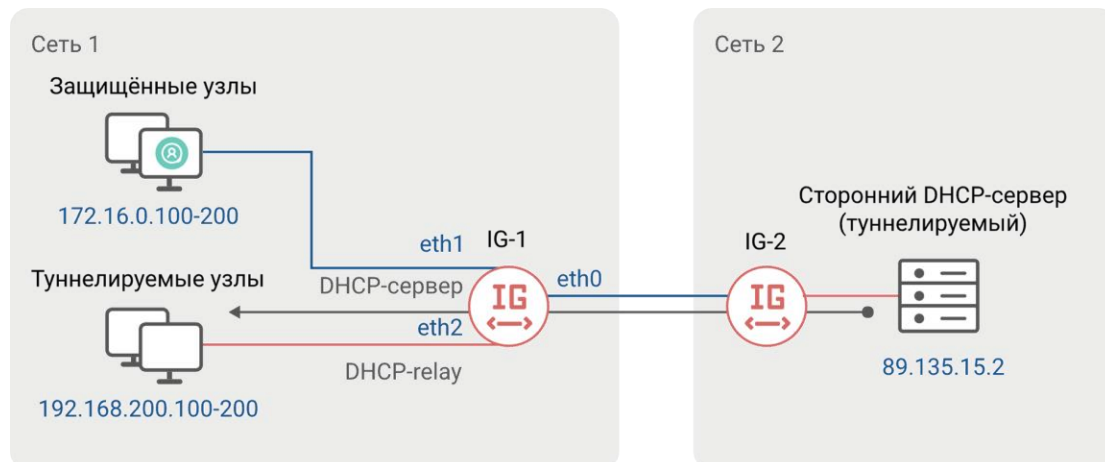


Рисунок 33. Одновременное использование DHCP-сервера и DHCP-relay

Между координатором IG-1 и туннелируемым DHCP-сервером настроена видимость по реальным IP-адресам.

Задача

Настроить координатор IG-1 таким образом, чтобы он работал одновременно как агент DHCP-relay для своих туннелируемых узлов и как DHCP-сервер для защищённых узлов сети 1.

Настройка

На координаторе IG-1:

- 1 Настройте DHCP-relay, принимающий данные от DHCP-сервера, стоящего за координатором IG-2:
 - 1.1 Задайте сетевой интерфейс eth0 как внешний интерфейс службы DHCP-relay:

```
hostname# inet dhcp relay add external-interface eth0 server 89.135.15.2
```
 - 1.2 Задайте сетевой интерфейс eth2 как слушающий интерфейс для туннелируемых узлов:

```
hostname# inet dhcp relay add listen-interface eth2
```
 - 1.3 Запустите службу DHCP-relay:

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

1.4 Включите автоматический запуск службы DHCP-relay при загрузке ViPNet Coordinator IG:

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

2 Выполните команды для настройки DHCP-сервера:

2.1 Задайте рабочий сетевой интерфейс DHCP-сервера:

```
hostname# inet dhcp server add interface eth1
```

2.2 Задайте диапазон IP-адресов, выделяемых DHCP-сервером для защищённых узлов:

```
hostname# inet dhcp server add range 172.16.0.100 172.16.0.200 interface eth1
```

2.3 Запустите DHCP-сервер:

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

2.4 Включите автоматический запуск DHCP-сервера при загрузке ViPNet Coordinator IG:

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

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

Рассмотрим пример, когда у организации есть несколько офисов, при этом в центральном офисе установлены корпоративные DNS- и NTP-серверы. Требуется, чтобы к этим серверам могли обращаться клиенты, находящиеся в удаленном офисе:

- 1 В сетевых настройках клиентов задайте IP-адрес ViPNet Coordinator IG в качестве DNS- и NTP-серверов. Клиенты удаленного офиса будут направлять запросы на разрешение доменных имен и на синхронизацию времени на ViPNet Coordinator IG.

- 2 На ViPNet Coordinator IG настройте обращение к корпоративному DNS-серверу:

- 2.1 Проверьте текущий список DNS-серверов пересылки:

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

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

- 2.2 Добавьте IP-адрес корпоративного DNS-сервера:

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

- 2.3 Проверьте текущие параметры DNS-сервера:

```
hostname# inet show dns
```

- 2.4 Запустите DNS-сервер:

```
hostname# inet dns start
```

- 2.5 Включите автоматический запуск DNS-сервера при загрузке ViPNet Coordinator IG:

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

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

- 3 На ViPNet Coordinator IG настройте обращение к корпоративному NTP-серверу:

- 3.1 Проверьте текущий список NTP-серверов:

```
hostname# inet ntp list
```

По умолчанию для синхронизации времени используются публичные NTP-серверы из кластера `vniiftri.ru`.

- 3.2 Добавьте IP-адрес или доменное имя корпоративного NTP-сервера:

```
hostname# inet ntp add server {<IP-адрес> | <доменное имя>}
```

3.3 Проверьте текущие параметры NTP-сервера:

```
hostname# inet show ntp
```

3.4 Запустите NTP-сервер:

```
hostname# inet ntp start
```

3.5 Включите автоматический запуск NTP-сервера при загрузке ViPNet Coordinator IG:

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

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

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

Защита соединения между удалёнными сегментами сети на канальном уровне модели OSI

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



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

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

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

Подробнее о технологии L2OverIP см. в документе «Настройка с помощью командного интерпретатора», раздел «Настройка туннелирования на канальном уровне по технологии L2OverIP».

Настройка L2OverIP при отсутствии VLAN

Исходные данные

В организации два удалённых друг от друга сегмента сети (сегмент 1 и сегмент 2). На границе каждого сегмента установлен ViPNet Coordinator IG (IG-1 в сегменте 1 и IG-2 в сегменте 2).

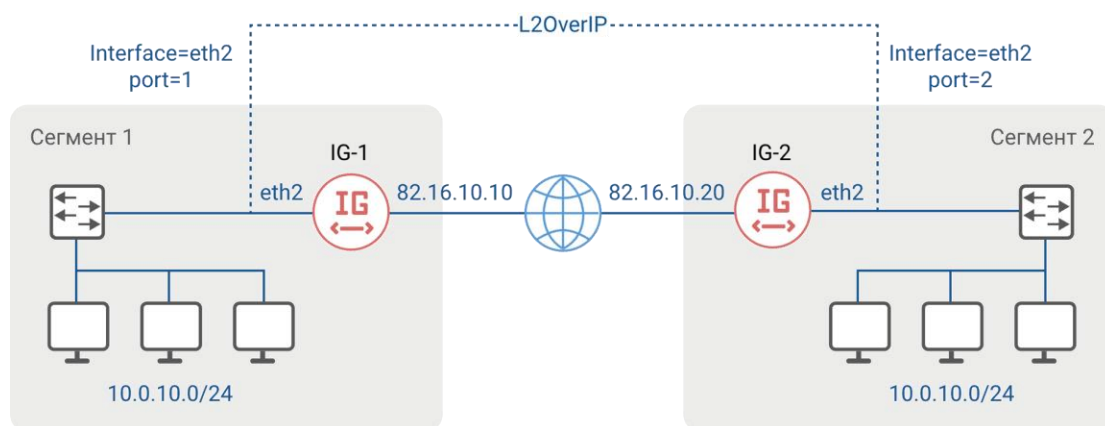


Рисунок 34. Защита соединения удалённых сегментов сети на канальном уровне модели OSI

Задача

Объединить 2 сегмента сети с помощью технологии L2OverIP.

Настройка

На каждом ViPNet Coordinator IG:

- 1 Задайте статический IP-адрес на внешнем сетевом интерфейсе, который будет использоваться для связи с удалённым сегментом по L2OverIP:

- На IG-1:

```
hostname# inet ifconfig eth0 address 82.16.10.10 netmask 255.255.255.0
```

- На IG-2:

```
hostname# inet ifconfig eth0 address 82.16.10.20 netmask 255.255.255.0
```

- 2 Для узлов обоих сегментов укажите шлюз по умолчанию — IP-адрес интерфейса eth2.

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

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

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

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

- На IG-1:

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

- На IG-2:

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

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

5 Задайте параметры удалённого сегмента сети, указав его номер порта и IP-адрес видимости удалённого ViPNet Coordinator IG:

- На IG-1:

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

- На IG-2:

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

Если настроена видимость по виртуальным адресам, укажите виртуальный IP-адрес вместо реального. Если ViPNet Coordinator IG имеет более одного адреса видимости для удалённого ViPNet Coordinator IG, используйте реальную видимость и реальные IP-адреса. Подробнее см. в документе «Настройка с помощью командного интерпретатора», раздел «Настройка L2OverIP».

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

Внимание! Если в процессе работы вы поменяете видимость удалённого узла, измените его адрес в настройках L2OverIP:



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

После изменения адреса удалённого узла перезапустите службу L2OverIP:

```
hostname# iplir set l2overip mode none
```

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

6 Добавьте сетевые фильтры защищённой сети, разрешающие любые соединения по протоколу 97:

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

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

7 Включите L2OverIP:

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

8 Убедитесь в корректности настроек L2OverIP:

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

После настройки и включения L2OverIP на всех ViPNet Coordinator IG, взаимодействие между узлами удалённых сегментов сети будет защищено на канальном уровне модели OSI.

Настройка L2OverIP в случае использования VLAN

С помощью функции L2OverIP вы можете объединить сегменты сети, состоящие из нескольких виртуальных локальных сетей (VLAN).

Исходные данные

Имеется два удалённых друг от друга сегмента сети (сегмент 1 и сегмент 2), в каждом из которых есть по три VLAN с одинаковыми номерами. Виртуальные сети в каждом сегменте объединены с помощью коммутатора в транковый порт, к которому подключается один из интерфейсов ViPNet Coordinator IG.

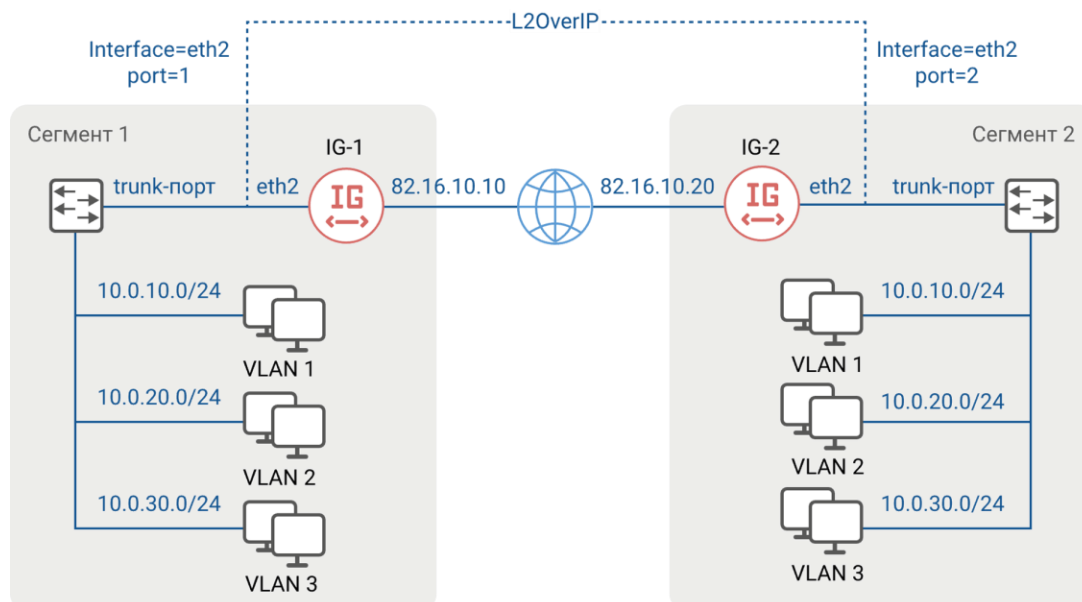


Рисунок 35. Схема объединения сегментов с одинаковым числом VLAN

Задача

Объединить 2 сегмента сети с помощью технологии L2OverIP.

Настройка

На каждом ViPNet Coordinator IG:

- 1 Разделите интерфейс `eth2`, подключённый к коммутатору, на виртуальные интерфейсы для VLAN с номерами 10, 20, 30 и выполните необходимые настройки (см. [Организация обработки трафика из нескольких VLAN](#)).
- 2 Задайте статический IP-адрес на внешнем сетевом интерфейсе, который будет использоваться для связи с удалённым сегментом по L2OverIP:
 - На IG-1:

```
hostname# inet ifconfig eth0 address 82.16.10.10 netmask 255.255.255.0
```
 - На IG-2:

```
hostname# inet ifconfig eth0 address 82.16.10.20 netmask 255.255.255.0
```
- 3 Для узлов обоих сегментов укажите шлюз по умолчанию — IP-адрес интерфейса `eth2`.
- 4 Укажите сетевой интерфейс, сегмент сети которого будет объединяться с другим удалённым сегментом с помощью функции L2OverIP:

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

5 Укажите параметры локального сегмента сети:

- На IG-1:

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

- На IG-2:

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

6 Укажите параметры удалённого сегмента сети:

- На IG-1:

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

- На IG-2:

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

Если настроена видимость по виртуальным адресам, укажите виртуальный IP-адрес вместо реального. Если ViPNet Coordinator IG имеет более одного адреса видимости для удалённого ViPNet Coordinator IG, используйте реальную видимость и реальные IP-адреса. Подробнее см. в документе «Настройка с помощью командного интерпретатора», раздел «Настройка L2OverIP».

7 Добавьте сетевые фильтры защищённой сети, разрешающие любые соединения по протоколу 97:

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

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

8 Включите функцию L2OverIP:

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

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

Если в приведённом примере требуется объединить только одну виртуальную сеть одного сегмента с той же виртуальной сетью другого сегмента, то в качестве рабочего интерфейса для функции L2OverIP вместо физического интерфейса `eth2` укажите виртуальный интерфейс для соответствующей VLAN (например, `eth2.10`).

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

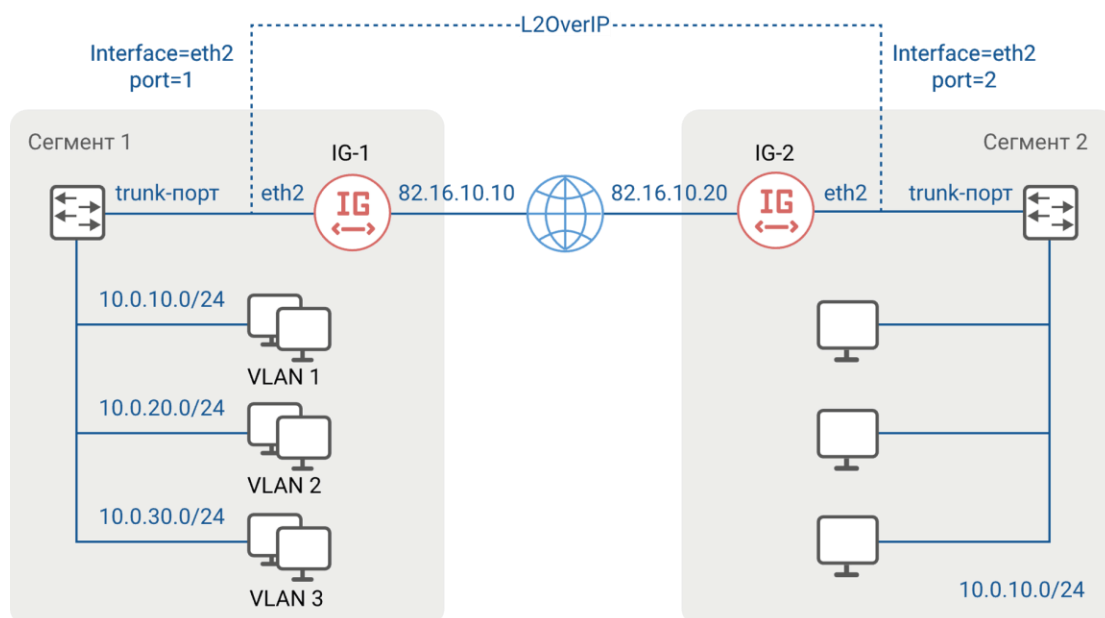


Рисунок 36. Схема объединения одной VLAN с сегментом без VLAN

В этом случае измените настройку следующим образом:

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

```
hostname# iplir set l2overip interface eth2.10
```

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

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

Настройка L2OverIP для обеспечения работоспособности протоколов динамической маршрутизации

Технология L2OverIP также может использоваться для объединения удаленных сегментов сети с целью организации между ними динамической маршрутизации по протоколу OSPF.

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

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

Исходные данные

Имеется два удалённых друг от друга сегмента сети, в каждом из которых создана разветвлённая структура узлов с несколькими маршрутизаторами. Каждый сегмент сети подключён к ViPNet Coordinator IG, установленному на границе сегмента. На всех маршрутизаторах обоих сегментов поддерживается работа по протоколу OSPF.

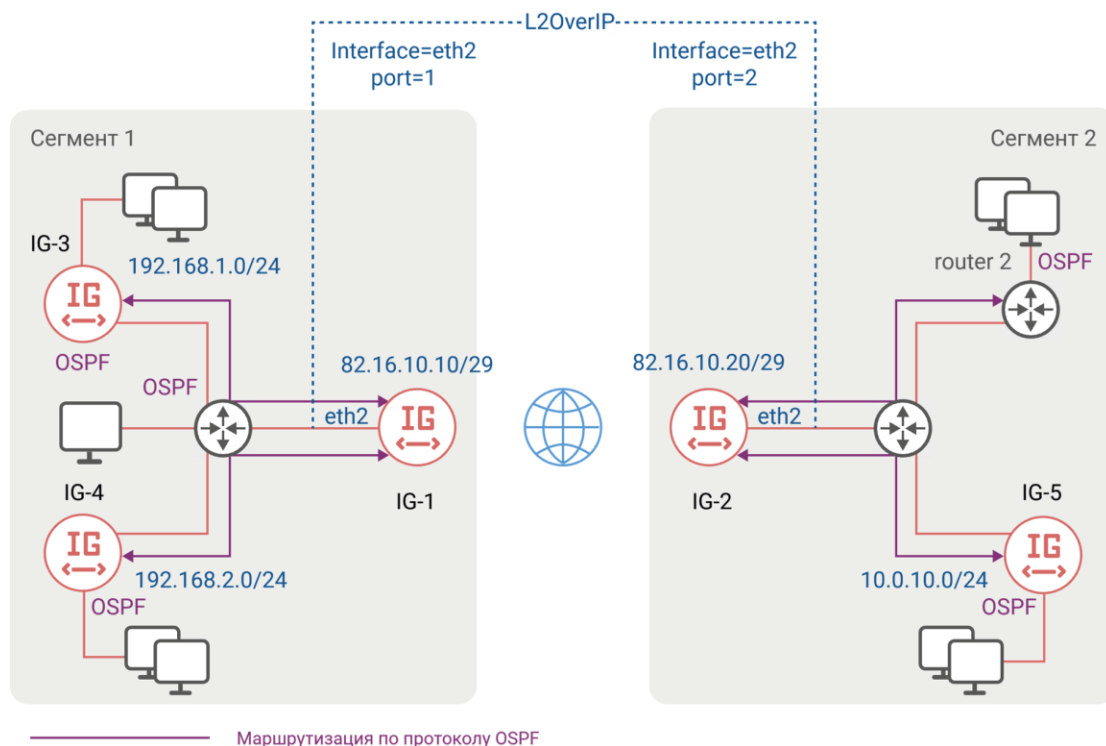


Рисунок 37. Схема работы L2OverIP и динамической маршрутизации по протоколу OSPF

Задача

- Объединить 2 сегмента сети с помощью технологии L2OverIP.
- Обеспечить работу протокола динамической маршрутизации OSPF.

Порядок действий

Чтобы организовать маршрутизацию по протоколу OSPF, требуется:

- 1 Настроить параметры L2OverIP на координаторах IG-1 и IG-2 (см. [Настройка параметров L2OverIP](#)).
- 2 Настроить работу по протоколу OSPF на маршрутизаторах router1, router2 и координаторах IG-3, IG-4, IG-5 (см. [Настройка протокола OSPF](#)).



Примечание. На IG-1 и IG-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

На координаторах IG-1 и IG-2:

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

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

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

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

- На IG-1:

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

- На IG-2:

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

- 3 Задайте параметры удалённого сегмента сети, указав номер порта и актуальный адрес видимости удалённого ViPNet Coordinator IG:

- На IG-1:

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

- На IG-2:

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

Если настроена видимость по виртуальным адресам, укажите виртуальный IP-адрес вместо реального. Если ViPNet Coordinator IG имеет более одного адреса видимости для удалённого ViPNet Coordinator IG, используйте реальную видимость и реальные IP-адреса. Подробнее см. в документе «Настройка с помощью командного интерпретатора», раздел «Настройка L2OverIP».

- 4 На каждом координаторе добавьте сетевые фильтры защищённой сети, разрешающие любые соединения по протоколу 97:

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

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

- 5 На каждом координаторе включите функцию L2OverIP:

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

- 6 На каждом координаторе убедитесь в корректности выполненных настроек функции L2OverIP:

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

Настройка протокола OSPF

На координаторах IG-3, IG-4, IG-5:

1 Включите использование протокола OSPF:

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

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

○ На IG-3:

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

○ На IG-4:

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

○ На IG-5:

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



Примечание. В рассматриваемом примере все координаторы, работающие по протоколу OSPF, принадлежат одной области маршрутизации.

3 Создайте фильтры открытой сети, 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, создайте следующие фильтры:

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

```
hostname# firewall local add 2 rule "Rule 2" src @any dst @multicast service @OSPF pass
```

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

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

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

Имя координатора	Название	Источник	Назначение	Протокол	Действие
	Rule 4	<IP-адрес router2>	@multicast	IP:89	pass

- 4 Создайте фильтр открытой сети, 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 «Справочник команд и конфигурационных файлов»). 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.

Настройка резервирования каналов с помощью BGP

Исходные данные

Корпоративная сеть компании состоит из двух удалённых сегментов (AS 1 и AS 2), подключённых к сети провайдера (AS 3). В каждом сегменте установлен ViPNet Coordinator IG100 I4 или ViPNet Coordinator IG1000 Q1.

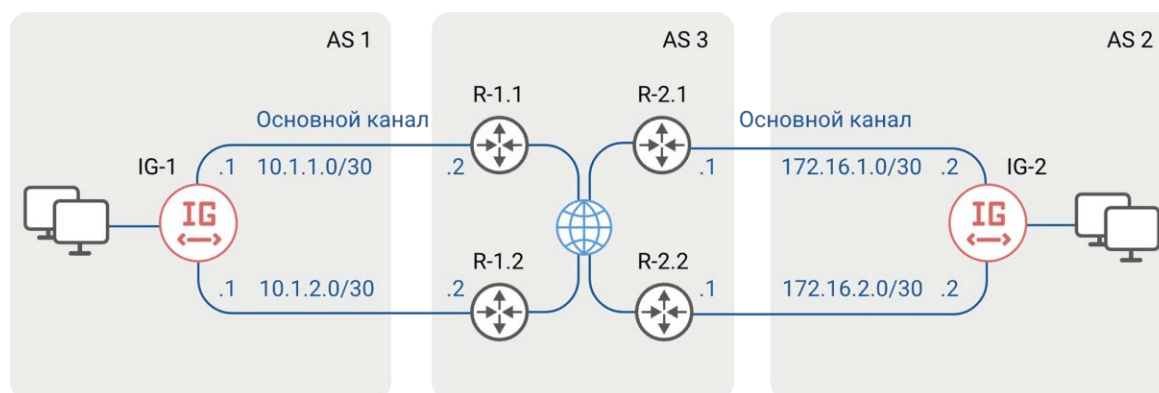


Рисунок 38. Подключение сегментов корпоративной сети (AS 1 и AS 2) к сети провайдера (AS 3)

Задача

Настроить резервирование каналов связи между сегментами AS 1 и AS 2 с помощью протокола BGP, чтобы:

- трафик между сегментом и провайдером передавался по выбранному основному каналу;
- при недоступности основного канала трафик перенаправлялся на резервный канал;
- при восстановлении работоспособности основного канала трафик перенаправлялся обратно на основной канал.

Настройка

Для выбора основного канала используется вес, назначаемый маршрутам, поступающим от провайдера. По умолчанию вес маршрутов равен нулю. С помощью карты маршрутов необходимо назначить маршрутам от маршрутизаторов R-1.1 и R-2.1 вес 100, чтобы выбрать основные каналы. Если основной канал станет недоступным, трафик будет перенаправлен на резервный. Если основной канал станет доступным, трафик будет перенаправлен на основной.

1 Настройте BGP на IG-1:

1.1 Включите BGP:

```
hostname# inet bgp on
```

1.2 Задайте номер автономной системы:

```
hostname# inet bgp router as 1
```

1.3 Настройте соседство с маршрутизаторами R-1.1 и R-1.2:

```
hostname# inet bgp neighbor add 10.1.1.2 remote-as 3
```

```
hostname# inet bgp neighbor add 10.1.2.2 remote-as 3
```

1.4 Проверьте, что с R-1.1 и R-1.2 установлены сессии:

```
hostname# inet bgp show summary
```

1.5 Проверьте, что в BGP-таблице есть маршруты до маршрутизаторов R-1.1 и R-1.2:

```
hostname# inet bgp show
```

1.6 Создайте карту маршрутов с именем SET_WEIGHT_100:

```
hostname# inet route-map add SET_WEIGHT_100
```

1.7 Добавьте в карту маршрутов блок 1:

```
hostname# inet route-map SET_WEIGHT_100 permit 1
```

1.8 Добавьте в блок 1 карты маршрутов действие «установить вес 100»:

```
hostname# inet route-map SET_WEIGHT_100 1 set weight 100
```

1.9 Назначьте карту маршрутов SET_WEIGHT_100 для входящих маршрутов от R-1.1:

```
hostname# inet bgp neighbor 10.1.1.2 set route-map SET_WEIGHT_100 in
```

2 Настройте BGP на IG-2:

2.1 Включите BGP:

```
hostname# inet bgp on
```

2.2 Задайте номер автономной системы:

```
hostname# inet bgp router as 2
```

2.3 Настройте соседство с маршрутизаторами R-2.1 и R-2.2:

```
hostname# inet bgp neighbor add 172.16.1.1 remote-as 3
```

```
hostname# inet bgp neighbor add 172.16.2.1 remote-as 3
```

2.4 Проверьте, что с R-2.1 и R-2.2 установлены сессии:

```
hostname# inet bgp show summary
```

2.5 Проверьте, что в BGP-таблице есть маршруты до маршрутизаторов R-2.1 и R-2.2:

```
hostname# inet bgp show
```

2.6 Создайте карту маршрутов с именем SET_WEIGHT_100:

```
hostname# inet route-map add SET_WEIGHT_100
```

2.7 Добавьте в карту маршрутов блок 1:

```
hostname# inet route-map SET_WEIGHT_100 permit 1
```

2.8 Добавьте в блок 1 карты маршрутов действие «установить вес 100»:

```
hostname# inet route-map SET_WEIGHT_100 1 set weight 100
```

2.9 Назначьте карту маршрутов SET_WEIGHT_100 для входящих маршрутов от R-2.1:

```
hostname# inet bgp neighbor 172.16.1.1 set route-map SET_WEIGHT_100 in
```

Использование ViPNet Coordinator IG в качестве защищённого интернет-шлюза

ViPNet Coordinator IG1000 Q1 может использоваться как **защищённый интернет-шлюз**, который позволяет связанным с ним клиентам переключаться между режимами работы в корпоративной сети:

- Работа в интернете. Блокируется любое взаимодействие с другими сетевыми узлами, кроме защищённого интернет-шлюза.
- Работа в корпоративной сети. Блокируется взаимодействие с защищённым интернет-шлюзом, доступ в интернет невозможен.

Чтобы использовать ViPNet Coordinator IG в качестве защищённого интернет-шлюза:

- 1 В ViPNet Prime включите функцию **Защищенный интернет шлюз** для вашего ViPNet Coordinator IG.
- 2 Настройте работу прокси-сервера ViPNet Coordinator IG:
 - 2.1 Завершите работу прокси-сервера, если он запущен:

```
hostname# service http-proxy stop
```
 - 2.2 Укажите сетевой интерфейс и порт, на котором прокси-сервер будет принимать запросы от клиентов, например интерфейс `eth2` и порт `3128`:

```
hostname# service http-proxy listen-address add eth2 3128
```
 - 2.3 Укажите внешний интерфейс, который используется для подключения к интернету:

```
hostname# service http-proxy external-address set eth0
```
 - 2.4 Добавьте сетевые фильтры и правила трансляции, необходимые для работы прокси-сервера:

```
hostname# service http-proxy fw-rules apply
```
 - 2.5 Включите автоматический запуск прокси-сервера при загрузке ViPNet Coordinator IG:

```
hostname# service http-proxy mode on
```
 - 2.6 Запустите службу прокси-сервера:

```
hostname# service http-proxy start
```
- 3 Добавьте локальный фильтр открытой сети, запрещающий соединения со стороны интернет-шлюза в защищённую сеть:

```
hostname# firewall local add src interface eth2 dst @PrivateNetworkIP drop
```
- 4 В настройках браузера клиентов, связанных с ViPNet Coordinator IG, укажите IP-адрес и порт прокси-сервера ViPNet Coordinator IG.

Для переключения режима работы клиента в корпоративной сети используются конфигурации ViPNet Client:

- Конфигурация «Открытый интернет» разрешает доступ в интернет и изолирует узел от других узлов корпоративной сети.
- Любая другая конфигурация запрещает доступ узла в интернет и разрешает доступ к узлам корпоративной сети.

Подробнее о работе с конфигурациями см. «ViPNet Client 4. Руководство пользователя».



Термины и сокращения

Modbus-преобразователь

Устройство, выполняющее преобразование интерфейса Ethernet в интерфейс RS-485 и обратно для обеспечения взаимодействия Ethernet- и Modbus-устройств.

TCP-туннель

Способ соединения клиентов ViPNet, находящихся во внешних сетях, со своим сервером соединений, а затем и с другими узлами сети ViPNet по протоколу TCP. Используется в том случае, если соединение по протоколу UDP заблокировано провайдерами услуг интернета.

TCP-туннель настраивается на координаторе, который является для клиента сервером соединений.

ViPNet Prime

ПО для централизованного управления решениями ViPNet. Позволяет управлять конфигурацией сети (включая устройства, пользователей и лицензии), централизованно обновлять ПО ViPNet и выполнять мониторинг состояния сети ViPNet.

Включает в себя основные функциональные модули:

- ViPNet VPN — модуль управления топологией сети, регистрирует защищаемые устройства и задает связи между ними.
- ViPNet Network Visibility System — модуль мониторинга состояния сети ViPNet и входящих в нее устройств.
- ViPNet Policy Management — модуль централизованного управления политиками безопасности узлов сети ViPNet.

Администратор сети ViPNet

Лицо, отвечающее за управление сетью ViPNet, создание и обновление справочников и ключей для сетевых узлов ViPNet, настройку межсетевого взаимодействия с доверенными сетями.

Адреса видимости

IP-адреса, виртуальные или реальные, по которым данный узел видит остальные узлы сети ViPNet и по которым приложения отправляют свой трафик.

Защищенный интернет-шлюз (открытый интернет)

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

Кластер горячего резервирования

Кластер горячего резервирования состоит из двух взаимосвязанных ViPNet Coordinator IG, один из которых (активный) выполняет функции координатора сети ViPNet, а другой (пассивный) находится в режиме ожидания. В случае сбоев, критичных для работоспособности ПО ViPNet Coordinator IG на активном ViPNet Coordinator IG, пассивный ViPNet Coordinator IG переключается в активный режим для выполнения функций. При этом сбойный ViPNet Coordinator IG 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 ViPNet, представляющий собой компьютер с установленным программным обеспечением координатора (ViPNet Coordinator) или программно-аппаратный комплекс. В сети ViPNet-координатор выполняет серверные функции, а также маршрутизацию трафика и служебной информации.

Сетевой узел ViPNet

Узел, на котором установлено программное обеспечение ViPNet, зарегистрированный в управляющем ПО.

Сеть ViPNet

Логическая сеть, организованная с помощью программного обеспечения ViPNet и представляющая собой совокупность защищённых узлов ViPNet. Сеть ViPNet имеет наложенную

маршрутизацию, обеспечивающую взаимодействие узлов сети. Каждая сеть ViPNet имеет свой уникальный номер.

Туннелирование

Технология для защиты соединений между устройствами локальных сетей, которые связаны через интернет или другие публичные сети. Шифрование трафика устройств выполняется координаторами, установленными на границах локальных сетей.

Туннелируемый узел

Узел, на котором не установлено программное обеспечение ViPNet с функцией шифрования трафика на сетевом уровне, но его трафик на потенциально опасном участке сети зашифровывается и расшифровывается на координаторе, за которым он стоит.

Шлюз Modbus

Программно-аппаратная функциональность, выполняющая преобразование трафика интерфейса Ethernet в интерфейсы RS-232 и RS-485 и обратно для обеспечения взаимодействия Ethernet- и Modbus-устройств.