Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Лесосибирский педагогический институт –
филиал Сибирского федерального университета

Кафедра высшей математики, информатики и естествознания

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Н.Ф, Романцова
подпись
« 8 » июня 2018 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 Информационные системы и технологии

Разработка корпоративной сети предприятия под управлением ОС Linux

Руководитель _____ доцент, канд.филос.наук М.В.Румянцев
подпись, дата

Выпускник _____ А.А.Игнатьев
подпись, дата

Лесосибирск 2018

РЕФЕРАТ

Выпускная квалификационная работа по теме «Организация корпоративной сети предприятия на базе ОС Linux» содержит 69 страниц текстового документа, 2 приложения, 40 использованных источников.

КОРПОРАТИВНАЯ СЕТЬ, ЧАСТНАЯ СЕТЬ, VPN, VYOS, TCP/IP, OSI, IPSEC, LINUX

Объект исследования: корпоративная сеть.

Цель исследования: разработка корпоративной сети предприятия.

Задачи исследования:

– проанализировать специальную и научную литературу по теме исследования;

– выявить требования к корпоративной сети;

– сформировать структуру и методы реализации корпоративной сети на основе выявленных требований;

– реализовать корпоративную сеть предприятия.

В ходе выполнения выпускной квалификационной работы проанализирована специальная и научная литература по теме исследования.

В результате проведения исследования были выявлены требования к корпоративной сети, сформирована структура и обоснованы методы реализации корпоративной сети на основе выявленных требований.

Разработана и апробирована корпоративная сеть на предприятии.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

На сегодняшний день интернет является самым популярным средством передачи данных в мире.

Его просторы размещают более одного миллиарда веб и бесчисленное количество менее понятных для обычного пользователя серверов. АТС, базы данных, файловые хранилища не могут функционировать или теряют смысл без возможности доступа, которую обеспечивает сеть. Естественно, что при текущих скоростях работы глобальной сети и предоставляемых ее средствами сервисах, большинство компаний стараются организовать общий доступ к корпоративным ресурсам через интернет, вне зависимости от расположения офисов и расстояний между ними. Кроме того, наличие КС позволяет управлять трафиком и настраивать политики доступа в зависимости от нужд и необходимостей каждого из отделов. Исходя из вышесказанного, создание и поддержка КС является необходимой мерой для успешного ведения бизнеса.

Объект исследования: корпоративная сеть.

Предмет исследования – особенности организации корпоративной сети предприятия.

Цель исследования: разработка корпоративной сети предприятия.

Задачи:

– проанализировать специальную и научную литературу по теме исследования;

– выявить требования к корпоративной сети;

– сформировать структуру и методы реализации корпоративной сети на основе выявленных требований;

– реализовать корпоративную сеть предприятия.

Создание корпоративной сети является основным способом обеспечить доступ к актуальной и необходимой информации внутри предприятия в реальном времени. Абсолютно любое клиент-серверное приложение требует, помимо клиента и сервера, канала связи между ними. Фактически, при

необходимости, можно связать любые два компьютера (не важно, являются они клиентскими или серверными) посредством интернета, словно они находятся в одной локальной сети. Кроме того, появится возможность разграничивать доступ к определенным типам ресурсов как в глобальной сети (интернет) так и в локальной сети предприятия.

1 Теоретические основы разработки сети

Исторически сложилось, что базовая эталонная модель взаимодействия открытых систем – сетевая модель OSI является общепринятой моделью стека сетевых протоколов, определяющей различные уровни взаимодействия систем. На каждый уровень накладываются определенные функции, которые исполняются в зависимости от возложенных на него задач. Всего таких уровней семь, описание в научной литературе обычно начинается с последнего, не станет исключением и данная работа.

1.1 OSI

Условно модель OSI делится на две глобальные части: уровни хоста и уровни среды передачи данных:

I. Уровни хоста (хостом здесь и далее называется любое устройство, подключенное к локальной или глобальной сети) включают:

– прикладной уровень. Используется для доступа к сетевым службам.

– представительский уровень. Иначе, именуемый уровнем представления (представление и шифрование данных).

– сеансовый уровень. Исходя из названия, используется для управления сеансом связи.

– транспортный. Прямая связь между конечными пунктами и надежность.

II. Уровни среды передачи данных:

– сетевой уровень определяет маршрут и логическую адресацию;

– канальный уровень определяет физическую адресацию;

– физический уровень работает со средой передачи данных, сигналами и двоичными данными.

Нижние уровни модели OSI (с 1 по 3) управляют физической доставкой данных по сети и реализуются в виде аппаратных средств и программного обеспечения.

Верхние уровни модели OSI (с 4 по 7) обеспечивают точную доставку данных между приложениями, работающими на сетевых узлах, и обычно реализуются только на программном уровне.

Таблица 1 – Модель OSI

	Уровень	**Тип обрабатываемых данных**	**Функции**
7	**Уровень приложений**	Пользовательские данные	Предоставление сервисов для сетевых приложений
6	**Уровень представлений**	Пользовательские данные	Общий формат представления данных, сжатие и шифрование
5	**Сеансовый уровень**	Пользовательские данные	Установление, управление и завершение сессий между приложениями
4	**Транспортный уровень**	Сегменты/дейтаграммы	Адресация процессов, сегментация/ повторная сборка данных, управление потоком, надежная доставка
3	**Сетевой уровень**	Пакеты/дейтаграммы	Передача сообщений между удаленными устройствами, выбор наилучшего маршрута, логическая адресация
2	**Канальный уровень**	Кадры	Доступ к среде передачи, передача сообщений между локальными устройствами, физическая адресация
1	**Физический уровень**	Биты	Передача электрических и оптических сигналов между устройствами

Модель OSI не описывает службы и протоколы, используемые на каждом уровне, она определяет набор действий, которые должен выполнить уровень для передачи информации между узлами.

Модель OSI определяет схему обмена данными между сетевыми узлами, но сама не является способом такого обмена. Обмен данными становится возможным благодаря *протоколам*.

Протокол – это формальный набор правил и соглашений, регламентирующий обмен информацией между узлами по сети. Он реализует функции одного или нескольких уровней модели OSI[9].

Однако физическое соединение устройств выполняется только на физическом уровне модели OSI, следовательно, чтобы данные были переданы по сети другому устройству, они должны «спуститься» с уровня приложений на физический уровень внутри передающего узла. Когда данные будут переданы по каналу связи, физический уровень устройства-получателя извлечет их из среды передачи и передаст вышележащему уровню. Таким образом, реальное взаимодействие одноименных уровней происходит *по вертикали* посредством взаимодействия с соседними уровнями (нижележащим и вышележащим) своего *стека протоколов*.

Стек протоколов – совокупность протоколов разных уровней. Наиболее известным является стек протоколов TCP/IP. Правила и процедуры, которые отвечают за взаимодействие между соседними уровнями, называются *интерфейсами.*

Инкапсуляция – это процесс, при котором к данным добавляется служебная информация определенного протокола (уровня) перед отправкой в сеть[4].

Уровень приложений *(Application layer)* – это седьмой, самый близкий к пользователю уровень модели OSI. Он отличается от других уровней тем, что не предоставляет услуги ни одному другому уровню модели OSI, а только обслуживает прикладные процессы, находящиеся вне пределов модели OSI. Примерами могут служить Web-браузер, который является прикладным процессом, запущенным на компьютере и использующим сервисы, предоставляемые протоколом прикладного уровня HTTP (Hypertext Transfer Protocol); почтовый клиент, использующий сервисы протокола POP3 (Post Office Protocol Version 3).

Уровень представлений *(Presentation layer)* – шестой уровень модели OSI. Он отвечает за то, чтобы информация, посылаемая уровнем приложений одной системы, могла быть прочитана уровнем приложений другой системы. При необходимости уровень представлений преобразует форматы данных путем использования общего формата представления информации. Также он

может выполнять сжатие (распаковку) данных с целью повышения пропускной способности сети.

Сеансовый уровень *(Session layer)* – пятый уровень модели OSI. Как следует из названия, он позволяет двум прикладным процессам устанавливать, управлять и завершать сеансы связи (сессии) друг с другом. Сеансовый уровень синхронизирует диалог между прикладными процессами и отвечает за восстановление аварийно прерванных сеансов связи. Технологии сеансового уровня часто реализованы в виде набора программных средств, называемых *application program interfaces* (API, прикладной программный интерфейс). API предоставляют набор сервисов, позволяющих программистам разрабатывать сетевые приложения, не заботясь о транспортировке, адресации и доставке данных. Эти функции выполняют нижележащие уровни модели OSI.

На транспортном уровне *(Transport layer)* выполняется целый ряд функций. Наиболее важными из них являются *контроль ошибок*, их *исправление* и *управление потоком данных*. Транспортный уровень отвечает за надежную работу служб межсетевой передачи данных, функции которой выполняются незаметно для программ более высокого уровня.

Сетевой уровень *(Network layer)* – третий уровень модели OSI. Он является одним из самых важных уровней модели OSI и отвечает за соединение узлов, расположенных в географически удаленных друг от друга сетях. Сетевой уровень выполняет две основные функции – логическую адресацию и маршрутизацию. Каждому устройству, подключенному к сети, назначается логический адрес, который также называют адресом 3 уровня. Он используется для маршрутизации пакетов.

Маршрутизация – это процесс определения наилучшего маршрута передачи информации от отправителя к получателю, когда отправитель и получатель находятся в разных сетях, соединенных произвольным образом. Также на сетевом уровне решаются задачи управления потоком данных и диагностики ошибок передачи. Сетевой уровень выполняет инкапсуляцию сегментов, полученных от транспортного уровня в *пакеты* (также называемые

дейтаграммами). Основным протоколом сетевого уровня является протокол *IP (Internet Protocol)* [5].

Канальный уровень *(Data link layer)* – второй уровень модели OSI. Он обеспечивает сетевым узлам доступ к среде передачи и решает вопросы физической адресации (в противоположность сетевой или логической адресации), обнаружения и коррекции ошибок, упорядоченной доставки кадров, логической топологии. Канальный уровень завершает процесс инкапсуляции и помещает дейтаграммы (пакеты), полученные с сетевого уровня в *кадры*.

Физический уровень (*Physical layer*) – первый уровень модели OSI. На физическом уровне выполняются наиболее важные функции передачи данных по сравнению со всеми другими уровнями. К физическому уровню относятся все соединители, кабели, спецификации частот, требования к расстояниям и задержкам при распространении сигналов, регламентируемые напряжения, другими словами, все физические параметры.

Стоит отметить, что модель OSI так и не воплотилась в жизнь в виду своей многоуровневости и сложности, хотя де-юре является стандартом. Самый популярный используемый стек протоколов на сегодняшний день – TCP/IP, созданный в 1972 и представленный в июле 1976 года. Конкретней об особенностях реализации, связи с моделью OSI и конкретных протоколах ниже.

1.2 Стек протоколов TCP/IP

Стек протоколов TCP/IP включает в себя четыре уровня, полностью реализующие функциональные возможности модели OSI (таблица 2):

Таблица 2 –- Сравнение модели OSI и стека протоколов TCP/IP

OSI	TCP/IP	Протоколы
7) Прикладной уровень 6)Представительский уровень 5) Сеансовый уровень	4) Прикладной уровень	HTTP, FTP, SSH и т.д.
4) Транспортный	3)Транспортный уровень	TCP, UDP
3) Сетевой	2) Уровень Интернет	IP, NAT, OSPF и т. д.
2) Канальный 1) Физический	1) Уровень доступа к среде	Ethernet, IEEE 802.11, Token Ring и т. д.

В следующих параграфах рассмотрим каждый из уровней.

1.3 Канальный уровень

Для обеспечения адресации узлов в локальной сети в заголовке кадров должны присутствовать адрес отправителя и адрес получателя. Большинство протоколов канального уровня семейства IEEE 802 для идентификации устройств используют физический адрес или *MAC-адрес* (MAC address).

MAC-адрес (Media Access Control) – это уникальный идентификатор, который присваивается каждому сетевому устройству во время изготовления. Он позволяет уникально идентифицировать каждый узел сети и доставлять данные только этому узлу[6].

На сегодняшний день технология Ethernet является самой распространенной технологией локальных сетей благодаря своей простоте и универсальности. В процессе стандартизации кадр Ethernet приобрел следующий вид:

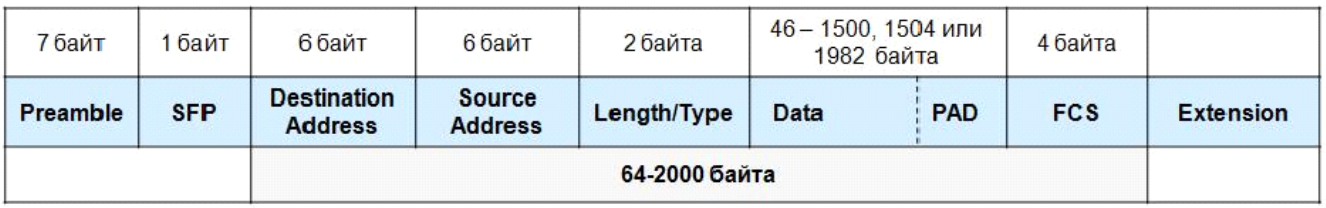

Рисунок 1 – Кадр Ethernet

Кадр содержит семь обязательных полей:

– *Preamble* (преамбула) – состоит из семи синхронизирующихся байт 10101010;

– *Start-of-Frame-Delimiter* (SFP, начальный ограничитель кадра) – содержит значение 10101011. Эта комбинация указывает на то, что следующий байт – это начало заголовка кадра;

– *Destination Address* (DA, адрес назначение) – MAC-адрес получателя кадра;

– *Source Address* (SA, адрес источника) – MAC-адрес отправителя кадра;

– *Length/Type* (длина/тип) – а) если значение меньше или равно 0x05DC (1500 в десятичной системе счисления), то поле указывает на длину поля данных в кадре (интерпретируется как длина); б) если значение больше или равно 0x0600 (1536 в десятичной системе счисления), то поле указывает на тип протокола, вложившего пакет в поле данных кадра (интерпретируется как тип);

– *Data* (данные) – поле данных переменной длины. Минимальная длина поля 46 байт, максимальная длина поля –1500 байт (для стандартных кадров), 1504 байт (для кадров, содержащих тег протокола IEEE 802.1Q), 1982 байт (для расширенных (envelope) кадров);

– *Pad* (Padding, заполнение) – состоит из такого количества байт заполнителей, которое обеспечивает минимальную длину поля данных в 46 байт. Это обеспечивает корректное распознавание коллизий при работе протокола CSMA/CD. Если длина поля данных достаточна, поле заполнения в кадре отсутствует;

– *Frame Check Sequence* (FCS, поле контрольной суммы) – содержит контрольную сумму кадра. Служит для проверки и проверяет не искажен ли кадр. Значение поля вычисляется на основе содержимого полей DA, SA, Length/Type, поля данных и заполнения с помощью 32-разрядного циклического избыточного кода (Cyclic Redundancy Code, CRC);

– Поле *Extension* (расширение) следует за полем FCS и состоит из последовательности битов, которые отличаются от битов данных и используются для выполнения процедур сетевого управления. Если эти процедуры не требуются, длина поля будет равна нулю. Это поле не используется при вычислении контрольной суммы кадра.

Коммутируемая сеть Ethernet (*Ethernet switched network*) – сеть Ethernet, сегменты которой соединены мостами или коммутаторами [17].

Коммутаторы локальных сетей обрабатывают кадры на основе *алгоритма прозрачного моста (transparent bridge*), который определен стандартом IEEE 802.1D. Процесс работы алгоритма прозрачного моста начинается с построения *таблицы коммутации* (Forwarding DataBase, FDB) или *таблицы MAC-адресов*.

```
DES-3528:5#show fdb
Command: show fdb

 Unicast MAC Address Aging Time  = 300

 VID  VLAN Name                         MAC Address       Port  Type
 ---- --------------------------------- ----------------- ----- ----------

 1    default                           1C-AF-F7-4C-5C-70 CPU   Self
 1    default                           20-6A-8A-72-A5-82 1     Dynamic
 1    default                           20-6A-8A-73-7C-8C 9     Permanent

Total Entries: 3
```

Рисунок 2 – Таблица MAC-адресов

Согласно представленному рисунку коммутатор перенаправляет запрос не всем устройствам в сети, а к конкретному порту, за которым находится определенный мак-адрес, кроме случая отправки широковещательных сообщений.

1.3.1 Виртуальные локальные сети (VLAN)

В соответствии с логикой работы алгоритма прозрачного моста коммутатор рассылает широковещательные кадры через все порты (за исключением порта-приемника такого кадра). Таким образом, все устройства сети, построенной на коммутаторах, находятся в одном *широковещательном домене*. Широковещательный домен – это область распространения широковещательных кадров. Широковещательные кадры используются при

работе многих сетевых протоколов, таких как ARP или DHCP. Большой объем широковещательных кадров в сети, особенно крупной, приводит к нерациональному использованию полосы пропускания. Проблема ограничения распространения широковещательного трафика в сетях, построенных на коммутаторах, решается с помощью технологии *виртуальных локальных сетей* (Virtual LAN, VLAN).

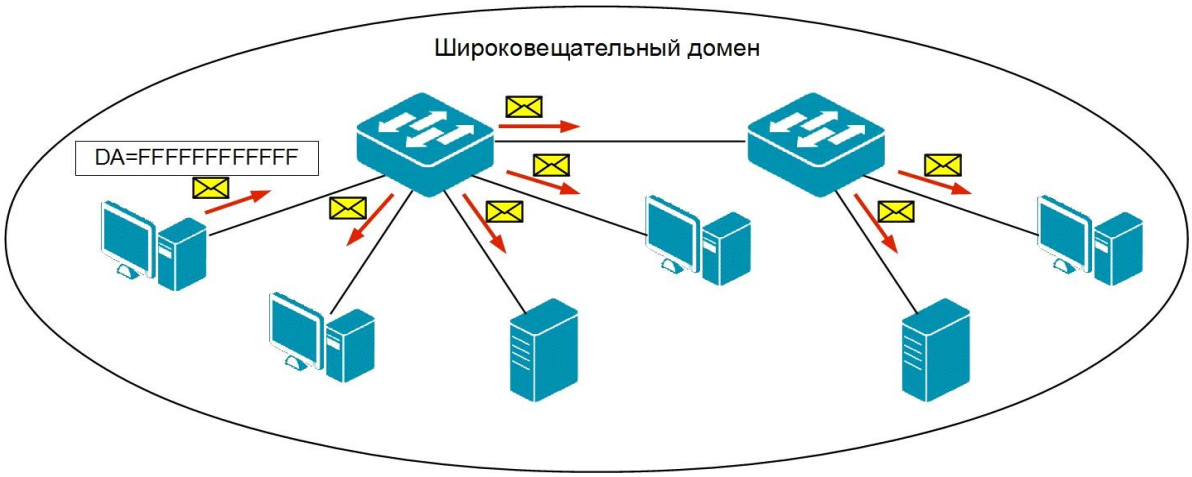

Рисунок 3 – Широковещательный домен

Виртуальной локальной сетью называется логическая группа узлов сети, трафик которой, в том числе и широковещательный, полностью изолирован от других узлов сети на канальном уровне[10]. Это означает, что передача кадров между разными виртуальными сетями на основании MAC-адреса невозможна независимо от типа адреса – индивидуального, группового или широковещательного. В то же время внутри виртуальной сети кадры передаются по технологии коммутации, то есть только на тот порт, который связан с MAC-адресом назначения кадра. Таким образом, с помощью виртуальных сетей решается проблема распространения широковещательных кадров и вызываемых ими следствий, которые могут развиться в широковещательные штормы и существенно снизить производительность сети.

VLAN обладают следующими преимуществами:

– гибкость внедрения – VLAN являются эффективным способом группировки сетевых пользователей в виртуальные рабочие группы независимо от их физического размещения в сети;

– ограничивают распространение широковещательного трафика, что увеличивает полосу пропускания, доступную для пользователя;

– позволяют повысить безопасность сети, определив с помощью фильтров, настроенных на коммутаторе или маршрутизаторе, политику взаимодействия пользователей из разных виртуальных сетей.

1.4 Сетевой уровень

При построении сетей передачи данных часто возникает задача организации связи между различными сетями или подсетями, которые образуют *составную сеть*. Так, например, в локальных сетях, логически сегментированных с использованием виртуальных локальных сетей (VLAN), администраторам часто требуется организовать передачу данных между ними. Эта задача решается с помощью функций *сетевого уровня* (network layer).

Различные сети могут быть построены с использованием различных протоколов канального и физического уровня. Таким образом, они используют различные форматы кадров, методы доступа к среде передачи, методы модуляции и кодирования. Для того чтобы соединить такие сети, нужен общий межсетевой уровень, использующий понятный всем нижележащим сетям протокол.

Основным протоколом сетевого уровня является протокол IP (Internet Protocol), который позволяет передавать данные в сетях TCP/IP между узлами составной сети и выполняет четыре основные функции:

– адресацию узлов;

– инкапсуляцию данных;

– фрагментацию и последующую сборку пакетов;

– маршрутизацию.

Протокол IP не гарантирует надежной доставки пакета до адресата, эта функция выполняется протоколами более высокого уровня.

1.4.1 Протокол IP версии 4

Данные, передаваемые с использованием протокола IPv4, помещаются в сообщения, называемые **пакетами** или **дейтаграммами** [7]. Протокол IPv4 использует пакет, который условно можно разделить на заголовок длиной, как правило, 20 байт и данные. Заголовок содержит адресные и управляющие поля, а в поле *Данные* находится непосредственно информация, которая передается через составную сеть. В отличие от формата некоторых других протоколов, например Ethernet, пакет IPv4 не содержит следующего за полем *Данные* контрольной суммы всего пакета.

Рисунок 4 – Структура пакета IPv4 [12]

Пакет IPv4 состоит из следующих полей:

– Версия (Version) – для IPv4 значение поля равно 4;

– Длина заголовка (IHL, Internet Header Length) – указывает на начало блока данных в пакете. Обычно значение для этого поля равно 5;

– Тип сервиса (ToS, Type of Service) – содержит информацию, требуемую для обеспечения функций качества обслуживания (QoS);

– Общая длина (TL, Total Length) – общая длина пакета с учетом заголовка и поля данных;

– Идентификатор пакета (Identification) – используется для распознавания пакетов, образовавшихся путем фрагментации исходного пакета;

– Флаги (Flag) – содержит признаки, связанные с фрагментацией пакета;

– Смещение фрагмента (Fragment Offset) – значение, определяющее позицию фрагмента в потоке данных;

– Время жизни (TTL, Time to Live) – временной интервал, в течение которого пакет может перемещаться по сети маршрутизаторами;

– Протокол (Protocol) – указывает, какому протоколу верхнего уровня принадлежит информация, размещенная в поле данных пакета;

– Контрольная сумма (Header Checksum) – рассчитывается по заголовку и позволяет определить целостность заголовка пакета;

– Адрес источника (Source IP Address) и адрес назначения (Destination IP Address) – указывают отправителя и получателя пакета;

– Опции (Options) – необязательное поле, может использоваться при отладке работы сети;

– Данные (Data) – данные передаваемые в пакете: или полное сообщение, полученное от вышележащего уровня или его фрагмент.

Заголовок IPv4, как правило, имеет длину 20 байт. При использовании необязательного поля Опции (Options), длина заголовка может быть увеличена в зависимости от количества опций, но всегда остается кратной 32 битам.

Основной задачей протокола IP является передача данных между устройствами составной сети, для чего необходима информация о расположении адресата. Идентифицировать адресата и определить маршрут до него позволяет IP-адрес.

В отличие от физического адреса (MAC-адреса), который присваивается каждому сетевому устройству во время изготовления и позволяет уникально идентифицировать каждый узел сети, IP-адрес идентифицирует *сетевой интерфейс* (интерфейс подключения к сети), а не само устройство.

Любое устройство, которое передает данные, используя сетевой уровень, будет иметь как минимум один уникальный IP-адрес для сетевого интерфейса. Например, таким сетевым узлам, как компьютеры (если установлена одна

сетевая карта) и сетевые принт-серверы обычно присваивают один IP-адрес. Маршрутизаторам или коммутаторам третьего уровня может быть присвоено более одного IP-адреса, т.к. они могут использоваться для соединения нескольких сетей.

1.4.2 Таблица маршрутизации

Как уже было сказано в параграфе 1.1, *маршрутизация* – это процесс определения наилучшего маршрута передачи информации от отправителя к получателю, когда отправитель и получатель находятся в разных сетях, соединенных произвольным образом. Для выбора маршрута роутер использует таблицу маршрутизации.

Таблица маршрутизации – таблица, состоящая из сетевых маршрутов и предназначенная для определения наилучшего пути передачи сетевого пакета [14]. Каждая запись в таблице маршрутизации состоит, как правило, из таких полей:

– *адрес сети назначения* (destination);

– *маска сети назначения* (netmask, genmask);

– *адрес шлюза* (gateway), за исключением тех случаев, когда описывается в маршрут непосредственно доступную (directly connected) сеть, в этом случае вместо адреса шлюза обычно указываются 0.0.0.0;

– *метрика маршрута* (не всегда).

Пример таблицы маршрутизации:

default via 192.168.0.1 dev enp3s0 proto dhcp metric 100

default via 192.168.0.1 dev enp3s0 proto dhcp src 192.168.0.100 metric 1024

192.168.0.0/24 dev enp3s0 proto kernel scope link src 192.168.0.100

192.168.0.0/24 dev enp3s0 proto kernel scope link src 192.168.0.100 metric 100

192.168.0.1 dev enp3s0 proto dhcp scope link src 192.168.0.100 metric 1024

При отправке сетевого пакета, система смотрит, по какому именно маршруту он должен быть отправлен, основываясь на таблице маршрутизации.

Как правило, выбирается наиболее конкретный (то есть, с наиболее длинной сетевой маской) маршрут из тех, которые соответствуют адресу отправителя. Если ни один из маршрутов не подходит, пакет уничтожается, а его отправителю возвращается сообщение *No route to host*.

1.4.3 NAT

NAT(Network Address Translation) – преобразование сетевых адресов – метод отображения одного пространства IP-адресов в другое, с изменением сетевой адресной информации в заголовке IP пакетов, проходя через маршрутизатор [8]. Свою популярность этот протокол обрел с целью сохранения глобального адресного пространства в связи с исчерпанием адресов IPv4. Один интернет-маршрутизируемый IP-адрес шлюза NAT может использоваться для всей частной сети.

Подмена IP - метод, скрывающий все пространство IP-адресов, обычно состоящий из частных IP-адресов, находящихся за одним устройством, имеющим доступ в интернет. Скрываемый адрес изменяется в единственный (общедоступный) IP-адрес как «новый» исходный адрес исходящего пакета IP, таким образом, что конечное устройство, на которое отправлен пакет, работает с роутером, имеющим внешний IP-адрес.

1.5 Транспортный уровень

1.5.1 TCP – Transmission Control Protocol

Обмен данными, ориентированный на соединения, может использовать надежную связь, для обеспечения которой протокол четвертого уровня посылает подтверждения о получении данных и запрашивает повторную передачу, если данные не получены или искажены. Протокол TCP использует именно такую надежную связь. TCP используется в таких прикладных протоколах, как HTTP, FTP, SMTP и Telnet.

Протокол TCP требует, чтобы перед отправкой сообщения было открыто соединение. Серверное приложение должно выполнить так называемое пассивное открытие (passive open), чтобы создать соединение с известным номером порта, и, вместо того чтобы отправлять вызов в сеть, сервер переходит в ожидание поступления входящих запросов. Клиентское приложение должно выполнить активное открытие (active open), отправив серверному приложению синхронизирующий порядковый номер (SYN), идентифицирующий соединение. Клиентское приложение может использовать динамический номер порта в качестве локального порта.

Сервер должен отправить клиенту подтверждение (ACK) вместе с порядковым номером (SYN) сервера. В свою очередь клиент отвечает АСК, и соединение устанавливается.

После этого может начаться процесс отправки и получения сообщений. При получении сообщения в ответ всегда отправляется сообщение АСК. Если до получения АСК отправителем истекает тайм-аут, сообщение помещается в очередь на повторную передачу.

В рамках выпускной квалификационной работы избыточно рассматривать полный набор полей заголовка TCP, остановимся на первых двух:

Таблица 3 – Поля заголовка TCP

Поле	**Длина**	**Описание**
Порт источника	2 байта	Номер порта источника
Порт назначения	2 байта	Номер порта назначения

При установлении соединения хост-отправитель открывает соединение на определенном порту, и ожидает ответа от хоста-получателя. Хост, получивший запрос от хоста-отправителя, узнает порт, на который следует отправить ответ. После обмена служебной информацией на хосте-отправителе и хосте-получателе устанавливается соединение с указанными в заголовках портами, по которым в дальнейшем и будет передаваться запрашиваемая информация.

1.5.2 UDP – User Datagram Protocol

В отличие от TCP, UDP – очень быстрый протокол, поскольку в нем определен самый минимальный механизм, необходимый для передачи данных. Конечно, он имеет некоторые недостатки. Сообщения поступают в любом порядке, и то, которое отправлено первым, может быть получено последним. Доставка сообщений UDP вовсе не гарантируется, сообщение может потеряться, и могут быть получены две копии одного и того же сообщения. Последний случай возникает, если для отправки сообщений в один адрес использовать два разных маршрута [15].

UDP не требует открывать соединение, и данные могут быть отправлены сразу же, как только они подготовлены. UDP не отправляет подтверждающие сообщения, поэтому данные могут быть получены или потеряны.

Полный заголовок UDP выглядит следующим образом:

Таблица 4 – Поля заголовка UDP

Поле	**Длина**	**Описание**
Порт источника	2 байта	Указание порта источника для UDP необязательно. Если это поле используется, получатель может отправить ответ этому порту.
Порт назначения	2 байта	Номер порта назначения
Длина	2 байта	Длина сообщения, включая заголовок и данные.
Контрольная сумма	2 байта	Контрольная сумма заголовка и данных для проверки

UDP – это быстрый протокол, не гарантирующий доставки. Если требуется поддержание порядка сообщений и надежная доставка, нужно использовать TCP. UDP предназначен для широковещательной и групповой передачи данных.

1.6 VPN

Любая виртуальная частная сеть подразумевает использование туннеля. Туннель – это логический интерфейс, роль которого состоит в том, чтобы инкапсулировать пакеты одного протокола (passenger protocol), с помощью второго (carrier protocol) и передать его по третьему (transport protocol) [19]. В роли протокола, который занимается инкапсуляцией могут выступать протоколы GRE, IPIP, PPP, работающие на разных уровнях модели OSI.

В вопросе выбора уровня реализации защищенного канала существует несколько противоречивых аргументов: с одной стороны, за выбор верхних уровней говорит их независимость от вида транспортировки (выбора протокола сетевого и канального уровней), с другой стороны, для каждого приложения необходима отдельная настройка и конфигурация. Плюсом в выборе нижних уровней является их универсальность и наглядность для приложений, минусом – зависимость от выбора конкретного протокола (например, PPP или Ethernet). Компромиссом в выборе уровня является IPsec: он располагается на сетевом уровне, используя самый распространённый протокол этого уровня – IP. Это делает IPsec более гибким, так что он может использоваться для защиты любых протоколов, базирующихся на TCP и UDP. В то же время, он прозрачен для большинства приложений.

1.6.1 IPSec

IPsec был разработан с целью повышения безопасности IP протокола. Следует отметить, что IPsec предлагает набор алгоритмов и механизмов, а не готовое решение, задача настройщика выбрать нужное и применить в зависимости от поставленных целей. Представлены возможности шифрования, аутентификации и функции для сохранности целостности сообщения [21].

Протоколы IPSec обеспечивают управление доступом, целостность вне соединения, аутентификацию источника данных, защиту от воспроизведения, конфиденциальность и частичную защиту от анализа трафика.

IPsec включает в себя:

– безопасное сокрытие данных (ESP);

– аутентификационный заголовок (AH);

– контексты (ассоциации) безопасности (SA);

– управление ключами (IKE);

– режимы работы: туннельный и транспортный.

Опишем принцип работы каждого из параметров, а затем процесс установления соединения в целом.

AH (Authentication Header) – протокол заголовка идентификации, обеспечивающий целостность передаваемых данных методом проверки того, что ни один бит в защищаемой части пакета не был изменён во время передачи. Использование AH может вызвать проблемы, например, при прохождении пакета через NAT устройство. Так как в пакет при проходе через NAT меняется отправитель, то контрольная сумма AH станет неверной. Также стоит отметить, что AH разрабатывался только для обеспечения целостности. Он не обеспечивает шифрование содержимого пакета.

ESP (Encapsulating Security Protocol) – инкапсулирующий протокол безопасности, обеспечивающий и целостность и конфиденциальность. В транспортном режиме, ESP заголовок находится между оригинальным IP транспортных протоколов. В режиме туннеля заголовок ESP размещается между новым IP заголовком и полностью зашифрованным оригинальным IP пакетом.

Так как AH и ESP работают на сетевом уровне, они имеют собственные ID протокола, AH зарезервирован за ID 51, а ESP за ID 50.

Третий протокол, используемый IPSec – это IKE(Internet Key Exchange protocol), предназначенный для обмена ключами между двумя узлами VPN. Несмотря присутствующую возможность генерировать ключи вручную, наиболее безопасным и масштабируемым вариантом будет автоматизация этого процесса с помощью IKE. Для обеспечения безопасности, ключи должны часто меняться, и удобнее доверить этот процесс автоматизации. IKE использует 500 порт UDP[22].

SA (Security Association) – это термин IPSec для обозначения соединения. При настроенном VPN, для каждого используемого протокола создается одна SA пара (т.е. одна для AH и одна для ESP). SA создаются парами, т.к. каждая SA – это однонаправленное соединение, а данные необходимо передавать в двух направлениях. Полученные SA пары хранятся на каждом узле. Если ваш узел имеет SA, значит VPN туннель был установлен успешно.

Т.к. каждый узел способен устанавливать несколько туннелей с другими узлами, каждый SA имеет уникальный номер, позволяющий определить к какому узлу он относится. Это номер называется SPI (Security Parameter Index) или индекс параметра безопасности.

Каждый узел IPSec также имеет вторую БД – SPD или Security Policy Database (БД политики безопасности). Она содержит настроенную вами политику узла.

Транспортный режим работы

В этом варианте механизмы безопасности применяются только для протоколов, начиная с транспортного (TCP) уровня и выше, оставляя данные самого сетевого уровня (заголовок IP) без дополнительной защиты. Места размещения дополнительной информации, вставляемой протоколами в пакет, представлены в соответствии с рисунком.

Рисунок 5 – Транспортный режим работы IPSec

Туннельный режим работы

Этот режим интересен тем, что обеспечивает защиту в том числе данных сетевого уровня путем добавления нового IP-заголовка. После определения ассоциаций безопасности (например, между двумя шлюзами) истинные адреса хостов отправления и назначения (и другие служебные поля) полностью защищаются от модификаций для AH или вообще скрываются для ESP, а в новый заголовок выставляются адреса и другие данные для шлюзов (отправления/получения). На рисунке 6 представлены преимущества и недостатки обоих протоколов. ESP обеспечивает сокрытие данных, но не полную аутентификацию всего пакета. AH полностью аутентифицирует, но не скрывает данные. В этом причина того, что для обеспечения высокого уровня безопасности, применение протоколов совмещается.

Рисунок 6 – Туннельный режим работы IPSec

Основная сложность управления политиками IPsec в срезе маршрутизации является тот факт, что необходимо настраивать политику для каждой сети. Решить эту проблему можно созданием туннеля без функций защиты, и обрабатывать IPsec только туннель, а определять возможность доступа к локальным ресурсам с помощью политик безопасности на сетевом уровне при помощи сетевого экрана. Таким образом, мы достигнем уменьшения количества точек отказа, упростим настройку и устранение

проблем. Одним из протоколов, обеспечивающих необходимый нам туннель, является GRE.

1.6.2 GRE

GRE (generic routing encapsulation – общая инкапсуляция маршрутов) – разработанный CISCO протокол туннелирования сетевых пакетов. Основное назначение – инкапсуляция пакетов сетевого уровня модели OSI в IP-пакеты. Номер протокола в IP – 47. Протокол не поддерживает никаких режимов аутентификации или шифрования, его задача доставка пакетов. Таким образом, используя GRE+IPsec, можно добиться сбалансированного решения по соотношению сложность настройки/безопасность без ущерба для конфеденциальности передаваемых данных[15].

1.7 Firewall

Межсетевой экран, сетевой экран – программный или программно-аппаратный элемент компьютерной сети, осуществляющий контроль и фильтрацию проходящего через него сетевого трафика в соответствии с заданными правилами [4].

Сетевые экраны могут работать с канальным, сетевым и транспортным уровнями модели OSI. Суть работы сетевого экрана сводится к следующему: на некоторый интерфейс приходит запрос, содержащий информацию любого из вышеописанных уровней, то есть порт, ip-адрес и mac-адрес источника и порт, ip-адрес и mac-адрес получателя. Задача firewall обработать приходящий запрос, сравнить с записью правил обработки, и при обнаружении совпадения, совершить указанные в правиле действия. Firewall имеет только два действия – отбросить и пропустить. Но, благодаря обработке информации о сегменте, дает нам безграничные возможности в ограничении нежелательного доступа.

Предположим, что на хосте 192.168.2.2 находится веб-сервис (порт 80). Нам нужно разрешить доступ только хосту с ip-адресом 192.168.2.1, а все

остальные запросы на сервер запретить. В общем случае, правила будут выглядеть таким образом:

1) разрешить запросы на хост 192.168.2.2 по порту 80 для хоста 192.168.2.1;

2) запретить запросы на хост 192.168.2.2 по порту 80 всем хостам.

Сетевые экраны обрабатывают свои таблицы по порядку, начиная с первого. Как только находится совпадение по назначению, происходит проверка остальных условий. Если все условия совпадают, то сетевой экран обрабатывает запрос заданным образом, в данном случае если приходит запрос с хоста 192.168.2.1, обработка завершается на первом правиле и пакет продолжает свой путь до назначенной цели. Если же хост-отправитель отличен от 192.168.2.1, первое правило пропускается и обработка происходит согласно второму правилу, то есть пакет отбрасывается, и на хост-отправитель не приходит никакого ответа.

Сетевой экран является мерой контроля происходящего в сети, в том числе защиты от различного типа атак.

Таким образом, вышеизложенное позволяет сделать вывод о том, что для организации корпоративной сети нам необходим маршрутизатор с поддержкой VLAN, NAT, IPSec, GRE, Firewall.

2 Практическая реализация корпоративной сети предприятия

В качестве программного маршрутизатора для разработки корпоративной сети был выбран основанный на Debian дистрибутив VyOS, поддерживающий:

– VLANs: 802.1q;

– статическую и динамическую маршрутизацию;

– межсетевой экран;

– туннели PPPoE, GRE, IPIP;

– VPN: Site-to-site IPsec for IPv4 and IPv6, L2TP/IPsec server, PPTP server;

– NAT;

– DHCP-server;

– удобный интерфейс командной строки [18];

помимо вышеперечисленного ОС Linux позволяет добавлять собственные скрипты, что упрощает администрирование маршрутизатора.

2.1 Схема сети

В данный момент в организации находится два офиса: в Красноярске, имеющий выделенный ip-адрес 20.20.20.20 и Лесосибирске, с адресом 30.30.30.30. В Красноярске располагаются: сервер, руководство, бухгалтерия и менеджеры. В Лесосибирском филиале присутствуют только менеджеры. В связи с этим было решено установить следующее разделение сети:

Таблица 5 – Схема сети

Группа	VLAN	Сеть
Красноярск		192.168.0.0/21
Руководство	10	192.168.0.0/24
Бухгалтеры	11	192.168.2.0/24
Менеджеры	12	192.168.3.0/24
Серверы	20	192.168.1.0/24
***	***	***
Лесосибирск		192.168.8.0/23
Менеджеры	12	192.168.8.0/24

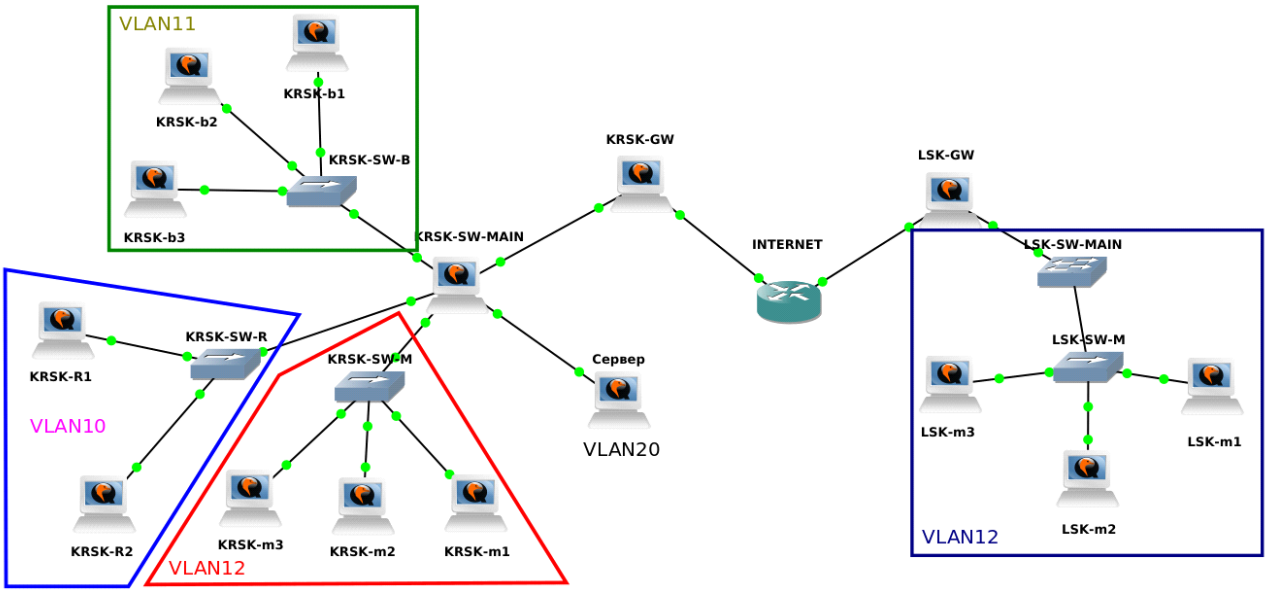

Рисунок 7 – Схема разделения на VLAN

Для каждого отдела был выделен свой VLAN, и, как следствие, своя сеть для дальнейших настроек межсетевого экрана.

Закономерный вопрос: почему используется сеть 192.168.8.0/24, а не 192.168.4.0/24 в городе Лесосибирске. Ответ: используя агрегацию (суммаризацию) маршрутов – процесс объединения сети с большей маской из нескольких меньших, необходимого для упрощения восприятия таблицы маршрутов и уменьшения количества записей в ней, мы можем прописать один маршрут для города Красноярска из города Лесосибирска, который будет выглядеть так (в нашем конкретном случае):

route 192.168.0.0/21 next-hop 10.0.0.2

вместо 8 записей вида:

route 192.168.0.0/24 next-hop 10.0.0.2

route 192.168.1.0/24 next-hop 10.0.0.2

route 192.168.2.0/24 next-hop 10.0.0.2

route 192.168.3.0/24 next-hop 10.0.0.2

route 192.168.4.0/24 next-hop 10.0.0.2

route 192.168.5.0/24 next-hop 10.0.0.2

route 192.168.6.0/24 next-hop 10.0.0.2

route 192.168.7.0/24 next-hop 10.0.0.2.

При этом не исчерпывается лимит сетей, т.к. запись вида 192.168.0.0/21 включает в себя сети со 192.168.0.0/24 до 192.168.7.0/24, и, если появится необходимость добавить еще одну сеть для очередного отдела или по любой другой причине, в запасе остается еще 4 сети с маской /24.

Таким же образом построен и маршрут /23 в сторону Лесосибирска, одна сеть 192.168.9.0/24 оставлена на случай расширения филиала.

2.2 DHCP

Получение настроек обеспечивает протокол DHCP. DHCP (Dynamic Host Configuration Protocol/протокол динамической конфигурации узла) – это сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP, таким образом нет необходимости настраивать каждый компьютер вручную, при подключении к сети они самостоятельно получают необходимые им настройки.

На текущий момент конфигурация выглядит так:

Красноярск

```
interfaces {
   ethernet eth0 {
      address 20.20.20.20/27
      description WAN
      duplex auto
      hw-id aa:aa:aa:ff:ff:f1
      smp_affinity auto
      speed auto
   } [39]
   ethernet eth1 {
      duplex auto
      smp_affinity auto
      speed auto
      vif 10 {
```

```
            address 192.168.0.1/24
            description admin
        }
        vif 11 {
            address 192.168.2.1/24
            description buhg
        }
        vif 12 {
            address 192.168.3.1/24
            description managers
        }
        vif 20 {
            address 192.168.1.1/24
            description server
        }
    } [32]
    loopback lo {
    }
}
    dhcp-server {
        disabled false
        shared-network-name ADMINS {
            authoritative disable
            subnet 192.168.0.0/24 {
                default-router 192.168.0.1
                lease 86400
                start 192.168.0.2 {
                    stop 192.168.0.254
                } [
                static-mapping ADMIN {
```

```
            ip-address 192.168.0.24
            mac-address 00:01:1b:c5:c5:00
        }
    }
}
shared-network-name BUHG {
    authoritative disable
    subnet 192.168.2.1/24 {
        default-router 192.168.2.1
        lease 86400
        start 192.168.2.2 {
            stop 192.168.2.254
        }
    }
}
shared-network-name MANAGERS {
    authoritative disable
    subnet 192.168.3.1/24 {
        default-router 192.168.3.1
        lease 86400
        start 192.168.3.2 {
            stop 192.168.3.254
        }
    }
}
shared-network-name SERVERS {
    authoritative disable
    subnet 192.168.1.0/24 {
        default-router 192.168.1.1
        lease 86400
```

```
            start 192.168.1.2 {
               stop 192.168.1.254
            }
            static-mapping 1.2 {
               ip-address 192.168.1.2
               mac-address 00:01:1b:c8:f7:00
            }
         }
      }
   }
```

Следующие строки конфигурации в DHCP:

```
static-mapping 1.2 {
        ip-address 192.168.1.2
        mac-address 00:01:1b:c8:f7:00
}
```

выполняют функцию привязки ip-адреса к mac-адресу хоста, запрашивающего настройки. То есть, он будет всегда получать IP 192.168.1.2, что, в конкретном случае сервера, необходимо для постоянного доступа сотрудников.

Лесосибирск:

```
interfaces {
   ethernet eth0 {
      address 30.30.30.30/27
      description internet
      duplex auto
      }
      hw-id aa:aa:aa:ff:ff:f2
      smp_affinity auto
      speed auto
   }
   ethernet eth1 {
```

```
        duplex auto
        smp_affinity auto
        speed auto
        vif 12 {
            address 192.168.8.1/24
            description MANAGERS
        }
    }
    loopback lo {
    }
}
service {
    dhcp-server {
        disabled false
        shared-network-name MANAGERS {
            authoritative disable
            subnet 192.168.8.0/24 {
                default-router 192.168.8.1
                lease 86400
                server-identifier 192.168.8.1
                start 192.168.8.2 {
                    stop 192.168.8.254
                }
            }
        }
    }
```

2.3 NAT и Routing

После настроек сети необходимо обеспечить доступ во вне, с помощью маршрутизации и технологии NAT.

Так как у нас имеется только один шлюз доступа в интернет с каждой стороны, пропишем шлюз по умолчанию с обеих сторон:

Красноярск. Провайдер предоставляет нам IP-адрес 20.20.20.20/27 со шлюзом 20.20.20.1:

```
static {
    route 0.0.0.0/0 {
        next-hop 20.20.20.1 {
        }
    }
```

Лесосибирск

По аналогии с Красноярском: IP-адрес 30.30.30.30/27 со шлюзом 30.30.30.1

```
    static {
        route 0.0.0.0/0 {
            next-hop 30.30.30.1 {
            }
        }
```

Настройки NAT на примере Красноярска:

```
source {
    rule 1 {
        description WAN
        outbound-interface eth0
        translation {
            address 20.20.20.20
        }
        } [34]
```

Всем пакетам, выходящим из сети через интерфейс eth0, присваивается source address 20.20.20.20. Это необходимо для маршрутизации пакетов в Internet.

Те же действия на Лесосибирском маршрутизаторе:

```
source {
```

```
    rule 1 {
        description WAN
        outbound-interface eth0
        translation {
            address 30.30.30.30
        }
    }
}
```

Таким образом, весь выходящий трафик маршрутизируется через шлюзы и всем устройствам внутри сети обеспечен доступ в интернет.

2.4 VPN

Для конфигурации GRE-туннеля на двух сторонах пропишем соответствующие настройки. Так как туннель GRE это соединение точка-точка, нам достаточно будет двух IP-адресов из «серой» сети, например 10.0.0.0/30. Сконфигурируем два туннельных интерфейса с адресами 10.0.0.1 и 10.0.0.2:

```
tunnel tun0 {
    address 10.0.0.1/30
    description LSK
    encapsulation gre
    local-ip 20.20.20.20
    mtu 1400
    multicast disable
    remote-ip 30.30.30.30
} [38]
```

В конфигурации указаны внешние адреса двух офисов – local ip – текущего, remote ip – удаленного, для установления туннеля.

С обратной стороны проведем аналогичную настройку:

```
tunnel tun1 {
    address 10.0.0.2/30
```

description KRSK

encapsulation gre

local-ip 30.30.30.30

mtu 1400

multicast disable

remote-ip 20.20.20.20

}

Проверим соединение с помощью команды ping и рассмотрим, что происходит внутри передаваемого блока данных:

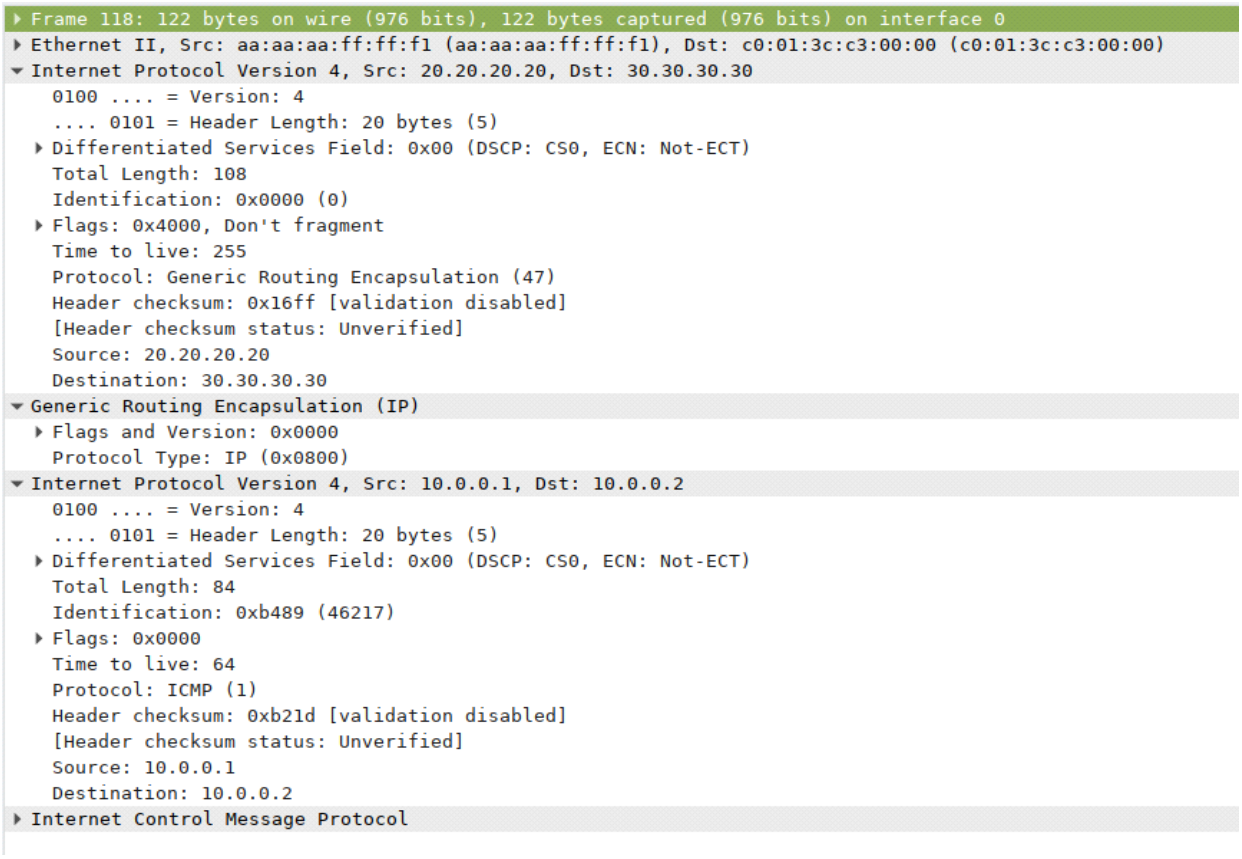

Рисунок 8 – Пинг без IPSec

Наружный заголовок отображает внешние IP-адреса, внутри него мы обнаруживаем протокол GRE, адреса туннелей и icmp запросы (команда ping) в поле данных. Связь между офисами установлена, приступим к организации шифрования с помощью IPSec.

IPSec. Красноярск:

vpn {

```
ipsec {
    esp-group ESP {
        compression disable -  Без сжатия
        lifetime 3600 – время жизни ключа
        mode tunnel – туннельный или транспортный режим
        pfs enable – режим постоянной перегенерации и смены ключей
        proposal 1 {
            encryption aes128 - шифрование
            hash sha1 - хеширование
        }
    } [38]
```

Настройка осуществляется в несколько этапов, сначала мы определили настройки для ESP, затем IKE:

```
    ike-group IKE {
        lifetime 28800
        proposal 1 {
```

dh-group 2 – выбор используемой длины ключа протоколом Диффи-Хеллмана

```
            encryption aes128
            hash sha1
        }
    }
```

В дальнейшей конфигурации указаны интерфейс, обрабатываемый IPsec, локальный и удаленный IP-адреса, метод аутентификации, используемые группы и gre в качестве туннеля.

```
            ipsec-interfaces {
                interface eth0
            }
            site-to-site {
                peer 30.30.30.30 {
```

```
            authentication {
                mode pre-shared-secret
                pre-shared-secret **********
            }
            connection-type initiate
            default-esp-group ESP
            ike-group IKE
            local-address 20.20.20.20
            tunnel 1 {
                allow-nat-networks disable
                allow-public-networks disable
                protocol gre
            }
        }
    }
}
```

Лесосибирск:

```
vpn {
    ipsec {
        esp-group ESP {
            compression disable
            lifetime 3600
            mode tunnel
            pfs enable
            proposal 1 {
                encryption aes128
                hash sha1
            }
        }
```

```
        ike-group IKE {
            lifetime 28800
            proposal 1 {
                dh-group 2
                encryption aes128
                hash sha1
            }
        }
        ipsec-interfaces {
            interface eth0
        }
        site-to-site {
            peer 20.20.20.20 {
                authentication {
                    mode pre-shared-secret
                    pre-shared-secret **********
                }
                connection-type initiate
                default-esp-group ESP
                ike-group IKE
                local-address 30.30.30.30
                tunnel 1 {
                    allow-nat-networks disable
                    allow-public-networks disable
                    protocol gre
                }
            }
        }
    }
}
```

Запустим точно такой же пинг

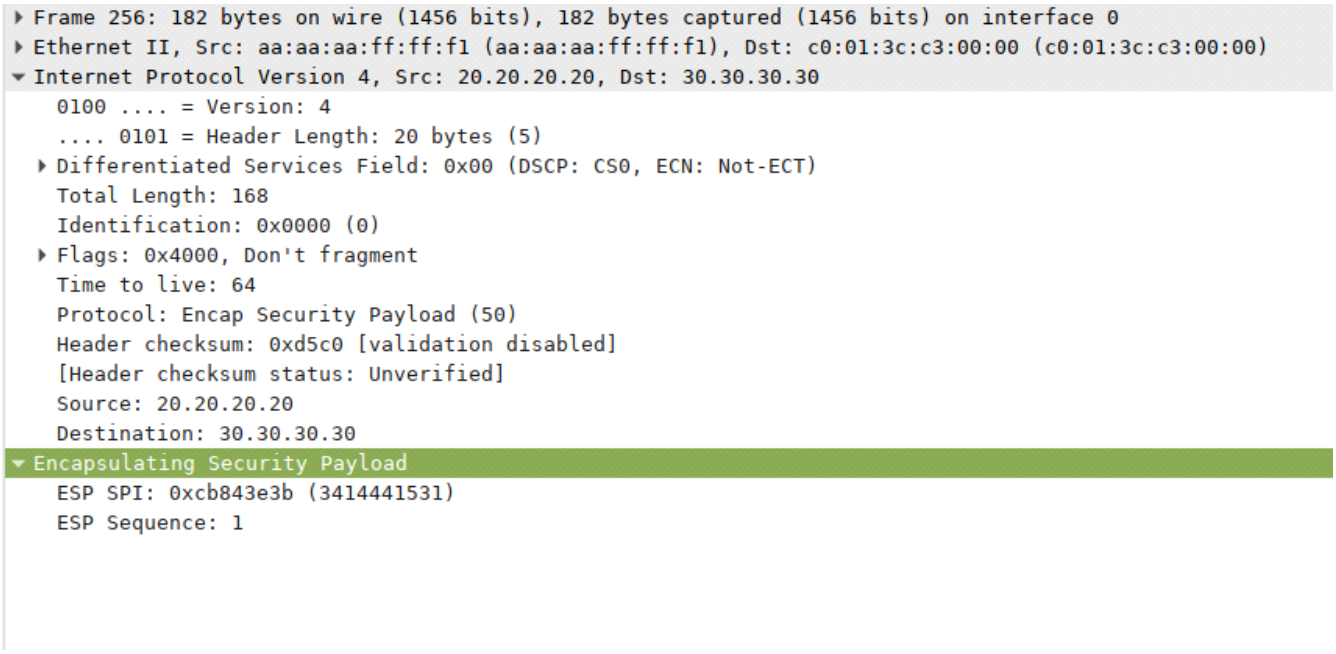

Рисунок 9 – пинг с IPSec

В результате вся передающаяся информация – зашифрована, при прослушке злоумышленник увидит только заголовок IP.

Пропишем маршрут для удаленного доступа к Красноярскому серверу из Лесосибирска:

```
protocols {
    static {
        route 192.168.0.0/21 {
            next-hop 10.0.0.1 {
            }
        }
    } [30]
```

Все запросы, направленные в сеть 192.168.0.0/21 будут перенаправляться в туннель и там достигать своего назначения. Для работоспособности необходимо прописать также маршрут в Красноярске, иначе роутер ничего не будет знать о сети 192.168.8.0/23 и пакеты будут отброшены.

```
protocols {
    static {
```

```
        route 192.168.8.0/23 {
            next-hop 10.0.0.2 {
            }
        }
    }
}
```

Имеется доступ в интернет и защищенный канал передачи данных между офисами, необходимо настроить политики безопасности. В VyOS нет понятия ACL, для любого управления трафиком используется Firewall.

2.5 Firewall

В VyOS сетевой экран использует имена групп, которым можно задать определенные параметры, а затем использовать на любом интерфейсе. В Красноярском филиале создадим три группы:

LAN-LOCAL - будет использоваться для политик безопасности доступа к маршрутизатору изнутри офиса,

LAN-OUT - будет использоваться для политик безопасности доступа из локальной сети наружу,

WAN-LOCAL - будет использоваться для политик безопасности доступа на маршрутизатор извне [31].

Помимо названий политик используются так называемые группы – список IP адресов или целых сетей, на которые данные правила будут действовать. Конечная конфигурация устанавливается на необходимый нам интерфейс, в необходимом нам направлении:

in – входящий трафик на интерфейс

out – выходящий трафик с интерфейса

local – трафик, предназначенный непосредственно узлу.

Для начала запретим доступ на наш роутер извне:

```
name WAN-LOCAL {
```

```
default-action drop – действие по умолчанию
rule 10 {
   action accept – действие правила
   source {
      group {
         address 30.30.30.30
      }
   }
}
```

Разрешает доступ на узел только ip-адресу 30.30.30.30. все остальные входящие пакеты будут отбрасываться. Устанавливаем правило на local-доступ интерфейса eth0:

```
ethernet eth0 {
   address 20.20.20.20/27
   description WAN
   duplex auto
   firewall {
      local {
         name WAN-LOCAL
      }
   }
```

С использованием группы LAN-OUT запретим общение локальных сетей, оставим только доступ к серверу с адресом 192.168.1.2:

```
name LAN-OUT {
   default-action accept
   rule 9 {
      action accept
      destination {
         address 192.168.1.2
      }
```

```
    }
    rule 10 {
        action drop
        destination {
            address 192.168.0.0/20
            }
        }
    }
}
```

Глядя на данную конфигурацию может возникнуть закономерный вопрос: зачем в группе с default-action accept создавать правило с тем же параметром? Параметр default-action применяется в самом конце списка, все остальные параметры применяются по порядку от меньшего идентификатора rule к большему, таким образом, если мы оставим только запись action drop destination address 192.168.0.0/20, в нее автоматически попадает наш сервер 192.168.1.2. Во избежание блокировки доступа на сервер и помещается правило с разрешением перед правилом с запрещением.

Группа LAN-LOCAL будет совсем простой, в ней мы просто запретим весь доступ на маршрутизатор:

```
name LAN-LOCAL {
    default-action drop
}
```

Применим данные параметры ко всем необходимым нам интерфейсам:

```
vif 11 {
        address 192.168.2.1/24
        description buhg
        firewall {
            in {
                name LAN-OUT
            }
```

```
        local {
            name LAN-LOCAL
        }
        out {
        }
    }
}
vif 12 {
    address 192.168.3.1/24
    description managers
    firewall {
        in {
            name LAN-OUT
        }
        local {
            name LAN-LOCAL
        }
    }
}
```

Таким образом, в процессе организации корпоративной сети был обеспечен шифрованный канал связи между двумя филиалами, организовано автоматическое получение настроек всем хостам в сети, обеспечена защита от доступа пользователей на маршрутизатор и в другие сегменты сети изнутри согласно принятым политиками доступа. Имеют доступ только сервер и руководство (системный администратор). Аналогичным образом маршрутизатор настроен в городе Лесосибирске.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы получены следующие результаты и сделаны соответствующие выводы:

а) на основе анализа специальной и научной литературы выявлены требования к корпоративной сети:

- сеть должны быть сегментирована;
- сегменты должны быть изолированы согласно политикам доступа;
- трафик между офисами должен шифроваться;

б) сформирована структура корпоративной сети, включающая:

- разделение сети на сегменты;
- организацию доступа к серверу и ограничение доступа в другие сегменты сети;
- организация защищенного канала между двумя офисами;

в) выделены основные методы реализации корпоративной сети:

- разделение сети с помощью технологии VLAN;
- реализация политик доступа методом разрешения/запрещения доступа с использованием сетевого экрана(firewall);
- организация туннеля GRE между удаленными офисами и организация шифрования методами группы протоколов IPSec.

г) с использованием программного маршрутизатора VyOS реализована корпоративная сеть, удовлетворяющая выделенным требованиям.

По материалам выпускной квалификационной работы опубликована научная статья на тему «Особенности использования VPN вида сеть-сеть» в электронном сборнике статей по материалам IV международной студенческой научно-практической конференции «Технические и математические науки. Студенческий научный форум».

Таким образом, цель исследования достигнута, задачи выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Брайан, У. Внутреннее устройство Linux / У. Брайан. – Санкт-Петербург : Питер, 2016 г. – 384 с.

2. Бреснахэн, К. Linux на практике / Б. Кристин, Б. Ричард. – Санкт-Петербург : Питер, 2017 г. – 384 с.

3. Колисниченко, Д. Linux-сервер своими руками / Д. Колисниченко. – Сакт-Петербург : Наука и техника, 2006 г. – 752 с.

4. Куроуз, Д. Компьютерные сети. Настольная книга системного администратора / Д. Куроуз, Т. Росс. – Москва : Эксмо, 2016 г. – 912 с.

5. Куроуз, Д. Компьютерные сети. Нисходящий подход / Д. Куроуз, Т. Росс. – Москва : Эксмо, 2016 г. – 912 с.

6. Лора, А. TCP/IP. Учебный курс / А. Лора, Э. Титтел – Санкт-Петербург : БХВ–Петербург, 2003 г. – 960 с.

7. Ногл, М. TCP/IP. Иллюстрированный учебник / М. Ногл. – Москва : ДМК-Пресс, 2007 г. – 491 с.

8. Одом, У. Официальное руководство Cisco по подготовке к сертификационным экзаменам CCNA ICND2 200-101. Маршрутизация и коммутация / У. Одом. – Москва : Вильямс, 2016 г. – 736 с.

9. Олифер, В. Компьютерные сети. Принципы, технологии, протоколы / В. Олифер, Н. Олифер. – Санкт-Петербург : Питер, 2016 г. – 992 с.

10. Паркер, Т. TCP/IP. Для профессионалов, 3-е изд. / Т. Паркер, К. Сиян. – Санкт-Петербург : Питер, 2004 г. – 859 с.

11. Робачевский, А. Интернет изнутри. Экосистема глобальной сети / А. Робачевский. – Москва : Альпина Паблишер, 2017 г. – 224 с.

12. Сергеев, А. Основы локальных компьютерных сетей / А. Сергеев. – Санкт-Петербург : Лань, 2016 г. – 184 с.

13. Снейдер, Й. Эффективное программирование TCP/IP / Й. Снейдер. – Москва : ДМК-Пресс, 2007 г. – 321 с.

14. Таненбаум, Э. Компьютерные сети / Э. Таненбаум, Д. Уэзеролл. – Санкт-Петербург : Питер, 2016 г. – 960 с.

15. Фейт, С. TCP/IP. Архитектура, протоколы, реализация / С. Фейт. – Москва : Лори, 2009 г. – 424 с.

16. Фленов, М. Linux глазами хакера / М. Фленов. – Санкт-Петербург : БХВ-Петербург, 2016 г. – 432 с.

17. Хант, К. TCP/IP. Сетевое администрирование / К. Хант. – Изд. 3-е. Пер. с англ. – Санкт-Петербург : Сим-вол-Плюс, 2007 г. – 816 с.

18. «VyOS Wiki» [Электронный ресурс]. – Режим доступа: https://wiki.vyos.net/wiki/

19. «Виды VPN-соединений (PPTP, L2TP, IPSec, SSL)» [Электронный ресурс]. – Режим доступа: https://kb.zyxel.ru/hc/ru/articles

20. «Калейдоскоп VPN-технологий» [Электронный ресурс]. – Режим доступа: https://cyberleninka.ru/article/v/kaleydoskop-vpn-tehnologiy

21. «Методическое пособие. IPSec» [Электронный ресурс]. – Режим доступа: http://dfe.karelia.ru/koi/posob/security/index.html

22. «Методическое пособие. IPSec» [Электронный ресурс]. – Режим доступа: http://dfe.karelia.ru/koi/posob/security/index.html

23. «Networking: tunneling» [Электронный ресурс]. – Режим доступа: https://wiki.linuxfoundation.org/networking/tunneling

24. «OpenVPN. Documentation» [Электронный ресурс]. – Режим доступа: https://openvpn.net/index.php/open-source/documentation.html

25. «Vyatta System. Basic routing» [Электронный ресурс]. – Режим доступа: http://docs.huihoo.com/vyatta/6.5/Vyatta-BasicRouting_6.5R1_v01.pdf

26. «Vyatta System. Basic System» [Электронный ресурс]. – Режим доступа: http://docs.huihoo.com/vyatta/6.5/Vyatta-PolicyBasedRouting_6.5R1_v01.pdf

27. «Vyatta System. BGP» [Электронный ресурс]. – Режим доступа: http://docs.huihoo.com/vyatta/6.5/Vyatta-BGP_6.5R1_v01.pdf

28. «Vyatta System. Bridging» [Электронный ресурс]. – Режим доступа: http://docs.huihoo.com/vyatta/6.5/Vyatta-Bridging_6.5R1_v01.pdf

29. «Vyatta System. Connection management» [Электронный ресурс]. – Режим доступа: http://docs.huihoo.com/vyatta/6.5/Vyatta-ConnectionManagement_6.5R1_v01.pdf

30. «Vyatta System. Encapsulations» [Электронный ресурс]. – Режим доступа: http://docs.huihoo.com/vyatta/6.5/Vyatta-Encapsulations_6.5R1_v01.pdf

31. «Vyatta System. Firewall» [Электронный ресурс]. – Режим доступа: http://docs.huihoo.com/vyatta/6.5/Vyatta-Firewall_6.5R1_v01.pdf

32. «Vyatta System. LAN Interfaces» [Электронный ресурс]. – Режим доступа. – http://docs.huihoo.com/vyatta/6.5/Vyatta-LANInterfaces_6.5R1_v01.pdf

33. «Vyatta System. Security» [Электронный ресурс]. – Режим доступа: http://docs.huihoo.com/vyatta/6.5/Vyatta-Security_6.5R1_v01.pdf

34. «Vyatta System. NAT» [Электронный ресурс]. – Режим доступа: http://docs.huihoo.com/vyatta/6.5/Vyatta-NAT_6.5R1_v01.pdf

35. «Vyatta System. Policy Based Routing» [Электронный ресурс]. – Режим доступа:http://docs.huihoo.com/vyatta/6.5/Vyatta-PolicyBasedRouting_6.5R1_v01.pdf

36. «Vyatta System. QoS» [Электронный ресурс]. – Режим доступа: http://docs.huihoo.com/vyatta/6.5/Vyatta-QoS_6.5R1_v01.pdf

37. «Vyatta System. Routing polices» [Электронный ресурс]. – Режим доступа: http://docs.huihoo.com/vyatta/6.5/Vyatta-RoutingPolicies_6.5R1_v01.pdf

38. «Vyatta System. VPN» [Электронный ресурс]. – Режим доступа: http://docs.huihoo.com/vyatta/6.5/Vyatta-VPN_6.5R1_v01.pdf

39. «Vyatta System. WAN Interfaces» [Электронный ресурс]. – Режим доступа: http://docs.huihoo.com/vyatta/6.5/Vyatta-WANInterfaces_6.5R1_v01.pdf

40. «Vyatta System. Policy Tunnels» [Электронный ресурс]. – Режим доступа: http://docs.huihoo.com/vyatta/6.5/Vyatta-Tunnels_6.5R1_v01.pdf

ПРИЛОЖЕНИЕ А

Конфигурация KRSK-GW

```
firewall {
    all-ping enable
    broadcast-ping disable
    config-trap disable
    group {
        address-group myhosts {
            address 20.20.20.20
        }
        network-group KRSK {
            network 192.168.0.0/20
        }
    }
    ipv6-receive-redirects disable
    ipv6-src-route disable
    ip-src-route disable
    log-martians enable
    name LAN-LOCAL {
        default-action drop
    }
    name LAN-OUT {
        default-action accept
        rule 9 {
            action accept
            destination {
                address 192.168.1.2
            }
        }
```

```
    rule 10 {
        action drop
        destination {
            group {
                network-group KRSK
            }
        }
        log enable
    }
}
name WAN-LOCAL {
    default-action drop
    enable-default-log
    rule 10 {
        action accept
        source {
            group {
                address-group myhosts
            }
        }
    }
}
receive-redirects disable
send-redirects enable
source-validation disable
state-policy {
    established {
        action accept
    }
}
```

```
    syn-cookies disable
}
interfaces {
    ethernet eth0 {
        address 30.30.30.30/27
        description internet
        duplex auto
        firewall {
            local {
                name WAN-LOCAL
            }
        }
        hw-id aa:aa:aa:ff:ff:f2
        smp_affinity auto
        speed auto
    }
    ethernet eth1 {
        duplex auto
        smp_affinity auto
        speed auto
        vif 12 {
            address 192.168.8.1/24
            description MANAGERS
            firewall {
                in {
                    name LAN-OUT
                }
                local {
                    name LAN-LOCAL
                }
```

```
            }
        }
    }
    loopback lo {
    }
    tunnel tun1 {
        address 10.0.0.2/30
        description KRSK
        encapsulation gre
        local-ip 30.30.30.30
        mtu 1400
        multicast disable
        remote-ip 20.20.20.20
    }
}
nat {
    source {
        rule 1 {
            description WAN
            outbound-interface eth0
            translation {
                address 30.30.30.30
            }
        }
    }
}
protocols {
    static {
        route 0.0.0.0/0 {
            next-hop 30.30.30.30 {
```

```
            }
        }
        route 192.168.0.0/21 {
            next-hop 10.0.0.1 {
            }
        }
    }
}
service {
    dhcp-server {
        disabled false
        shared-network-name MANAGERS {
            authoritative disable
            subnet 192.168.8.0/24 {
                default-router 192.168.8.1
                lease 86400
                server-identifier 192.168.8.1
                start 192.168.8.2 {
                    stop 192.168.8.254
                }
            }
        }
    }
    ssh {
        port 22
    }
}
system {
    config-management {
        commit-revisions 20
```

```
}
console {
    device ttyS0 {
        speed 9600
    }
}
host-name LSK-GW
login {
    user vyos {
        authentication {
            encrypted-password **********
            public-keys vyos@LSK-GW {
                key **********
                type ssh-rsa
            }
        }
        level admin
    }
}
ntp {
    server 0.pool.ntp.org {
    }
    server 1.pool.ntp.org {
    }
    server 2.pool.ntp.org {
    }
}
package {
    auto-sync 1
    repository community {
```

```
            components main
            distribution hydrogen
            password **********
            url http://packages.vyos.net/vyos
            username ""
        }
    }
    syslog {
        global {
            facility all {
                level notice
            }
            facility protocols {
                level debug
            }
        }
    }
    time-zone UTC
}
vpn {
    ipsec {
        esp-group ESP {
            compression disable
            lifetime 3600
            mode tunnel
            pfs enable
            proposal 1 {
                encryption aes128
                hash sha1
            }
```

```
    }
    ike-group IKE {
        lifetime 28800
        proposal 1 {
            dh-group 2
            encryption aes128
            hash sha1
        }
    }
    ipsec-interfaces {
        interface eth0
    }
    site-to-site {
        peer 20.20.20.20 {
            authentication {
                mode pre-shared-secret
                pre-shared-secret **********
            }
            connection-type initiate
            default-esp-group ESP
            ike-group IKE
            local-address 30.30.30.30
            tunnel 1 {
                allow-nat-networks disable
                allow-public-networks disable
                protocol gre
            }
        }
    }
}
```

ПРИЛОЖЕНИЕ Б

Конфигурация LSK-GW

```
firewall {
    all-ping enable
    broadcast-ping disable
    config-trap disable
    group {
        address-group myhosts {
            address 30.30.30.30
        }
        network-group LAN {
            network 192.168.0.0/20
            network 30.30.30.30/32
        }
    }
    ipv6-receive-redirects disable
    ipv6-src-route disable
    ip-src-route disable
    log-martians enable
    name LAN-LOCAL {
        default-action drop
    }
    name LAN-OUT {
        default-action accept
        rule 9 {
            action accept
            destination {
                address 192.168.1.2
            }
```

```
    }
    rule 10 {
        action drop
        destination {
            address 192.168.0.0/20
        }
        log enable
    }
}
name WAN-LOCAL {
    default-action drop
    rule 10 {
        action accept
        source {
            address 30.30.30.30
        }
    }
}
receive-redirects disable
send-redirects enable
source-validation disable
state-policy {
    established {
        action accept
    }
    related {
        action accept
    }
}
syn-cookies enable
```

```
}
interfaces {
    ethernet eth0 {
        address 20.20.20.20/27
        description WAN
        duplex auto
        firewall {
            local {
                name WAN-LOCAL
            }
        }
        hw-id aa:aa:aa:ff:ff:f1
        smp_affinity auto
        speed auto
    }
    ethernet eth1 {
        duplex auto
        smp_affinity auto
        speed auto
        vif 10 {
            address 192.168.0.1/24
            description admin
        }
        vif 11 {
            address 192.168.2.1/24
            description buhg
            firewall {
                in {
                    name LAN-OUT
                }
```

```
            local {
                name LAN-LOCAL
            }
            out {
            }
        }
    }
    vif 12 {
        address 192.168.3.1/24
        description managers
        firewall {
            in {
                name LAN-OUT
            }
            local {
                name LAN-LOCAL
            }
        }
    }
    vif 20 {
        address 192.168.1.1/24
        description server
    }
}
loopback lo {
}
tunnel tun0 {
    address 10.0.0.1/30
    description LSK
    encapsulation gre
```

```
        local-ip 20.20.20.20
        mtu 1400
        multicast disable
        remote-ip 30.30.30.30
    }
}
nat {
    source {
        rule 1 {
            description INWAN
            outbound-interface eth0
            translation {
                address 20.20.20.20
            }
        }
    }
}
protocols {
    static {
        route 0.0.0.0/0 {
            next-hop 20.20.20.1 {
            }
        }
        route 192.168.8.0/23 {
            next-hop 10.0.0.2 {
            }
        }
    }
}
service {
```

```
dhcp-server {
    disabled false
    shared-network-name ADMINS {
        authoritative disable
        subnet 192.168.0.0/24 {
            default-router 192.168.0.1
            lease 86400
            start 192.168.0.2 {
                stop 192.168.0.254
            }
            static-mapping ADMIN {
                ip-address 192.168.0.24
                mac-address 00:01:1b:c5:c5:00
            }
        }
    }
    shared-network-name BUHG {
        authoritative disable
        subnet 192.168.2.1/24 {
            default-router 192.168.2.1
            lease 86400
            start 192.168.2.2 {
                stop 192.168.2.254
            }
        }
    }
    shared-network-name MANAGERS {
        authoritative disable
        subnet 192.168.3.1/24 {
            default-router 192.168.3.1
```

```
            lease 86400
            start 192.168.3.2 {
                stop 192.168.3.254
            }
        }
    }
    shared-network-name SERVERS {
        authoritative disable
        subnet 192.168.1.0/24 {
            default-router 192.168.1.1
            lease 86400
            start 192.168.1.2 {
                stop 192.168.1.254
            }
            static-mapping 1.2 {
                ip-address 192.168.1.2
                mac-address 00:01:1b:c8:f7:00
            }
        }
    }
  }
  ssh {
    port 22
  }
}
system {
  config-management {
    commit-revisions 20
  }
  console {
```

```
    device ttyS0 {
        speed 9600
    }
}
host-name KRSK-GW
ip {
    arp {
        table-size 8192
    }
}
login {
    user vyos {
        authentication {
            encrypted-password **********
            public-keys ssh {
                key **********
                type ssh-rsa
            }
        }
        level admin
    }
}
ntp {
    server 0.pool.ntp.org {
    }
    server 1.pool.ntp.org {
    }
    server 2.pool.ntp.org {
    }
}
```

```
    package {
        auto-sync 1
        repository community {
            components main
            distribution hydrogen
            password **********
            url http://packages.vyos.net/vyos
            username ""
        }
    }
    syslog {
        global {
            facility all {
                level notice
            }
            facility protocols {
                level debug
            }
        }
    }
    time-zone UTC
}
vpn {
    ipsec {
        esp-group ESP {
            compression disable
            lifetime 3600
            mode tunnel
            pfs enable
            proposal 1 {
```

```
        encryption aes128
        hash sha1
    }
}
ike-group IKE {
    lifetime 28800
    proposal 1 {
        dh-group 2
        encryption aes128
        hash sha1
    }
}
ipsec-interfaces {
    interface eth0
}
site-to-site {
    peer 30.30.30.30 {
        authentication {
            mode pre-shared-secret
            pre-shared-secret **********
        }
        connection-type initiate
        default-esp-group ESP
        ike-group IKE
        local-address 20.20.20.20
        tunnel 1 {
            allow-nat-networks disable
            allow-public-networks disable
            protocol gre
        }
```

```
            }
        }
    }
}
```