Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Е. Ю. Агеев

СЕТЕВЫЕ ТЕХНОЛОГИИ ВЫСОКОСКОРОСТНОЙ ПЕРЕДАЧИ ДАННЫХ

Методические указания для выполнения практических работ для студентов технических направлений подготовки и специальностей

Томск
2024

УДК 681.3.069
ББК 32.971.35-02
А 23

Рецензент:
Крюков Я.В., доцент кафедры телекоммуникаций и основ радиотехники ТУСУРа, кандидат технических наук

А 23 Сетевые технологии высокоскоростной передачи данных: Методические указания для выполнения практических работ для студентов технических направлений подготовки и специальностей / Е. Ю. Агеев – Томск: Томск. гос. ун-т систем управления и радиоэлектроники, 2024. – 62 с.

Настоящие учебно-методическое пособие содержит указания по выполнению практических работ. Данный практикум имеет целью закрепить и расширить теоретические знания студентов в области настройки сетевого оборудования и взаимодействия устройств в сети, показать зависимость такого параметра, как скорость передачи данных от множества факторов.

Одобрено на заседании кафедры ТОР, протокол № 4 от 30 ноября 2023 г.

УДК 681.3.069
ББК 32.971.35-02

Оглавление

ВВЕДЕНИЕ....4
Практическая работа № 1....5
Практическая работа № 2....10
Практическая работа № 3....24
Практическая работа № 4....42
Практическая работа № 5....54
ЗАКЛЮЧЕНИЕ....60
СПИСОК ЛИТЕРАТУРЫ....61

ВВЕДЕНИЕ

Данный практикум имеет целью закрепить и расширить теоретические знания студентов в области компьютерных сетевых технологий, предоставляя возможность настройки и анализа сетевого оборудования, интерфейсов и сервисов в эмулируемых средах. Несмотря на то, что работа выполняется не на реальном оборудовании, интерфейс взаимодействия близок к реальному и позволяет проверить и применить полученные в ходе изучения курса знания.

Практикум предназначен для студентов технических вузов и содержит описание следующих работ:

1) Управление качеством обслуживания в компьютерной сети;
2) Контроль перегрузки в протоколе tcp;
3) Настройка контроллера управления беспроводной сетью;
4) Настройка ipsec соединения точка-точка;
5) Настройка списков контроля доступа в заданной топологии сети.

Практические работы данного перечня выполняются в программах-симуляторах компьютерной сети Cisco Packet Tracer и Mininet. Также в работах используется вспомогательное ПО виртуализации Oracle VirtualBox, программа iperf для генерации трафика и программа Wireshark для захвата и анализа пакетов.

Практическая работа № 1

«Управление качеством обслуживания в компьютерной сети»

1. Теоретический материал

Quality of Service (QoS) — технология предоставления различным классам трафика различных приоритетов в обслуживании. Любая приоритезация имеет смысл только в том случае, когда возникает очередь на обслуживание. Именно там, в очереди, можно «проскользнуть» первым, используя своё право. Очередь же образуется там, где узко (обычно такие места называются «бутылочным горлышком», bottle-neck). Типичное «горлышко» — выход в Интернет офиса, где компьютеры, подключенные к сети как минимум на скорости 100 Мбит/сек, все используют канал к провайдеру, который редко превышает 100 МБит/сек, а часто составляет мизерные 1-2-10МБит/сек. на всех.

Разумеется, приоритезация не решит всех проблем если «горлышко» слишком узкое. Рано или поздно начнет переполняется физический буфер интерфейса, куда помещаются все пакеты, собирающиеся выйти через этот интерфейс. И тогда новопришедшие пакеты будут уничтожены, даже если они сверхнужные. Поэтому, если очередь на интерфейсе в среднем превышает 20% от максимального своего размера (на маршрутизаторах Cisco максимальный размер очереди составляет, как правило, 128-256 пакетов), нужно подумать над дизайном сети, проложить дополнительные маршруты или расширить полосу пропускания канала до провайдера.

Маркировка пакетов.

В служебных заголовках различных сетевых протоколов (Ethernet, IP, ATM, MPLS и др.) присутствуют специальные поля, выделенные для маркирования трафика. Это необходимо для последующей обработки в соответствии с установленными приоритетами в очередях.

В технологии Ethernet это поле Class of Service (CoS) — 3 бита. Позволяет разделить трафик на 8 типов по приоритетам

В протоколе IP есть 2 способа: старый и новый. В старом было поле «тип сервиса», ToS размер которого составлял 8 бит, однако использовались 3 бита подполя IP Precedence. Они копировались в поле CoS Ethernet заголовка.

Позднее был определен новый стандарт RFC2474. Поле ToS было переименовано в DiffServ, и дополнительно выделено 6 бит для поля Differencial Service Code Point (DSCP), в котором можно передавать требуемые для данного типа трафика параметры.

Маркировать данные лучше всего ближе к источнику этих данных. По этой причине большинство IP-телефонов самостоятельно добавляют в IP-заголовок голосовых пакетов поле DSCP = EF или CS5. Многие приложения также маркируют трафик самостоятельно в надежде, что их пакеты будут обработаны приоритетно. Например, этим «грешат» torrent-сети.

Очереди.

Даже если мы не используем никаких технологий приоритезации, это не значит, что не возникает очередей. В узком месте очередь возникнет в любом случае и будет предоставлять стандартный механизм FIFO (First In First Out). Такая очередь, очевидно, позволит не уничтожать пакеты сразу, сохраняя их до отправки в буфере, но никаких преференций, скажем, голосовому трафику не предоставит.

Если хочется предоставить некоторому выделенному классу абсолютный приоритет (т.е. пакеты из этого класса всегда будут обрабатываться первыми), то такая технология называется Priority queuing. Все пакеты, находящиеся в физическом исходящем буфере интерфейса, будут разделены на 2 логических очереди и пакеты из привилегированной очереди будут отсылаться, пока она не опустеет. Только после этого начнут передаваться пакеты из второй очереди. Эта технология простая, довольно грубая, её можно считать устаревшей, т.к. обработка неприоритетного трафика будет постоянно останавливаться. На

маршрутизаторах Cisco можно создать 4 очереди с разными приоритетами. В них соблюдается строгая иерархия: пакеты из менее привилегированных очередей не будут обслуживаться до тех пор, пока не опустеют все очереди с более высоким приоритетом.

Справедливая очередь (Fair Queuing). Технология, которая позволяет каждому классу трафика предоставить одинаковые права. На практике редко используется, т.к. мало даёт с точки зрения улучшения качества сервиса.

Взвешенная справедливая очередь (Weighted Fair Queuing, WFQ). Технология, которая предоставляет разным классам трафика разные права (можно сказать, что «вес» у разных очередей разный), но одновременно обслуживает все очереди. «На пальцах» это выглядит так: все пакеты делятся на логические очереди, используя в качестве критерия поле IP Precedence. Это же поле задаёт и приоритет. Далее маршрутизатор вычисляет, пакет из какой очереди требуется «быстрее» передать и передаёт именно его. Наглядно описанный процесс представлен на рисунке 1.1.

Рисунок 1.1 – Очереди с разным приоритетом

Расчет выполняется по формуле:

$$dT = (t(i) - t(0)) / (1 + IPP) \quad (1.1)$$

где IPP – значение поля IP Precedence;

$t(i)$ – время, требуемое на реальную передачу пакета интерфейсом. Можно вычислить, как: $L / Speed$, где L – длина пакета, а $Speed$ – скорость передачи интерфейса.

Такая очередь по умолчанию включена на всех интерфейсах маршрутизаторов Cisco, кроме интерфейсов точка-точка (инкапсуляция HDLC или PPP).

WFQ имеет ряд минусов: она использует уже маркированные ранее пакеты, и не позволяет самостоятельно определять классы трафика и выделяемую полосу. Мало того, сегодня, как правило, уже никто не маркирует полем IP Precedence, поэтому пакеты идут немаркированные, т.е. все попадают в одну очередь.

Развитием WFQ стала взвешенная справедливая очередь, основанная на классах (Class-Based Weighted Fair Queuing, CBWFQ). В этой очереди администратор сам задаёт классы трафика, следуя различным критериям, например, используя списки контроля доступа, ACL, как шаблон или анализируя заголовки протоколов. Далее, для этих классов определяется «вес» и пакеты их очередей обслуживаются, соразмерно весу (больше вес — больше пакетов из этой очереди уйдёт в единицу времени). Однако, такая очередь не обеспечивает первоочередной обработки наиболее важных пакетов (как правило голосовых или пакетов других интерактивных приложений). Поэтому появился гибрид Priority и Class-Based Weighted Fair Queuing — PQ-CBWFQ, также известный как, Low Latency Queuing (LLQ). В этой технологии можно задать до 4х приоритетных очередей, остальные классы обслуживать по механизму CBWFQ.

LLQ — наиболее удобный, гибкий и часто используемый механизм. Он требует настройки классов, настройки политики и применения политики на интерфейсе.

2. Ход выполнения работы

Работа выполняется в программе Cisco Packet Tracer версии 8.1 и выше. Настройте в программе топологию согласно рисунку 2.1.

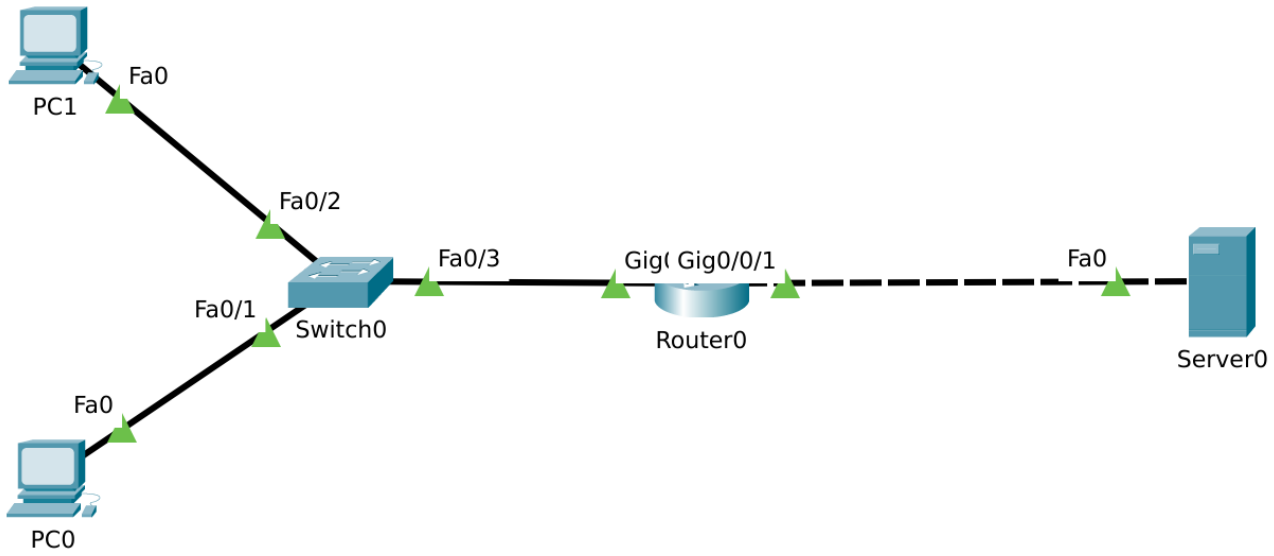

Рисунок 2.1 – Топология выполняемой работы

Задайте адреса на устройствах, согласно индивидуальному варианту, представленному в таблице 2.1. Здесь задана исходная сеть и требования по ее разбиению на две подсети заданного размера. Интерфейс GigabitEthernet0/0/0 маршрутизатора подключен к одной подсети данного разбиения, а интерфейс GigabitEthernet0/0/1 к другой.

Таблица 2.1 – Индивидуальное задание

Вариант	Исходная сеть (блок адресов)	Количество компьютеров	
		А	Б
1	118.7.50.0 /25	7	9
2	39.221.98.0 /25	8	5
3	88.27.252.0 /23	30	9
4	81.104.216.0 /21	48	120
5	7.50.128.0 /19	267	176
6	89.151.32.0 /19	311	246
7	126.61.74.0 /23	8	61
8	36.121.96.0 /19	311	696
9	28.54.64.0 /19	957	153
10	67.253.1.0 /20	365	116
11	77.75.0.0 /18	338	830
12	5.63.168.0 /21	119	61
13	85.123.72.0 /21	189	51

Окончание таблицы 2.1

14	72.241.3.0 /25	12	7

15	87.228.68.0 /22	26	45
16	46.41.64.0 /18	384	1535
17	57.214.86.0 /23	63	9
18	74.30.128.0 /19	346	179
19	88.61.128.0 /20	366	77
20	10.58.180.0 /22	30	92
21	112.56.76.0 /22	23	114
22	2.78.160.0 /19	214	443
23	30.182.64.0 /18	624	1700
24	75.39.128.0 /19	625	219
25	98.115.89.37 /21	48	119
26	35.163.168.0 /21	119	60

Далее для примера считается что слева расположена сеть 192.168.2.0/24, а справа 192.168.1.0/24, Настройте интерфейсы маршрутизатора командами:

Router(config)#interface g0/0/0

Router(config-if)#ip address 192.168.2.1 255.255.255.0

Router(config-if)#no shutdown

Router(config)#interface g0/0/1

Router(config-if)#ip address 192.168.1.1 255.255.255.0

Router(config-if)#no shutdown

Задайте компьютерам соответствующие адреса и настройте адрес шлюза по умолчанию. Настроим на маршрутизаторе маркировку поступающего на интерфейс GigabitEthernet0/0/0 трафика. Создадим правило, определяющее веб-трафик. Сделать это можно создав расширенный список контроля доступа с указанием номера порта 80:

Router(config)#ip access-list extended WEB_BROWSING

Router(config-ext-nacl)#permit tcp any any eq 80

Затем создадим класс для трафика, назовем его WEB_BROWSING_CLASS и сопоставим его списку доступа.

Router(config-ext-nacl)class-map WEB_BROWSING_CLASS

Router(config-cmap)#match access-group name WEB_BROWSING

Теперь создадим политику, назовем ее CLASSIFY_WEB и потребуем в ней использовать только что созданную карту классов.

Router(config-cmap)#policy-map CLASSIFY_WEB

Router(config-pmap)#class WEB_BROWSING_CLASS

Теперь созданную политику нужно применить на интерфейс:

Router(config-pmap)#interface GigabitEthernet 0/0/0

Router(config-if)#service-policy input CLASSIFY_WEB

Выйдите из режима настройки интерфейса нажатием клавиш Ctrl+Z или введением команды end. Проверьте настроенную политику командой:

Router#show policy-map interface g0/0/0

Вы должны увидеть картину, похожую на рисунке 2.2.

```
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show policy-map interface g0/0/0
 GigabitEthernet0/0/0

  Service-policy input: CLASSIFY_WEB

    Class-map: class-default (match-any)
      75 packets, 3497 bytes
      5 minute offered rate 98 bps, drop rate 0 bps
      Match: any
```

Рисунок 2.2 – Проверка настроенной политики классификации трафика

Сколько веб-пакетов попало под настроенную политику в вашем случае?

Откройте приложение «Генератор трафика» на вкладке Desktop компьютера PC0 и настройте параметры генератора, как показано на рисунке 2.3, с учетом коррекции адресов компьютера и веб-сервера согласно вашего варианта.

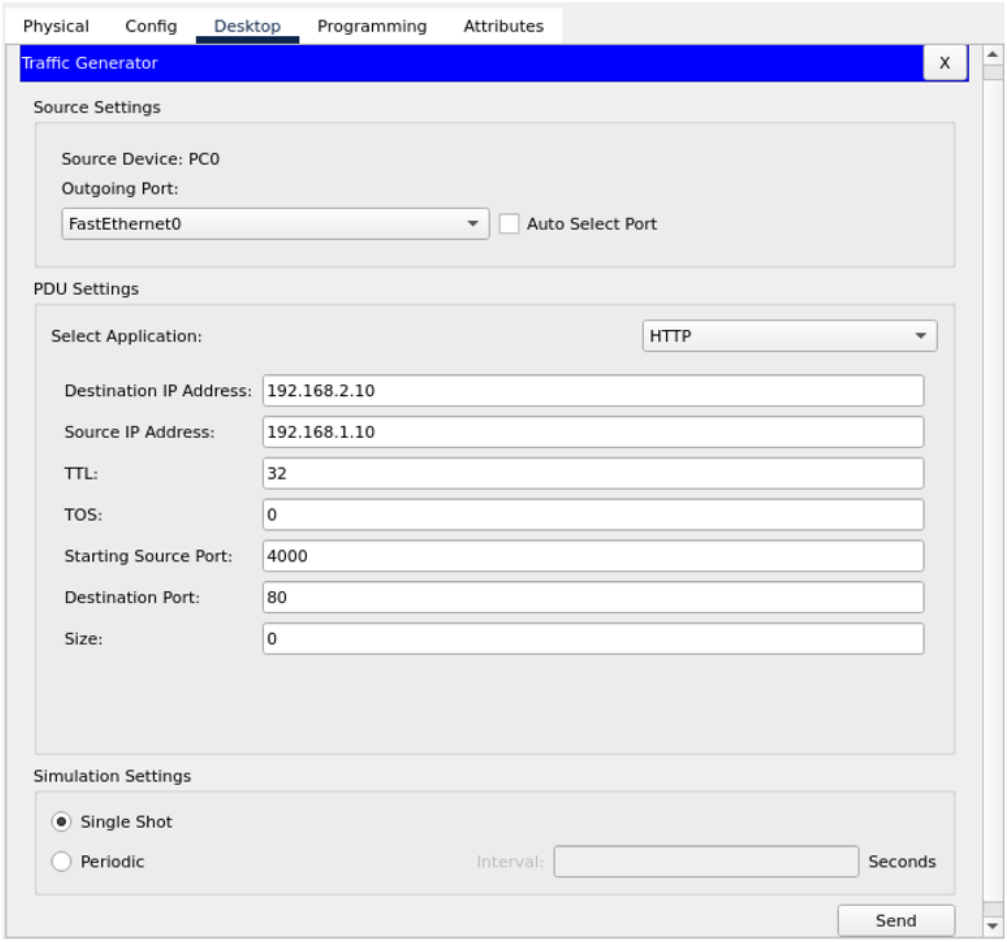

Рисунок 2.3 – Настройка генератора трафика на отправку веб-запроса

Нажмите клавишу Send и еще раз выведите информацию о настроенной политике ранее указанной командой. Как изменилось число пакетов, попавших под политику?

Конфигурацию маршрутизатора можно сохранить командой

copy running-config startup-config. Сохраните изменения в файле Cisco Packet Tracer и загрузите его на проверку.

Практическая работа № 2

«Контроль перегрузки в протоколе TCP»

Mininet — эмулятор компьютерной сети в ОС Linux, созданный на основе развитых в Linux средств изолирования процессов, таких как namespaces, cgroups, а также утилиты классификации и управления трафиком tc из пакета iproute2, установленного по умолчанию в любом современном дистрибутиве Linux.

Mininet позволяет эмулировать такие устройства как компьютеры, коммутаторы, маршрутизаторы, а также контроллеры протокола OpenFlow для программного управления конфигурацией сетевых устройств (SDN-контроллер) [1].

Гибкость и удобство использования Mininet сделало его востребованным и широко используемым в учебном процессе во многих учебных заведениях, например, в таких университетах как Стенфорд и MIT [2, 3].

С помощью Mininet можно познакомиться с устройством и функционированием компьютерных сетей без необходимости использования какого-либо сетевого оборудования, получив, в то же время, результат, близкий к реальной картине в реальной сети. Основная функциональность Mininet реализована на Python, за исключением некоторых утилит, написанных на Си.

1. Теоретический материал

Протокол TCP

Протокол TCP функционирует на четвертом уровне OSI (Layer 4). Он составляет 80-90 процентов современного сетевого трафика. Сегмент TCP состоит из заголовка и блока данных. Структура заголовка состоит из довольно большого числа полей, представленного на рисунке 1.1.

Хотя TCP пересылает данные в СЕГМЕНТАХ, но это БАЙТ-ОРИЕНТИРОВАННЫЙ протокол (Byte-stream), поэтому поток данных контролируется ПОБАЙТНО. С целью контроля потока данных (в байтах внутри сегмента) байты нумеруются в возрастающем порядке. Заголовок TCP содержит поле "номер последовательности" (Sequence Number - SN) – 32 битный номер первого текущего байта в сегменте.

Рисунок 1.1 – Структура заголовка сегмента TCP

При этом сегмент передаваемых данных TCP можно условно представить следующим образом:

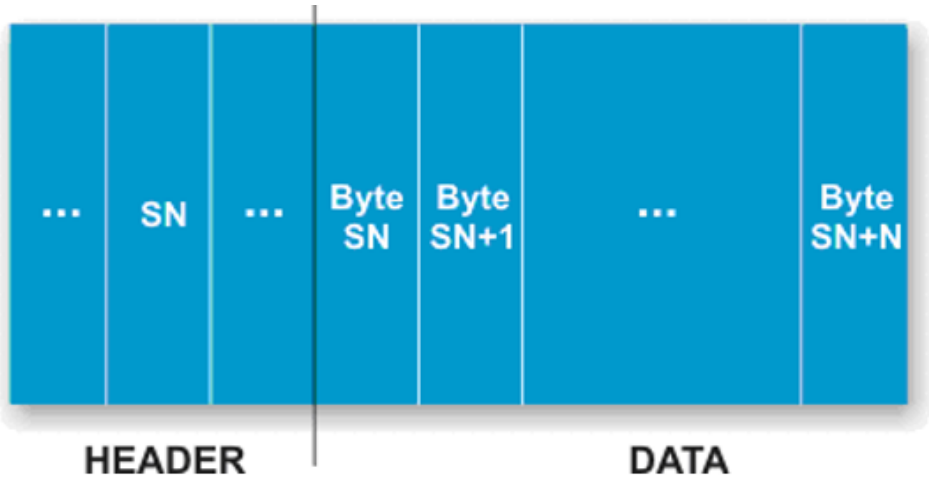

Рисунок 1.2 – Условное представление потока байт в сегменте

В самом начале соединения, когда еще ничего не передавалось, SN называется "начальным номером последовательности" (Initial Sequence Number — ISN). Гарантированная доставка передаваемых данных в TCP обеспечивается квитированием, то есть подтверждением приемной стороной получения данных. В каждом сегменте, идущем от получателя к отправителю, содержится 32 битный номер СЛЕДУЮЩЕГО ожидаемого к приему байта в потоке на стороне приемника (Acknowledge Number - AN). Это число отражает факт успешного получения отправленных ранее (AN-1) байт.

Установление TCP-соединения

Передатчик (Sender) посылает в сторону приемника (Receiver) пакет с установленным флагом Synchronization (SYN), что означает инициацию установления соединения передатчик => приемник. Кроме того, пакет содержит стартовый номер байтовой последовательности (Initial Sequence Number - ISN_A), равный произвольно выбранному числу (алгоритм выбора ISN зависит от реализации протокола, в общем случае он псевдослучаен). Процедура «трехстороннего рукопожатия» при старте TCP-сессии представлена на рисунке 1.3.

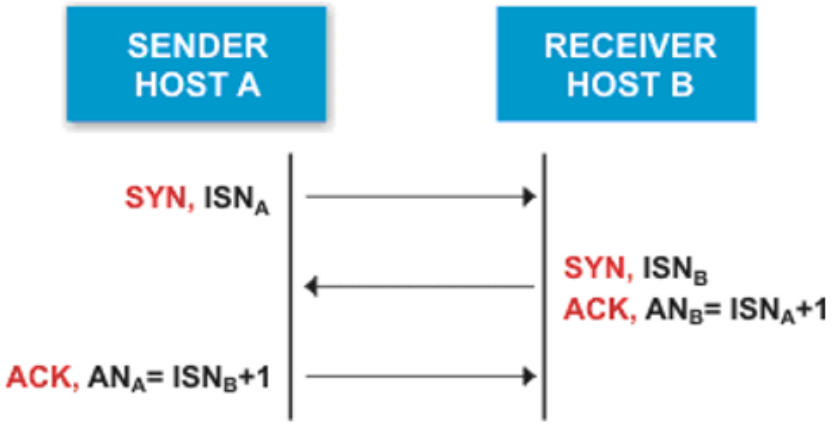

Рисунок 1.3 – Процедура «трехстороннего рукопожатия» при старте TCP-сессии

Приемник (Receiver) посылает передатчику (Sender) в ответ на его SYN пакет с флагом Acknowledge (ACK), что означает согласие на обмен данными. Дополнительно приемник сообщает номер следующего ожидаемого от передатчика байта в потоке данных, заполняя поле Acknowledge Number (AN_B) (значением ISN_A+1, что очевидно, так как пакет с ISN_A уже получен).

Кроме того, TCP – дуплексный протокол, поэтому приемник, в свою очередь, открывает соединение приемник => передатчик, устанавливая в том же пакете с ACK флаг SYN и указывая свой ISN_B. Передатчик, получив ACK от приемника, считает канал передатчик => приемник установленным и может дальше передавать сегменты с байтами, начиная их нумерацию с ISN_A+1. Однако, он получил еще и SYN от приемника и отвечает на него пакетом с ACK и AN равным ISN_B + 1.

Приемник, получив от передатчика подтверждение принятого им SYN в виде ACK пакета с $AN_A = ISN_B + 1$, считает канал приемник => передатчик установленным. В этот момент дуплексный канал связи приемник <=> передатчик налажен и дальше передаются только подтверждения приема очередной порции байт (пакеты только с ACK). Саму процедуру установления соединения TCP часто называют «трехсторонним рукопожатием» (3-way Handshake). Завершиться обмен может штатно (пакет с флагом FIN), когда данные закончатся, или принудительно (пакет с флагом RST).

Окно управления потоком TCP

Обязательное подтверждение каждого сегмента сильно замедляет скорость передачи данных. Поэтому вводится «окно управления потоком» (WINDOW SIZE или просто WINDOW), этим термином называется число сегментов, которое отправитель посылает, не дожидаясь подтверждения (то есть ACK). Технология называется "скользящее окно" (Sliding Window) и образно может трактоваться как перемещение (скольжение) окна передачи данных по потоку байтов.

Размер окна сообщается приемной стороной (поле WINDOW или «размер окна» в заголовке). Кроме того, размер окна может меняться в процессе передачи. Каждое подтверждение (ACK) содержит анонс окна, которое приемник может обработать. Если подтверждение (ACK) на порцию пакетов не получено, передатчик снижает размер окна и повторно посылает сегменты, на которые не пришло подтверждение. Изменение размера окна в процессе передачи представлено на рисунке 1.4.

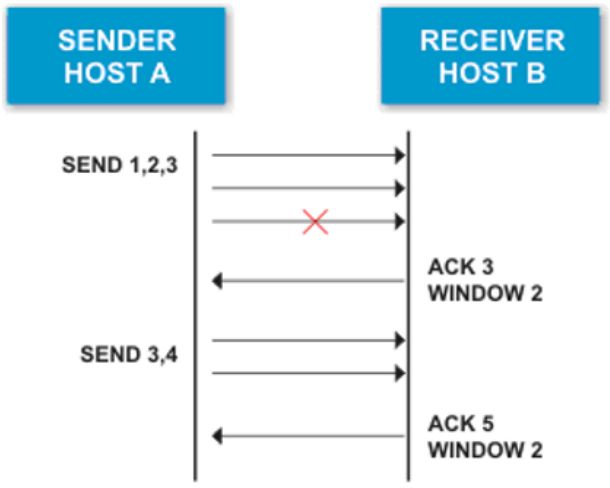

Рисунок 1.4 – Изменение размера окна в процессе передачи

В примере на рис. 4 мы видим, что WINDOW сначала равно 3, а затем приемник не смог обработать один пакет и изменил размер окна до 2.

Управление перегрузками, Congestion control

Предотвращение перегрузок TCP реализуется двумя алгоритмами, представленными на рисунке 1.5.

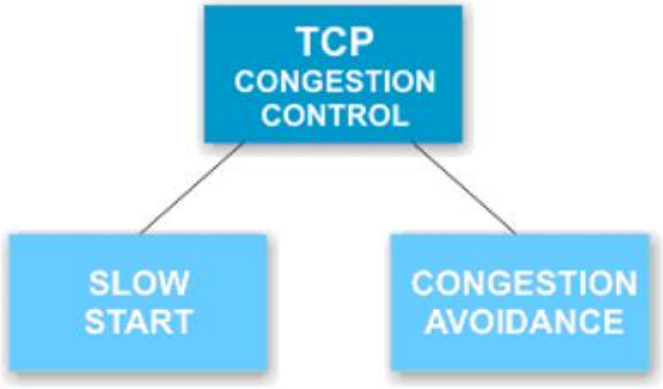

Рисунок 1.5 – Контроль перегрузок в TCP

Медленный старт. Slow Start.

Когда передатчик получает от приемника подтверждение ACK, дополнительно приемник передает размер окна WINDOW, которое характеризует его (приемника) возможности по получению пакетов (размер буферов и прочее). WINDOW измеряется в сегментах. Первоначальная реализация протокола TCP предполагала, что передатчик начнет передавать данные по технологии плавающего окна с числом передаваемых без подтверждения сегментов равным WINDOW.

В современных реализациях протокола TCP параметр WINDOW модифицирован. Окно, объявляемое приемником, как максимальное число сегментов, отправляемых без подтверждения, которое он готов обработать называется не просто WINDOW, а «объявленное приемником окно» - RECEIVER'S ADVERTISED WINDOW. Для его обозначения используется аббревиатура rwnd. Окно, используемое передатчиком, как правило, имеет меньшее значение, оно лишь стремиться достигнуть rwnd и имеет название – окно перегрузки, CONGESTION WINDOW (cwnd). Именно cwnd определяет число передаваемых без квитирования (подтверждения) сегментов.

В начале передачи данных CONGESTION WINDOW = 1 и передается один сегмент. После каждого успешного подтверждения CONGESTION WINDOW удваивается, а вместе с ним удваивается число передаваемых без подтверждения сегментов. Математически это означает экспоненциальный рост. Увеличение CONGESTION WINDOW происходит до значения rwnd. Изменение cwnd после старта соединения представлено на рисунке 1.6.

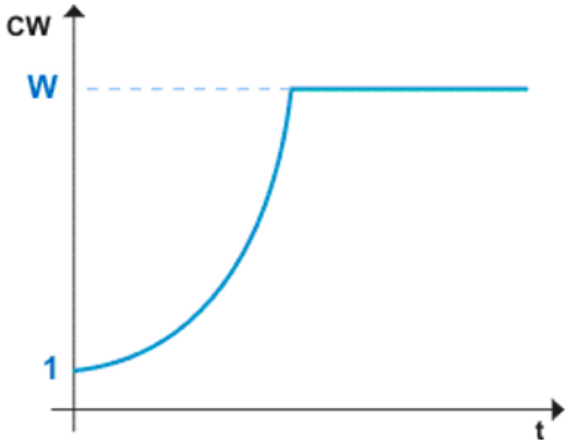

Рисунок 1.6 – Изменение cwnd после старта соединения

В случае перегрузки передатчик не получает очередное подтверждение и сбрасывает CONGESTION WINDOW до 1. Изменение cwnd при возникновении перегрузки представлено на рисунке 1.7.

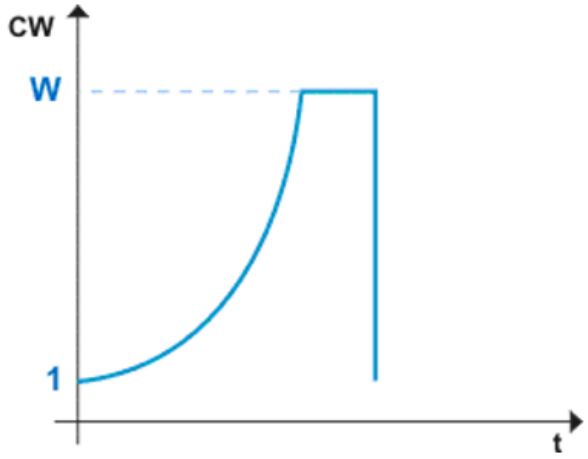

Рисунок 1.7 – Изменение cwnd при возникновении перегрузки

При перегрузке задействуется специальный дополнительный параметр - «пороговое значение медленного старта», Slow Start Threshold Size (SSTHRESH). По умолчанию ssthresh=65636, но после детектирования перегрузки принимает значение rwnd/2 или WINDOW/2 и в дальнейшем является верхней границей экспоненциального роста cwnd (когда CONGESTION WINDOW удваивается). После достижения порогового значения CONGESTION WINDOW растет уже не экспоненциально, а линейно (с коэффициентом 1/cwnd) снова до значения rwnd. Изменение динамики cwnd после перегрузки представлено на рисунке 1.8.

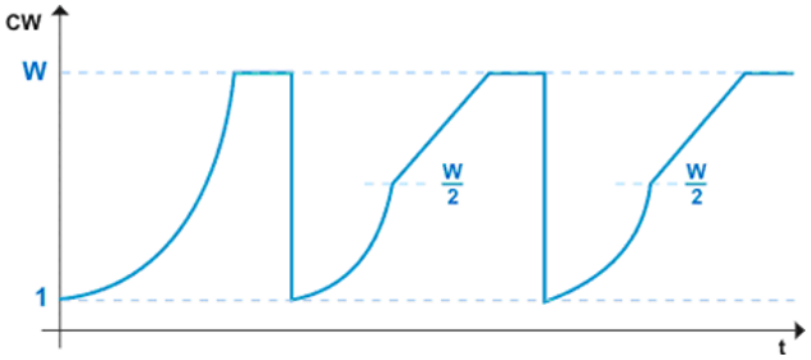

Рисунок 1.8 – Изменение динамики cwnd после перегрузки

Область линейного увеличения cwnd часто называется режимом исключения перегрузки (congestion avoidance) или AIMD (Additive Increase, Multiple Decrease). В случае, если cwnd так и не дойдет до WINDOW (проблемы в сети, подтверждение не пришло), то пороговое значение ssthresh еще уменьшится в два раза от своего предыдущего значения, и так далее. Изменение порога ssthresh при плохом соединении показано на рисунке 1.9.

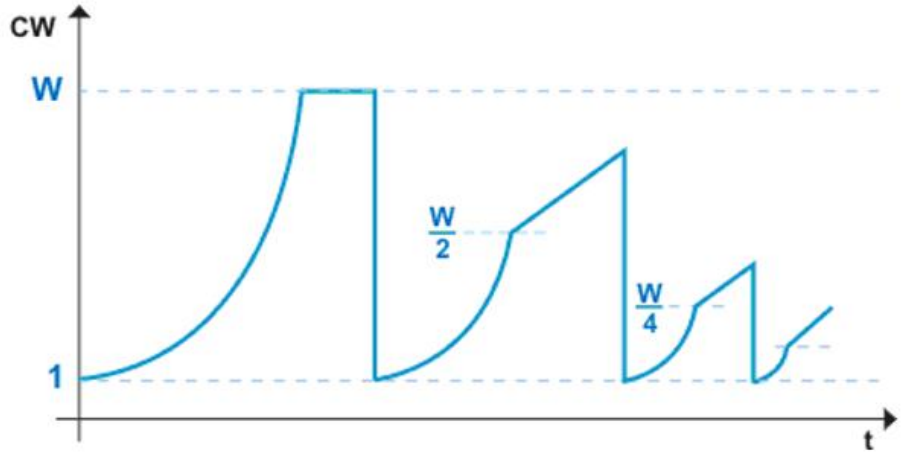

Рисунок 1.9 – Изменение порога ssthresh при плохом соединении

Таймеры TCP

Таймер повторной передачи.

Для детектирования потери сегмента и неполучения соответствующего ACK используется специальный таймер - таймер повторной передачи (Retransmission Time Out - RTO). Данный таймер принимает некоторое стартовое значение (RTO) в момент посылки TCP-сегмента получателю и начинает уменьшаться. В случае, если данный таймер окажется сброшенным в ноль до момента получения подтверждения (ACK), то переданный пакет считается потерянным и требуется повторная передача. Кроме того, начинается процедура пересчета ssthresh.

Так как маршруты пакетов могут быть разными, время их прохождения варьируется. Поэтому величина RTO не может быть фиксированной и должна вычисляться для каждого логического соединения. Предварительно для каждого пакета TCP измеряется величина, называемая временем отклика (Round Trip Time - RTT) – интервал времени от момента посылки TCP пакета до момента получения подтверждения на него. На основе RTT вычисляется значение сглаженной величины RTT (Smoothed RTT - SRTT), что обеспечивает фильтрацию резких выбросов. А уже от SRTT вычисляется RTO. Таким образом, RTO является функцией от RTT, что позволяет учитывать временные особенности каждого соединения.

В случае, если и после повторной передачи пакета, опять не придет подтверждение за интервал RTO, то попытки передачи будут повторяться (ограниченное число раз, до 12). При этом параметр RTO будет экспоненциально увеличиваться (протокол предполагает ухудшение условий и на всякий случай повышает RTT). И только после неудачи всех попыток происходит аварийное закрытие соединения.

Таймер возобновления передачи

В ходе взаимодействия возможно следующее. Приемная сторона уведомляет передатчик о невозможности приема (установив размер окна в ноль), передатчик останавливает на некоторое время обмен, приемник через некоторое время желает возобновить прием пакетов и посылает пакет с НЕНУЛЕВЫМ ОКНОМ, но тот по какой-то причине теряется. Передатчик так и будет считать, что приемник не может принимать пакеты, а приемник не дождется от передатчика подтверждения на возобновление передачи. Возникает тупиковая ситуация.

С целью предотвращения подобного, для каждого соединения вводится таймер возобновления передачи (Persistent Timer - PT). Он взводится в момент получения пакета с нулевым размером окна. Если до момента обнуления таймера так и не придет разрешения на передачу, произойдет отправка одного "разведывательного" пакета.

Основы работы в командной строке Mininet

Работа с виртуальной сетью mininet, а именно развертывание сети желаемой топологии, изменение различных параметров хостов или коммутаторов и т. п., выполняется в интерпретаторе команд – mn. Откройте окно терминала (окно командной строки) и выполните команду:

```
$ sudo mn
```

Запущенный без параметров командой mn Mininet перейдет в режим интерпретации команд. При этом по умолчанию будет создана минимальная сеть, состоящая из двух хостов (h1, h2), коммутатора (s1) и OpenFlow-контроллера (c1). Результат выполнения команды, как он отображается на экране терминала:

```
$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 1 switches
s1
*** Starting CLI:
mininet>
```

Можно видеть, что приглашение командной строки изменилось и мы находимся теперь в интерпретаторе команд Mininet. С помощью команды help в интерпретаторе можно получить краткую справку по доступным командам:

```
mininet> help
```

Построение сети в эмуляторе Mininet начинается с топологии. Топология определяет сколько будет хостов, коммутаторов, а так же каким образом они будут объедены в сеть.
Шаблоны четырех базовых топологий уже заданы в программе и могут быть задействованы из командной строки при запуске Mininet. Опишем кратко эти топологии:

1. Топология mininal. Используется по умолчанию при запуске mn без параметров. В этом случае создаются два хоста, подключенные к одному коммутатору, который, в свою очередь управляется OpenFlow-контроллером.

2. Топология single. Как и в случае с minimal, все хосты подключаются к одному коммутатору, но можно указать их количество. Соответствующая команда показана ниже. Цифра в команде задает число хостов (компьютеров):

```
$ sudo mn —topo single,24
```

3. Топология linear. Описывает сеть, в которой все хосты подключены к собственным коммутаторам, которые, в свою очередь, соединены между собой:

```
$ sudo mn —topo linear,6
```

Указывая в команде число, мы задаем и количество узлов, и количество коммутаторов одновременно.

4. Топология tree. Древовидная топология здесь в качестве параметров можно указать глубину иерархии коммутаторов (depth), а так же число подключенных к ним хостов (fanout).

```
$ sudo mn —topo tree,depth=3,fanout=4
```

Последняя топология задает наиболее сложную иерархию соединений. В случае выбора трех уровней эта иерархия будет соответствовать общепринятой иерархии построения крупной локальной сети предприятия, где верхний уровень – магистральный, средний –

уровень распределения и нижний – уровень доступа к сети. Такая структура отражена и в стандартах на проектирование структурированных кабельных систем, являющихся фундаментом локальной сети [6].

Можно построить и свою собственную топологию произвольной сложности и конфигурации. Описать эту топологию нужно будет на языке программирования Python. Следующая команда показывает, как запускается эмуляция такой сети произвольной топологии:

```
$ sudo mn —custom /<путь к скрипту>/special_topology.py —topo mytopo
```

Более подробное описание этого вопроса можно найти здесь [7].

Рассмотрим несколько наиболее полезных команд Mininet.

Вывести список всех устройств можно с помощью команды nodes:

```
mininet> nodes
available nodes are:
h1 h2 c0 s1
```

Посмотреть топологию сети, а именно сопоставление портов коммутатора и хостов можно с помощью команды net:

```
mininet> net
c0
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
```

Вывести конфигурацию сетевого интерфейса конкретного устройства можно с помощью команды ifconfig перед которой необходимо указать имя конкретного узла:

```
mininet> h1 ifconfig
```

Любой из портов устройства можно выключить или включить, в команде указывается какие два устройства этот линк соединяет:

```
mininet> link s1 h1 down
mininet> link s1 h1 up
```

Посмотреть таблицу маршрутизации конкретного хоста можно с использованием стандартной Linux команды route:

```
mininet> h1 route
```

Можно выполнить проверку связи между устройствами отправкой эхо-запросов командой ping:

```
mininet> h1 ping h2
```

Предусмотрена специальная версия команды ping, которая реализует отправку эхо-запросов между всеми хостами сети:

```
mininet> pingall
```

Вообще говоря, так как Mininet разворачивается на ОС Linux, на каждом из хостов, указывая предварительно его имя, можно выполнять большинство стандартных команд Linux. Например, посмотреть запущенные на устройстве процессы с помощью команды ps:

```
mininet> s1 ps
```

Можно завершить любой из процессов с помощью команды kill, протестировать пропускную способность между узлами с командой iperf:

```
mininet> iperf h1 h2
```

Открыть окно терминального доступа к любому из узлов можно командой:

```
mininet> xterm h1
```

Разумеется, успешное выполнение последней команды возможно в том случае, когда в ОС Linux работает сервер X Window.

2. Ход выполнения работы

В этой лабораторной работе мы изучим динамику TCP в небольшой, «домашней» локальной сети. На рисунке 2.1 показана «типичная» топология «домашней» сети с одним маршрутизатором и подключенным к нему пользовательским компьютером.

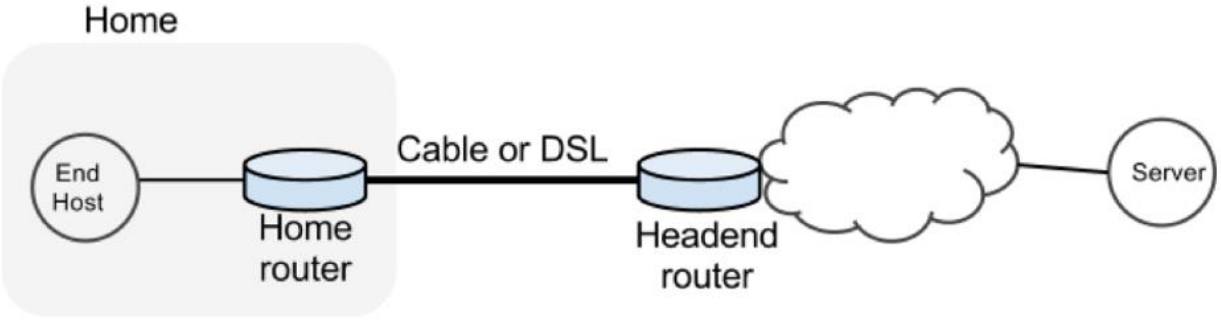

Рисунок 2.1 – «Домашняя» локальная сеть

Домашний маршрутизатор подключен через кабель или DSL к маршрутизатору провайдера Headend router, который обеспечивает доступ в Интернет к различным сервисам, там расположенным. Мы изучим что происходит, когда мы загружаем данные с удаленного сервера на конечный хост в нашей домашней сети.

В реальной сети, на самом деле, довольно сложно измерить влияние такого параметра TCP-соединения как окно перегрузки (Congestion Window), обозначаемое сокращением **cwnd** (прежде всего, потому что эта информация о конфигурации сервера, пользователю недоступная) и занятость буфера (потому что он приватен для маршрутизатора). Эмулируя сеть в Mininet мы можем провести все измерения.

Цели лабораторной работы:

- Узнать на практических примерах о влиянии окна перегрузки cwnd и буфера памяти на маршрутизаторе на динамику передачи данных в реальной сети.
- Узнать почему увеличение размера буфера памяти маршрутизатора может привести к снижению производительности в домашних сетях. Эту проблему часто называют «раздуванием буфера».

Научиться использовать Mininet для повторения, прерывания и продления экспериментов.

1. Настройка и запуск сетевой топологии для проведения эксперимента.

На рисунке 2.2 показана сетевая топология, состоящая из маршрутизатора с имеющимся на нем настраиваемом программном буфере для временного хранения передаваемых данных и двух подключенных к маршрутизатору устройств: компьютер и сервер. Линки, которыми подключены компьютер и сервер, имеют разные скорости.

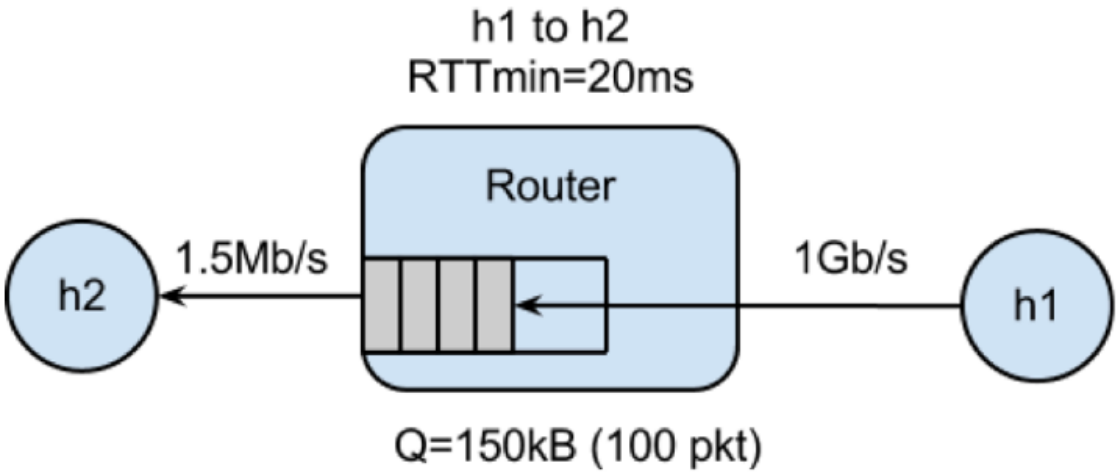

Рисунок 2.2 – Экспериментальная сетевая топология лабораторной работы

Для реализации такой специальной топологии потребуется использование специального варианта custom создания сетевой топологии в Mininet. Такой специальный вариант уже подготовлен и размещен на ресурсе bitbucket.org. Для развертывания копии этой топологии достаточно выполнить клонирование файлов проекта командой:

git clone https://bitbucket.org/huangty/cs144_bufferbloat.git

Откройте окно терминала, убедитесь, что вы в домашней директории, и выполните указанную команду. Она создаст в вашем домашнем каталоге подкаталог **cs144_bufferbloat** со всеми файлами проекта.

Перейдите в указанный каталог и выполните скрипт **run.sh** командами:

> cd cs144_bufferbloat/

> sudo ./run.sh

Если в операционной системе предустановлен Mininet, то он будет запущен и необходимая топология будет создана.

2. Тестирование задержки при отправке эхо-запроса и скачивании веб-страницы

Выполнение скрипта run.sh завершается проверкой соединения и отображает среднюю величину задержки получения сигнала между хостами h1 и h2. Повторим это измерение командой:

mininet> h1 ping -c 10 h2

Данная команда создает 10 эхо-запросов с хоста h1 к хосту h2. Убедимся, что задержка RTT (round-trip time) действительно составляет 20 миллисекунд, как и показано на рисунке 1.1.

Скачайте индексную веб-страницу с веб-сайта, размещенного на h1 командой:

mininet> h2 wget http://10.0.0.1

За какое время скачалась веб-страница?

Нарисуйте на графике, представленном на рисунке 2.3, как, по вашему мнению, меняется во времени значение окна перегрузки на сервере h1.

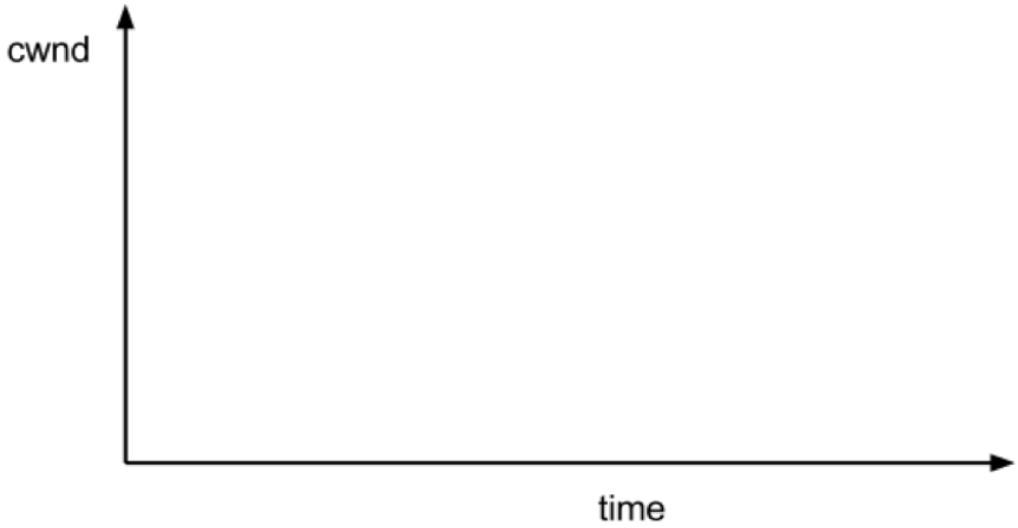

Рисунок 2.3 – Зависимость величины окна перегрузки во времени

Выполните команду:

mininet> exit

Эта команда завершит выполнение сценария mininet и вернет окно командной строки. Выполните команду запуска сценария повторно:

> sudo ./run.sh

Запустите в фоновом режиме программу-анализатор сетевых пакетов Wireshark на интерфейсе хоста h2:

mininet> h2 wireshark&

Эта команда откроет дополнительное окно Wireshark, а в текущей командной строке вернет управление.

В окне Wireshark выберите интерфейс h2 и запустите захват пакетов, как это показано на рисунке 2.4.

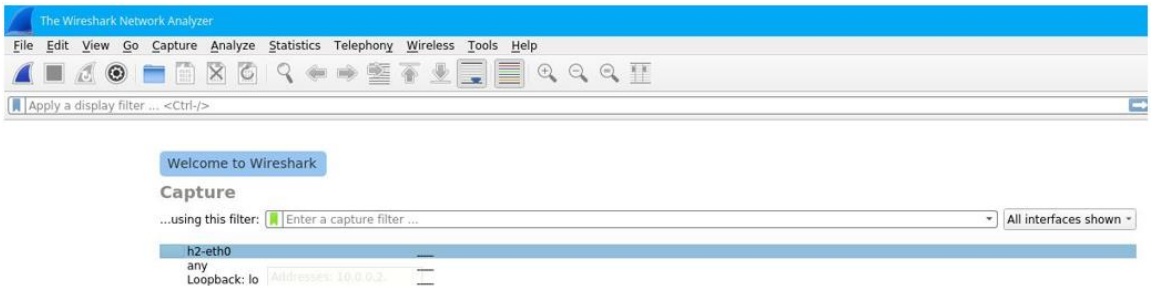

Рисунок 2.4 – Выбор интерфейса виртуального хоста h2

Вернитесь в окно терминала и повторно выполните команду загрузки веб-странички с h1 на h2:

mininet> h2 wget http://10.0.0.1

В окне захвата Wireshark отобразятся проходящие пакеты. Выделите самый первый пакет, затем выберите из меню «Статистика» (Statistics) пункт «TCP Stream Graphs», в этом пункте откройте первый подпункт «Time Sequense». На графике отображается временная последовательность передаваемых сегментов и можно видеть значение окна перегрузки и его изменения. Сравните эти значения с теми, которые вы нарисовали из теоретических предположений.

3. Влияние на задержки интенсивного трафика, например, потокового видео

Чтобы увидеть, как меняется динамика передачи данных при увеличении интенсивности и объема данных (при этом TCP должен переходить в фазу устранения перегрузки AIMD (Additive Increase/Multiplicative Decrease)) в отличии от режима быстрого старта при малых объемах высокой интенсивности, мы повторим эксперимент с измерением задержки, предварительно создав в этой сети режим передачи «потока потокового видео». Вместо того, чтобы реально смотреть видео на вашем компьютере, мы эмулируем долгоживущий поток интенсивного трафика, аналогичный передаче видеопотока. Для генерации потока используется команда iperf. В лабораторной работе мы запустим программный скрипт со всеми необходимыми настройками для iperf следующим образом:
mininet> h1 ./iperf.sh

Указанный скрипт создаст постоянный поток, занимающий практически всю полосу пропускания канала между h2 и маршрутизатором. Занимаемую потоком полосу пропускания можно видеть в реальном времени, если выполнить команду:
mininet> h2 tail -f ./iperf-recv.txt

Чтобы выйти из режима отображения занимаемой полосы пропускания нажмите комбинацию клавиш CTRL - C

Посмотрим, как влияет созданный нами поток трафика на задержку распространения по сети сигнала. Повторим отправку эхо-запроса между хостами h1 и h2:
mininet> h1 ping -c 10 h2

Как изменилась задержка получения ответов на эхо-запросы? Меняется ли она во времени? Попробуйте нарисовать изменение во времени окна перегрузки для этого случая на рисунке 2.5

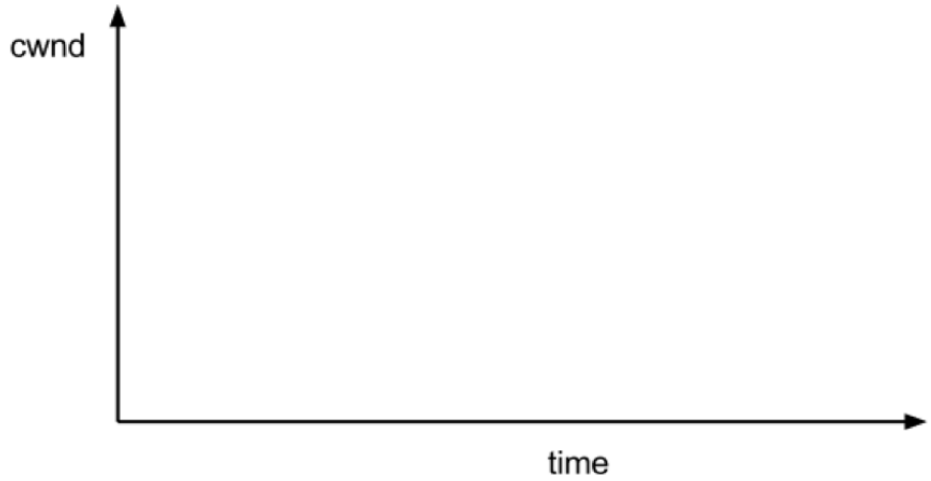

Рисунок 2.4 – Изменение cwnd при передаче видеопотока

Попробуйте еще раз загрузить веб-страницу с веб-сервера:
mininet> h2 wget http://10.0.0.1

Как долго выполнялась загрузка в этот раз? Объясните наблюдаемый результат.

Завершите выполнение сценария:
mininet> exit
Выполните запуск еще раз:
> sudo ./run.sh
Запустите эмуляцию загрузки канала видеопотоком:
mininet> h1 ./iperf.sh
Запустите в фоновом режиме программу-анализатор сетевых пакетов Wireshark на интерфейсе хоста h2:
mininet> h2 wireshark&
Включите захват пакетов на интерфейсе h2. Еще раз загрузите веб-страницу с веб-сервера:
mininet> h2 wget http://10.0.0.1

По завершению загрузки остановите захват пакетов в Wireshark (Поскольку эмуляция потока видео продолжает работать, пакеты будут непрерывно захватываться). Среди захваченных пакетов найдите блок, соответствующий загрузке веб-странички (в таком блоке будет указан протокол передачи гипертекста HTTP, и он будет выделяться цветом). Выберите и выделите в этом блоке любой пакет. Перейдите в меню «Статистика» (Statistics) пункт «TCP Stream Graphs», в этом пункте откройте первый подпункт «Time Sequense». Определите по отображаемым графикам значения окна перегрузки, сравните их с теми, который вы нарисовали из теоретических предположений и с теми, что вы наблюдали ранее. Изучите другие подпункты и отображаемые ими графики. Открывать их можно, не возвращаясь в исходное меню, а прямо в окне графика, выбирая в левом нижнем углу соответствующий пункт из раскрывающегося меню.

Как изменяется окно перегрузки в этом случае?

4. Влияние размера буфера на скорость загрузки

Завершите сценарий Mininet:

```
mininet> exit
```

Выполните запуск на этот раз измененного сценария:

```
sudo ./run-minq.sh
```

В этом сценарии размер буфера маршрутизатора уменьшен со 100 пакетов до 20.

Запустите в фоновом режиме программу-анализатор сетевых пакетов Wireshark на интерфейсе хоста h2:

```
mininet> h2 wireshark&
```

Запустите последовательно:

1) загрузку веб-страницы:

```
mininet> h2 wget http://10.0.0.1
```

2) Включение эмуляции видеопотока с h1 на h2:

```
mininet> h1 ./iperf.sh
```

3) Повторную загрузку веб-страницы:

```
mininet> h2 wget http://10.0.0.1
```

По завершению загрузки остановите захват пакетов в Wireshark.

Как изменилось время загрузки веб-страницы без параллельного видеопотока и с ним для данного случая уменьшенного в 5 раз буфера памяти? Поясните почему.

Проанализируйте графики TCP Stream Graphs для двух вариантов загрузки веб-страницы с новым значением буфера. Что можете сказать о значении окна перегрузки?

5. Разделение трафика по типам и формирование отдельных очередей

В рассмотренных выше примерах все сетевые пакеты попадали в общую очередь, т.к. маршрутизатор не определял их тип. Что если мы будем формировать отдельные очереди для каждого типа трафика, как это показано на рисунке 2.5.

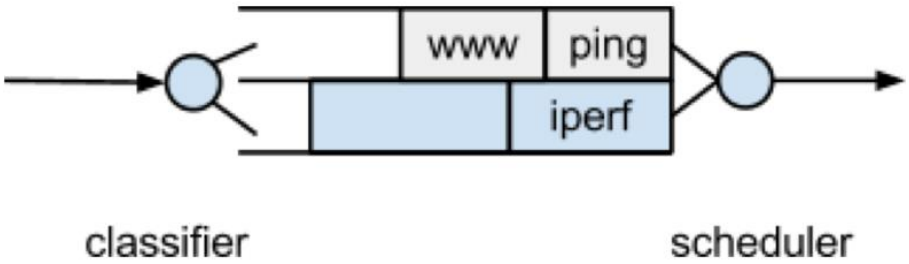

Рисунок 2.5 – Отдельные очереди для разных типов трафика

Перезапустите Mininet:

```
mininet> exit
```

Запустите его снова, на этот раз со специальной версией маршрутизатора, которая умеет определять типы трафика и формирует разные очереди (имеет отдельные буферы памяти) для разных типов трафика:

sudo ./run-diff.sh

Запустите в фоновом режиме программу-анализатор сетевых пакетов Wireshark на интерфейсе хоста h2:

mininet> h2 wireshark&

И повторите все измерения:

1) загрузку веб-страницы:

mininet> h2 wget http://10.0.0.1

2) Включение эмуляции видеопотока с h1 на h2:

mininet> h1 ./iperf.sh

3) Повторную загрузку веб-страницы:

mininet> h2 wget http://10.0.0.1

По завершению загрузки остановите захват пакетов в Wireshark.

Как повлияло разделение трафика по типам и отдельная буферизация на время загрузки веб-страницы без параллельного видеопотока и с ним? Поясните почему.

3. Контрольные вопросы к работе:

1. Что такое CWND и RWND?
2. Как меняется окно перегрузки в режиме медленного старта?
3. Как определяется возникновение перегрузки?
4. Что такое AIMD, для чего он предложен?
5. Буферы сетевых устройств используют схему первый_вошел-первым_вышел (FIFO). Предполагается, что размер этих буферов соответствует произведению RTT*B (B - полоса пропускания, RTT – время двойного оборота). Если последнее условие нарушено, пропускная способность неизбежно понизится и будет определяться размером буфера, а не полосой пропускания канала. Поясните почему.
6. Почему уменьшение размера буфера уменьшает время загрузки?

Практическая работа № 3

«Настройка контроллера управления беспроводной сетью»

1. Теоретический материал

Аутентификация в беспроводной сети имеет специфику. Здесь аутентифицируется не пользователь, а устройство, получающее доступ к сети. Стандарт IEEE 802.11 предусматривает два механизма аутентификации беспроводных абонентов: открытую аутентификацию (Open Authentication) и аутентификацию с общим ключом (Shared Key Authentication). В аутентификации в беспроводных сетях также широко используются два других механизма, выходящих за рамки стандарта 802.11, а именно назначение идентификатора беспроводной локальной сети (Service Set Identifier - SSID) и аутентификация абонента по его MAC-адресу (MAC Address Authentication). Идентификатор беспроводной локальной сети (SSID) представляет собой конфигурационный атрибут беспроводной сети, позволяющий логически отличать сети друг от друга. В общем случае абонент беспроводной сети должен задать у себя соответствующий SSID для того, чтобы получить доступ к требуемой беспроводной локальной сети.

Процесс аутентификации абонента беспроводной локальной сети IEEE 802.11 состоит из следующих этапов, представленных на рисунке 1.1:

Абонент (Client) посылает кадр Probe Request во все радиоканалы.

Каждая точка радиодоступа (Access Point - AP), в зоне радиовидимости которой находится абонент, посылает в ответ кадр Probe Response.

Абонент выбирает предпочтительную для него точку радиодоступа (обычно по уровню сигнала) и посылает в обслуживаемый ею радиоканал запрос на аутентификацию (Authentication Request).

Точка радиодоступа посылает подтверждение аутентификации (Authentication Reply).

В случае успешной аутентификации абонент посылает точке радиодоступа кадр ассоциации (Association Request).

Точка радиодоступа посылает в ответ кадр подтверждения ассоциации (Association Response).

Абонент может теперь осуществлять обмен пользовательским трафиком с точкой радиодоступа и проводной сетью.

Рисунок 1.1 – Процесс аутентификации абонента в беспроводной сети

При активизации беспроводный абонент начинает поиск точек радиодоступа в своей зоне радиовидимости с помощью управляющих фреймов Probe Request. Фреймы Probe Request посылаются в каждый из радиоканалов, поддерживаемых абонентским радиоинтерфейсом, чтобы найти все точки радиодоступа с необходимыми клиенту идентификатором SSID и поддерживаемыми скоростями радиообмена. Каждая точка радиодоступа из находящихся в зоне радиовидимости абонента, удовлетворяющая запрашиваемым во фрейме Probe Request параметрам, отвечает фреймом Probe Response, содержащим синхронизирующую информацию и данные о текущей загрузке точки радиодоступа. Абонент определяет, с какой точкой радиодоступа он будет работать, путем сопоставления поддерживаемых ими скоростей радиообмена и загрузки. После того как предпочтительная точка радиодоступа определена, абонент переходит в фазу аутентификации.

Открытая аутентификация, по сути, не является алгоритмом аутентификации в привычном понимании. Точка радиодоступа удовлетворит любой запрос открытой аутентификации. На первый взгляд использование этого алгоритма может показаться бессмысленным, однако следует учитывать, что разработанные в 1997 году методы аутентификации IEEE 802.11 ориентированы на быстрое логическое подключение к беспроводной локальной сети. Вдобавок к этому многие IEEE 802.11-совместимые устройства представляют собой портативные блоки сбора информации (сканеры штрих-кодов и т. п.), не имеющие достаточной процессорной мощности, необходимой для реализации сложных алгоритмов аутентификации.

В процессе открытой аутентификации происходит обмен сообщениями двух типов:

- запрос аутентификации (Authentication Request);
- подтверждение аутентификации (Authentication Response).

Таким образом, при открытой аутентификации возможен доступ любого абонента к беспроводной локальной сети. Если в беспроводной сети шифрование не используется, любой абонент, знающий идентификатор SSID точки радиодоступа, получит доступ к сети. При использовании точками радиодоступа шифрования WEP сами ключи шифрования становятся средством контроля доступа. Если абонент не располагает корректным WEP-ключом, то даже в случае успешной аутентификации он не сможет ни передавать данные через точку радиодоступа, ни расшифровывать данные, переданные точкой радиодоступа. Аутентификация с общим ключом представлена на рисунке 1.2.

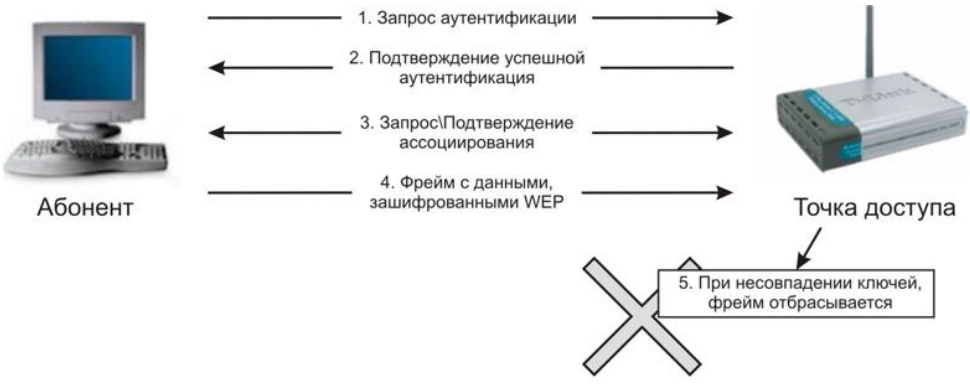

Рисунок 1.2 – Аутентификация с общим ключом

Аутентификация с общим ключом является вторым методом аутентификации стандарта IEEE 802.11. Она использует одинаковый ключ шифрования WEP, который вводится в конфигурацию на точке беспроводного доступа и на клиенте. Процесс аутентификации иллюстрирует рисунок 1.3:

- Абонент посылает точке радиодоступа запрос аутентификации, указывая при этом необходимость использования режима аутентификации с общим ключом.
- Точка радиодоступа посылает подтверждение аутентификации, содержащее Challenge Text.
- Абонент шифрует Challenge Text своим статическим WEP-ключом и посылает точке радиодоступа запрос аутентификации.
- Если точка радиодоступа в состоянии успешно расшифровать запрос аутентификации и содержащийся в нем Challenge Text, она посылает абоненту подтверждение аутентификации, таким образом предоставляя доступ к сети.

Рисунок 1.3 – Аутентификация WEP

Защита беспроводного соединения с помощью ключей WEP оказалась уязвимой к атакам «грубой силой», т.к. длина ключевой последовательности была недостаточно большой, а сам ключ не изменялся. Злоумышленник, набрав некоторое число пакетов при прослушивании радиосети, может подобрать ключ шифрования перебором всех возможных комбинаций. Поэтому позднее были предложены улучшенные методы защиты WPA, WPA2. Несмотря на улучшение алгоритмов защиты и смену ключей шифрования при каждой отправке нового пакета в беспроводную сеть, оба метода также основывались на аутентификации с общим ключом, англ. PSK – Pre Shared Key. PSK также известен, как четырёхэтапное установление связи, поскольку именно столько сообщений необходимо передать между точкой беспроводного доступа и подсоединяющимся устройством, чтобы подтвердить, что они договорились по поводу пароля, при том, что ни одна из сторон не сообщает его другой. До 2016 года PSK казался безопасным, а потом была открыта атака с переустановкой ключа (Key Reinstallation Attacks, KRACK). В 2018г был предложен новый стандарт защиты: WPA3, в котором применен новый метод аутентификации устройства, SAE, Simultaneous Authentication of Equals. Это разновидность dragonfly handshake [рукопожатия по методу стрекозы], поэтому можно встретить название WPA3 Dragonfly. На рисунке 1.4 представлены стандарты безопасности беспроводных сетей по времени введения

Рисунок 1.4 – Стандарты безопасности беспроводных сетей по времени введения

Несмотря на усложнение алгоритма для защиты от словарных атак и то что WPA3 пока не получил широкого распространения, в 2019 году исследователи показали слабости нового протокола в статье Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd.

Поэтому защита современной беспроводной сети никогда не строится только на аутентификации устройств, дополнительно применяются традиционные методы аутентификации пользователей.

При создании большой распределенной беспроводной сети настройка множества точек радиодоступа становится очень трудоемкой задачей. Для решения этой проблемы были созданы специальные устройства — контроллеры беспроводной сети, WLC - Wireless LAN Controller, а также «упрощенные» точки беспроводного доступа, LAP - Lightweight Access Point, специально предназначенные для подключения к контроллеру. Некоторые модели LAP могут работать самостоятельно, без подключения к контроллеру, но большинство не могут. Контроллер берет на себя большую часть функций управления, в том числе аутентификацию подключающихся клиентов. Между контроллером и LAP, подключенными в общую локальную сеть предприятия, создается защищенное туннельное соединение по специальному протоколу CAPWAP, как показано на рисунке 1.5.

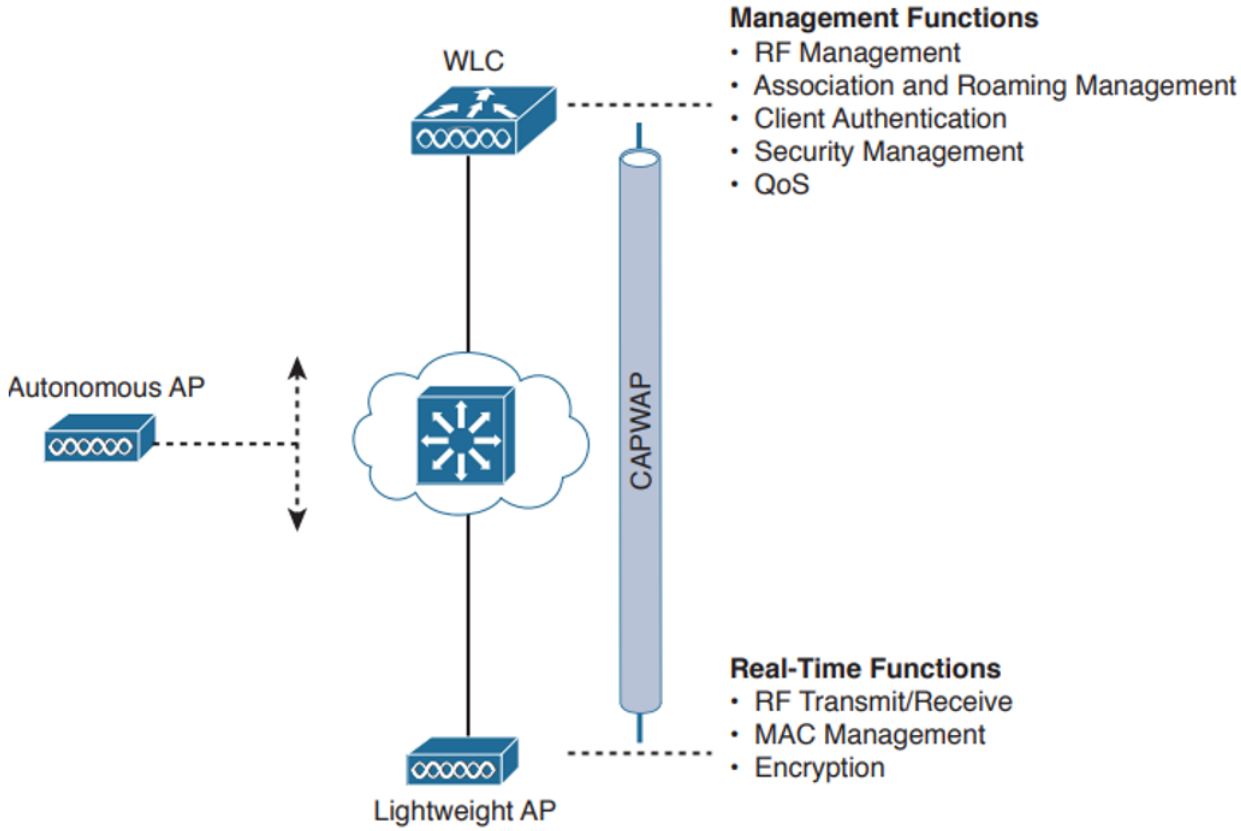

Рисунок 1.5 – Туннель CAPWAP между WLC и LAP

Контроллер может поддерживать одновременную работу с десятками точек радиодоступа.

Еще один шаг в развитии технологий аутентификации в беспроводной сети был сделан с развитием «облачных» сервисов. Можно сказать, что наиболее современным способом управления беспроводной сетью являются облачные сервисы подобные Cisco Meraki, FortiGate NGFW и др. Это комплексные решения, не ориентированные только на поддержку беспроводных сетей, но включающие их как один из элементов. Облачный сервис и специализированные устройства безопасности: security appliances, позволяют быстро развернуть защищенную информационную инфраструктуру предприятия, соединяющую географически распределенные подразделения криптографически защищенными VPN-туннелями. Настройка механизмов защиты, в том числе беспроводных сетей и аутентификации в них, выполняется через веб-интерфейс облачного сервиса. На рисунке 1.6 представлен облачный сервис управления решениями безопасности.

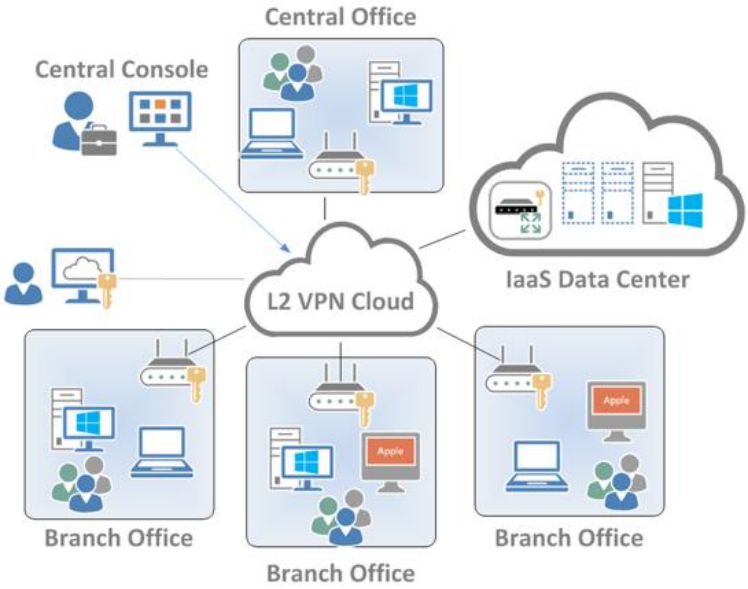

Рисунок 1.6 – Облачный сервис управления решениями безопасности

2. Ход выполнения работы

Для выполнения работы нужно использовать версию Cisco Packet Tracer 8.2.1. Вы настраиваете локальную сеть с участком беспроводной сети под управлением локального контроллера WLC.

Создайте топологию, как показано на рисунке 2.1.

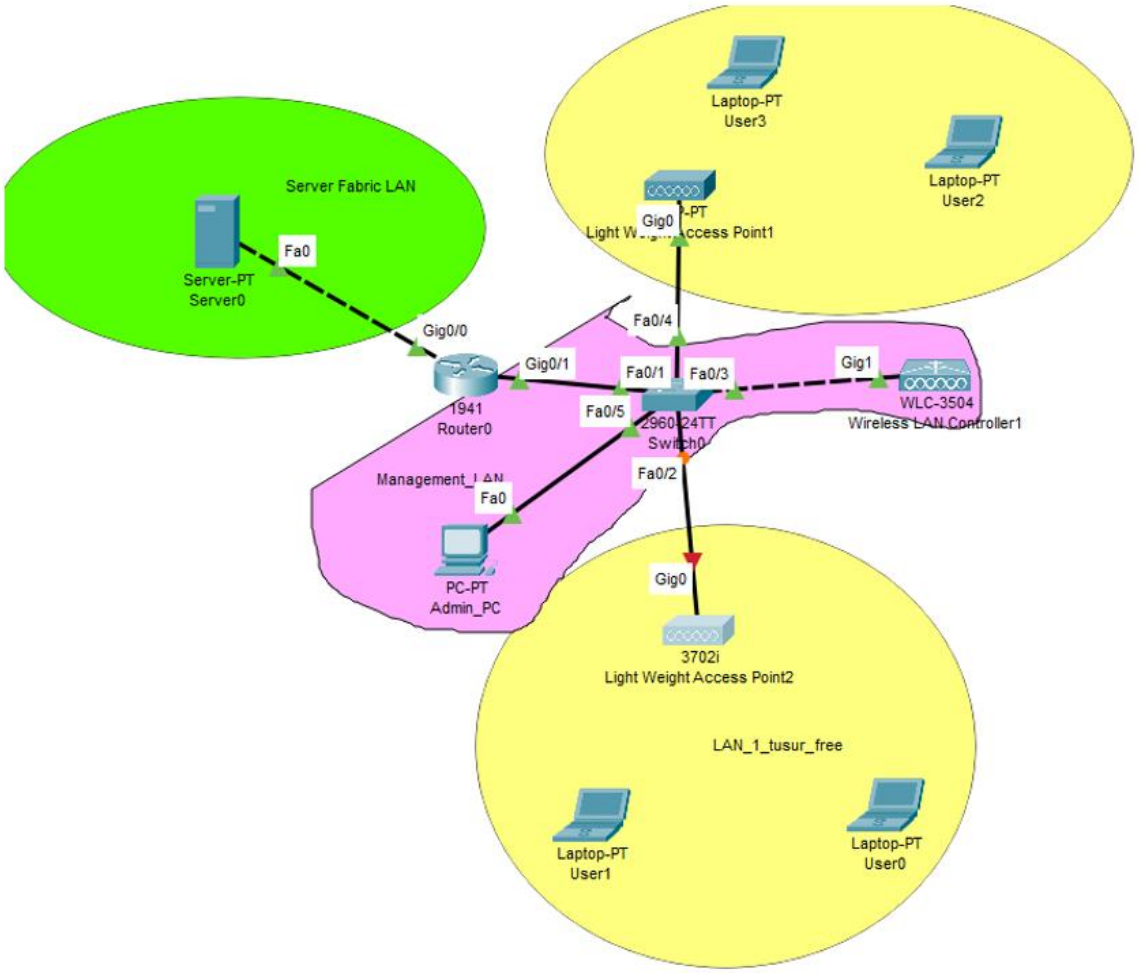

Рисунок 2.1 – Топология практической работы часть 1

Беспроводные устройства находятся в группе сетевого оборудования, рис. 8. В качестве контроллера беспроводной сети возьмите WLC-3504, а в качестве LAP – 3702i. Гигабитный порт контроллера №1 подключите к гигабитному порту коммутатора. Ко второму гигабитному порту подключите компьютер администратора. Это важно!

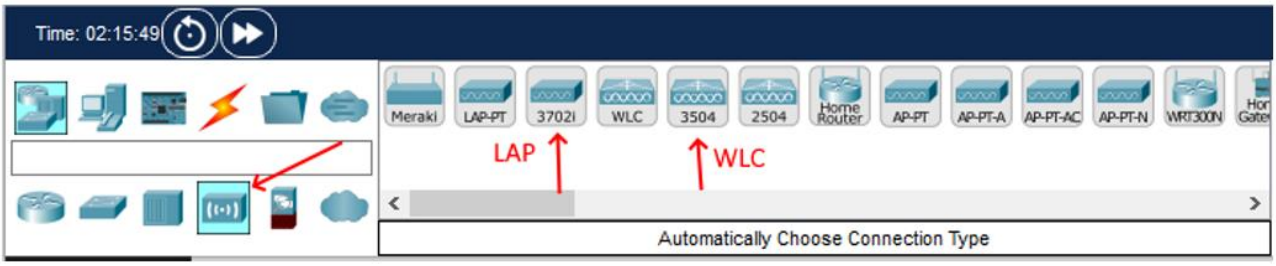

Рисунок 2.2 – Выбор беспроводных устройств

Точка беспроводного доступа не имеет встроенного источника питания и для ее включения нужно подключить внешний источник перетаскиванием его рисунка снизу на точку доступа на вкладке физических свойств, так чтобы разъем шнура подключения попал в разъем питания на устройстве. Подключение источника питания к LAP представлено на рисунке 2.3.

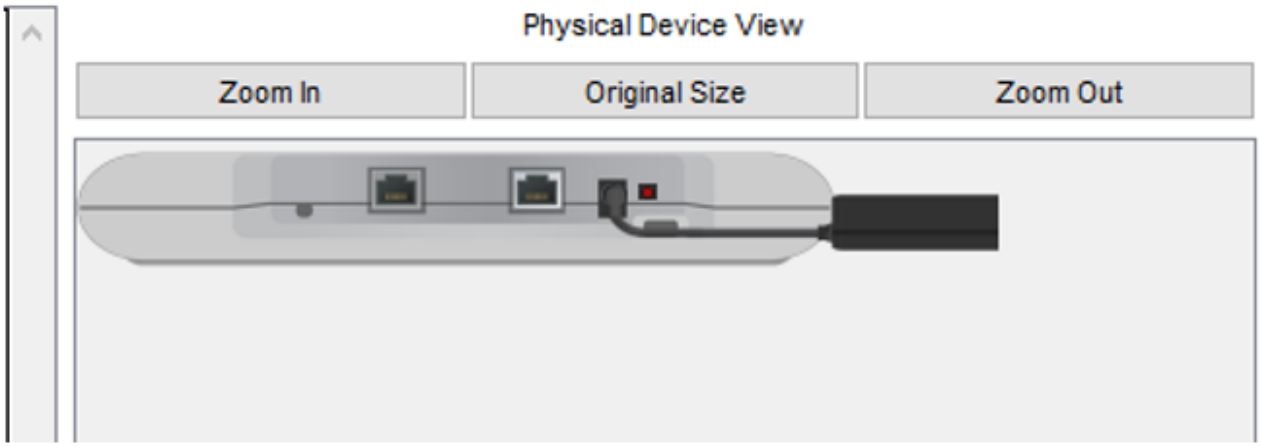

Рисунок 2.3 – Подключение источника питания к LAP

В топологии задействованы три VLAN:

1. VLAN-10 – для пользовательских компьютеров в беспроводной сети;
2. VLAN-20 – то же, что VLAN-10;
3. VLAN-99 – для управления.

Каждой VLAN сопоставляется подсеть на сетевом уровне. Для VLAN10 и VLAN20 это две подсети из вашей практики №1, VLAN-99 использует подсеть 172.16.99.0 /24.

Кроме того, в топологии есть подсеть серверной фермы, представленной одним веб-сервером. В ней используется адресный блок 172.16.0.0 /24.

Настройте сетевые интерфейсы сервера и маршрутизатора для подсети серверной фермы.

Интерфейс маршрутизатора в подсети управления будет выполнять маршрутизацию между всеми VLAN. Создайте на нем соответствующие подынтерфейсы и задайте первый доступный адрес, например, 172.16.99.1 для подынтерфейса, соответствующего VLAN-99.

Также на маршрутизаторе включите DHCP-сервис для абонентов беспроводной сети, подключающихся к точкам LAP в соответствующие VLAN. Это можно сделать командами в глобальной конфигурации маршрутизатора:
ip dhcp pool <вашеФИО>_vlan10
network <ваша сеть №1>
default-gateway <адрес шлюза в вашей сети №1>
dns-server 8.8.8.8 (можно указать как адрес шлюза)
ip dhcp excluded-address <первый и последний адрес диапазона, который вы хотите убрать из заданной сети, например 172.16.99.1 172.16.99.100, адреса начнут выдаваться с 172.16.99.101>. Исключите 50 адресов. На маршрутизаторе должно быть настроено два пула адресов: для VLAN-10 и VLAN-20.

Все интерфейсы на коммутаторе, к которым подключены маршрутизатор, контроллер WLC, точки LAP, компьютер администратора, переведите в режим транка командой:
switchport mode trunk в контексте настройки интерфейса.

По умолчанию контроллер беспроводной сети имеет настройки интерфейса управления, как показано на рис. 10. На нем нет никаких аккаунтов до первого подключения. Настройте интерфейс компьютера администратора (Admin_PC) на рис. 7 в ту же сеть, откройте вкладку Desktop и браузер для подключения к WLC. Введите в адресную строку адрес контроллера, нажмите Enter или кнопку Go. Подождите некоторое время. Можно нажать кнопку ускорения включения интерфейсов (двойная стрелка слева внизу). Быстрое моргание светодиодов на линиях подключения компьютера и контроллера говорит о нормальном процессе. В конце концов откроется окно создания нового аккаунта администратора. Настройки интерфейса управления WLC по умолчанию представлены на рисунке 2.4.

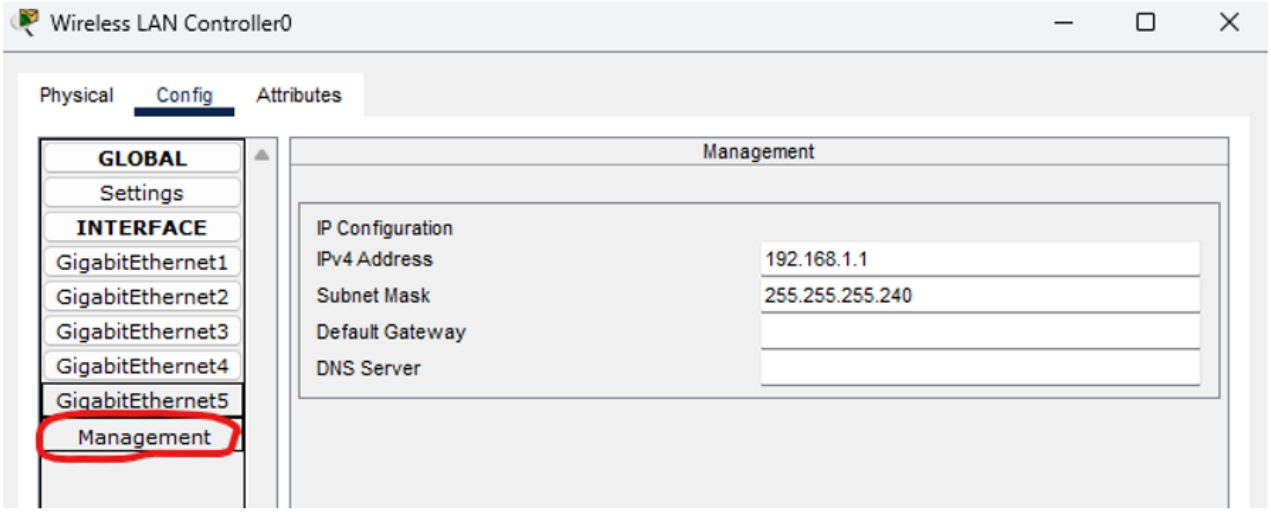

Рисунок 2.4 – Настройки интерфейса управления WLC по умолчанию

Введите в этом окне логин и пароль. Для пароля задана политика: обязательно заглавные и прописные символы, длина не менее восьми, например, логин admin, пароль Tusur123. Окно создания аккаунта администратора WLC представлено на рисунке 2.5.

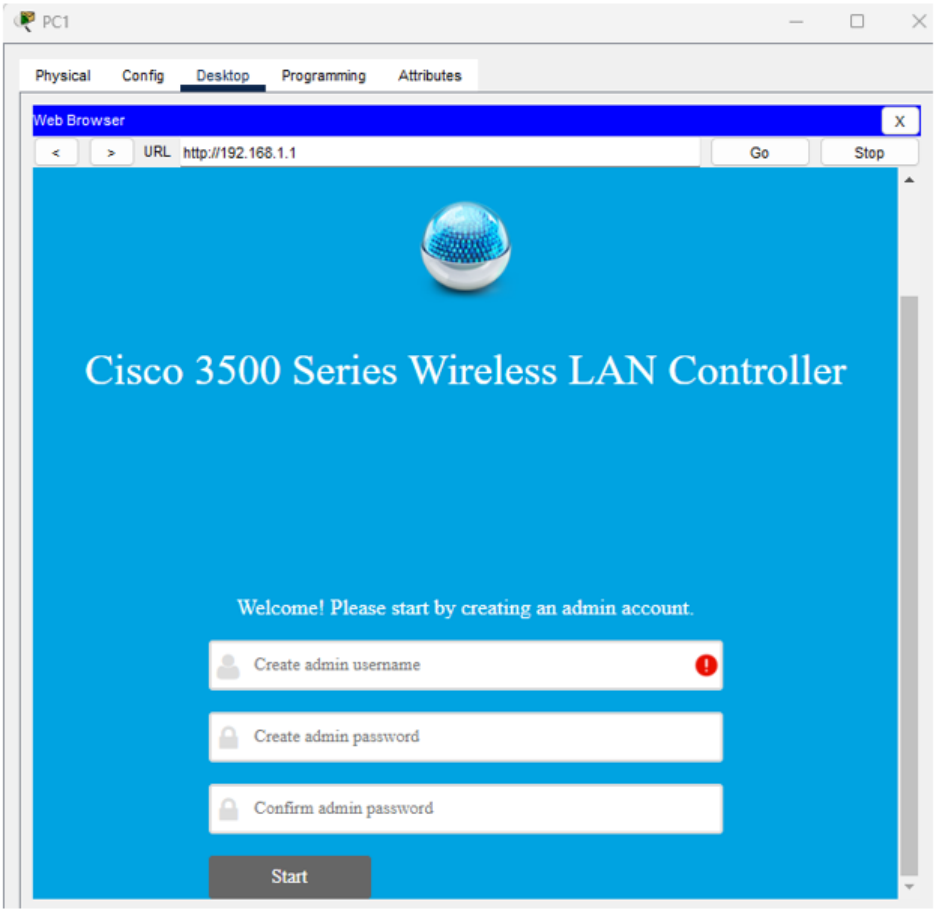

Рисунок 2.5 – Окно создания аккаунта администратора WLC

После нажатия клавиши Start откроется первое окно мастера настройки, как на рисунке 2.6. Здесь необходимо задать имя: поле System Name, оно может быть любым. Адрес

управления. Первоначально использовался адрес 192.168.1.1, но мы определили, что VLAN управления будет 99 и для нее будет использоваться адресное пространство 172.16.99.0/24, поэтому адрес и маску нужно задать соответствующие. То же касается и шлюза по умолчанию. В поле Management VLAN ID тоже следовало бы поставить 99, но для Cisco Packet Tracer здесь должно быть только 1, поэтому замените 0 на 1. Нажмите клавишу Next. Откроется окно следующего шага, как на рисунке 2.7, где необходимо создать первую WLAN, задав для нее имя, уровень защиты и пароль доступа. Задайте имя в виде <ваше ФИО>_vlan10, и пароль, например, 12345678. WPA2 Personal оставьте как есть и нажмите клавишу Next.

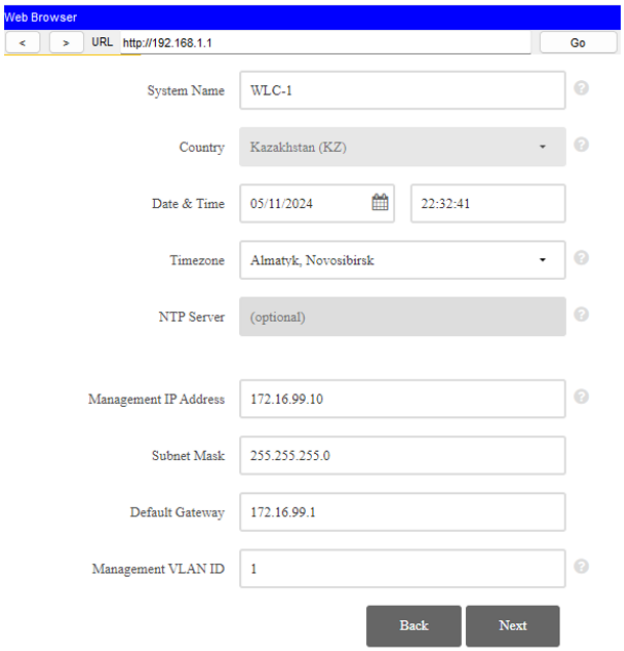

Рисунок 2.6 – Стартовое окно настройки WLC

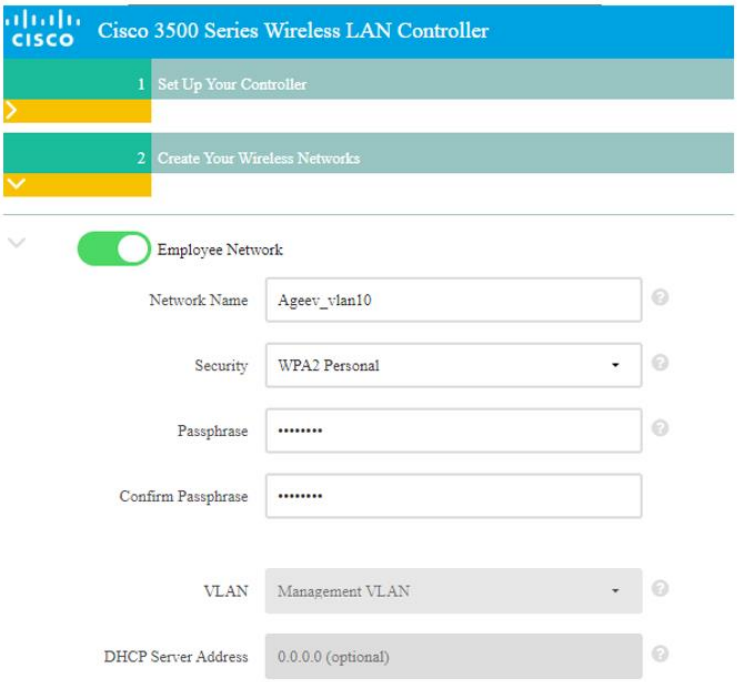

Рисунок 2.7 – Настройка первой беспроводной сети

На следующем и последнем окне мастера настройки не нужно ничего менять, просто нажмите Next. Откроется сводное окно, демонстрирующее все заданные настройки, вы можете

их еще раз проверить и, если нет никаких ошибок, нажмите клавишу Apply. Выскочит еще одно окошко с подтверждением, в котором нужно нажать ОК. После чего настройки применятся, но будет висеть крутящееся колесо с просьбой подождать. Ждать не нужно, нужно закрыть окно браузера, иначе это колесо будет крутиться до бесконечности. Так как мы задали новый адрес управления на WLC, нужно изменить адрес на компьютере администратора, задав его в сети 172.16.99.0/24.

Вновь откройте окно браузера на компьютере и введите заданный при настройке адрес управления, только сначала укажите протокол https в адресной строке, т. к. теперь, когда на контроллере есть аккаунт админа, разрешается только защищенное подключение. Повторное подключение к контроллеру в защищенном режиме представлено на рисунке 2.8.

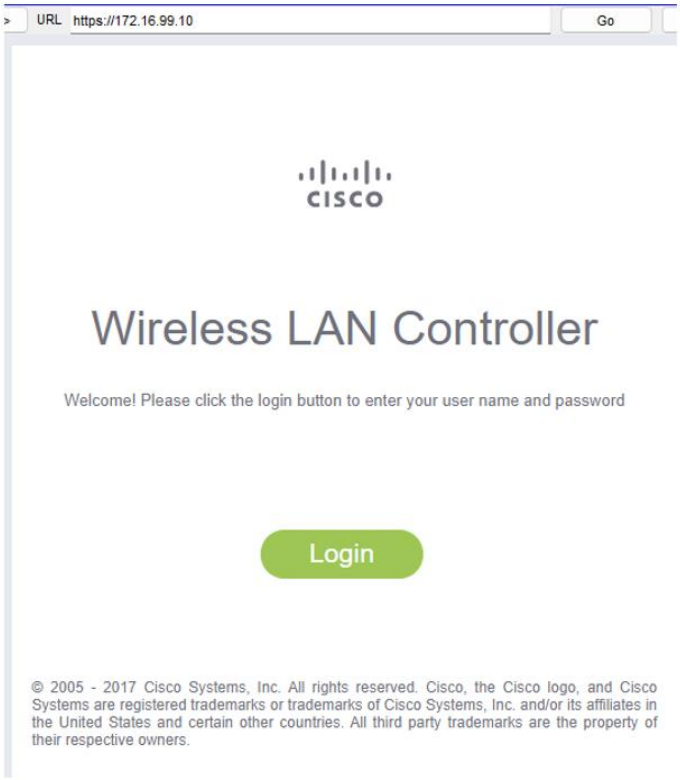

Рисунок 2.8 – Повторное подключение к контроллеру в защищенном режиме

Нажмите кнопку Login и в открывшемся окне введите заданный ранее логин и пароль административного доступа. Откроется окно веб-интерфейса настройки контроллера, как на рисунке 2.9. Мы видим здесь сводную информацию по устройству. Например, отмечается что контроллер поддерживает до 150 точек доступа. Указана версия ПО, загрузка процессора, работа вентилятора, температура устройства и другая системная информация. На странице сводной информации есть информация о подключенных точках доступа. Вы можете видеть, что сейчас их 0. Зайдите в раздел настройки точек доступа и укажите адрес управления WLC, как показано на рисунке 2.10.

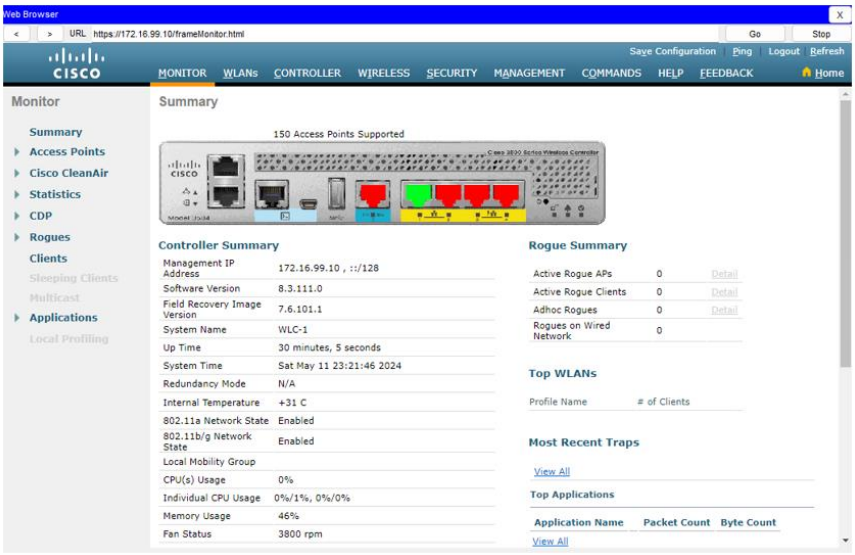

Рисунок 2.9 – Стартовая страница веб-интерфейса WLC

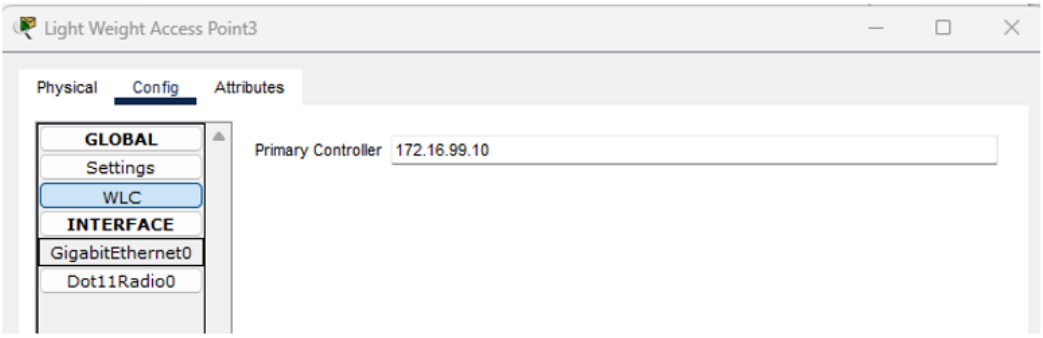

Рисунок 2.10 – Задание адреса WLC на точке доступа

Откройте пункт меню Controller в веб-интерфейсе управления контроллером. Выберите пункт Internal DHCP Server, как показано на рисунке 2.11, выберите пункт DHCP Scope. Удалите дефолтовый диапазон адресов. Создайте новый с адресным пространством сети управления. На рисунке 2.11 настройка адресного диапазона и других параметров DHCP.

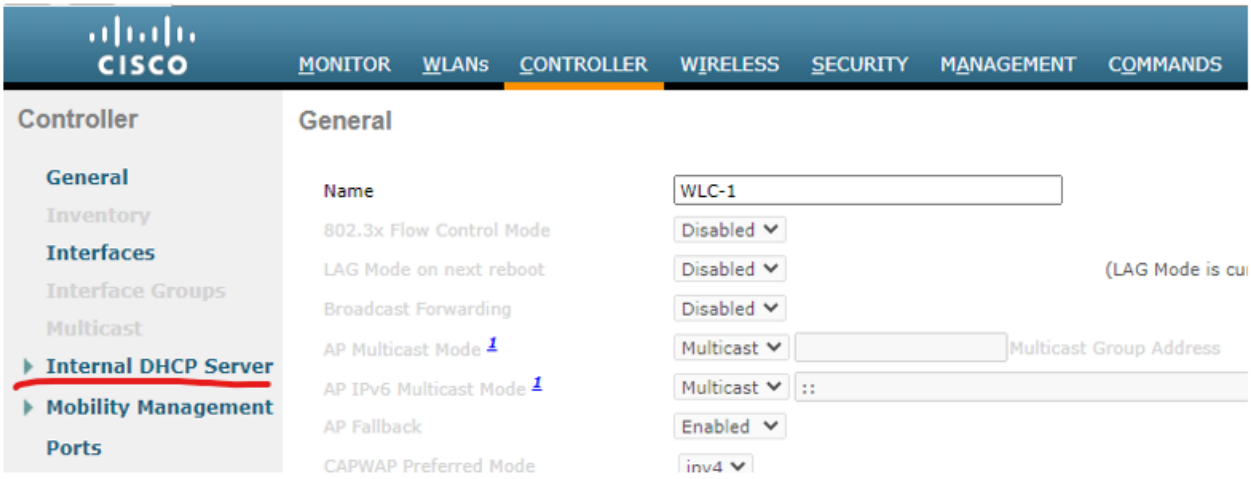

Рисунок 2.11 – Настройка встроенного DHCP-сервера на WLC

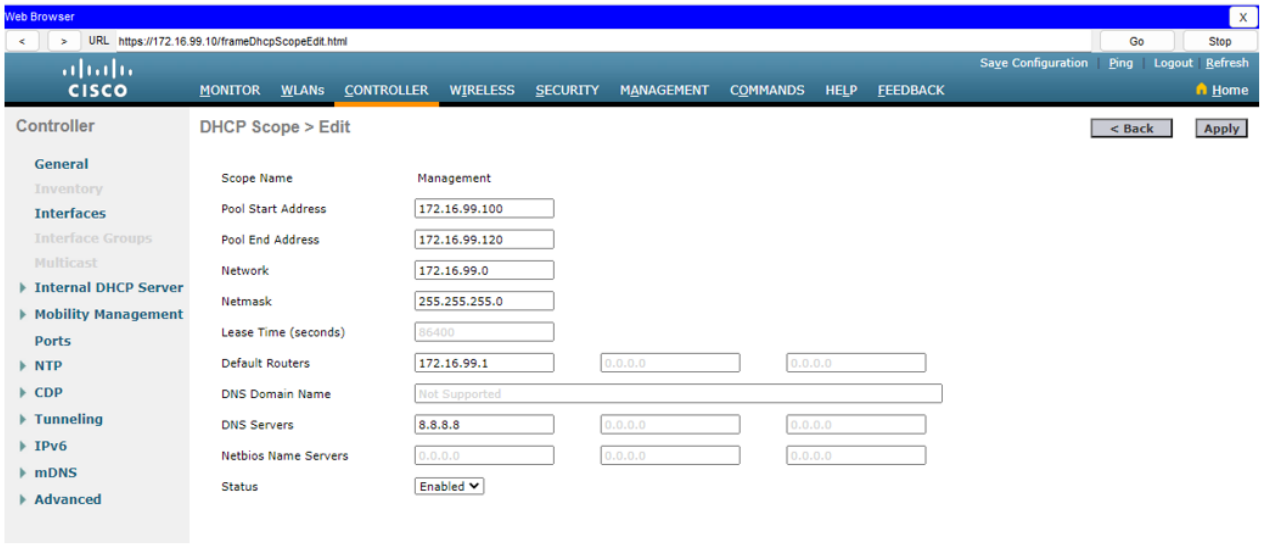

Рисунок 2.12 – Настройка адресного диапазона и других параметров DHCP

Нажмите клавишу Apply в правом верхнем углу. На точках LAP выберите получить конфигурацию IP по DHCP, как это показано на рисунке 2.13.

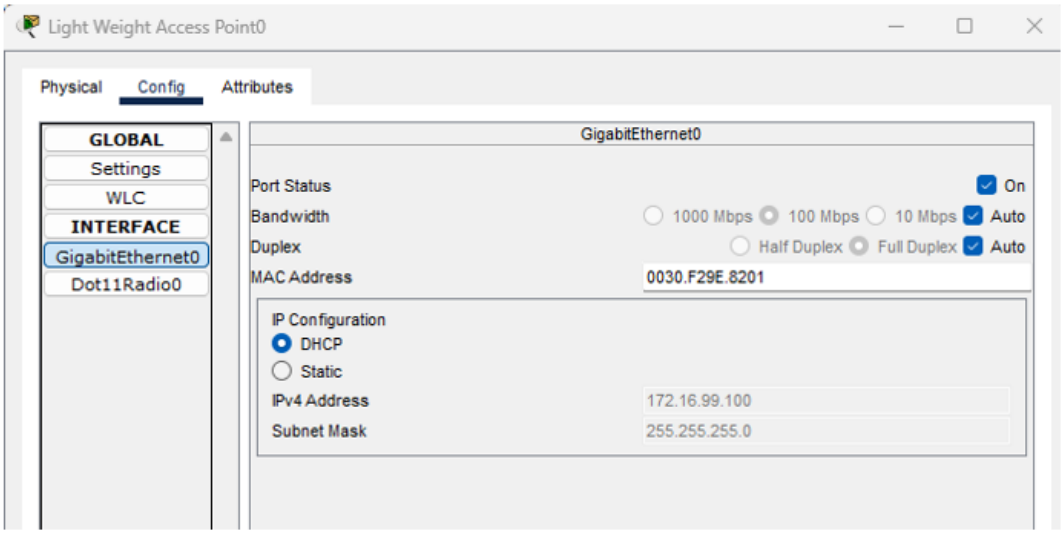

Рисунок 2.13 – Получение адреса на точках доступа от сервера DHCP на WLC

На вкладке Monitor главного меню управления вы теперь видите две подключенные точки, как на рисунке 2.14.

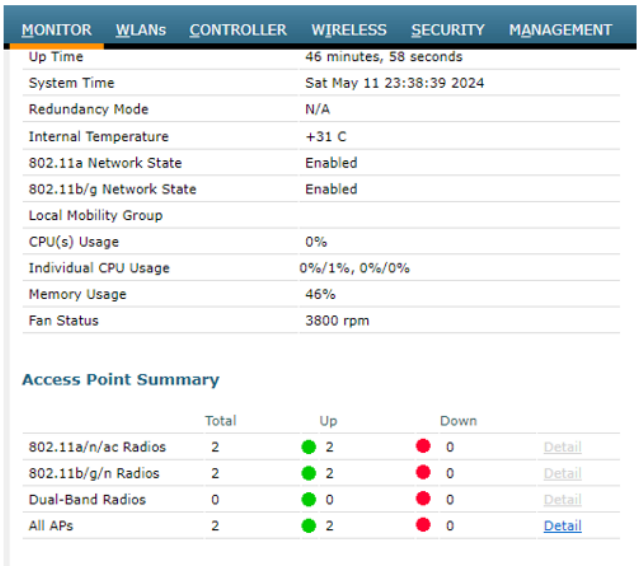

Рисунок 2.14 – В разделе Access Point Summary отображается число подключенных LAP

При переходе по ссылке Detail в этом разделе можно видеть имя и адрес точек доступа, как на рисунке 2.15.

Рисунок 2.15 – Подробная информация о подключенных точках доступа

Во всплывающем окне, появляющемся при наезде мышкой на LAP можно видеть, что между точкой беспроводного доступа и WLC сформирован CAPWAP туннель.

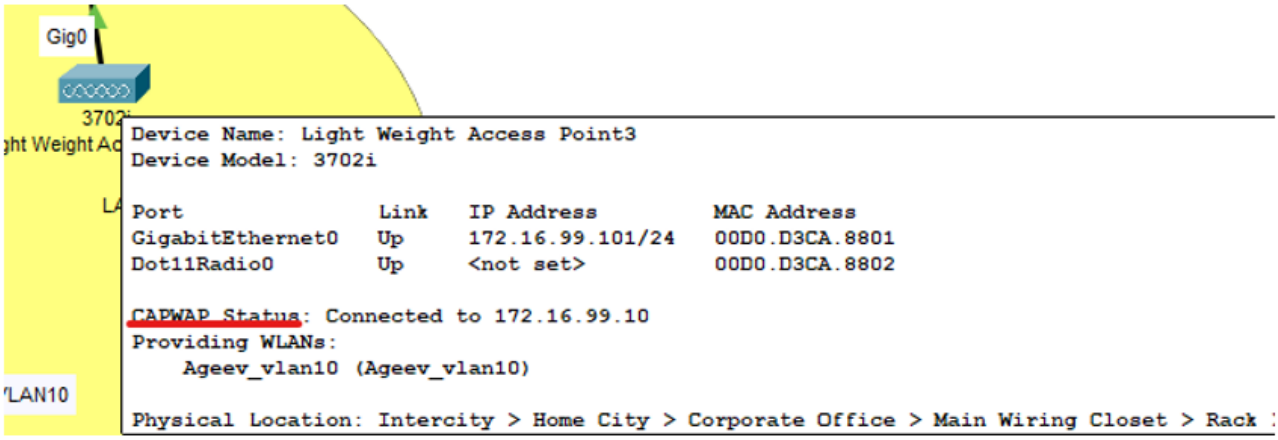

Рисунок 2.16 – Туннельное подключение LAP к WLC

Интерфейсы WLC (Wi-Fi контроллера) — это логические и физические порты, через которые контроллер взаимодействует с другими сетевыми устройствами и управляет беспроводными точками доступа. Основные интерфейсы WLC включают:

1. Интерфейс управления (Management Interface) — это логический интерфейс, через

который контроллер "общается" с внешним миром (кроме точек доступа). Обычно настраивается статический IP-адрес для доступа к веб-интерфейсу и CLI контроллера.

2. Динамические интерфейсы (Dynamic Interfaces) — это VLAN-интерфейсы, которые используются для туннелирования клиентского трафика. Для каждого SSID настраивается свой динамический интерфейс с уникальным VLAN.

3. Портал гостевого доступа (Guest-Anchor) - специальный интерфейс для туннелирования трафика гостевых клиентов. Позволяет сегментировать гостевой трафик.

4. Физические порты Ethernet - используются для подключения контроллера к проводной сети. Например, порт 10/100/1000 Ethernet.

Важное значение имеет виртуальный интерфейс WLC, он используется для служебных функций, например, ретрансляции трафика DHCP, перевода подключающихся клиентов на страничку аутентификации и др. Откройте в меню Controller слева пункт Interfaces. На рисунке 2.17 представлены интерфейсы на WLC.

Создайте динамические интерфейсы для VLAN-10 и 20, чтобы обеспечить связь VLAN и соответствующих им WLAN. Нажмите кнопку New… и задайте имя интерфейса и соответствующую ему VLAN. На рисунке 2.18 представлено создание нового динамического интерфейса.

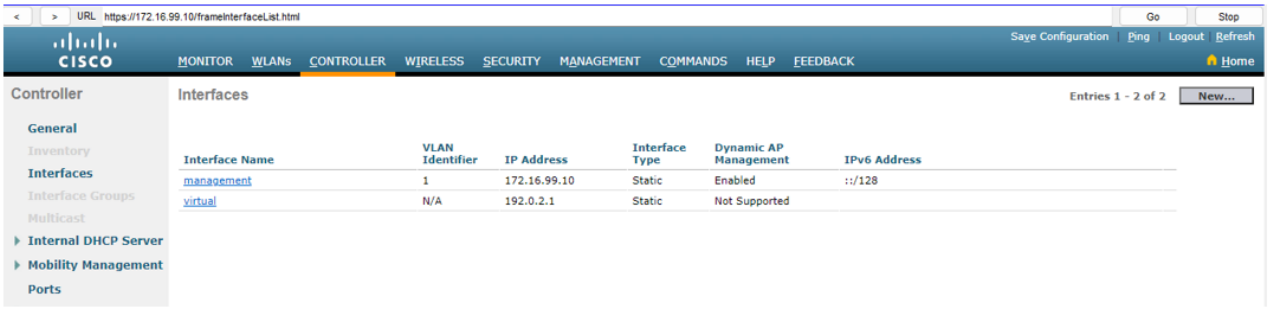

Рисунок 2.17 – Интерфейсы на WLC

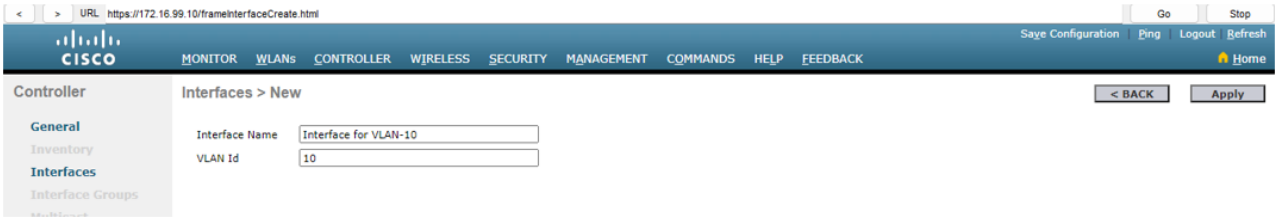

Рисунок 2.18 – Создание нового динамического интерфейса

На следующей страничке, которая открывается после нажатия кнопки Apply для создания интерфейса, нужно добавить конфигурационную информацию, как показано на рисунке 2.19.

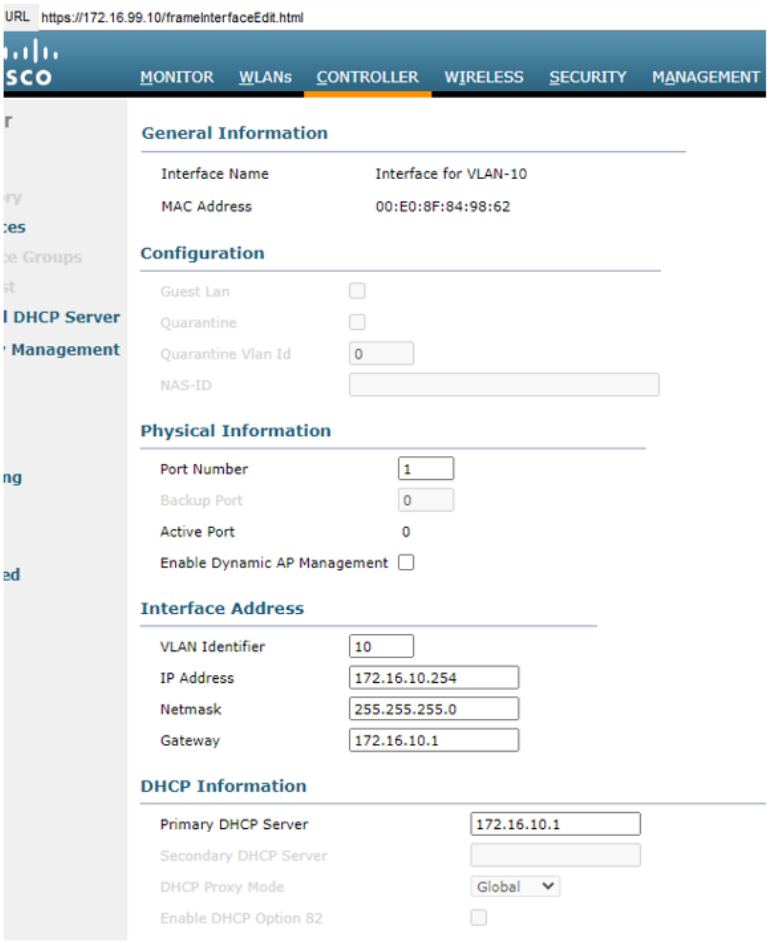

Рисунок 2.19 – Настройка созданного на WLC интерфейса

• Номер порта в разделе физической информации. Это номер физического интерфейса на WLC, с которым будет связан создаваемый динамический интерфейс. Если вы выбрали для подключения WLC первый порт GigabitEthernet, то нужно ввести номер 1.

• Адрес создаваемого интерфейса. Интерфейс будет находится в той подсети, которую вы определили для VLAN-10. У вас это ваша персональная подсеть из практики 1. В примере для наглядности показана подсеть 172.16.10.0/24. Шлюзом в другие сети является подынтерфейс маршрутизатора, который вы создали для VLAN-10.

• Главный DHCP сервер. Здесь мы указываем внешний DHCP сервер, чтобы продемонстрировать возможности использования как внутреннего, так и внешнего сервера DHCP. Этот сервер вы должны были настроить на маршрутизаторе для обоих VLAN. Поэтому указываем соответствующий VLAN интерфейс маршрутизатора.

Нажимаем клавишу Apply и подтверждаем внесение изменений. Аналогичным образом создаем второй интерфейс для VLAN-20. Теперь на вкладке Interfaces должны отображаться созданные интерфейсы, как показано на рисунке 2.20.

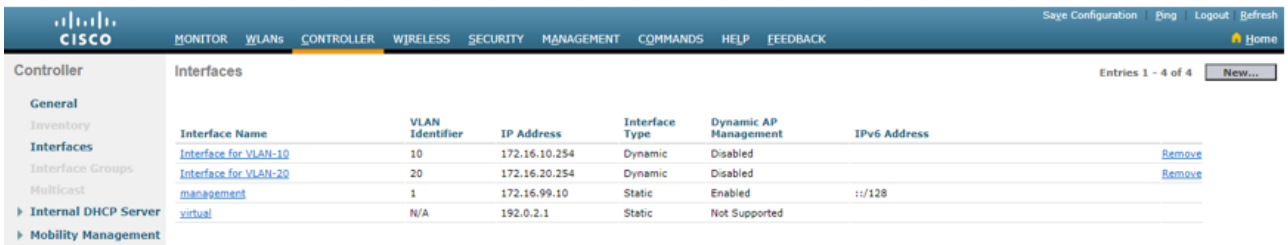

Рисунок 2.20 – Просмотр добавленных интерфейсов

Перейдите в пункт меню WLANs. Здесь отображается первая беспроводная сеть, созданная в мастере первоначальной настройки. На рисунке 2.21 представлен просмотр настроенных на WLC беспроводных сетей.

Рисунок 2.21 – Просмотр настроенных на WLC беспроводных сетей

Перейдите по ссылке с идентификатором первой WLAN в ее настройки и исправьте интерфейс с management на интерфейс для соответствующей VLAN, как показано на рисунке 2.22. Дополнительно, перейдите на вкладку Advanced прокрутите вниз страничку и включите параметры FlexConnect Local Switching и FlexConnect Local Auth. Эти параметры позволяют более эффективно обрабатывать трафик в реальных условиях на реальных сетях. В данном случае это больше акцентирование внимания на этих параметрах для обучающегося и необходимое условие чтобы настроенная в Packet Tracer беспроводная сеть правильно работала.

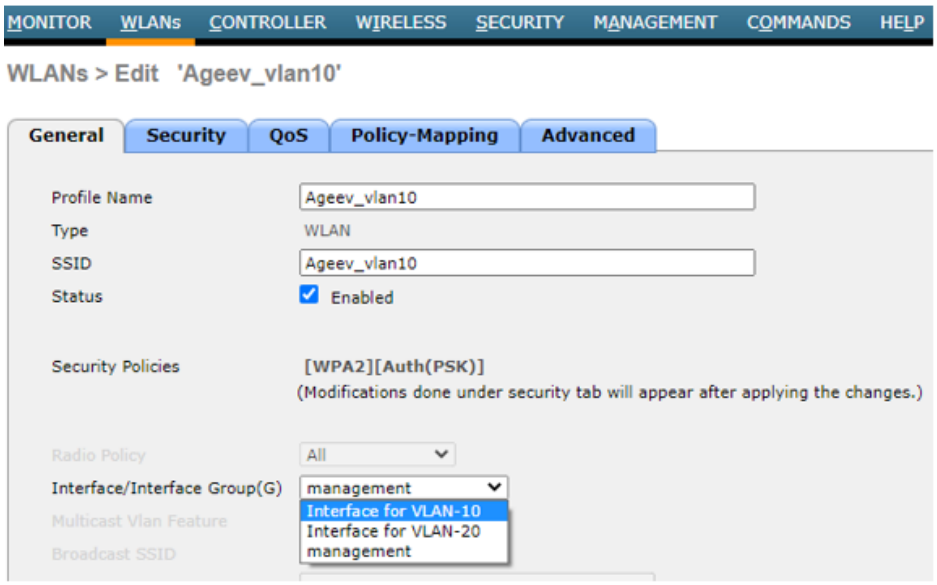

Рисунок 2.22 – Изменение интерфейса для WLAN

Вернитесь в главное меню WLANs и создайте еще одну беспроводную сеть для VLAN-20, нажатием на клавишу «Create New» **Go**. Задайте для нее такие же настройки как для VLAN-10.

На ноутбуках в сетевой топологии замените сетевой адаптер с проводного на беспроводной и откройте приложение PC Wireless на вкладке Desktop. Вид приложения для поиска и подключения к беспроводной сети на рабочем столе компьютера представлен на рисунке 2.23.

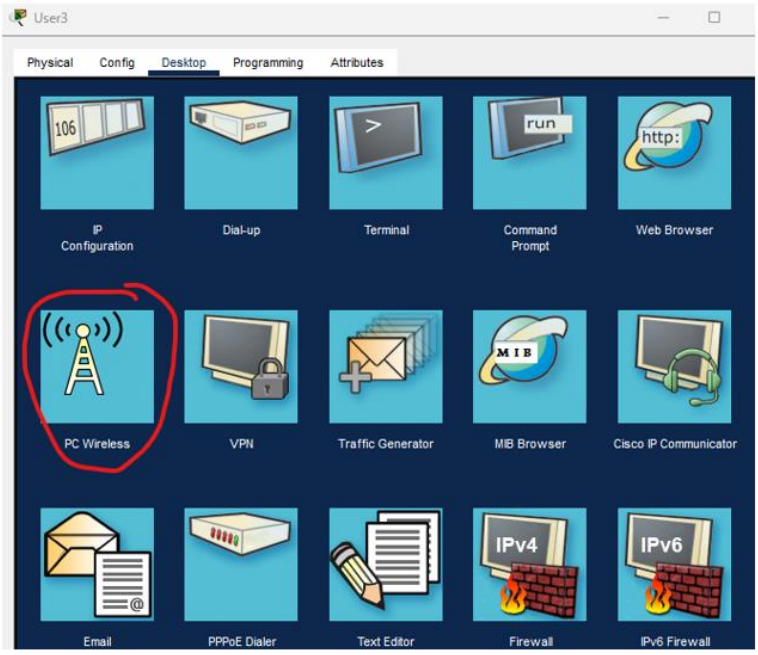

Рисунок 2.23 – Приложение для поиска и подключения к беспроводной сети на рабочем столе компьютера

Перейдите на вкладку Connect и дождитесь появления в окне списка доступных Wi-Fi сетей, подключитесь к сети, соответствующей заданной VLAN. В настройках сетевого адаптера задайте получение конфигурации по DHCP. Какой адрес получил компьютер?

Подключите к беспроводной сети все четыре ноутбука, как показано на рисунке 2.24.

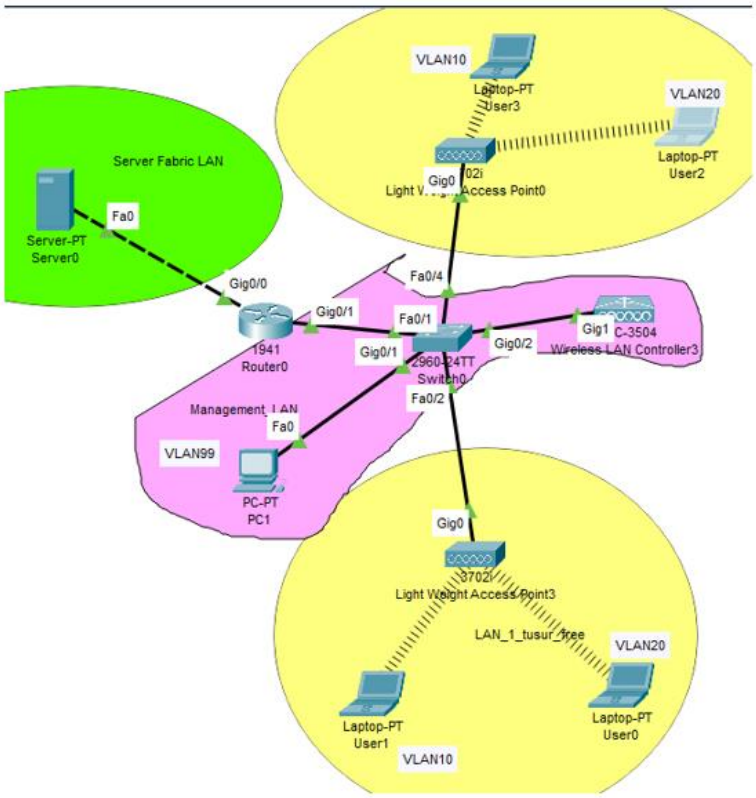

Рисунок 2.24 – Подключение ноутбуков к беспроводной сети

Проверьте что ноутбуки могут обращаться к компьютерам в других VLAN с помощью эхо-запросов и все они могут открыть страничку на веб-сервере. Открытие индексной страницы на веб-сервере представлено на рисунке 2.25.

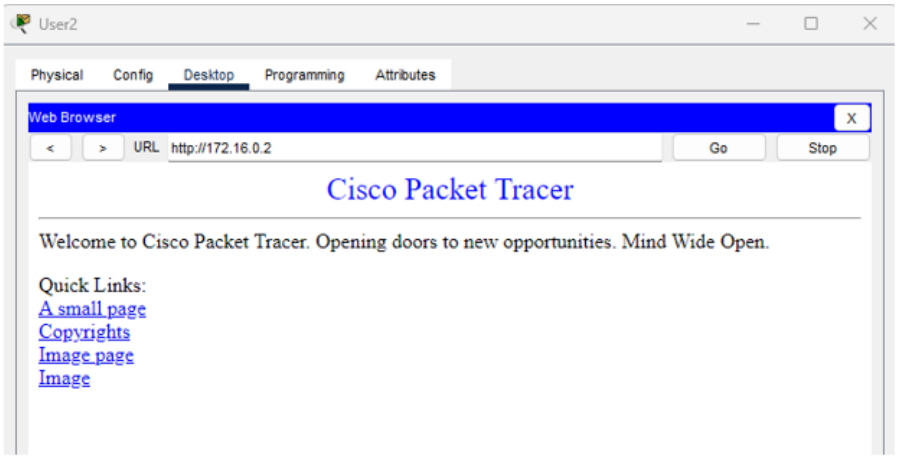

Рисунок 2.25 – Открытие индексной страницы на веб-сервере

Выполненную работу сохраните в файле с вашей фамилией и номером группы и загрузите на проверку. Отчета по работе не требуется, отчетом является файл Cisco Packet Tracer с настроенной топологией.

Практическая работа № 4

«Настройка IPSEC соединения точка_точка»

1. Теоретический материал

VPN основанный на технологии IPsec включается в два этапа. На первом работает протокол Internet Key Exchange (IKE), на втором протоколы IPsec (AH/ESP/both).

Первый этап соединения (IKE) является фазой согласования, во время которой две VPN-точки выбирают какие методы будут использоваться для защиты IP трафика, посылаемого между ними. Помимо этого, IKE также используется для управления соединениями, для этого вводится понятие Security Associations (SA) для каждого соединения. SA направлены только в одну сторону, поэтому типичное IPsec соединение использует два SA.

Вторая часть – это те IP данные, которые необходимо зашифровать и аутентифицировать перед передачей методами, согласованными в первой части (IKE). Существуют разные протоколы IPsec, которые могут быть использованы: AH, ESP или оба.

Последовательность установления VPN через IPsec можно кратко описать как:

- IKE согласовывает защиту уровня IKE;
- IKE согласовывает защиту уровня IPsec;
- защищаемые данные передаются через VPN IPsec.

IKE, Internet Key Exchange

Для шифрования и аутентификации данных требуется выбрать способ шифрования/аутентификации (алгоритм) и ключи, используемые в них. Задача Internet Key Exchange protocol, IKE, в этом случае сводится к распространению данных "ключей сессии" и согласованию алгоритмов, которыми будут защищаться данные между VPN-точками.

Основные задачи IKE:

- Аутентификация VPN-точками друг друга;
- Организация новых IPsec соединений (через создание SA пар);
- Управление текущими соединениями.

IKE ведет учет соединений путем назначения каждому из них некого Security Associations, SA. SA описывает параметры конкретного соединения, включая IPsec протокол (AH/ESP или оба), ключи сессии, используемые для шифрования/дешифрования и/или аутентификации данных. SA является однонаправленной, поэтому используется несколько SA на одно соединение. В большинстве случаев, когда используется только ESP или AH, создаются только две SA для каждого из подключений, одна для входящего трафика, а вторая для исходящего. Когда ESP и AH используются вместе, SA требуется четыре.

Процесс согласования IKE проходит через несколько этапов (фаз). Данные фазы включают:

- IKE первой фазы (IKE Phase-1):

а) Согласовывается защита самого IKE (ISAKMP tunnel).

- IKE второй фазы (IKE Phase-2):

а) Согласовывается защита IPsec;

б) Получение данных из первой фазы для формирования ключей сессии.

Соединения IKE и IPsec ограничены по продолжительности (в секундах) или по количеству переданных данных (в килобайтах). Это сделано для повышения защищенности. После истечения времени безопасная ассоциация согласовывается снова.

Продолжительность IPsec подключения, как правило, короче IKE. Поэтому, когда заканчивается срок IPsec соединения, новое IPsec соединение пересоздается через вторую фазу согласования. Первая фаза согласования используется только при пересоздании IKE подключения.

Для согласования IKE вводится понятие IKE предложение (IKE Proposal) – это предложение того, как защитить данные. VPN-точка инициализирующая IPsec подключение отправляет список (предложение) в котором указаны разные методы защиты подключения. Переговоры могут вестись как об установлении нового IPsec соединения, так и об установлении нового IKE соединения. В случае IPsec защищаемыми данными является тот трафик, что отправлен чрез VPN-туннель, а в случае IKE защищаемые данные – данные самих согласований IKE.

VPN-точка получившая список (предложение), выбирает из него наиболее подходящее и указывает его в ответе. Если ни одно из предложений не может быть выбрано, VPN шлюз отвечает отказом. Предложение содержит всю необходимую информацию для выбора алгоритма шифрования и аутентификации и пр.

IKE первой фазы – согласование защиты IKE (ISAKMP Tunnel)

На первой фазе согласования VPN-точки аутентифицируют друг друга на основе общего ключа (Pre-Shared Key). Для аутентификации используются хэш алгоритм: MD5, SHA-1, SHA-2.

Однако перед тем как аутентифицировать друг друга, чтобы не передавать информацию открытым текстом, VPN-точки выполняют обмен списками предложений (Proposals), описанный ранее. Только после того как устраивающее обеих VPN-точек предложение выбрано, происходит аутентификация VPN-точками друг друга.

Аутентификацию можно осуществлять разными способами: через общие ключи (Pre-Shared Keys), сертификаты или шифрование с открытым ключом. Общие ключи являются наиболее распространенным способом аутентификации.

Согласование IKE первой фазы может происходить в одном из двух режимов: main (основной) и aggressive (агресивный). Основной режим более длительный, но зато и более защищенный. В его процессе происходит обмен шестью сообщениями. Агрессивный режим происходит быстрее, ограничиваясь тремя сообщениями.

Основная работа первой фазы IKE лежит в выработке общего ключа по процедуре Диффи-Хеллмана. Он основан на шифровании с открытым ключом, каждая из сторон шифрует аутентификационный параметр (Pre-Shared Key) открытым ключом соседа, который получив данное сообщение расшифровывает его своим закрытым ключом. Другой способ аутентификации сторон — использование сертификатов.

IKE второй фазы – согласование защиты IPsec.

Во второй фазе осуществляется выбор способа защиты IPsec подключения.

Для работы второй фазы используется материал (keying material), извлеченный из обмена ключами Диффи-Хеллмана (Diffie-Hellman key exchange), произошедшего на первой фазе. На основе этого материала создаются ключи сессии (session keys), использующиеся для защиты данных в VPN-туннеле.

Если используется механизм *Perfect Forwarding Secrecy (PFS)*, то для каждого согласования второй фазы будет использоваться новый обмен ключами Диффи-Хеллмана. Несколько снижая скорость работы, данная процедура гарантирует, что ключи сессии независимы друг от друга, что повышает защиту, поскольку даже если произойдет компроментация одного из ключей, он не сможет быть использован для подбора остальных.

Режим работы второй фазы согласования IKE только один, он называется quick mode — быстрый режим. В процессе согласования второй фазы происходит обмен тремя сообщениями. По окончании второй фазы, устанавливается VPN-подключение, т.е туннель начинает работать, готов передавать пользовательские данные.

Параметры IKE.

Во время установления соединения используются несколько параметров, без согласования которых невозможно установить VPN-подключение.

- Идентификация конечных узлов;

Каким образом узлы аутентифицируют друг друга. Наиболее часто используется общий ключ. Аутентификация, основанная на общем ключе, использует алгоритм Диффи-Хеллмана.

• Локальная и удаленная сеть/хост;

Определяет трафик, который будет пускаться через VPN-туннель.

• Режим туннеля или транспорта.

IPsec может работать в двух режимах: туннельном и транспортном. Выбор режима зависит от защищаемых объектов.

Туннельный режим применяется для защиты между удаленными объектами, т.е. IP-пакет полностью инкапсулируется в новый и для наблюдателя со стороны будет видно только соединение между двумя VPN-точками. Реальные IP-адреса источника и получателя будут видны только после декапсуляции пакета при приеме его на VPN-точке получения. Таким образом туннельный режим чаще всего используется для VPN-подключений.

Транспортный режим защищает данные IP-пакета (TCP, UDP и протоколы верхних уровней), а сам заголовок оригинального IP-пакета будет сохранен. Таким образом для наблюдателя будет виден оригинальный источник и назначение, но не передаваемые данные. Данный режим наиболее часто используется при защите соединение в локальной сети между хостами.

• Удаленный шлюз;

VPN-точка получатель защищенного соединения, которая будет расшифровывать/аутентифицировать данные с другой стороны и отправлять их к окончательному месту назначения.

• Режим работы IKE;

IKE согласование может работать в двух режимах: *основной* и *агрессивном*. Разница между ними заключается в том, что в агрессивном режиме используется меньшее кол-во пакетом, что позволяет достичь более быстрого установления соединения. С другой стороны, агрессивный режим не передает некоторые параметры согласования, такие как Диффи-Хеллман группы и PFS, что требует предварительной идентичной настройки их на точках участницах подключения.

• IPsec протоколы;

Существует два протокола IPsec: Authentication Header (AH) и Encapsulating Security Payload (ESP), которые выполняют функции шифрования и аутентификации. ESP позволяет шифровать, аутентифицировать по отдельности или одновременно. AH позволяет только аутентифицировать. Разница с ESP аутентификацией в том, что AH аутентифицирует также и внешний IP заголовок, позволяя подтвердить, что пакет прибыл действительно от источника указанного в нем.

• IKE шифрование;

Указывает используемый алгоритм шифрования IKE и его ключи. Поддерживаются разные симметричные алгоритмы шифрования, например: DES, 3DES, AES.

• IKE аутентификация;

Алгоритм аутентификации используемый в IKE согласовании. Могут быть: SHA, MD5.

• IKE Диффи-Хеллмана (DH) группы;

Используемая DF группа для обмена ключами в IKE. Чем больше группа, тем больше размер ключей обмена.

• Продолжительность жизни IKE подключения;

Указывается как по времени (секундах), так и по размеру переданных данных (килобайтах). Как только один из счетчиков достигнет порогового значения запускается новая первая фаза. Если с момента создания IKE соединения не было передано никаких данных, никаких новых подключений не будет создано до тех пор, пока одна из сторон не захочет создать VPN соединение.

• PFS;

При отключенном PFS материал для создания ключей будет извлечен в первой фазе согласования IKE в момент обмена ключей. Во второй фазе согласования IKE ключи сессии будут созданы основываясь на полученном материале. При включенном PFS при создании новых ключей сессии материал для них будет использоваться каждый раз новый. Таким образом при компромате ключа, на основе него невозможно создать новые ключи. PFS может быть использован в двух режимах: первый PFS на ключах (PFS on keys), будет запускать новый обмен ключами в первой фазе IKE каждый раз, когда запускается согласование второй фазы. Второй режим PFS на идентификаторах (PFS on identities), будет удалять SA первой фазы каждый раз, после прохождения согласования второй фазы, гарантируя тем самым, что ни одно согласование второй фазы не будет зашифровано идентичным предыдущему ключом.

• IPsec DH группы;

Данные DF группы аналогичны использующимся в IKE, только используются для PFS.

• IPsec шифрование;

Алгоритм, использующийся для шифрования данных. Используется в случае использования ESP в режиме шифрования. Пример алгоритмов: DES, 3DES, AES.

• IPsec аутентификация;

Алгоритм, используемый для аутентификации передаваемых данных. Используется в случае AH или ESP в режиме аутентификации. Пример алгоритмов: SHA, MD5.

• Время жизни IPsec;

Время жизни VPN соединения указывается как по времени (секундах) так и по размеру переданных данных (килобайты). Счетчик первым достигнувший лимита запустит пересоздание ключей сессии. Если с момента создания IKE соединения не было передано никаких данных, никаких новых подключений не будет создано до тех пор, пока одна из сторон не захочет создать VPN соединение.

Методы аутентификации IKE:

1. Ручной режим;

Самый простой из методов, при котором IKE не используется, а ключи аутентификации и шифрования, а также некоторые другие параметры задаются вручную на обеих точках VPN подключения.

2. Через общие ключи (Pre-Shared Keys, PSK);

Заранее введенный общий ключ на обоих точках VPN соединения. Отличие от предыдущего метода в том, что используется IKE, что позволяет аутентифицировать конечные точки и использовать меняющиеся ключи сессии, вместо фиксированных ключей шифрования.

3. Сертификаты.

Каждая точка VPN использует: свой приватный ключ, свой открытый ключ, свой сертификат включающий свой открытый ключ и подписанный доверенным центром сертификации. В отличие от предыдущего метода позволяет избежать ввода одного общего ключа на всех точках VPN соединения, заменяя его личными сертификатами, подписанными доверенным центром.

Протоколы IPsec/

IPsec протоколы используются для защиты передаваемых данных. Выбор протокола и его ключей происходит при согласовании IKE.

AH (Authentication Header)/

AH предоставляет возможно аутентифицировать передаваемые данные. Для этого используется криптографическая хэш-функция по отношению к данным содержащимся в IP-пакете. Вывод данной функции (хэш) передается вместе с пакетом и позволяет удаленной VPN точке подтвердить целостность оригинального IP-пакета, подтверждая, что он не был изменен по пути. Помимо данных IP-пакета, AH также аутентифицирует часть его заголовка.

В режиме транспорта, AH встраивает свой заголовок после оригинального IP пакета.

В режиме туннеля AH встраивает свой заголовок после внешнего (нового) IP-заголовка и перед внутренним (оригинальным) IP заголовком.

ESP (Encapsulating Security Payload).

ESP протокол используется для шифрования, для аутентификации или и того, и другого по отношению к IP пакету.

В режиме транспорта ESP протокол вставляет свой заголовок после оригинально IP заголовка.

В режиме туннеля ESP заголовок находится после внешнего (нового) IP заголовка и перед внутренним (оригинальным).

Два основных различия между ESP и AH:

- ESP помимо аутентификации предоставляет еще возможность шифрования (AH этого не предоставляет)
- ESP в режиме туннеля аутентифицирует только оригинальный IP заголовок (AH аутентифицирует также внешний).

2. Ход выполнения работы

Работа выполняется в программе Cisco Packet Tracer версии 6.2 или выше. Соберите топологию сети, как показано на рисунке 2.1. В качестве маршрутизаторов используйте устройства 1941. Для организации последовательных соединений добавьте в них дополнительный модуль HWIC-2T на вкладке «физический вид» из группы модулей в левом столбце.

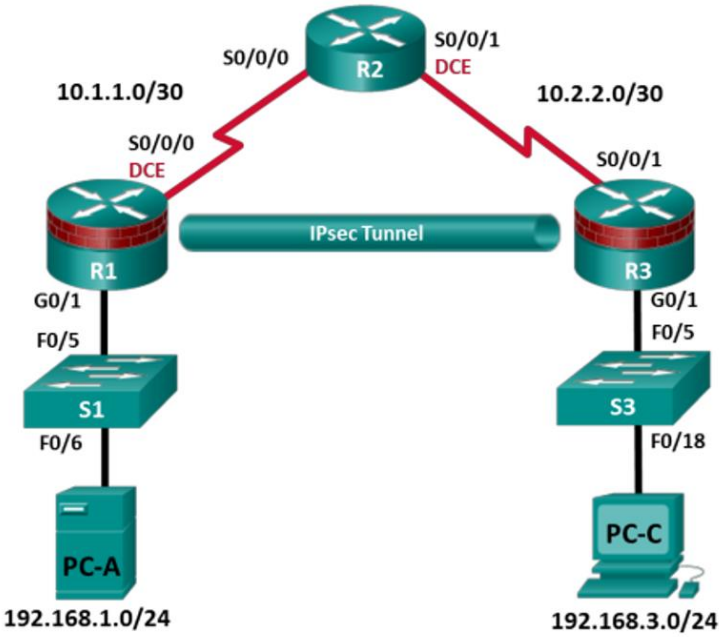

Рисунок 2.1 – Топология сети

Настройте IP-адреса на интерфейсах маршрутизаторов и на компьютерах, согласно таблице адресов, показанной ниже. Учитывайте, что названия интерфейсов в вашей топологии могут не совпадать с названиями в таблице. Делайте соответствующую коррекцию.

Таблица 2.1 – Таблица IP-адресов

Устройство	Интерфейс	1P-адрес	Маска подсети	Шлюз по умолчанию	Порт коммутатора
R1	G0/1	192.168.1.1	255.255.255.0	Н/П	S1 F0/5
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	Н/П	Н/П
R2	S0/0/0	10.1.1.2	255.255.255.252	Н/П	Н/П
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	Н/П	Н/П
R3	G0/1	192.168.3.1	255.255.255.0	Н/П	S3 F0/5
	S0/0/1	10.2.2.1	255.255.255.252	Н/П	Н/П
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1	S1 F0/6
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1	S3 F0/18

Для получения полной связности, настройте протокол OSPF на маршрутизаторах R1, R2 и R3. На маршрутизаторе R1 используйте следующие команды:

```
R1(config)# router ospf 1
R1(config-router)# network 192.168.1.0 0.0.0.255 area 0
R1(config-router)# network 10.1.1.0 0.0.0.3 area 0
```

На маршрутизаторе R2 используйте следующие команды:

```
R2(config)# router ospf 1
R2(config-router)# network 10.1.1.0 0.0.0.3 area 0
R2(config-router)# network 10.2.2.0 0.0.0.3 area 0
```

На маршрутизаторе R3 используйте следующие команды.

```
R3(config)# router ospf 1
R3(config-router)# network 192.168.3.0 0.0.0.255 area 0
R3(config-router)# network 10.2.2.0 0.0.0.3 area 0
```

После настройки протокола убедитесь, что он работает. Выведите таблицу маршрутизации командой:

```
show ip route
```

в контексте администратора. Проверьте базовую связь по сети. Отправьте эхо-запрос с маршрутизатора R1 на интерфейс Fa0/1 маршрутизатора R3 по IP-адресу 192.168.3.1. Если запрос завершается с ошибкой, измените значения основных параметров устройства перед тем, как продолжить работу.

Отправьте эхо-запрос с компьютера PC-A в локальной сети маршрутизатора R1 на компьютер PC-C в локальной сети маршрутизатора R3.

Если запрос завершается с ошибкой, измените значения основных параметров устройства перед тем, как продолжить работу.

Если эхо-запрос с компьютера PC-A на компьютер PC-C выполнен успешно, это означает, что протокол маршрутизации OSPF настроен правильно и работает корректно.

Включение на маршрутизаторе Cisco программного пакета Security Technology, реализующего криптографические функции.

1. На маршрутизаторе R1 введите команду show version для просмотра сведений о лицензии Security Technology Package.

2. Если пакет Security Technology не активирован, сделайте это с помощью следующей команды:

```
R1(config)# license boot module c1900 technology-package securityk9
```

3. Примите условия лицензионного соглашения с конечным пользователем.

4. Сохраните текущую конфигурацию командой copy run start и перезагрузите маршрутизатор командой reload, чтобы активировать лицензию Technology Package.

5. Убедитесь, что пакет Security Technology включен, с помощью команды show version.

Повторите эти действия на всех маршрутизаторах.

Включите политики распределения ключей IKE на маршрутизаторах R1 и R3.

IPsec – это открытая платформа, поддерживающая обмен протоколами безопасности по мере появления новых технологий и алгоритмов шифрования. В процессе реализации сети IPsec VPN особое значение играют две операции конфигурирования:

1) Настройка параметров протокола обмена ключами безопасности Internet Key Exchange (IKE), это фаза 1 установления соединения IPsec.

2) Настройка параметров IPsec, фаза 2 соединения.

Проверка того, что протокол IKE поддерживается и включен.

Фаза 1 IKE определяет метод обмена ключами, используемый для передачи политик IKE между узлами и проверки этих политик. На фазе 2 IKE узлы обмениваются и сопоставляют политики IPsec для аутентификации и шифрования передаваемых данных.

Чтобы IPsec работал, необходимо сначала включить протокол IKE. IKE по умолчанию включен в образах IOS с наборами криптографических функций. Если этот протокол выключен, его можно включить с помощью команды: crypto isakmp enable.

Эта команда позволяет также проверить, что операционная система маршрутизатора поддерживает IKE и что этот протокол включен. Выполните

```
R1(config)# crypto isakmp enable
R3(config)# crypto isakmp enable
```

Если вы не можете выполнить эту команду на маршрутизаторе, необходимо обновить образ IOS до версии, которая содержит криптографические сервисы Cisco.

Установите политику ISAKMP и ознакомьтесь с доступными опциями.

Для обеспечения согласования на фазе 1 IKE необходимо создать политику ISAKMP и настроить ассоциацию узлов, применяющую эту политику. Политика ISAKMP определяет алгоритмы аутентификации и шифрования, а также хеш-функцию, используемую для отправки управляющего трафика между двумя конечными устройствами VPN. Как только ассоциация безопасности ISAKMP будет принята узлами IKE, фаза 1 IKE будет завершена.

Введите в режиме глобальной настройки команду crypto isakmp policy number на маршрутизаторе R1 для политики 10.

```
R1(config)# crypto isakmp policy 10
```

Степень конфиденциальности канала управления между двумя конечными устройствами определяется алгоритмом шифрования. Хеш-алгоритм контролирует целостность данных, то есть проверяет, что данные, полученные из узла, не были несанкционированно изменены при пересылке. Тип аутентификации гарантирует, что пакет был отправлен и подписан на удаленном узле. Для создания секретного ключа, используемого совместно узлами, но не пересылаемого по сети, используется группа (протокол) Диффи-Хеллмана (Diffie-Hellman).

Мы задали для политики ISAKMP приоритет 10. Используйте тип аутентификации «предварительно введенные ключи» pre-shared key, алгоритм шифрования aes 256, алгоритм хеширования sha и группу Diffie-Hellman 14 для обмена ключами. Установите время действия политики на 3600 секунд (один час).

Старые версии Cisco IOS не поддерживают шифрование AES 256 и SHA в качестве хеш-алгоритма.

Замените указанные алгоритмы шифрования и хеширования на любые, поддерживаемые вашим маршрутизатором.

```
R1(config-isakmp)# hash sha
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# group 1
R1(config-isakmp)# lifetime 3600
R1(config-isakmp)# encryption aes 256
R1(config-isakmp)# end
```

Убедитесь также, что аналогичные изменения были внесены на маршрутизаторе R3.

```
R3(config)# crypto isakmp policy 10
R3(config-isakmp)# hash sha
R3(config-isakmp)# authentication pre-share
R3(config-isakmp)# group 1
R3(config-isakmp)# lifetime 3600
R3(config-isakmp)# encryption aes 256
R3(config-isakmp)# end
```

Проверьте политику IKE с помощью команды show crypto isakmp policy.

```
R1# show crypto isakmp policy
Global IKE policy
Protection suite of priority 10
encryption algorithm:AES - Advanced Encryption Standard (256 bit keys).
hash algorithm:Secure Hash Standard
authentication method:Pre-Shared Key
Diffie-Hellman group:#1 (768 bit)
lifetime:3600 seconds, no volume limit
```

Настройте общие ключи.

На каждом маршрутизаторе, подключенном к другому конечному устройству в сети VPN, должен быть сконфигурирован ключ, так как в качестве метода аутентификации в политике IKE применяются общие ключи. Для успешной аутентификации эти ключи должны совпадать. Для ввода общего ключа применяется команда:

crypto isakmp key <key-string> address <ip-address>

в режиме глобальной настройки. Это должен быть IP-адрес удаленного интерфейса, который узел будет использовать для маршрутизации трафика на локальный маршрутизатор.

В качестве IP-адреса удаленного устройства в VPN также можно применять каждый IP-адрес, используемый для настройки узлов IKE. Настройте общий ключ TusurPish2023 на маршрутизаторе R1. Эта команда указывает на IP-адрес интерфейса S0/0/1 удаленного маршрутизатора R3.

```
R1(config)# crypto isakmp key TusurPish2023 address 10.2.2.1
```

Настройте общий ключ TusurPish2023 на маршрутизаторе R3. Данная команда относится к маршрутизатору R3 и указывает на IP-адрес интерфейса S0/0/0 маршрутизатора R1.

```
R3(config)# crypto isakmp key TusurPish2023 address 10.1.1.1
```

Настройте набор преобразований и время жизни IPsec.

Набор преобразований Ipsec – это еще один криптографический параметр, который маршрутизаторы согласуют друг с другом для создания ассоциации безопасности. Для создания набора преобразований IPsec используйте команду crypto ipsec transform-set <tag>. Для просмотра доступных параметров используйте ?.

```
R1(config)# crypto ipsec transform-set 50 ?
```

На маршрутизаторах R1 и R3 создайте набор преобразований с тегом 50, используйте преобразование ESP с шифрованием AES 256 и протоколом ESP и хеш-функцией SHA. Наборы преобразований должны совпадать.

```
R1(config)# crypto ipsec transform-set 50 esp-aes 256 esp-sha-hmac
R1(cfg-crypto-trans)# exit
R3(config)# crypto ipsec transform-set 50 esp-aes 256 esp-sha-hmac
R3(cfg-crypto-trans)# exit
```

Какую функцию выполняет набор преобразований IPsec?

Вы можете изменить время жизни ассоциации безопасности IPSec и не использовать значение по умолчанию 3600 секунд. На маршрутизаторах R1 и R3 установите время жизни ассоциации безопасности 30 минут или 1800 секунд.

```
R1(config)# crypto ipsec security-association lifetime seconds 1800
R3(config)# crypto ipsec security-association lifetime seconds 1800
```

Определите «интересный» трафик.

Чтобы использовать шифрование IPsec в сети VPN, необходимо задать расширенные списки доступа, с помощью которых маршрутизатор сможет понимать, какой трафик следует шифровать. Если сеанс IPsec настроен правильно, то пакет, разрешаемый в списке доступа, который применяется для определения трафика IPsec, будет шифроваться. Пакет, отклоняемый одним из таких списков доступа, не отбрасывается, а отправляется незашифрованным. Так же, как и в любом другом списке доступа, в конце имеется оператор неявного отклонения deny any. Это означает, что по умолчанию для трафика шифрование не выполняется. Если ассоциация безопасности IPsec сконфигурирована неправильно, трафик не шифруется и передается в нешифрованном виде.

В нашем сценарии с позиции маршрутизатора R1 мы хотим зашифровать трафик, поступающий из локальной сети Ethernet маршрутизатора R1 в локальную сеть Ethernet маршрутизатора R3, или наоборот, если смотреть со стороны маршрутизатора R3. Данные списки доступа используются для исходящего трафика на интерфейсах устройств VPN и должны зеркально отражать друг друга.

Сконфигурируйте список ACL для «интересного» трафика в IPSec VPN на маршрутизаторе R1.

```
R1(config)# access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255
```

Сконфигурируйте симметричный список ACL для «интересного» трафика в IPSec VPN на маршрутизаторе R3.

```
R3(config)# access-list 101 permit ip 192.168.3.0 0.0.0.255 192.168.1.0 0.0.0.255
```

Проверяет ли IPsec наличие зеркальных списков доступа как необходимое условие для согласования ассоциации безопасности?

Создайте и примените криптографическую карту.

Криптографическая карта ассоциирует трафик, соответствующий списку доступа, с узлом и различными параметрами IKE и IPsec. После создания криптографической карты ее можно применить к одному или нескольким интерфейсам. Интерфейсы, к которым такая карта применяется, должны быть подключены к узлу IPsec.

Для создания криптографической карты используйте команду:

```
crypto map <name> <sequence-num> <type>
```

в режиме глобальной настройки, чтобы войти в режим настройки криптографической карты для заданного порядкового номера. В одной криптографической карте может быть несколько криптографических операторов, которые анализируются в порядке возрастания номеров. Войдите в режим настройки криптографический карты на маршрутизаторе R1. Используйте тип ipsec-isakmp, чтобы указать, что для установления ассоциаций безопасности IPsec будет применяться IKE.

Создайте криптографическую карту на маршрутизаторе R1, назовите ее PishMAP и укажите 10 в качестве порядкового номера. После ввода команды будет выведено сообщение.

```
R1(config)# crypto map PishMAP 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer and a valid access list have been configured.
```

Используйте команду match address <access-list> для указания списка доступа, определяющего трафик, который нужно шифровать.

```
R1(config-crypto-map)# match address 101
```

Требуется указать IP-адрес или имя хоста для узла. Укажите интерфейс конечного устройства VPN маршрутизатора R3 с помощью следующей команды.

```
R1(config-crypto-map)# set peer 10.2.2.1
```

Используйте команду set transform-set <tag>, чтобы четко указать набор преобразований, который должен использоваться с этим узлом. Установите тип perfect forwarding secrecy с помощью команды set pfs <type> и измените время жизни ассоциации безопасности IPsec, заданное по умолчанию, с помощью команды:
set security-association lifetime seconds <seconds>.

```
R1(config-crypto-map)# set pfs group1
R1(config-crypto-map)# set transform-set 50
R1(config-crypto-map)# set security-association lifetime seconds 900
R1(config-crypto-map)# exit
```

Создайте такую же криптографическую карту на маршрутизаторе R3.

```
R3(config)# crypto map PishMAP 10 ipsec-isakmp
R3(config-crypto-map)# match address 101
R3(config-crypto-map)# set peer 10.1.1.1
R3(config-crypto-map)# set pfs group1
R3(config-crypto-map)# set transform-set 50
R3(config-crypto-map)# set security-association lifetime seconds 900
R3(config-crypto-map)# exit
```

Примените криптографическую карту к интерфейсам.

Ассоциации SA будут установлены только после активации криптографической карты «интересным» трафиком. Маршрутизатор сгенерирует уведомление о том, что шифрование теперь активно. Примените криптографические карты к соответствующим интерфейсам на маршрутизаторах R1 и R3.

```
R1(config)# interface S0/0/0
R1(config-if)# crypto map PishMAP
*Jan 28 04:09:09.150: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
R1(config)# end
R3(config)# interface S0/0/1
R3(config-if)# crypto map PishMAP
*Jan 28 04:10:54.138: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON

R3(config)# end
```

Проверьте конфигурацию IPsec на маршрутизаторах R1 и R3.

Ранее вы использовали команду show crypto isakmp policy для отображения настроенных на маршрутизаторе политик ISAKMP. Команда show crypto ipsec transform-set отображает настроенные политики IPsec в виде наборов преобразований. Пример вывода настроенных политик IPsec представлен на риснуке 2.2, а пример вывода настроенных политик Ipsec на втором маршрутизаторе представлен на рисунке 2.3.

R1# show crypto ipsec transform-set

```
Transform set 50: { esp-256-aes esp-sha-hmac  }
   will negotiate = { Tunnel,  },

Transform set #$!default_transform_set_1: { esp-aes esp-sha-hmac  }
   will negotiate = { Transport,  },

Transform set #$!default_transform_set_0: { esp-3des esp-sha-hmac  }
   will negotiate = { Transport,  },
```

Рисунок 2.2 – Пример вывода настроенных политик IPsec

R3# show crypto ipsec transform-set

```
Transform set 50: { esp-256-aes esp-sha-hmac  }
   will negotiate = { Tunnel,  },

Transform set #$!default_transform_set_1: { esp-aes esp-sha-hmac  }
   will negotiate = { Transport,  },

Transform set #$!default_transform_set_0: { esp-3des esp-sha-hmac  }
   will negotiate = { Transport,  },
```

Рисунок 2.3 – Пример вывода настроенных политик Ipsec на втором маршрутизаторе

Используйте команду show crypto map для отображения криптографических карт, которые будут применены на маршрутизаторе.

R1# show crypto map

Проверка работы IPsec VPN.

Отобразите ассоциации безопасности ISAKMP.

Команда show crypto isakmp sa показывает, что в данный момент ассоциации IKE SA не существуют. После отправки «интересного» трафика выходные данные команды изменятся.

R1# show crypto isakmp sa

Отобразите ассоциации безопасности IPsec.

Команда show crypto ipsec sa показывает неиспользуемую ассоциацию SA между маршрутизаторами R1 и R3. Количество переданных пакетов равно нулю, и в нижней части выходных данных ассоциации безопасности не указаны.

R1# show crypto ipsec sa

Почему не было согласовано ни одной SA?

Создайте «неинтересный» тестовый трафик и проверьте результаты.

Отправьте эхо-запрос с маршрутизатора R1 на интерфейс S0/0/1 маршрутизатора R3 по IP-адресу 10.2.2.1. Этот запрос должен быть выполнен успешно.

Введите команду show crypto isakmp sa.

Отправьте эхо-запрос с маршрутизатора R1 на интерфейс G0/1 маршрутизатора R3 по IP-адресу 192.168.3.1. Этот запрос должен быть выполнен успешно.

Введите команду show crypto isakmp sa еще раз. Была ли создана SA для этих эхо-запросов?

Поясните ответ.

Создайте «интересный» тестовый трафик и проверьте результаты.

Отправьте расширенный эхо-запрос с маршрутизатора R1 на интерфейс G0/1 маршрутизатора R3 по IP-адресу 192.168.3.1. Расширенный эхо-запрос позволяет контролировать адрес источника пакетов.

Ответьте так, как показано в примере, представленном на рисунке 2.4. Нажмите клавишу Enter для принятия значений по умолчанию везде, кроме тех позиций, где показан другой ответ.

```
R1# ping
Protocol [ip]:
Target IP address: 192.168.3.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 192.168.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:

Packet sent with a source address of 192.168.1.1
..!!!
```

Рисунок 2.4 – Отправка эхо-запроса с помощью расширенной команды ping

Снова введите команду show crypto isakmp sa.

R1# show crypto isakmp sa

Почему на этот раз между маршрутизаторами R1 и R3 была создана ассоциация SA?

Что является конечными точками туннеля IPSec VPN?

Отправьте эхо-запрос с компьютера PC-A на компьютер PC-C. Если он был выполнен успешно, введите команду show crypto ipsec sa. Сколько пакетов было преобразовано между маршрутизаторами R1 и R3?

R1# show crypto ipsec sa

Загрузите файл с выполненной в Cisco Packet Tracer работой на проверку.

Практическая работа № 5

«Настройка списков контроля доступа в заданной топологии сети»

1. Теоретический материал

Списки контроля доступа предоставляют возможности фильтрации пакетов для управления потоком трафика. На любом маршрутизаторе Cisco можно настроить простой межсетевой экран, который позволяет фильтровать трафик с помощью списков контроля доступа — ACL, Access Control Lists. Использование списков контроля доступа позволяет фильтровать трафик, блокируя или пропуская только определенные пакеты.

Список контроля доступа ACL — это последовательный список разрешающих или запрещающих операторов, называемых записями контроля доступа (ACE). При прохождении сетевого трафика через интерфейс, где действует список контроля доступа, маршрутизатор последовательно сопоставляет информацию из пакета с каждой записью в списке контроля доступа на предмет соответствия.

ACL-списки IPv4 используют шаблонные маски, механизм действий которых представлен ан рисунке 1.1.

Рисунок 1.1 – Механизм действия шаблонной маски

Шаблонная маска — это 32-разрядная двоичная строка, используемая маршрутизатором для определения битов адреса, которые будут рассматриваться на предмет совпадения в списке контроля доступа. Ноль в маске означает бит, который будет проверяться, а единица — бит, который можно игнорировать.

Можно настроить списки контроля доступа для входящего и исходящего трафика, рис. 2. Пакеты, создаваемые самим маршрутизатором, не могут фильтроваться списками контроля доступа на этом маршрутизаторе, но могут другими маршрутизаторами, для которых они будут, например, входящим трафиком.

Рисунок 1.2 – На маршрутизаторах Cisco правила фильтрации можно создать для входящего и исходящего трафика

В конце любого созданного списка контроля доступа есть запись запрета всего трафика: deny any. Эта запись автоматически вставляется в конец каждого списка, хотя и не видна в выводе команды show, если мы просматриваем существующий список. Поэтому любой список контроля доступа должен содержать хотя бы одно разрешающее правило, иначе он будет блокировать весь трафик.

Список контроля доступа просматривается маршрутизатором последовательно строка за строкой до тех пор, пока параметры проходящего пакета не подпадут под действие какого-либо из правил. Если это происходит, проверка прекращается и выполняется заданное действие. Т.е. список контроля не обязательно просматривается до конца. Рекомендуемая практика говорит о том, что мы должны сначала записать более частные правила в списке, а затем более общие. Например, сначала запрет для пакетов от конкретного хоста, а затем разрешение для пакетов от всех хостов. Если изменить порядок правил и сначала поместить правило, разрешающее трафик для всех хостов, то правило запрета для конкретного хоста никогда не сработает, т. к. все пакеты будут подпадать под первое правило, разрешающее трафик.

Можно настроить только один список контроля доступа для каждого протокола, направления и интерфейса. Например, в случае маршрутизатора с двумя интерфейсами при использовании двух протоколов, как на рисунке 1.3, можно настроить восемь списков контроля доступа. Если попытаться создать второй список контроля доступа на том интерфейсе и в том же направлении, где уже действует другой список, то маршрутизатор не сообщит об ошибке.

Рисунок 1.3 – Ограничения на число списков контроля доступа

Вместо этого он просто применит на интерфейс последний созданный список.

При создании списков контроля доступа могут указываться не только IP-адреса или их диапазоны, в расширенных списках контроля могут применяться адреса транспортного уровня — логические порты. Организацией по регистрации имен в Интернете, IANA, логические порты подразделяются на три группы, как показано на рисунке 1.4.

Номера портов

Диапазон номеров портов	Группа портов
От 0 до 1023	От 0 до 1023
От 1024 до 49151	Зарегистрированные порты
От 49152 до 65535	Частные и/или динамические порты

Рисунок 1.4 – Деление логических портов на три группы

Порты в диапазоне от 0 до 1023 относятся к группе «Хорошо известных портов» (Well Known Ports), закрепленных за распространенными сервисами. Диапазон от 1024 до 49151 это так называемые «Зарегистрированные порты», Registered Ports, они также закрепляются IANA за конкретными сервисами по запросам производителей программного обеспечения. Наконец, последняя группа - «Эфемерные или динамические порты», Ephemeral Ports, не контролируются IANA и могут использоваться свободно кем угодно, часто пользовательскими приложениями. Деление логических портов транспортного уровня на группы не является стандартом и жестким требованием, это рекомендуемая практика. Не все производители ПО следовали ей в 1990-е, 2000-е годы, но сейчас можно сказать, что это общепринятый подход.

На рисунке 1.5 показан пример работы расширенного списка контроля доступа для сравнительно простого сценария: есть две локальные сети A и B с определенными адресными диапазонами. Для сети A разрешен только веб-трафик, весь остальной запрещен, для сети B, наоборот, весь трафик разрешен, кроме веб-трафика на 80 порт.

Рисунок 1.5 – Пример работы алгоритма проверки пакета на соответствие правилам списка

На маршрутизаторах Cisco можно настроить стандартные или расширенные списки контроля. Стандартные ACL-списки можно использовать для разрешения или отклонения прохождения трафика только на основе IPv4-адресов источника. Место назначения пакета и порты, участвующие в передаче данных, не проверяются. В примере, представленном на рисунке 1.6, разрешен весь трафик от устройств в сети 192.168.30.0/24.

```
access-list 10 permit 192.168.30.0 0.0.0.255
```

Рисунок 1.6 – Настройка стандартного списка контроля доступа

Из-за неявного правила deny any в конце списка этот список контроля доступа блокирует весь трафик, за исключением трафика из сети 192.168.30.0/24. Команда создания списка контроля доступа выполняется в режиме глобальной конфигурации.

Расширенные списки контроля доступа фильтруют IPv4-пакеты, исходя из нескольких признаков:

- тип протокола;
- IPv4-адрес источника;
- IPv4-адрес назначения;
- TCP или UDP порты источника;
- TCP или UDP порты назначения;
- дополнительная информация о типе протокола для оптимизированного контроля.

На рис. 7 ACL-список 103 разрешает трафику с любого адреса сети 192.168.30.0/24 идти в любую IPv4-сеть, если порт назначения — 80 (HTTP).

```
access-list 103 permit tcp 192.168.30.0 0.0.0.255 any eq 80
```

Рисунок 1.7 – Настройка расширенного списка контроля доступа

Настраиваемый список стандартным или расширенным определяется заданным номером. Диапазон номеров для стандартных и расширенных списков контроля доступа представлен ан рисунке 1.8.

```
Router(config)#access-list ?
  <1-99>     IP standard access list
  <100-199>  IP extended access list
```

Рисунок 1.8 – Диапазон номеров для стандартных и расширенных списков контроля доступа

Номера от 1 до 99 и от 1300 до 1999 задают стандартный нумерованный список контроля, номера от 100 до 199 и от 2000 до 2699 расширенный нумерованный список контроля доступа.

2. Ход выполнения работы

Настройте топологию согласно рисунку 2.1.

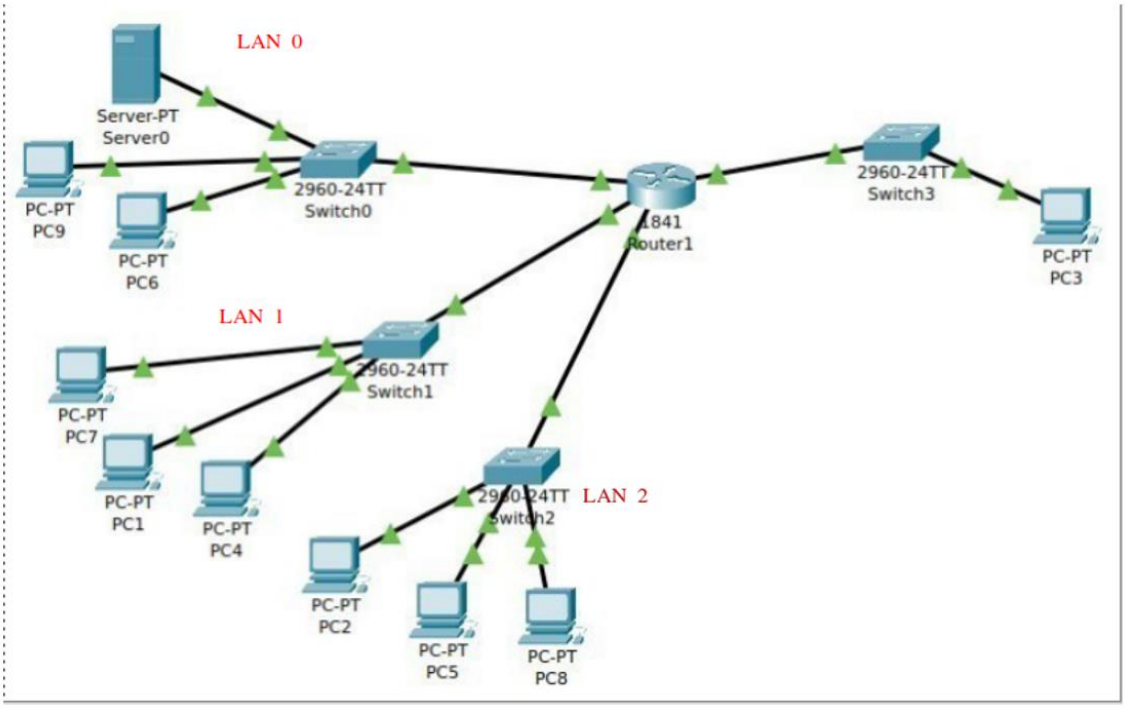

Рисунок 2.1 – Топология сети выполняемой работы

IP-адрес компьютера PC3 задайте 209.165.200.225, на интерфейсе маршрутизатора в той же сети настройте 209.165.200.226 или другой адрес из той же сети. Устройства, подключенные к коммутаторам Switch0, Switch1 и Switch2, настройте с IP-адресами из вашего варианта разбиения сети на подсети. Варианты представлены в таблице 2.1.

Таблица 2.1 – Варианты

Вариант	Исходная сеть (блок адресов)	Количество компьютеров в отделах		
		А	Б	В
1	118.7.50.0 /25	7	9	27
2	39.221.98.0 /25	8	5	18
3	88.27.252.0 /23	30	9	46
4	81.104.216.0 /21	48	120	249
5	7.50.128.0 /19	267	176	678
6	89.151.32.0 /19	311	246	806
7	126.61.74.0 /23	8	61	17
8	36.121.96.0 /19	311	696	226
9	28.54.64.0 /19	957	153	274
10	67.253.1.0 /20	365	116	508
11	77.75.0.0 /18	338	830	1403
12	5.63.168.0 /21	119	61	226
13	85.123.72.0 /21	189	51	72
14	72.241.3.0 /25	12	7	3
15	87.228.68.0 /22	26	45	71
16	46.41.64.0 /18	384	1535	675
17	57.214.86.0 /23	63	9	21
18	74.30.128.0 /19	346	179	732
19	88.61.128.0 /20	366	77	130
20	10.58.180.0 /22	30	92	43

Окончание таблицы 2.1

21	112.56.76.0 /22	23	114	60
22	2.78.160.0 /19	214	443	525
23	30.182.64.0 /18	624	1700	358
24	75.39.128.0 /19	625	219	372
25	98.115.89.37 /21	48	119	250
26	35.163.168.0 /21	119	60	224

Будем считать далее, что Switch0 объединяет устройства из LAN0, Switch1 из LAN1 и Switch2 из LAN2. Настройте на маршрутизаторе:

1. список контроля доступа, запрещающий доступ всем устройствам из LAN2 к любым устройствам в LAN0. Компьютеры в LAN1, компьютер PC3 и другие устройства, не входящие в LAN2 не должны попадать под ограничения.

2. список контроля доступа, разрешающий доступ к PC3 только одному компьютеру из LAN1, например, компьютеру PC1 на рис. 9. Доступ к другим устройствам в той же сети, где расположен PC3 не должен быть ограничен. Устройства из других локальных сетей, кроме LAN1, также должны свободно отправлять пакеты любым устройствам в сети на Switch3.

3. на сервере в локальной сети LAN0 включите веб-сервис (вкладка Services, сервис HTTP. Он может быть включен по умолчанию, убедитесь, что радиокнопка в положении On). Настройте список контроля доступа, разрешающий доступ к веб-серверу только для компьютера PC3. Для всех остальных доступ должен быть запрещен.

Все три условия должны выполняться одновременно в финальной конфигурации.

Отчет по работе и файл Cisco Packet Tracer с выполненной работой загрузите на проверку.

ЗАКЛЮЧЕНИЕ

Успешно пройденный курс позволяет приобрести навыки работы с сетевым оборудованием, анализа конфигурационных параметров и характеристик сетевого трафика. При успешном усвоении курса студенты получат ясное представление о взаимодействии уровней референсной модели OSI в реальных компьютерных сетях, влиянии на скорость передачи данных различных факторов, как физических характеристик канала передачи и оборудования, так и конфигурационных настроек. Курс позволяет применить на практике полученные ранее теоретические знания, что при успешном усвоении материала позволит самостоятельно работать с компьютерными сетями различных типов.

СПИСОК ЛИТЕРАТУРЫ

1. Руденко А. Эмулятор компьютерной сети / А. Руденко // Системный администратор. – 2013. - № 12 (133). – С. 18 – 21.

2. CS244: Advanced Topics in Networking : сайт / Stanford. – URL: http://web.stanford.edu/class/cs244/ . – Режим доступа: свободный. (дата обращения 22.11.2023).

3. Manya Ghobadi and Mohammad Alizadeh. 6.829: Computer Networks : сайт / Canvas. – URL: http://web.mit.edu/6.829/www/currentsemester/. – Режим доступа: свободный. (дата обращения 22.11.2023).

4. Старая песня о главном. TCP/IP : сайт / nag. – URL: https://nag.ru/material/3490 . – Режим доступа: свободный. (дата обращения 22.11.2023).

5. Модели реализации протокола TCP и его перспективы : сайт / book.itep – URL: http://book.itep.ru/4/44/tcp.htm . – Режим доступа: свободный. (дата обращения 22.11.2023).

6. ГОСТ Р 53246-2008 Информационные технологии (ИТ). Системы кабельные структурированные. Проектирование основных узлов системы. Общие требования : сайт / Электронный фонд правовых и нормативно-технических документов – URL: http://book.itep.ru/4/44/tcp.htm . – Режим доступа: свободный. (дата обращения 22.11.2023).

7. Introduction to Mininet : сайт / GitHub – URL: https://github.com/mininet/mininet/wiki/Introduction-to-Mininet#custom . – Режим доступа: свободный. (дата обращения 22.11.2023).