Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО «МГТУ «СТАНКИН»)

С.Е. Сосенушкин, М.В. Левин

ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫЕ СЕТИ. РЕШЕНИЕ ЗАДАЧ

Рекомендовано кафедрой информационных систем ФГБОУ ВО «МГТУ «СТАНКИН» в качестве учебного пособия для студентов технических вузов, обучающихся по направлению подготовки 09.03.02 «Информационные системы и технологии»

Москва
2019

УДК 004.72 (075)
ББК 32.973
С661

Рецензент – канд. техн. наук Сутягин М.В., заместитель директора филиала ЧОУ ДПО «Газпром корпоративный институт» в Москве

Сосенушкин С.Е., Левин М.В.

С661 Информационно-телекоммуникационные сети. Решение задач: учеб. пособие / С.Е. Сосенушкин, М.В. Левин. – М.: ФГБОУ ВО МГТУ «СТАНКИН», 2018. – 87 с.: ил.

ISBN 987-5-7028-0612-9

Содержит теоретические сведения, практические задания и методические рекомендации к самостоятельной работе студентов при изучении дисциплины «Информационно-телекоммуникационные сети». Раскрыты основные темы, рассматриваемые на практических занятиях по указанной дисциплине, соответствующие принципы, технологии и протоколы. Разобраны решения типовых задач, приведены упражнения для самостоятельного выполнения.

Предназначено для студентов третьего курса, обучающихся по направлению подготовки 09.03.02 «Информационные системы и технологии».

УДК 004.72 (075)
ББК 32.973

ISBN 987-5-7028-0612-9

Содержание

Семинар 1.
Сетевые модели OSI и TCP/IP. Инкапсуляция. Адресация. Назначение сетевых устройств

1.1. Цель и задачи семинара

Цель семинара – закрепление обучающимися знаний по вводным разделам курса в соответствии с темой занятия. На семинаре обсуждается структура простейших сетей на основе компьютеров, коммутаторов, маршрутизаторов, различные типы сетевых кабелей и способы адресации устройств в сети.

1.2. Теоретическая часть

1.2.1. Структура простейшей сети

Рассмотрим задачу по созданию простейшей сети. Допустим, есть два компьютера, оснащенных стандартными сетевыми адаптерами Ethernet. Как соединить их между собой, какое оборудование для этого нужно и как оно должно быть настроено?

Рассмотрим несколько вариантов решения задачи от простого к сложному. Самый простой из них: напрямую соединить два компьютера сетевым кабелем (рис. 1.1).

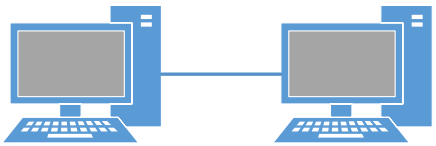


Рис. 1. Простейшая сеть

Вопрос 1.1. Какой именно кабель необходимо использовать?

Соединяем, получаем простейшую сеть из двух конечных устройств, непосредственно подключенных друг к другу. Для передачи между ними данных им необходима адресация.

Вопрос 1.2. Какие адреса есть у компьютеров по умолчанию? Что они собой представляют? Как их узнать?

Действительно, физические адреса позволяют адресовать устройства на канальном уровне в пределах домашней сети. С физическими адресами работают коммутаторы и сетевые адаптеры, но не пользователи и не прикладное ПО. Допустим, у компьютера А будет физический адрес MAC-A, а у компьютера В – MAC-B. Этих адресов, казалось бы, достаточно для передачи кадров между компьютерами средствами канального уровня, но это не так.

Вопрос 1.3. Что такое стек протоколов?

Для обеспечения возможности передачи данных необходимо закрыть протоколами все уровни взаимодействия, т.к. сетевые компоненты каждого уровня могут взаимодействовать только с соседними уровнями домашнего узла.

Вопрос 1.4. Как называется набор формализованных правил такого взаимодействия (между сетевыми компонентами соседних уровней одного узла)?

Соответственно, для передачи данных нам потребуются протоколы всех уровней стека.

Вопрос 1.5. Какой стек протоколов лежит в основе современного Интернета? Сколько в нем уровней? Назовите их.

Для того, чтобы передать данные вниз по стеку (а данные появляются на верхнем уровне – уровне приложений), потребуется в том числе протокол сетевого уровня IP (Internet Protocol), использующий собственную схему адресации узлов, называемую IP-адресацией. В отличие от физического адреса, логический адрес узла может меняться при перемещении этого узла из одной сети в другую. Аналогом физического (MAC) адреса сетевого устройства в реальной жизни может служить имя. Оно присваивается человеку при рождении и меняется лишь в исключительных случаях с использованием специальных средств и процедур. Аналогом же логического (или сетевого, или IP) адреса может быть, например, адрес фактического проживания. Сегодня человек проживает по одному адресу, завтра переедет и адрес сменится, но имя останется тем же. Для доставки сообщения необходимо знать и адрес, и имя. При этом как и для MAC-адресов, для IP-адресов существует правило уникальности: при связи через сеть Интернет требуется глобальная уникальность адреса; без выхода в Интернет требуется уникальность адреса в пределах домашней сети.

В зависимости от назначения IP-адреса делятся на индивидуальные (unicast – служат для адресации конкретного узла сети), групповые (multicast – идентифицируют группу узлов, объединенную общим признаком, например, использующую общий протокол) и широковещательные (broadcast – идентифицирующие все узлы сети).

IP-адрес представляет собой 32-битное двоичное число, разделенное на 4 байта (октета). В структуре индивидуальных адресов выделяют две составные части: номер сети (N – network, характеризует сеть в целом и совпадает для всех узлов сети) и номер узла в данной сети (H – host). Сетевая часть служит для определения адреса сети, в которую необходимо доставить пакет, а узловая часть определяет тот конкретный узел, которому это сообщение предназначено. Примером сетевой и узловой части в повседневной жизни можно назвать отправление письма по адресу: 123456 Москва, Сетевая улица, дом 128, квартира 255, где 123456 Москва, Сетевая улица, дом 128 – это сетевая часть, общая для всех квартир данного дома, а квартира 255 – узловая часть. Таким образом, IP-адрес состоит из двух частей: номера сети (левые n бит) и номера узла (правые (32-n) бит). Чем больше значение n, тем меньше размер сети и количество узлов в ней, но тем больше таких сетей можно создать. Верно и обратное.

Не забегая вперед, отметим, что для настройки простейшей сети необходимо выбрать адреса и присвоить их компьютерам. Причем эти адреса должны иметь совпадающую сетевую часть. Выделение сетевой и узловой части IP-адреса будет подробно рассмотрено на следующем семинаре. Пока ограничимся выбором двух подряд идущих IP-адресов: 192.168.1.2 и 192.168.1.3 и сетевой маски 255.255.255.0 (назначение этого параметра будет разъяснено позднее). После присвоения указанных адресов сетевым интерфейсам компьютеров настройка сети будет завершена, а сама сеть готова к работе.

Адресация в простейшей сети

Устройство	IP-адрес	Маска
PC0	192.168.1.2	255.255.255.0
PC1	192.168.1.3	255.255.255.0

1.2.2. Сети на основе коммутатора

Усложним задачу. Как известно, одним из критериев качества сетей является масштабируемость.

Вопрос 1.6. Назовите остальные критерии качества сетей.

Допустим, возникла потребность подключить к сети еще 1 компьютер. Очевидно, что подключить его напрямую к сетевым двух первых узлов невозможно: интерфейсы заняты. Можно добавить во все три узла еще по одному интерфейсу и соединить узлы треугольником, каждый с каждым. Но это решение неверное, т.к. не обеспечивает масштабируемости. Для эффективного решения задачи необходимо добавить в топологию сетевое устройство. Существуют несколько различных типов сетевых устройств. В данном случае требуется устройство, способное соединить между собой узлы в пределах одной сети. Для этого предназначены сетевые коммутаторы. Добавим в сеть коммутатор и подключим к нему компьютеры.

Вопрос 1.7. Какие кабели используются для подключения компьютеров к коммутатору?

Получилась сеть, показанная на рисунке 1.2. Она полностью работоспособна: между компьютерами A и B возможна передача данных, причем для такой передачи достаточно возможностей канального уровня. Сетевые адреса компьютерам нужны лишь для обеспечения непрерывности передачи данных по стеку.

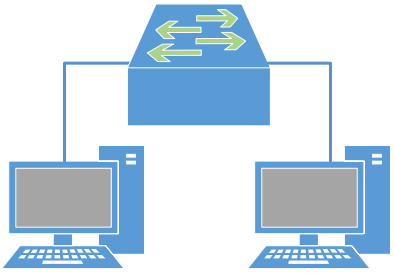


Рис. 1.2. Сеть на основе коммутатора

Снова усложним задачу. Допустим, возникла необходимость подключения к созданной локальной сети на основе коммутатора еще одной точно такой же, адреса компьютеров в которой будут: 192.168.2.2 и 192.168.2.3, маска 255.255.255.0. Рассмотрим несколько подходов.

Первый подход, простейший. Соединим коммутаторы между собой с помощью сетевого кабеля.

Вопрос 1.8. Какой кабель необходимо использовать для прямого соединения двух коммутаторов?

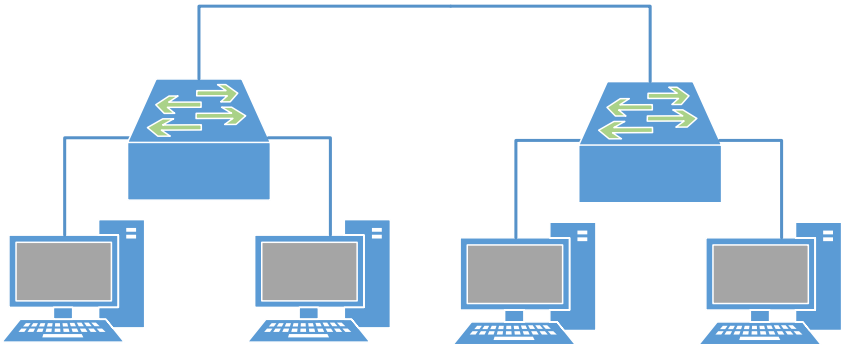


Рисунок 1.3. Масштабирование сети

Получившаяся сеть не работает. Вернее, работает каждый из сегментов по отдельности, но связь между первой и второй сетями нарушена. Это происходит из-за того, что компьютеры первой сети и компьютеры второй сети принадлежат разным сетям с точки зрения IP-адресации, т.к. сетевые части их IP-адресов не совпадают. Коммутаторы не в состоянии передавать данные из одной сети в другую, поэтому передача невозможна. Для устранения этой ошибки изменить адресацию так, чтобы компьютеры оказались в одной сети – то есть, чтобы совпала сетевая часть их IP-адресов. Для этого достаточно (первый вариант) изменить адреса устройств в одной из сетей (например, присвоить устройствам сети B адреса 192.168.1.4 и 192.168.1.5), оставив сетевую маску неизменной, или (второй вариант) увеличить размер сети, изменив для всех четырех устройств сетевую маску на 255.255.0.0, а адреса компьютеров не менять. В обоих случаях все устройства окажутся в одной сети, где для передачи данных достаточно функций канального уровня.

Вариант 1: Изменяем IP-адреса

Устройство	IP-адрес	Маска
PC0	192.168.1.2	255.255.255.0
PC1	192.168.1.3	255.255.255.0
PC2	**192.168.1.4**	255.255.255.0
PC3	**192.168.1.5**	255.255.255.0

Вариант 2: изменяем размер сети (маску)

Устройство	IP-адрес	Маска
PC0	192.168.1.2	**255.255.0.0**
PC1	192.168.1.3	**255.255.0.0**
PC2	192.168.2.2	**255.255.0.0**
PC3	192.168.2.3	**255.255.0.0**

1.2.3. Простейшая составная сеть

Существует и другой способ решения этой проблемы, не связанный со сменой адресации. Этот способ потребует создания составной сети – соединения двух сетей с помощью нового устройства – маршрутизатора (рисунок 1.4).

Вопрос 1.9. На каких уровнях OSI работает маршрутизатор?

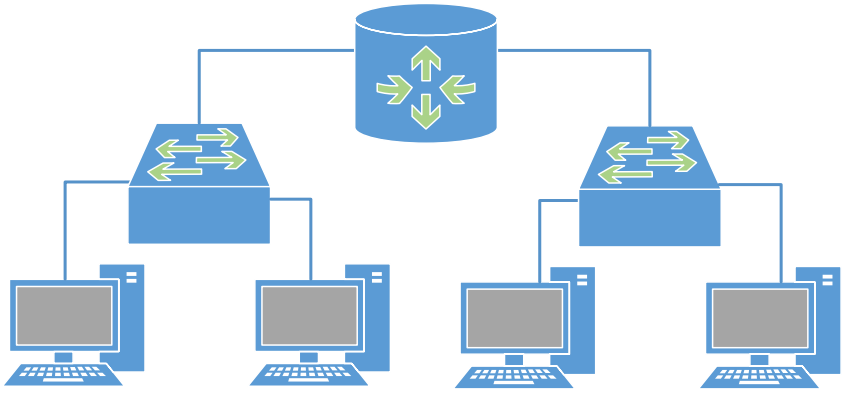


Рисунок 1.4. Простейшая составная сеть

Маршрутизатор взаимодействует с подключенными сетями с помощью интерфейсов, причем каждый интерфейс настраивается отдельно и независимо от других. Подобно сетевому адаптеру компьютера, интерфейсы маршрутизаторов имеют свои MAC- и IP-адреса, причем на одном маршрутизаторе не допускается настройка нескольких интерфейсов, принадлежащих одной сети. Интерфейсы могут служить для подключения маршрутизатора к разнородным сетям, т.е. относящимся к разным технологиям уровня интерфейсов, в том числе, локальным и глобальным, например, Gigabit Ethernet и Frame Relay.

Соединяем коммутаторы и маршрутизатор прямыми кабелями витая пара. На интерфейсах маршрутизатора необходимо настроить IP-адреса из соответствующих сетей и маски. Обратите внимание: адрес компьютера и адрес ближайшего интерфейса маршрутизатора должны принадлежать одной сети. Здесь и далее будем использовать для каждого интерфейса маршрутизатора первый из доступных IP-адресов соответствующего диапазона. Получившаяся составная сеть будет использовать технологии сетевого уровня для передачи данных из одной части сети в другую. Однако пока она не работает, т.к. ни один из компьютеров не может выйти за пределы своей домашней сети. Для обеспечения такой возможности настроим еще один из важнейших параметров

конфигурации любого сетевого интерфейса – основной шлюз (default gateway, шлюз по умолчанию). Шлюзом называется интерфейс маршрутизатора, ведущий из домашней сети в другие сети. Параметры настройки устройств приведены в таблице.

Устройство	IP-адрес	Маска	Основной шлюз
PC0	192.168.1.2	255.255.255.0	**192.168.1.1**
PC1	192.168.1.3	255.255.255.0	**192.168.1.1**
PC2	192.168.2.2	255.255.255.0	**192.168.2.1**
PC3	192.168.2.3	255.255.255.0	**192.168.2.1**
Router f0/0	**192.168.1.1**	**255.255.255.0**	–
Router f0/1	**192.168.2.1**	**255.255.255.0**	–

1.3. Ответы на вопросы семинара

1.1. ***Какой именно кабель необходимо использовать?*** *В телекоммуникациях используются медные и волоконно-оптические кабели. Для прямого соединения между собой сетевых адаптеров Ethernet следует использовать медный кабель витая пара. Этот кабель должен быть перекрестным (crossover), то есть терминированным на концах по двум разным схемам: A и B, в соответствии с международным стандартом ISO 11801.*

1.2. ***Какие адреса есть у компьютеров по умолчанию? Что они собой представляют?*** *Физические (MAC) адреса. MAC-адрес присваивается каждому сетевому интерфейсу производителем при выпуске устройства и не подлежит изменению штатными средствами. Он представляет собой 48-битное двоичное число, записываемое, как правило, в шестнадцатеричном формате, например, 000c.2ab6.e1dc. Существует несколько способов узнать MAC-адрес компьютера под управлением ОС Windows. Например, можно опросить сетевой адаптер, набрав в командной строке (command prompt) команду ipconfig с ключом /all. Другой способ: открыть окно настроек сетевого адаптера.*

1.3. ***Что такое стек протоколов?*** *Стеком протоколов называется иерархически организованный набор протоколов, достаточный для обеспечения взаимодействия узлов в сети.*

1.4. ***Как называется набор формализованных правил такого взаимодействия?*** *Формализованные правила, определяющие последовательность и формат сообщений, которыми обмениваются сетевые компоненты соседних уровней одного узла, называется сетевым интерфейсом.*

1.5. ***Какой стек протоколов лежит в основе современного Интернета? Сколько в нем уровней? Назовите их.*** *В основе современного Интернета лежит стек протоколов TCP/IP, получивший название благодаря двум ключевым протоколам сетевого и транспортного уровней. Модель TCP/IP выделяет четыре уровня протокольного взаимодействия (сверху вниз): уровень приложений, транспортный уровень, сетевой уровень, уровень сетевых интерфейсов.*

1.6. ***Назовите остальные критерии качества сетей.*** *Отказоустойчивость, масштабируемость, производительность, безопасность.*

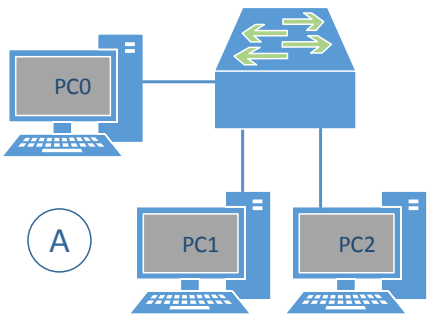
1.7. ***Какие кабели используются для подключения компьютеров к коммутатору?*** *Прямая (straight-through) витая пара.*

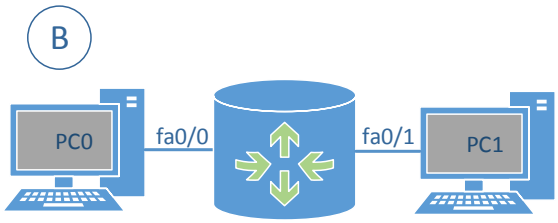
1.8. ***Какой именно кабель необходимо использовать для прямого соединения двух коммутаторов?*** *Перекрестная (crossover) витая пара.*

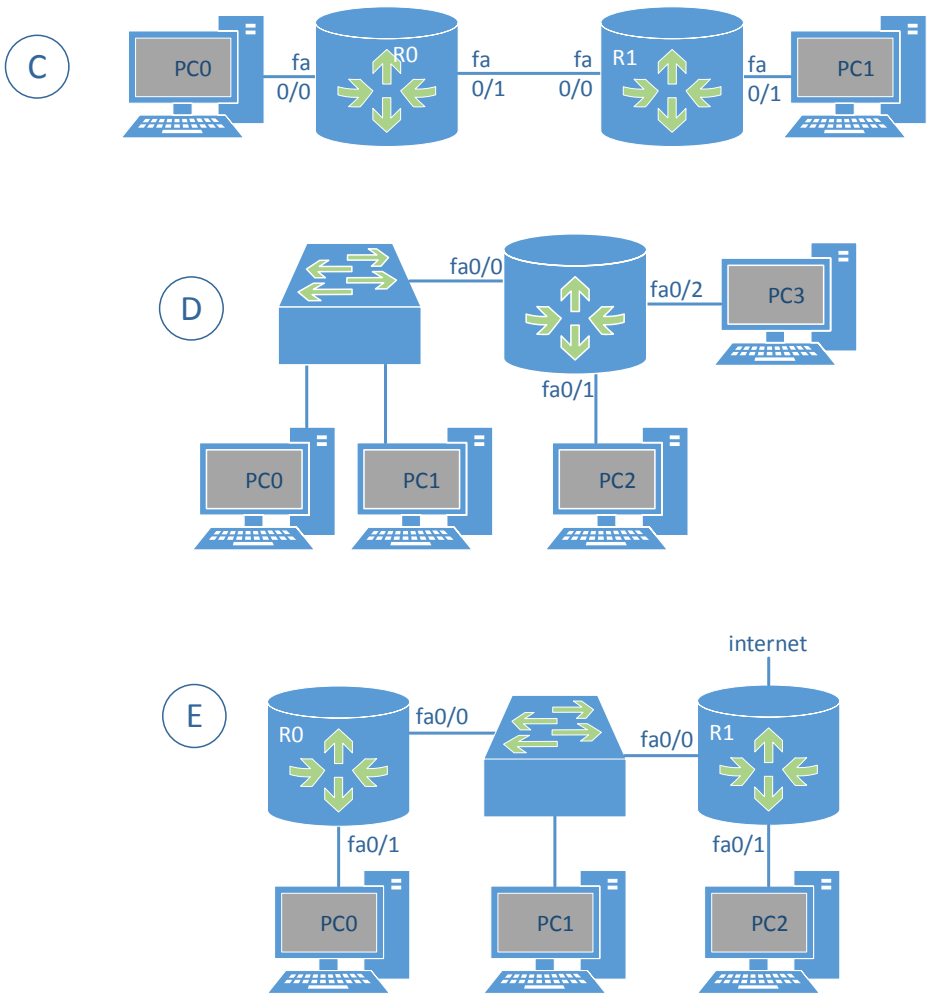
1.9. ***На каких уровнях OSI работает маршрутизатор?*** *Сетевой, канальный, физический.*

1.4. Тренировочные задания

Выберите подходящие адреса для сетевых устройств в сетевых топологиях, представленных ниже. Не используйте адреса вида 192.168.X.X.







1.5. Рекомендуемая литература и Интернет-ресурсы

1. Основы компьютерных сетей. Тема № 1. Основные сетевые термины и сетевые модели [электронный ресурс]. – Режим доступа: https://habr.com/post/307252/ – Заглавие с экрана. Дата обращения: 09.10.2018.

2. Основы компьютерных сетей. Тема № 4. Сетевые устройства и виды применяемых кабелей [электронный ресурс]. – Режим доступа: https://habr.com/post/312340/ – Заглавие с экрана. Дата обращения: 09.10.2018.

3. Компьютерные сети for dummies [электронный ресурс]. – Режим доступа: https://habr.com/post/335816/ – Заглавие с экрана. Дата обращения: 09.10.2018.

Семинар 2.
IP-адрес и маска подсети

2.1. Цель и задачи семинара

Семинар посвящен закреплению знаний о пространстве IPv4 адресов и формированию навыков расчета основных параметров IPv4 адресации. Обсуждаются следующие понятия: IP-адрес, сетевая маска, сеть и класс сети, подсеть и структура подсети, свободные и фиксированные биты, адрес подсети (subnet), адрес широковещательной рассылки (broadcast). На семинаре проводится разбор решения простейших типов задач на IP-адресацию: определение количества доступных адресов, расчет subnet и broadcast, подбор маски и др.

2.2. Теоретическая часть

IP-адрес (ай-пи адрес, сокращение от англ. Internet Protocol Address) – сетевой адрес узла в компьютерной сети, построенной по протоколу IP. При связи через сеть Интернет требуется глобальная уникальность адреса, в случае работы в локальной сети требуется уникальность адреса в пределах сети.

В зависимости от назначения IP-адреса делятся на индивидуальные (unicast – служат для адресации одного конкретного узла сети), групповые (multicast – идентифицируют группу узлов, объединенную общим признаком, например, использующую общий протокол) и широковещательные (broadcast – идентифицирующие все узлы сети).

IP-адрес представляет собой 32-битное двоичное число, разделенное на 4 байта (октета). В структуре индивидуальных адресов выделяют две составные части: номер сети (N – network, характеризует адрес сети в целом и совпадает для всех узлов сети) и номер узла в сети (H – host). Сетевая часть служит для определения адреса сети, в которую необходимо доставить пакет, а узловая часть определяет тот конкретный узел, которому это сообщение предназначено.

Примером сетевой и узловой части в повседневной жизни можно назвать отправление письма по адресу: 123456 Москва, Сетевая улица, дом 128, квартира 255, где 123456 Москва, Сетевая улица, дом 128 – это сетевая часть, общая для всех квартир данного дома, а квартира 255 – узловая часть.

Таким образом, IP-адрес состоит из двух частей: номера сети (левые ***n*** бит) и номера узла (правые (32-***n***) бит). Чем больше значение ***n***, тем меньше

размер сети и количество узлов в ней, но тем больше таких сетей можно создать. Верно и обратное.

В IP адресе каждый из четырех октетов состоит из 8 бит, каждый бит имеет значение 0 или 1. Четыре группы из 8 бит имеют одну и ту же серию допустимых десятичных значений – от 0 до 255 включительно. Значения каждого размещения бита соответствует степени двойки от 7 до 0: 128, 64, 32, 16, 8, 4, 2 и 1.

Если все 8 бит имеют значение 0 (0000 0000), значение октета равно 0.

Если все 8 бит имеют значение 1 (1111 1111), значение октета равно 255 (128 + 64 + 32 + 16 + 8 + 4 + 2 + 1).

Если среди 8 бит есть и единицы, и нули, то степени двойки, соответствующие единицам, складываются друг с другом. Например, значение октета 0010 0111 составляет $39 = 2^5 + 2^2 + 2^1 + 2^0 = (32 + 4 + 2 + 1)$, т.к. в данном случае единицы стоят на (считать начинаем справа налево с нуля) нулевом, первом, втором и пятом месте.

Таким образом, значение каждого из четырёх октетов находится в диапазоне от 0 до 255.

Пример: 192.168.1.12 перевести в двоичную систему.

$192 = 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^3 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$ = *1100 0000*

$168 = 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^3 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$ = *101 01000*

$1 = 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^3 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$ = *0000 0001*

$12 = 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^3 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$ = *0000 1010*

192.168.1.12 = 1100000. 1010100. 0000001. 0001010

Есть два способа определения того, сколько бит IP-адреса отводится на номер сети, а сколько на номер узла. Так, существует т.н. классовая адресация (INET) и бесклассовая адресация (CIDR). При классовой адресации значение ***n*** определяется классом сети, который в свою очередь зависит от старших бит IP-адреса.

В настоящее время чаще применяется бесклассовая адресация, при которой значение n определяется сетевой маской (netmask). Как и сам адрес, маска состоит из 32-х бит, разделенных на четыре октета. IP-адрес и маска подсети записываются друг под другом, как показано на рисунке 2.1.

Важнейшей особенностью сетевой маски является строгое правило группировки всех бит-единиц слева. Если в IP-адресе единицы и нули могут чередоваться друг с другом в любом октете и в любом порядке (например, 01101001), то в маске всегда и все единицы располагаются в старших битах,

строго подряд, а затем следуют нули. Тот бит, в котором в маске заканчиваются единицы и начинаются нули, и является границей между сетевой и узловой частями соответствующего IP-адреса. Таким образом, единицы в маске подсети определяют сетевую часть, а нули – узловую.

Рисунок 2.1. IP-адрес и маска подсети

Каждому биту в октете присваивается конкретное значение: у старшего бита, идущего первым, это значение равно 128, у младшего бита в октете это значение равно 1. Сумма октета, состоящего из всех единиц, равняется 255, октет, состоящий только из нулей, равен 0.

Маска подсети необходима для установления ограничения по количеству узлов в одной сети. Чем больше сетевая часть (количество единичных разрядов) в маске подсети, тем меньше количество узлов можно разместить в этой сети. Общее количество адресов в подсети вычисляется как 2 в степени, равной количеству нулей в сетевой маске.

Пример. Посчитаем количество адресов в сети с маской 255.255.255.0. Переводим маску в двоичный вид: 11111111 11111111 11111111 00000000 – 24 единицы, 8 нулей. Значит, в сети возможно 2^8 адресов, т.е. 256.

Кроме десятичной и двоичной форм записи маски, существует еще один способ обозначить ее размер. Этот тип записи называется префикс и имеет вид косой черты с числом, например, /19 и записывается в паре с IP-адресом – 10.96.47.0/19. Значение префикса соответствует количеству единиц в маске, а значит, и количеству бит сетевой части адреса. Так как маска подсети состоит

из 32-х битов, а первые 19 бит единицы, то двоичная запись такой маски будет выглядеть следующим образом: 11111111 11111111 11100000 00000000.

Пример. Десятичный вид маски подсети, записанный префиксом, для адреса 10.96.47.0/19, имеет следующий вид: 10.96.47.0 255.255.224.0.

Отметим, что особенность взаимного расположения бит в маске задает конечный диапазон допустимых значений каждого октета маски, приведенный в таблице 2.1.

2^7 128	2^6 64	2^5 32	2^4 16	2^3 8	2^2 4	2^1 2	2^0 1	Байт маски
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	255
1	**1**	**1**	**1**	**1**	**1**	**1**	**0**	254
1	**1**	**1**	**1**	**1**	**1**	**0**	**0**	252
1	**1**	**1**	**1**	**1**	**0**	**0**	**0**	248
1	**1**	**1**	**1**	**0**	**0**	**0**	**0**	240
1	**1**	**1**	**0**	**0**	**0**	**0**	**0**	224
1	**1**	**0**	**0**	**0**	**0**	**0**	**0**	192
1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	128
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	0

Таблица 2.1. Возможные значения октетов маски

Наложенная на конкретный IP-адрес, маска задает верхнюю и нижнюю границы подсети. Границами любой подсети служат минимальный адрес в сети, называемый также адресом сети (subnet, SN) и максимальный адрес в сети, который является широковещательным адресом по всем узлам, входящим в сеть (broadcast, BC). Для определения значений subnet и broadcast необходимо проверить, в каком октете маски подсети последовательность единиц заканчивается, и маска становится меньше 255. В нашем случае это третий октет, который имеет значение 224. Следующим действием необходимо расписать в двоичном виде третьи октеты маски подсети и IP-адреса один под одним. Третий октет маски под сети со значением 224, набирается суммированием первых трех старших бит октета, имеющих значение 128 + 64 +

32. Третий октет IP-адреса, равный 47, набирается суммированием значений 32 + 8 + 4 +2 + 1, во всех этих местах устанавливается значение 1, битам, которые не участвуют в суммировании, устанавливается значение 0. Затем по границе единиц и нулей в маске делим адрес на фиксированную часть (сеть) и свободную (узел).

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	128	64	32	16	8	4	2	1
47	**0**	**0**	**1**	**0**	**1**	**1**	**1**	**1**
224	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**

Маска нужна только для нахождения границы сетевой и узловой части. Граница найдена, маска более не понадобится. Для нахождения адресов SN и BC необходимо заполнить узловую часть всеми нулями (SN) и всеми единицами (BC) соответственно.

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	128	64	32	16	8	4	2	1
47	**0**	**0**	**1**	**0**	**1**	**1**	**1**	**1**
subnet	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**
broadcast	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**

Переводим полученные двоичные числа в десятичную форму. Получаем для subnet 2^5=32, для broadcast $2^5+2^4+2^3+2^2+2^1+2^0$=63. Вспомним, что это третий октет IP-адреса, и есть еще четвертый, который, так же, как и младшие биты третьего (после границы) в адресах SN и BC заполнен всеми нулями и всеми единицами соответственно. Следовательно, значение четвертого октета для SN равно нулю, а для BC – 255.

Получаем ответ: subnet: 10.96.32.0, broadcast: 10.96.63.255.

Результатом нахождения адресов subnet и broadcast является диапазон адресов, входящих в подсеть, которые можно использовать для присвоения их узлам в сети – больше первого и меньше второго. **Адреса со значением subnet и broadcast присваивать узлам запрещено.** Первый и последний из доступных для адресации узлов адреса называются hostmin и hostmax соответственно.

В протоколе IP существует несколько соглашений об особой интерпретации IP-адресов. Так, если весь IP-адрес состоит только из двоичных нулей, то он обозначает адрес того узла, который сгенерировал этот пакет; этот режим используется только в некоторых сообщениях ICMP. Если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен

рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такая рассылка называется ограниченным широковещательным сообщением (limited broadcast). Если в поле номера узла назначения стоят только единицы, то пакет имеющий такой адрес, рассылается всем узлам сети с заданным номером сети. Например, в сети 192.190.21.0 с маской 255.255.255.0 пакет с адресом 192.190.21.255 доставляется всем узлам этой сети. Такая рассылка называется широковещательным сообщением (broadcast).

Для изолированной сети её адрес может быть выбран администратором из специально зарезервированных для таких сетей блоков адресов (192.168.0.0/16, 172.16.0.0/12 или 10.0.0.0/8). Если же сеть должна работать как составная часть Интернета, то адрес сети выдаётся провайдером либо региональным интернет-регистратором (Regional Internet Registry, RIR). Региональные регистраторы получают номера автономных систем и большие блоки адресов у IANA, а затем выдают номера автономных систем и блоки адресов меньшего размера локальным интернет-регистраторам (Local Internet Registries, LIR), обычно являющимся крупными провайдерами.

IANA (от англ. Internet Assigned Numbers Authority – «Администрация адресного пространства Интернет») – американская некоммерческая организация, управляющая пространствами IP-адресов, доменов верхнего уровня.

2.3. Разбор задач

Задача 2.3.1.

Подсеть задана адресом 192.168.1.128 с маской 255.255.255.240. Найти максимальное количество узлов в подсети.

Решение.

Общее количество адресов в подсети определяется количеством свободных бит, т.е. количеством нулей в сетевой маске. Вычисляем:

$$240 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 1111\ 0000$$

Имеем N_0=4 свободных бита, значит общее количество адресов в сети равно 2^{N0} = 16, причем 2 из них (SUBNET и BROADCAST) являются служебными и не могут использоваться для адресации узлов. Следовательно, максимальное количество узлов в сети N_y = 16 – 2 = 14.

Ответ: 14 узлов.

Задача 2.3.2.

Подсеть задана IP-адресом входящего в нее узла и маской 172.16.114.159/19. Найти адрес подсети (SUBNET).

Решение.

SUBNET – первый по порядку адрес в подсети, у которого все свободные биты равны 0, вычисляется наложением маски, т.е. побитовой конъюнкцией любого IP-адреса, входящего в подсеть, и маски подсети.

Определим, в каком по счету октете маски проходит граница между фиксированными и свободными битами (между единицами и нулями). В маске 19 бит – это третий октет (в октетах по 8 бит, первый октет 0-7, второй 8-15, третий 16-23, четвертый 24-31). Записываем третий октет адреса и маски в двоичном виде:

$$114 = 0\cdot2^7+1\cdot2^6+1\cdot2^5+1\cdot2^4+0\cdot2^3+0\cdot2^2+1\cdot2^1+0\cdot2^0 = 0111\ 0010$$

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	128	64	32	16	8	4	2	1
114	**0**	**1**	**1**	**1**	**0**	**0**	**1**	**0**
маска 19	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
SN	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**

Выполняем побитовую конъюнкцию и получаем значение 3 октета *0110 0000*. Переводим в десятичную систему счисления:

$$0110\ 0000 = 0\cdot2^7+1\cdot2^6+1\cdot2^5+1\cdot2^4+0\cdot2^3+0\cdot2^2+0\cdot2^1+0\cdot2^0 = 96$$

Не забудем о том, что остальные октеты тоже подвергаются побитовой конъюнкции. Но первый и второй октеты маски заполнены единицами, следовательно, значения в IP-адресе остаются без изменений (фиксированные биты). Четвертый октет маски состоит из нулей (свободные биты), и при конъюнкции обнулится.

Ответ: 172.16.96.0.

Задача 2.3.3.

Подсеть задана IP-адресом входящего в нее узла и маской 172.16.114.159/19. Найти адрес широковещательной рассылки (BROADCAST) для данной подсети.

Решение.

BROADCAST – последний по порядку адрес в подсети. Для его вычисления необходимо заполнить все свободные биты единицами.

Определяем границу между фиксированными и свободными битами. Она проходит в третьем октете, т.к. в маске 19 бит (в октетах по 8 бит, первый октет 0-7, второй 8-15, третий 16-23, четвертый 24-31). Записываем третий октет адреса и маски в двоичном виде:

$$114 = 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 0111\ 0010$$

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	128	64	32	16	8	4	2	1
114	**0**	**1**	**1**	**1**	**0**	**0**	**1**	**0**
маска 19	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
ВС	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Заполняем свободные биты единицами и получаем значение 3 октета *0111 1111*. Переводим в десятичную систему счисления:

$$0111\ 1111 = 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 127$$

Не забудем о том, что последний (четвертый) октет целиком состоит из свободных бит, и их тоже необходимо заполнить единицами.

Ответ: 172.16.127.255.

Задача 2.3.4.

Даны адреса SUBNET 192.168.16.0 и BROADCAST 192.168.31.255. Определить маску подсети.

Решение.

SUBNET и BROADCAST – это первый и последний по порядку адреса в подсети. Они определяют границы подсети. Все совпадающие биты этих адресов будут фиксированными, следовательно, соответствующие биты маски будут единицами, а не совпадающие (свободные) – нулями.

Ищем различия. Первый (192) и второй (168) октеты полностью совпадают, различия начинаются в третьем октете. Переводим в двоичную систему счисления.

$$16 = 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 0001\ 0000$$

$$31 = 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 0001\ 1111$$

Видим, что старшие 4 бита в 3-ем октете совпадают, а младшие 4 – различаются, следовательно, общее количество фиксированных бит (и единиц в маске) равно 20 = 8 (первый октет целиком) + 8 (второй октет целиком) + 4 (в третьем октете). Получим третий октет маски *1111 0000*.

$1111\ 0000 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 240$

Последний октет целиком состоит из свободных бит, следовательно, соответствующие биты маски будут нулями.

Ответ: 255.255.240.0.

Задача 2.3.5.

Определите, входит ли узел с IP-адресом 172.16.156.140 в подсеть 172.16.0.0/17. Ответ обоснуйте.

Решение.

Границы любой подсети определяются адресами SUBNET и BROADCAST – первым и последним по порядку адресами в подсети. Для того, чтобы адрес входил в подсеть, он должен попадать в отрезок между этими адресами. Задача может быть решена различными способами.

1 способ решения.

Определим границы подсети и проверим, попадаем ли проверяемый адрес между ними. Нижняя граница задана в явном виде (адрес SUBNET дан). Определим верхнюю границы – найдем BROADCAST для данной подсети. Для этого заполним свободные биты единицами.

Определяем границу между фиксированными и свободными битами. Она проходит в третьем октете, т.к. в маске 17 бит (в октетах по 8 бит, первый октет 0-7, второй 8-15, третий 16-23, четвертый 24-31). Записываем третий октет адреса подсети и маски в двоичном виде:

$0 = 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 0000\ 0000$

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	128	64	32	16	8	4	2	1
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
маска 17	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
ВС	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Заполняем свободные биты единицами и получаем значение 3 октета *0111 1111*. Переводим в десятичную систему счисления:

$0111\ 1111 = 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 127$

Не забудем о том, что последний (четвертый) октет целиком состоит из свободных бит, и их тоже необходимо заполнить единицами. Получаем BROADCAST 172.16.127.255.

Итак, границы подсети: 172.16.0.0 – 172.16.127.255. Данный адрес 172.16.156.140 не входит в подсеть, т.к. не попадает между в отрезок между адресами SN и BC.

2 способ решения.

Принадлежность узла к подсети определяется по совпадению фиксированных бит IP-адресов: у всех адресов, входящих в подсеть (включая и первый адрес – SUBNET), они совпадают. Значит, данный адрес будет входить в подсеть тогда и только тогда, когда его фиксированная часть полностью совпадет с фиксированной частью адреса SUBNET (который гарантированно входит в подсеть).

Определяем границу между фиксированными и свободными битами. Она проходит в третьем октете, т.к. в маске 17 бит (в октетах по 8 бит, первый октет 0-7, второй 8-15, третий 16-23, четвертый 24-31). Записываем третий октет адреса подсети, проверяемого адреса и маски в двоичном виде:

$0 = 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$ = *0000 0000*

$156 = 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$ = *1001 1100*

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	128	64	32	16	8	4	2	1
SN=0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
IP=156	**1**	**0**	**0**	**1**	**1**	**1**	**0**	**0**
маска 17	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Как видно, фиксированные биты различаются, следовательно, эти адреса не могут принадлежать к одной сети.

3 способ решения.

Если узел принадлежит данной подсети, то его вычисленный адрес $SUBNET_2$ должен совпасть с адресом SUBNET из условия задачи.

SUBNET – первый по порядку адрес в подсети, у которого все свободные биты равны 0, вычисляется наложением маски, т.е. побитовой конъюнкцией IP-адреса и маски подсети.

Определим, в каком по счету октете маски проходит граница между фиксированными и свободными битами (между единицами и нулями). В маске 17 бит – это третий октет (в октетах по 8 бит, первый октет 0-7, второй 8-15, третий 16-23, четвертый 24-31). Записываем третий октет адреса и маски в двоичном виде:

$156 = 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$ = *1001 1100*

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	128	64	32	16	8	4	2	1
IP=156	**1**	**0**	**0**	**1**	**1**	**1**	**0**	**0**
маска 17	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
SN_2	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Выполняем побитовую конъюнкцию и получаем значение 3 октета *1000 0000*. Переводим в десятичную систему счисления:

$$1000\ 0000 = 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 128$$

Не забудем о том, что остальные октеты тоже подвергаются побитовой конъюнкции. Но первый и второй октеты маски заполнены единицами, следовательно, значения в IP-адресе остаются без изменений (фиксированные биты). Четвертый октет маски состоит из нулей (свободные биты), и при конъюнкции обнулится.

Вычисленный адрес $SUBNET_2$ = 172.16.128.0, данный по условию адрес SUBNET = 172.16.0.0. Адреса не совпадают, значит, подсети разные. Значит, адрес не принадлежит подсети.

Ответ: не принадлежит.

Задача 2.3.6.

Найти минимальную по размеру подсеть, в которую войдут адреса 192.168.26.32 и 192.168.31.169.

Решение.

Чтобы адреса входили в одну подсеть, их фиксированные биты должны совпадать. Необходимо определить маску так, чтобы она фиксировала совпадающие биты этих адресов.

Размер подсети определяется количеством узлов. Для того, чтобы он был минимальным, необходимо, чтобы в маске подсети было максимально возможное количество фиксированных бит (тогда будет меньше свободных, и меньше узлов: $N_y=2^{N0}$).

Ищем различия. Первый (192) и второй (168) октеты полностью совпадают, различия начинаются в третьих октетах. Переводим их в двоичную систему счисления.

$$26 = 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 0001\ 1010$$

$$31 = 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 0001\ 1111$$

Видим, что старшие 5 бит совпадают, а младшие 3 – различаются, следовательно, общее количество фиксированных бит (а значит, и единиц в маске) равно 21 = 8 (первый октет целиком) + 8 (второй октет целиком) + 5 (в третьем октете). Получим третий октет маски *1111 1000*.

$$1111\ 1000 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 240$$

Для вычисления адреса подсети (SUBNET) выполняем побитовую конъюнкцию любого из заданных в условии адресов на маску, т.е. обнуляем свободные биты (три младших в третьем октете). Получаем значение третьего октета *0001 1000 = 24*. Собираем адрес SN целиком: 192.168.24.0.

Ответ: 192.168.24.0/21.

2.4. Задачи для тренировки

Упражнение 2.4.1.

Подсеть задана адресом и маской подсети. Найти количество узлов, адреса SUBNET и BROADCAST:

1. 192.168.84.179 255.255.255.248
2. 34.76.169.200 255.255.255.192
3. 74.24.15.96 255.255.240.0
4. 117.236.137.19 255.255.255.252
5. 17.41.167.14/22
6. 178.35.12.48/18
7. 64.12.200.84/20
8. 85.176.11.23/17

Упражнение 2.4.2.

Найти минимальную по размеру подсеть, в которую войдут адреса:

1. 146.18.57.200 146.18.57.242
2. 15.24.48.190 15.24.74.10
3. 176.145.127.13 176.174.15.84
4. 185.24.75.17 185.24.75.96
5. 84.176.88.45 84.179.91.34

Упражнение 2.4.3.

Определить, входит ли адрес в подсеть:

1. 182.158.74.224/29 182.158.74.239

2. 34.65.13.64/27 34.65.13.102
3. 73.68.64.0/19 73.68.120.24
4. 118.47.240.192/28 118.47.240.255
5. 52.194.128.0/20 52.194.131.84

2.5. Рекомендуемая литература и Интернет-ресурсы

1. Основы компьютерных сетей. Тема № 3. Протоколы нижних уровней (транспортного, сетевого и канального) [электронный ресурс]. –Режим доступа: https://habr.com/post/308636/ – Заглавие с экрана. Дата обращения: 09.10.2018.

2. Основы компьютерных сетей. Тема № 5. Понятие IP адресации, масок подсетей и их расчет) [электронный ресурс]. – Режим доступа: https://habr.com/post/314484/ – Заглавие с экрана. Дата обращения: 09.10.2018.

3. Компьютерные сети for dummies [электронный ресурс]. – Режим доступа: https://habr.com/post/335816/ – Заглавие с экрана. Дата обращения: 09.10.2018.

Семинар 3.
Планирование пространства адресов. Разбиение на подсети

3.1. Цель и задачи семинара

На семинаре рассматриваются вопросы планирования адресного пространства сети при ее проектировании, сегментирование и разбиение на подсети, суммирование адресов. Рассматриваются примеры задач по планированию пространства адресов в информационно-телекоммуникационной сети условной компании, где необходимо спланировать пространство адресов с учетом ее структуры и количества рабочих мест в каждом отделе.

3.2. Теоретическая часть

3.2.1. Сегментирование сети

Планирование пространства IP-адресов – один из важнейших этапов разработки проекта будущей сети. Правильное распределение адресов позволит сильно упростить жизнь сетевых инженеров при внедрении проекта, тогда как ошибки планирования могут повлечь за собой не только неэффективную работу некоторых сетевых протоколов (например, протоколов маршрутизации), но серьезные риски информационной безопасности. И вот почему.

Представим себе организацию, состоящую из нескольких структурных подразделений (для краткости будем называть их отделами): менеджмент, бухгалтерия, кадры, логистика, продажи. Очевидно, что рядовому сотруднику отдела продаж со стажем работы в два дня совсем необязательно иметь доступ в сеть топ-менеджмента. Причина проста: он легко может оказаться «засланным казачком» от конкурентов, направленным в нашу организацию с какой-то вредоносной целью, например, коммерческий шпионаж. Если обобщить выводы из рассмотренного примера, получим одно из фундаментальных правил информационной безопасности: разрешен должен быть только тот трафик, который действительно необходим, все лишнее и необязательное следует запретить. Таким образом мы приходим к идее необходимости фильтрации (или хотя бы контроля) сетевого трафика не только на периметре, но и внутри корпоративной сети организации.

Вернемся к нашему примеру. Пусть в организации 200 рабочих мест, оснащенных компьютерами. Интернет-провайдер выделил нам блок адресов

192.168.1.0/24. Маска имеет 8 свободных бит, это 256 адресов – хватает для всех устройств, можно использовать этот блок адресов. Получим сеть, изображенную на рис. 3.1.

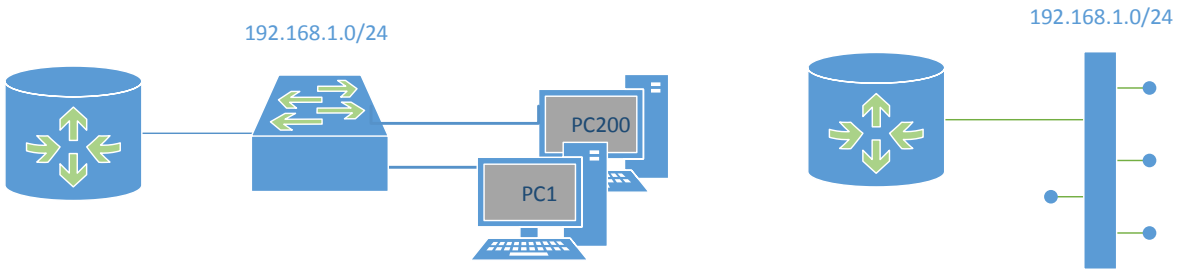


Рис. 3.1. Физическая (слева) и логическая (справа) топология общей сети

В рассмотренном примере все устройства принадлежат единой и единственной сети 192.168.1.0/24 с общим шлюзом. Доставка пакетов в этой сети осуществляется коммутатором (вернее, несколькими коммутаторами, т.к. устройств достаточно много – 200), т.е. средствами и протоколами канального уровня. Мы уже знаем, что коммутаторы не имеют гибких механизмов контроля и фильтрации трафика на интерфейсах, кроме грубого запрета отдельных MAC-адресов. Следовательно, наша задача средствами только канального уровня решена быть не может.

Однако эта задача может быть сравнительно несложно решена силами маршрутизатора, который проводит анализ заголовков пакета при его продвижении, следовательно, может вести учет и контроль проходящего трафика и даже выборочно блокировать его. Остается провести весь внутренний трафик через маршрутизатор. Но это возможно только в том случае, если узлы, передающие друг другу трафик, будут подключены к разным интерфейсам маршрутизатора. При этом мы помним, что несколько интерфейсов маршрутизатора не могут принадлежать одной и той же сети. Мы пришли к необходимости разбиения нашей общей сети 192.168.1.0/24 на подсети (рис.3.2).

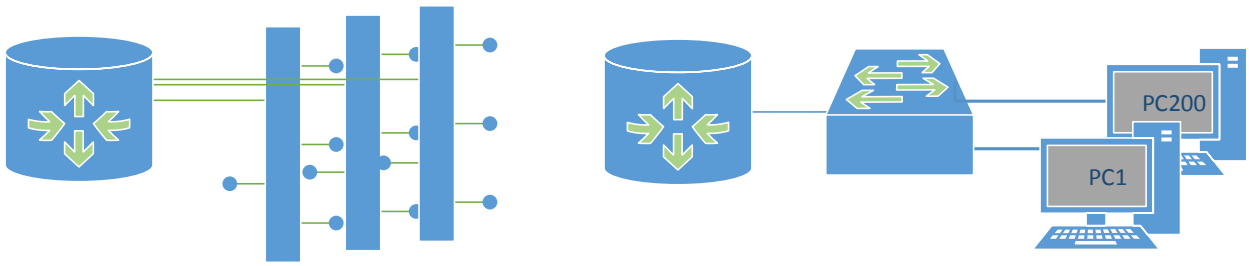


Рис. 3.2. Логическая (слева) и физическая (справа) топология сегментированной сети

Отметим, что современные протоколы канального уровня и программное обеспечение управляемых коммутаторов позволяет изменить логическую топологию сети без изменения физической – на основе технологии виртуальных локальных сетей (VLAN).

3.2.2. Виртуальные локальные сети (VLAN)

VLAN (аббр. от англ. virtual local area network) – виртуальная локальная компьютерная сеть, представляет собой группу хостов с общим набором требований, которые взаимодействуют так, как если бы они были подключены к общему широковещательному домену, независимо от их физического местонахождения. VLAN имеет те же свойства, что и физическая локальная сеть, но позволяет конечным узлам группироваться вместе, даже если они не находятся в одной физической сети. Такая реорганизация может быть сделана на основе программного обеспечения вместо физического перемещения устройств.

Идентификатором виртуальной сети является ее номер (десятичное число от 0 до 4095). Этот идентификатор добавляется в специальное поле заголовка кадра (тег), после чего тегированный кадр передается коммутаторами только по транковым портам (см. ниже) и по портам своего VLAN-а.

По умолчанию на каждом порту коммутатора имеется сеть VLAN1 или VLAN управления. Сеть управления не может быть удалена, однако могут быть созданы дополнительные сети VLAN и этим альтернативным VLAN могут быть дополнительно назначены порты.

Существует несколько способов определения членства в той или иной виртуальной сети для входящего на коммутатор нетегированного кадра.

1. По порту (англ. port-based, IEEE 802.1Q): порту коммутатора вручную назначается один VLAN. В случае, если одному порту должны соответствовать сразу несколько VLAN-ов (например, если домен VLAN распространяется на несколько коммутаторов), то этот порт должен быть членом транка. Коммутатор будет добавлять метки данной VLAN ко всем принятым кадрам, не имеющим никаких меток. VLAN, построенные на базе портов, имеют некоторые ограничения. Они очень просты в установке, но позволяют поддерживать для каждого порта только один VLAN.

Следовательно, такое решение неприемлемо при использовании концентраторов или в сетях с мощными серверами, к которым обращается много пользователей (сервер не удастся включить в разные VLAN). Кроме того, вносить изменения в VLAN на основе портов достаточно сложно, поскольку при каждом изменении требуется физическое переключение устройств. В рамках настоящего курса рассматриваются только port-based VLAN-ы.

2. По MAC-адресу (MAC-based): членство в VLAN основывается на MAC-адресе узла. Для этого необходимо, чтобы в таблице MAC-адресов коммутатора содержалась информация о том, какому VLAN принадлежит тот или иной MAC-адрес. Если сама таблица не поддерживает такое хранение, необходимо обеспечить коммутатору доступ к удаленной таблице соответствия (например, сервер VMPS – VLAN membership policy server).

3. По протоколу (Protocol-based): данные 3-4 уровня в заголовке пакета используются чтобы определить членство в VLAN. Например, IP-машины могут быть переведены в первую VLAN, а AppleTalk-машины во вторую. Основной недостаток этого метода в том, что он нарушает независимость уровней, поэтому, например, переход с IPv4 на IPv6 приведет к нарушению работоспособности сети.

4. Методом аутентификации (Authentication-based): устройства могут быть автоматически перемещены в VLAN, основываясь на данных аутентификации.

Внедрение технологии VLAN обладает рядом неоспоримых преимуществ:

- Облегчается перемещение, добавление устройств и изменение их соединений друг с другом.
- Достигается большая степень административного контроля вследствие наличия устройства, осуществляющего между сетями VLAN маршрутизацию на третьем уровне.
- Уменьшается потребление полосы пропускания по сравнению с ситуацией одного широковещательного домена.
- Сокращается непроизводственное использование CPU за счет сокращения пересылки широковещательных сообщений.
- Обеспечивается предотвращение широковещательных штормов и предотвращение петель.

3.2.3. Разбиение сети на подсети

Вернемся к нашему примеру. Мы остановились на необходимости разделить общую сеть 192.168.1.0/24 на несколько подсетей. Границами этих подсетей выберем границы отделов нашей организации: все компьютеры менеджмента разместим в одной подсети, бухгалтерии – в другой, и т.д. Пусть количество компьютеров распределено по отделам так, как показано в таблице 3.1.

Таблица 3.1. Распределение компьютеров по отделам

Наименование отдела	Количество рабочих мест в отделе
Менеджмент	10
Бухгалтерия	20
Кадры	20
Логистика	55
Продажи	110
ИТОГО	200

Итак, количество узлов в подсетях определено. Остается выбрать для каждой из них подходящие маски и рассчитать границы. Здесь важно отметить ключевое правило разбиения на подсети: его ***всегда*** следует начинать с наибольшей подсети, и далее рассматривать все подсети ***строго по убыванию*** числа узлов. Почему именно – выясним несколько позже.

Выбираем маску для подсети отдела продаж в 110 узлов. Во-первых, не следует забывать, что в каждой подсети помимо собственно адресов узлов есть два служебных адреса: SUBNET и BROADCAST, следовательно, адресов в подсети на самом деле должно быть минимум 112. Для данной подсети это никак не влияет на решение, но в некоторых случаях может оказаться важным, например, если узлов в сети должно быть 15 или 31. Ближайшая от 112 сверху степень двойки – $128 = 2^7$, в маске будет 7 свободных бит. Если общий адрес разбиваемой подсети 192.168.1.0/24, то (накладываем на него маску в $32 - 7 = 25$ бит) SUBNET равен 192.168.1.0, маска /25 или 255.255.255.128, BROADCAST 192.168.1.127.

Рассмотрим следующую подсеть. В порядке убывания количества узлов это будет подсеть отдела логистики с 55 узлами. Полное количество адресов равно $55 + 2 = 57$ адресов, это меньше (либо равно) $64 = 2^6$ – свободных бит в маске будет 6, а сама маска $32 – 6 = 26$ бит. Часть адресов исходной сети уже занята подсетью отдела продаж, ее последний адрес (BROADCAST)

192.168.1.127. Значит, первым свободным адресом будет следующий за ним – 192.168.1.128. Накладываем на него маску /26 и получаем для сети отдела логистики: SUBNET = 192.168.1.128, маска /26 или 255.255.255.192, BROADCAST 192.168.1.191.

По аналогии рассчитываем остальные 3 отдела, данные заносим в таблицу 3.2.

Таблица 3.2. Расчет разбиения на подсети

Наименование отдела	Количество адресов в подсети отдела	SUBNET и маска	BROADCAST
Менеджмент	**$5 + 2 \leq 8 = 2^3$**	**192.168.1.240/29**	**192.168.1.247**
Бухгалтерия	**$10 + 2 \leq 16 = 2^4$**	**192.168.1.224/28**	**192.168.1.239**
Кадры	**$20 + 2 \leq 32 = 2^5$**	**192.168.1.192/27**	**192.168.1.223**
Логистика	**$55 + 2 \leq 64 = 2^3$**	192.168.1.128/26	192.168.1.191
Продажи	**$110 + 2 \leq 8 = 2^3$**	192.168.1.0/25	192.168.1.127

Задача решена. Остается создать на маршрутизаторе соответствующие виртуальные интерфейсы (т.н. субинтерфейсы), настроить на коммутаторах виртуальные сети, и можно управлять трафиком при его продвижении маршрутизатором с интерфейса на интерфейс.

Но вернемся на шаг назад. Рассмотрим, что бы случилось, если бы мы рассчитывали адреса подсетей в другом порядке, например, в том, в котором они приведены в таблице. Начинаем с сети менеджмента. В ней 5 узлов (7 адресов), нужно 3 свободных бита, значение маски 29. Это первая подсеть, значит начинаем с начала общей сети – 192.168.1.0. Накладываем маску /29, рассчитываем SUBNET = 192.168.1.0, BROADCAST = 192.168.1.7. Переходим к следующей сети, это бухгалтерия, 10 узлов. Для 12 адресов потребуется 4 свободных бита, маска /28, накладываем ее на первый свободный адрес, оставшийся после первой сети – 192.168.1.8. При наложении получаем, что SUBNET опять равен 192.168.1.0. Это неверно, о чем обязательно сообщит маршрутизатор при попытке применить такие настройки. Чтобы не было пересечения первой и второй сети, следует пропустить адреса с 192.168.1.8 до 192.168.1.15 и начать сеть бухгалтерии с адреса 192.168.1.16 – тогда всем свободным битам будут соответствовать нули, как и должно быть в адресе SUBNET. Такой подход приведет к ряду негативных последствий, обусловленных тем, что пространстве адресов образуется промежуток из неиспользуемых адресов. Во-первых, это снижает эффективность расходования

адресов (в рассматриваемом примере их вообще не хватит для всех остальных подсетей). Во-вторых, создает проблемы маршрутизаторам при использовании адаптивных протоколов маршрутизации. В-третьих, повышает риски информационной безопасности. Именно поэтому крайне важно рассматривать подсети от большей к меньшей.

3.3. Тренировочные задания

Выделить в заданной сети подсети, подходящие для адресации указанного количества компьютеров соответственно:

№ п/п	Адрес сети	Количество узлов по подсетям
1.	192.168.1.128/25	2, 5, 30, 20, 10
2.	192.168.2.0/24	15, 15, 100, 50, 30
3.	192.168.3.224/27	2, 2, 7, 3
4.	192.168.0.0/16	200, 200, 100, 2000, 1000, 500
5.	172.16.0.0/12	3000, 500, 2000, 300, 1000
6.	10.0.80.0/20	50, 100, 1000, 500, 500
7.	10.0.0.0/8	3000, 2000, 2000, 15000, 7500
8.	128.16.254.0/23	300, 200, 100, 50, 20
9.	128.16.254.0/23	100, 100, 100, 30, 40, 50
10.	128.16.254.0/23	50, 50, 100, 10, 20, 30, 4, 4

3.4. Рекомендуемая литература и Интернет-ресурсы

1. Основы компьютерных сетей. Тема № 3. Протоколы нижних уровней (транспортного, сетевого и канального) [электронный ресурс]. –Режим доступа: https://habr.com/post/308636/ – Заглавие с экрана. Дата обращения: 09.10.2018.

2. Основы компьютерных сетей. Тема № 5. Понятие IP адресации, масок подсетей и их расчет) [электронный ресурс]. – Режим доступа: https://habr.com/post/314484/ – Заглавие с экрана. Дата обращения: 09.10.2018.

3. Компьютерные сети for dummies [электронный ресурс]. – Режим доступа: https://habr.com/post/335816/ – Заглавие с экрана. Дата обращения: 09.10.2018.

Семинар 4.
Бесклассовая адресация. CIDR и VLSM

4.1. Цель и задачи семинара

Заключительный семинар по IPv4 адресам посвящен разбору и решению сложных задач. На примерах иллюстрируются возможности построения пространства адресов, которые открывает применение технологии VLSM (использование масок переменной длины). Формируются навыки описания произвольного диапазона IP-адресов в формате CIDR. По итогам семинара проводится контрольная работа на решение задач рассмотренного типа. Примеры задач приведены в фонде оценочных средств дисциплины.

4.2. Теоретическая часть

На семинаре 4 не рассматривается новая теория – проводится углубление практических навыков по теоретическим разделам, рассмотренным на семинарах 2-3.

4.3. Разбор типовых задач

Задача 4.3.1 (повышенный уровень сложности)

Дан диапазон адресов: 192.168.0.0 – 192.168.113.239 (назовем его диапазоном A). Требуется описать данный диапазон адресов в формате CIDR с точностью до адреса (т.е. не захватить ни одного лишнего адреса и не потерять ни одного, входящего в диапазон).

Решение.

В задаче требуется найти описание диапазона адресов A в формате CIDR. Это означает, что ответом к задаче является некоторый набор подсетей, заданных адресами SUBNET и масками, таких что, рассмотренные совместно, они совпадают с диапазоном A. Решим задачу итерационно. На каждом шаге для нахождения очередного искомого CIDR-диапазона будем подбирать такую подсеть, для которой выполняются два условия:

1) Начало (левый край) описываемого пространства адресов A должно являться адресом SUBNET для этой подсети. SUBNET – самый первый по порядку адрес, входящий в подсеть. Выполнение этого условия гарантирует, что мы не захватим ни одного лишнего адреса слева.

2) BROADCAST для этой подсети должен быть максимально близок к концу (правому краю) описываемого диапазона адресов A, но не должен превышать (т.е. должен быть меньше либо равен). Выполнение этого условия гарантирует, что мы не захватим ни одного лишнего адреса справа.

При этом каждая найденная подсеть будет закрывать часть исходного диапазона адресов A. Для перехода к следующему шагу необходимо определить новый диапазон описываемых адресов B (затем C, D и т.д.) – часть исходного диапазона A, которая пока осталась незакрытой найденными CIDR-подсетями (рис. 4.1). Для этого будем сдвигать левый край исходного диапазона так, чтобы он был первым следующим за BROADCAST-ом последней из найденных подсетей (т.е. на единицу больше). На последней итерации BROADCAST должен совпасть с правым краем исходного диапазона – это признак завершения решения.

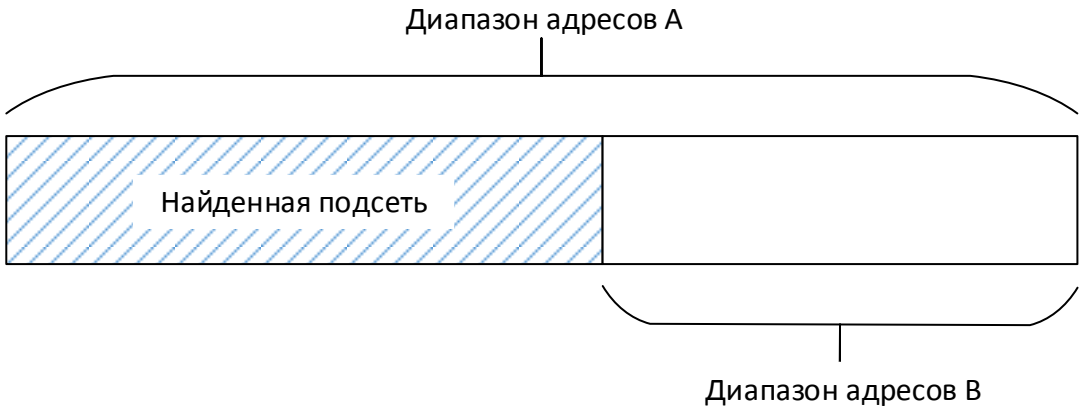


Рис. 4.1. Итерационный подход в решении задач на CIDR

Рассмотрим решение по шагам.

Шаг 1. Диапазон A: 192.168.0.0 – 192.168.113.239. Подберем такую подсеть, чтобы ее адрес SUBNET совпадал с 192.168.0.0, а BROADCAST не превышал 192.168.113.239. Отличительным признаком адреса SUBNET являются нули в свободных битах, следовательно, подбирая маску, мы должны выбирать только из тех значений, при которых ни одна из единиц адреса 192.168.0.0 не будет свободным битом.

192.168.0.0 = 1100 0000. 1010 1**000. 0000 0000. 0000 0000**

В конце адреса 18 нулей, значит, маска подсети не может быть короче, чем 32 – 18 = 13 бит, т.к. это привело бы к нарушению первого из условий, рассмотренных ранее.

При этом адрес BROADCAST не должен превышать правый край диапазона A: 192.168.113.239. Выбираем маску так, чтобы в третьем байте количество свободных бит позволяло максимально приблизиться от 0 к 113, но не превысить это значение. Рассмотрим все возможные маски, начиная с 13 бит, и выберем среди них первую подходящую.

- 192.168.0.0/13, BC = 192.175.255.255 – превышает 192.168.113.239;
- 192.168.0.0/14, BC = 192.171.255.255 – превышает 192.168.113.239;
- 192.168.0.0/15, BC = 192.169.255.255 – превышает 192.168.113.239;
- 192.168.0.0/16, BC = 192.168.255.255 – превышает 192.168.113.239;
- 192.168.0.0/17, BC = 192.168.127.255 – превышает 192.168.113.239;
- 192.168.0.0/18, BC = 192.168.63.255 – подходит.

Если присмотреться, можно сразу определить подходящее значение маски. Первые два байта начального и конечного адресов диапазона A (192.168) совпадают, отличия начинаются в третьем байте (0 и 113). Разница между значениями третьего байта начала и конца диапазона A равна 113 – 0 = 113. Подходящее значение маски должно обеспечить максимально близкое к 113 (меньшее или равное) количество значений в третьем байте. Количества значений определяются количеством свободных бит: все 8 свободны – будет $2^8 = 256$ значений, 7 – $2^7 = 128$ и т.д. В нашем случае необходимо 6 свободных бит, 64 значения, т.к. при свободных 7 битах различных значений третьего байта слишком много, что приведет к выходу адреса BROADCAST за правую границу диапазона A. Итак, 6 свободных бит в третьем байте – это маска длиной 18 бит, при которой BROADCAST = 192.168.63.255.

Сразу сделаем оговорку, что количества значений, ровно на 1 меньшие какой-либо степени двойки, требуют особой проверки. Подробнее это будет проиллюстрировано на последующих шагах решения задачи.

Первый из искомых диапазонов найден, это сеть 192.168.0.0/18, и она закрывает (или описывает) адреса диапазона A от левого края (192.168.0.0) и до своего BROADCAST – 192.168.63.255 (рис.4.1). Оставшийся неописанным диапазон B: 192.168.64.0 – 192.168.113.239. Переходим к следующему шагу.

Шаг 2. Диапазон B: 192.168.64.0 – 192.168.113.239. Необходимо подобрать такую подсеть, чтобы ее адрес SUBNET совпадал с 192.168.64.0, а BROADCAST не превышал 192.168.113.239.

Заметим, что начало диапазона B по сравнению с диапазоном A сдвинулось вправо, в нем больше единиц. Это значит, что более короткие маски, чем использована на первом шаге, рассматривать не имеет смысла.

Более того, маску той же длины использовать также нельзя, т.к. две сети с одинаковой маской (например, 18 бит) в сумме дадут одну сеть с маской на 1 бит короче (в примере это 17 бит), а мы ранее выяснили, что маска такой длины (17 бит) не подходит. Это подтверждается при рассмотрении двоичного представления адреса:

192.168.64.0 = 1100 0000. 1010 1000. 01**00 0000. 0000 0000**

Первые два байта начального и конечного адресов диапазона B (192.168) по-прежнему совпадают, отличия начинаются в третьем байте (64 и 113). Разница между значениями третьего байта начала и конца диапазона B равна 113 – 64 = 49. Подходящее значение маски должно обеспечить максимально близкое к 49 количество значений в третьем байте. Подбираем соответствующую степень двойки – это $2^5 = 32$. Значит, в третьем байте нужно оставить свободными 5 бит, что дает маску длиной 19.

192.168.64.0/19, BC = 192.168.95.255

Вторая подсеть найдена – 192.168.64.0/19, и она закрывает (или описывает) адреса диапазона B от левого края (192.168.64.0) и до своего BROADCAST – 192.168.95.255. Оставшийся неописанным диапазон адресов C: 192.168.96.0 – 192.168.113.239. Переходим к следующему шагу.

Шаг 3. Диапазон C: 192.168.96.0 – 192.168.113.239. Необходимо подобрать такую подсеть, чтобы ее адрес SUBNET совпадал с 192.168.96.0, а BROADCAST не превышал 192.168.113.239.

192.168.96.0 = 1100 0000. 1010 1000. 011**0 0000. 0000 0000**

В конце 13 нулей, можем рассматривать маски, начиная с 19 бит. Но маска /19 уже была использована на предыдущем шаге. Рассматриваем, начиная со следующего значения – 20 бит.

– 192.168.96.0/20, BC = 192.168.111.255 – подходит.

Рассуждая иначе, придем к такому же выводу. Первые два байта начального и конечного адресов диапазона C (192.168) по-прежнему совпадают, отличия начинаются в третьем байте (96 и 113). Разница между значениями третьего байта начала и конца диапазона C равна 113 – 96 = 17. Подходящее значение маски должно обеспечить максимально близкое к 17 количество значений в третьем байте. Подбираем соответствующую (ближайшую снизу) степень двойки – это $2^4 = 16$. Значит, в третьем байте нужно оставить свободными 4 бита, что дает маску длиной 20 бит и BROADCAST 192.168.111.255.

Третья подсеть найдена – 192.168.96.0/20, и она закрывает (или описывает) адреса диапазона C от левого края (192.168.96.0) и до своего BROADCAST – 192.168.111.255. Оставшийся неописанным диапазон адресов назовем D: 192.168.112.0 – 192.168.113.239. Переходим к следующему шагу.

Шаг 4. Диапазон D: 192.168.112.0 – 192.168.113.239. Необходимо подобрать такую подсеть, чтобы ее адрес SUBNET совпадал с 192.168.112.0, а BROADCAST не превышал бы 192.168.113.239.

192.168.112.0 = 1100 0000. 1010 1000. 0111 **0000. 0000 0000**

В конце 12 нулей, можем рассматривать маски, начиная с 20 бит. Но маска /20 уже была использована на предыдущем шаге. Рассматриваем, начиная со следующего значения – 21 бит.

- 192.168.112.0/21, BC = 192.168.119.255 – превышает 192.168.113.239;
- 192.168.112.0/22, BC = 192.168.115.255 – превышает 192.168.113.239;
- 192.168.112.0/23, BC = 192.169.113.255 – превышает 192.168.113.239;
- 192.168.112.0/24, BC = 192.168.112.255 – подходит.

Рассуждая иначе, придем к такому же выводу. Первые два байта начального и конечного адресов диапазона D (192.168) совпадают, отличия начинаются в третьем байте (112 и 113). Разница между значениями третьего байта начала и конца диапазона D равна 113 – 112 = 1. Подходящее значение маски должно обеспечить максимально близкое к 1 количество значений в третьем байте. Подбираем соответствующую (ближайшую снизу) степень двойки – это $2^0 = 1$. Значит, в третьем байте нужно оставить свободными 0 бит, что дает маску длиной 24 бита и BROADCAST 192.168.112.255.

Здесь необходимо сделать оговорку. Число 1 (113 – 112) это не только 2^0, но и $2^1 – 1$, а значит, требует особой проверки. Так, если бы значение четвертого байта в диапазоне D (а, значит, и в исходном диапазоне A) было равно 255, нам бы следовало выбрать количество свободных бит в третьем байте, равное не 0, а 1. Это обусловлено тем, что при подсчете количества значений счет начинается не с единицы, а с нуля.

Четвертая подсеть найдена – 192.168.112.0/24, и она описывает адреса диапазона D от левого края (192.168.112.0) и до своего BROADCAST – 192.168.112.255. Оставшийся неописанным диапазон адресов назовем E: 192.168.113.0 – 192.168.113.239. Переходим к следующему шагу.

Шаг 5. Диапазон E: 192.168.113.0 – 192.168.113.239. Необходимо подобрать такую подсеть, чтобы ее адрес SUBNET совпадал с 192.168.113.0, а BROADCAST не превышал бы 192.168.113.239.

192.168.113.0 = 1100 0000. 1010 1000. 0111 0001. **0000 0000**

В конце 8 нулей, можем рассматривать маски, начиная с 24 бит. Но маска /24 уже была использована на предыдущем шаге. Рассматриваем, начиная со следующего значения – 25 бит.

– 192.168.113.0/25, BC = 192.168.113.127 – подходит.

Рассуждая иначе, придем к такому же выводу. Теперь уже первые три байта начального и конечного адресов диапазона E (192.168.113) совпадают, отличия только в четвертом байте (0 и 239). Разница между значениями третьего байта начала и конца диапазона E равна 239 – 0 = 239. Подбираем ближайшую снизу степень двойки – это $2^7 = 128$. Значит, в четвертом байте нужно оставить свободными 7 бит, что дает маску длиной 25 бит и BROADCAST 192.168.113.127.

Пятая подсеть найдена – 192.168.113.0/25, и она описывает адреса диапазона E от левого края (192.168.113.0) и до своего BROADCAST – 192.168.113.127. Оставшийся неописанным диапазон адресов назовем F: 192.168.113.128 – 192.168.113.239. Переходим к следующему шагу.

Шаг 6. Диапазон F: 192.168.113.128 – 192.168.113.239. Необходимо подобрать такую подсеть, чтобы ее адрес SUBNET совпадал с 192.168.113.128, а BROADCAST не превышал бы 192.168.113.239.

192.168.113.128 = 1100 0000. 1010 1000. 0111 0001. 1**000 0000**

В конце 7 нулей, можем рассматривать маски, начиная с 25 бит. Но маска /25 уже была использована на предыдущем шаге. Рассматриваем, начиная со следующего значения – 26 бит.

– 192.168.113.128/26, BC = 192.168.113.191 – подходит.

Рассуждая иначе, придем к такому же выводу. Первые три байта начального и конечного адресов диапазона F (192.168.113) совпадают, отличия только в четвертом байте (128 и 239), разница между ними равна 239 – 128 = 111. Подбираем ближайшую снизу степень двойки – это $2^6 = 64$. Значит, в четвертом байте нужно оставить свободными 7 бит, что дает маску длиной 26 бит и BROADCAST 192.168.113.191.

Шестая подсеть найдена – 192.168.113.128/26, и она описывает адреса диапазона F от левого края (192.168.113.128) и до своего BROADCAST – 192.168.113.191. Оставшийся неописанным диапазон адресов назовем G: 192.168.113.192 – 192.168.113.239. Переходим к следующему шагу.

Шаг 7. Диапазон G: 192.168.113.192 – 192.168.113.239. Необходимо подобрать такую подсеть, чтобы ее адрес SUBNET совпадал с 192.168.113.192, а BROADCAST не превышал бы 192.168.113.239.

192.168.113.192 = 1100 0000. 1010 1000. 0111 0001. 11**00 0000**

В конце 6 нулей, можем рассматривать маски, начиная с 26 бит. Но маска /26 уже была использована на предыдущем шаге. Рассматриваем, начиная со следующего значения – 27 бит.

– 192.168.113.192/27, BC = 192.168.113.223 – подходит.

Рассуждая иначе, придем к такому же выводу. Первые три байта начального и конечного адресов диапазона G (192.168.113) совпадают, отличия только в четвертом байте (192 и 239), разница между ними равна 239 – 192 = 47. Подбираем ближайшую снизу степень двойки – это $2^5 = 32$. Значит, в четвертом байте нужно оставить свободными 5 бит, что дает маску длиной 27 бит и BROADCAST 192.168.113.223.

Седьмая подсеть найдена – 192.168.113.192/27, и она описывает адреса диапазона G от левого края (192.168.113.192) и до своего BROADCAST – 192.168.113.223. Оставшийся неописанным диапазон адресов назовем H: 192.168.113.224 – 192.168.113.239. Переходим к следующему шагу.

Шаг 8. Диапазон H: 192.168.113.224 – 192.168.113.239. Необходимо подобрать такую подсеть, чтобы ее адрес SUBNET совпадал с 192.168.113.224, а BROADCAST не превышал бы 192.168.113.239.

192.168.113.224 = 1100 0000. 1010 1000. 0111 0001. 111**0 0000**

В конце 5 нулей, можем рассматривать маски, начиная с 27 бит. Но маска /27 уже была использована на предыдущем шаге. Рассматриваем, начиная со следующего значения – 28 бит.

– 192.168.113.224/28, BC = 192.168.113.239 – подходит.

Рассуждая иначе, придем к такому же выводу. Первые три байта начального и конечного адресов диапазона H (192.168.113) совпадают, отличия только в четвертом байте (224 и 239), разница между ними равна 239 – 224 = 15. Обычно мы подбирали ближайшую снизу степень двойки – это $2^3 = 8$. Но значения, ровно на единицу меньше, чем степени двойки, требуют особой проверки, т.к. 15 – это шестнадцатое значение байта, если начинать считать не с единицы, а с нуля. Значит, в четвертом байте нужно оставить свободными не 3, а 4 бита, что дает маску длиной 28 бит и BROADCAST, в точности совпадающий с концом диапазона H (и A): 192.168.113.239.

Восьмая и последняя подсеть найдена – 192.168.113.224/28. Оставшихся неописанными адресов назовем не осталось. Задача решена.

Ответом к задаче являются все 8 найденных подсетей:

1. 192.168.0.0/18 (до 192.168.63.255);
2. 192.168.64.0/19 (до 192.168.96.255);
3. 192.168.96.0/20 (до 192.168.111.255);
4. 192.168.112.0/24 (до 192.168.112.255);
5. 192.168.113.0/25 (до 192.168.113.127);
6. 192.168.113.128/26 (до 192.168.113.191);
7. 192.168.113.191/27 (до 192.168.113.223);
8. 192.168.113.224/28 (до 192.168.113.239).

Заметим тенденцию: длина маски от шага к шагу увеличивается на 1 или более бит. Отметим, что найденное разбиение является единственно верным для данного диапазона адресов A.

Задача 4.3.2 (высокий уровень сложности)

Дан диапазон адресов: 191.189.216.100 – 192.168.113.239 (назовем его диапазоном A). Требуется описать данный диапазон адресов в формате CIDR с точностью до адреса (т.е. не захватить ни одного лишнего адреса и не потерять ни одного, входящего в диапазон).

Решение.

На первый взгляд, задача очень похожа на предыдущую. Есть диапазон адресов, заданный начальным и конечным значениями, нужно описать его набором подсетей. Существенное отличие заключается в левой границе диапазона A: совпадающих байт нет, отличия с первого байта. При этом ненулевые значения младших байт не позволяют применять короткие маски (см. условие 1 в разборе задачи 4.3.1), что приводит к существенному увеличению трудоемкости решения задачи.

Тем не менее, подход к решению задачи аналогичный – итерационный. На каждом шаге для нахождения очередного CIDR-диапазона будем подбирать такую подсеть, для которой выполняются два условия:

1) Начало (левый край) описываемого диапазона адресов A должно являться адресом SUBNET для этой подсети. SUBNET – самый первый по порядку адрес, входящий в подсеть. Выполнение этого условия гарантирует, что мы не захватим ни одного лишнего адреса слева.

2) BROADCAST для этой подсети должен быть максимально близок к концу (правому краю) описываемого диапазона адресов A, но не должен

превышать (т.е. должен быть меньше либо равен). Выполнение этого условия гарантирует, что мы не захватим ни одного лишнего адреса справа.

При этом каждая найденная подсеть будет закрывать часть исходного диапазона адресов A. Для перехода к следующему шагу необходимо определить новый диапазон описываемых адресов B (затем C, D и т.д.) – часть исходного диапазона A, которая пока осталась незакрытой найденными CIDR-подсетями (рис. 4.1). Для этого будем сдвигать левый край исходного диапазона так, чтобы он был первым следующим за BROADCAST-ом последней из найденных подсетей (т.е. на единицу больше). На последней итерации BROADCAST должен совпасть с правым краем исходного диапазона – это признак завершения решения.

Шаг 1. Диапазон A: 191.189.216.100 – 192.168.113.239. Подберем такую подсеть, чтобы ее адрес SUBNET совпадал с 191.189.216.100, а BROADCAST не превышал 192.168.113.239. Отличительным признаком адреса SUBNET являются нули в свободных битах, следовательно, подбирая маску, мы должны выбирать только из тех значений, при которых ни одна из единиц адреса 191.189.216.100 не будет свободным битом.

191.189.216.100 = 1011 1111. 1011 1101. 1101 1000. 0110 01**00**

В конце адреса всего 2 нуля, что ограничивает использование масок короче, чем 32 – 2 = 30 бит, т.к. это привело бы к нарушению первого из условий, рассмотренных ранее.

При этом адрес BROADCAST не должен превышать правый край диапазона A: 192.168.113.239. Очевидно, что при различиях уже в первом байте, маска 30 (а также все маски длиннее 8 бит) никак не может привести к превышению этого значения.

Первая подсеть найдена – 191.189.216.100/30 (до 191.189.216.103). Оставшийся неописанным диапазон адресов назовем B: 191.189.216.104 – 192.168.113.239. Переходим к следующему шагу.

Шаг 2. Диапазон: 191.189.216.104 – 192.168.113.239. На этом и последующих шагах рассуждаем аналогично предыдущему шагу.

191.189.216.104 = 1011 1111. 1011 1101. 1101 1000. 0110 1**000**

В конце адреса 3 нуля, можем использовать маску 32 – 3 = 29 бит. Маска длиннее 8 бит, поэтому правый край можем пока не проверять.

191.189.216.104/29, BC = 191.189.216.111 – вторая подсеть найдена.

Шаг 3. Диапазон: 191.189.216.112 – 192.168.113.239.

191.189.216.112 = 1011 1111. 1011 1101. 1101 1000. 0111 **0000**

В конце адреса 4 нуля, можем использовать маску 32 – 4 = 28 бит. Маска длиннее 8 бит, поэтому правый край можем пока не проверять.

191.189.216.112/28, BC = 191.189.216.127 – третья подсеть найдена.

Шаг 4. Диапазон: 191.189.216.128 – 192.168.113.239.

191.189.216.128 = 1011 1111. 1011 1101. 1101 1000. 1**000 0000**

В конце адреса 7 нулей, можем использовать маску 32 – 7 = 25 бит. Маска длиннее 8 бит, поэтому правый край можем пока не проверять.

191.189.216.128/25, BC = 191.189.216.255 – четвертая подсеть найдена.

Шаг 5. Диапазон: 191.189.217.0 – 192.168.113.239.

191.189.217.0 = 1011 1111. 1011 1101. 1101 1001. **0000 0000**

В конце адреса 8 нулей, можем использовать маску 32 – 8 = 24 бит. Маска длиннее 8 бит, поэтому правый край можем пока не проверять.

191.189.217.0/24, BC = 191.189.217.255. Пятая подсеть найдена.

Шаг 6. Диапазон: 191.189.218.0 – 192.168.113.239.

191.189.218.0 = 1011 1111. 1011 1101. 1101 101**0. 0000 0000**

В конце адреса 9 нулей, можем использовать маску 32 – 9 = 23 бита. Маска длиннее 8 бит, поэтому правый край можем пока не проверять.

191.189.218.0/23, BC = 191.189.219.255. Шестая подсеть найдена.

Шаг 7. Диапазон: 191.189.220.0 – 192.168.113.239.

191.189.220.0 = 1011 1111. 1011 1101. 1101 11**00. 0000 0000**

В конце адреса 10 нулей, можем использовать маску 32 – 10 = 22 бита. Маска длиннее 8 бит, поэтому правый край можем пока не проверять.

191.189.220.0/22, BC = 191.189.223.255. Седьмая подсеть найдена.

Шаг 8. Диапазон: 191.189.224.0 – 192.168.113.239.

191.189.224.0 = 1011 1111. 1011 1101. 111**0 0000. 0000 0000**

В конце адреса 13 нулей, можем использовать маску 32 – 13 = 19 бит. Маска длиннее 8 бит, поэтому правый край можем пока не проверять.

191.189.224.0/19, BC = 191.189.255.255. Восьмая подсеть найдена.

Шаг 9. Диапазон: 191.190.0.0 – 192.168.113.239.

191.190.0.0 = 1011 1111. 1011 111**0. 0000 0000. 0000 0000**

В конце адреса 17 нулей, можем использовать маску 32 – 17 = 15 бит. Маска длиннее 8 бит, поэтому правый край можем пока не проверять.

191.190.0.0/15, BC = 191.191.255.255. Девятая подсеть найдена.

Шаг 10. Диапазон: 191.192.0.0 – 192.168.113.239.

191.192.0.0 = 1011 1111. 11**00 0000. 0000 0000. 0000 0000**

В конце адреса 22 нуля, можем использовать маску 32 – 22 = 10 бит. Маска длиннее 8 бит, поэтому правый край можем пока не проверять.

191.192.0.0/10, BC = 191.255.255.255. Десятая подсеть найдена.

Шаг 11. Диапазон: 192.0.0.0 – 192.168.113.239.

192.0.0.0 = 11**00 0000. 0000 0000. 0000 0000. 0000 0000**

В конце адреса 30 нулей, можем использовать маску от 2 бит (32 – 30 = 2). Эта маска короче 8 бит, поэтому необходимо проверять, не выходит ли BROADCAST за правый край диапазона. Очевидно, что любая маска короче 8 бит не подойдет, т.к. приведет к отличиям в 1 бите между адресами SUBNET и BROADCAST, что противоречит условию задачи.

- 192.0.0.0/2, BC = 255.255.255.255 – превышает 192.168.113.239;
- 192.0.0.0/3, BC = 223.255.255.255 – превышает 192.168.113.239;
- 192.0.0.0/4, BC = 207.255.255.255 – превышает 192.168.113.239;
- 192.0.0.0/5, BC = 199.255.255.255 – превышает 192.168.113.239;
- 192.0.0.0/6, BC = 195.255.255.255 – превышает 192.168.113.239;
- 192.0.0.0/7, BC = 193.255.255.255 – превышает 192.168.113.239.

Действительно, рассматриваемый на данном шаге диапазон имеет совпадающие значения первого байта начального и конечного адреса, следовательно, можно рассматривать маски начиная с 8 бит и длиннее.

- 192.0.0.0/8, BC = 192.255.255.255 – превышает 192.168.113.239;
- 192.0.0.0/9, BC = 192.127.255.255 – подходит.

Рассуждая иначе, придем к такому же выводу. Первые байты начального и конечного адресов диапазона (192) совпадают, отличия во втором байте (0 и 168), разница между ними равна 168 – 0 = 168, ближайшая снизу степень двойки – $2^7 = 128$, следовательно, во втором байте 7 свободных бит, длина маски 9 бит, BROADCAST 192.127.255.255.

Одиннадцатая подсеть найдена. Заметим, что этот этап решения задачи характеризуется переходом от увеличения длины маски к ее уменьшению от шага к шагу, которая сохранится до конца решения задачи.

Шаг 12. Диапазон: 192.128.0.0 – 192.168.113.239.

192.128.0.0 = 1100 0000. 1**000 0000**. **0000 0000**. **0000 0000**

В конце 23 нуля, можем рассматривать маски, начиная от 9 бит. Но маска /9 уже была использована на предыдущем шаге. Рассматриваем, начиная со следующего значения – 10 бит.

- 192.128.0.0/10, BC = 192.192.255.255 – превышает 192.168.113.239;
- 192.128.0.0/11, BC = 192.159.255.255 – подходит.

Рассуждая иначе, придем к такому же выводу. Первые байты начального и конечного адресов диапазона (192) совпадают, отличия во втором байте (128 и 168), разница между ними равна 168 – 128 = 40, ближайшая снизу степень двойки – $2^5 = 32$, следовательно, во втором байте 5 свободных бит, длина маски 11 бит, BROADCAST 192.159.255.255.

Двенадцатая подсеть найдена.

Шаг 13. Диапазон: 192.160.0.0 – 192.168.113.239.

192.160.0.0 = 1100 0000. 101**0 0000**. **0000 0000**. **0000 0000**

В конце 21 нуль, можем рассматривать маски, начиная от 11 бит. Но маска /11 уже была использована на предыдущем шаге. Рассматриваем, начиная со следующего значения – 12 бит.

- 192.160.0.0/12, BC = 192.175.255.255 – превышает 192.168.113.239;
- 192.160.0.0/13, BC = 192.167.255.255 – подходит.

Рассуждая иначе, придем к такому же выводу. Первые байты начального и конечного адресов диапазона (192) совпадают, отличия во втором байте (160 и 168), разница между ними равна 168 – 160 = 8, ближайшая снизу степень двойки – $2^3 = 8$, следовательно, во втором байте 3 свободных бита, длина маски 13 бит, BROADCAST 192.167.255.255.

Двенадцатая подсеть найдена.

Шаг 14. Диапазон: 192.168.0.0 – 192.168.113.239. Заметим, что данный диапазон в точности совпадает с диапазоном адресов из предыдущей задачи 4.3.1. Дальнейшее решение задачи аналогично предыдущей и будет состоять из 8 шагов.

Ответом к задаче являются все найденные подсети (их 21):

1. 191.189.216.100/30 (до 191.189.216.103);
2. 191.189.216.104/29 (до 191.189.216.111);
3. 191.189.216.112/28 (до 191.189.216.127);
4. 191.189.216.128/25 (до 191.189.216.255);
5. 191.189.217.0/24 (до 191.189.217.255);
6. 191.189.218.0/23 (до 191.189.219.255);
7. 191.189.220.0/22 (до 191.189.223.255);
8. 191.189.224.0/19 (до 191.189.255.255);
9. 191.190.0.0/15 (до 191.191.255.255);
10. 191.192.0.0/10 (до 191.255.255.255);
11. 192.0.0.0/9 (до 192.127.255.255);
12. 192.128.0.0/11 (до 192.159.255.255);
13. 192.160.0.0/13 (до 192.167.255.255);
14. 192.168.0.0/18 (до 192.168.63.255);
15. 192.168.64.0/19 (до 192.168.96.255);
16. 192.168.96.0/20 (до 192.168.111.255);
17. 192.168.112.0/24 (до 192.168.112.255);
18. 192.168.113.0/25 (до 192.168.113.127);
19. 192.168.113.128/26 (до 192.168.113.191);
20. 192.168.113.191/27 (до 192.168.113.223);
21. 192.168.113.224/28 (до 192.168.113.239).

Заметим тенденцию: длина маски от шага к шагу сначала уменьшается, а затем увеличивается на 1 или более бит. Отметим, что, как и в предыдущей задаче, найденное разбиение является единственно верным для данного диапазона адресов А.

4.4. Тренировочные задания

Дан диапазон адресов, заданный начальным и конечным значениями. Требуется описать его в формате CIDR.

1. 192.168.0.0 – 192.168.112.205
2. 192.168.0.0 – 192.168.45.223
3. 10.0.0.0 – 12.56.173.95
4. 10.0.0.0 – 10.189.34.159
5. 128.0.0.0 – 129.46.11.151
6. 128.0.0.0 – 131.248.173.67
7. 172.16.0.0 – 171.30.232.191
8. 172.16.0.0 – 172.22.111.135

9. 192.0.0.0 – 200.15.255.255
10. 192.0.0.0 – 192.168.1.143
11. 127.211.183.80 – 128.16.54.159
12. 190.89.64.200 – 192.168.15.255
13. 172.6.116.160 – 172.17.114.127
14. 125.200.114.68 – 129.89.116.34
15. 8.14.167.60 – 10.0.190.31

4.5. Рекомендуемая литература и Интернет-ресурсы

1. Основы компьютерных сетей. Тема № 3. Протоколы нижних уровней (транспортного, сетевого и канального) [электронный ресурс]. –Режим доступа: https://habr.com/post/308636/ – Заглавие с экрана. Дата обращения: 09.10.2018.

2. Основы компьютерных сетей. Тема № 5. Понятие IP адресации, масок подсетей и их расчет) [электронный ресурс]. – Режим доступа: https://habr.com/post/314484/ – Заглавие с экрана. Дата обращения: 09.10.2018.

3. Компьютерные сети for dummies [электронный ресурс]. – Режим доступа: https://habr.com/post/335816/ – Заглавие с экрана. Дата обращения: 09.10.2018.

Семинар 5.
Протокол STP

5.1. Цель и задачи семинара

Семинар посвящен закреплению знаний по протоколу STP. Преподаватель обсуждает со студентами назначение и порядок работы протокола. Рассматривается пример работы протокола в информационно-телекоммуникационной сети условной компании. Проводится разбор и решение типовых задач выбора корневого коммутатора, корневых и назначенных портов. По итогам семинара проводится контрольная работа на решение задач рассмотренного типа. Примеры задач приведены в фонде оценочных средств дисциплины.

I think that I shall never see
A graph more lovely than a tree.
A tree whose crucial property
Is loop-free connectivity.

A tree that must be sure to span
So packets can reach every LAN.
First, the root must be selected.
By ID, it is elected.

Least-cost paths from root are traced.
In the tree, these paths are placed.
A mesh is made by folks like me,
Then bridges find a spanning tree.

Radia Perlman

5.2. Теоретическая часть

5.2.1. Широковещательный шторм

Часто для обеспечения стабильности работы сети в случае проблем со связью между коммутаторами (выход порта из строя, обрыв провода), используют избыточные связи (redundant links) – дополнительные соединения. Идея простая – если между коммутаторами по какой-то причине отказывает

канал связи, ему на замену используется запасной. Такое резервирование действительно позволяет повысить отказоустойчивость сети, но приводит к возникновению иных проблем, главная из которых – широковещательный шторм (broadcast storm).

Представим себе такую ситуацию: два коммутатора Sw1 и Sw2 соединены двумя кабелями (для определенности будем считать, что у них попарно соединены интерфейсы fa0/1 и fa0/24 соответственно; рис. 5.1).

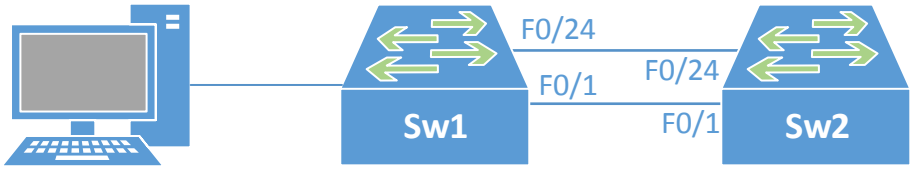


Рис. 5.1. Возникновение широковещательного шторма

Один из узлов, подключенных к коммутатору Sw1 (пусть это будет PC1), отправляет широковещательный кадр (broadcast; например, ARP-запрос). Раз кадр широковещательный, коммутатор Sw1 отправляет его на все порты, кроме того, с которого он был получен. Коммутатор Sw2 таким образом получает этот кадр сразу в два порта (fa0/1 и fa0/24) – для него это два разных кадра. При этом оба кадра широковещательные и должны быть отправлены со всех интерфейсов (кроме входного), но уже, получается, и обратно в те, с которых получил (кадр из fa0/24 отправляем в fa0/1, и наоборот). Коммутатор Sw1 действует точно так же и т.д.

Описанный сценарий называется широковещательным штормом; он намертво блокирует работу сети, т.к. коммутаторы вынуждены расходовать все ресурсы сети на обработку бесконечно размножающихся копий одного единственного широковещательного кадра. Для предотвращения такого развития событий и сохранения возможности повышения отказоустойчивости сети за счет использования избыточных связей служит протокол Spanning Tree.

5.2.2. Протокол Spanning Tree – общее

Spanning Tree Protocol (далее – STP) – сетевой протокол, основной задачей которого является приведение сети Ethernet с множественными связями к древовидной топологии, исключающей зацикливание кадров. Происходит это путём автоматического блокирования избыточных в данный момент связей для полной связности портов. Базовый протокол описан в стандарте IEEE 802.1D,

также существует множество версий и вариаций, таких как RSTP, MST, PVST и др., которые не будут рассмотрены в рамках настоящего курса. STP работает на втором уровне модели OSI; основан на одноимённом алгоритме (Spanning Tree Algorithm – STA), разработчиком которого является Радья Перлман, чье стихотворение использовано в качестве эпиграфа к данному разделу. Результатом его работы является граф в виде дерева (связный и без простых циклов).

Для обмена информацией между собой коммутаторы используют специальные пакеты, называемые BPDU (Bridge Protocol Data Units). BPDU бывают двух видов: конфигурационные (Configuration BPDU) и панические TCN (Topology Change Notification BPDU). Первые регулярно рассылаются одним из коммутаторов (корневым, см. ниже) и ретранслируются всеми остальными – они используются сбора сведений о топологии сети. Вторые, как понятно из названия, отсылаются в случае изменения топологии сети (проще говоря, подключении или отключении одного из коммутаторов). Конфигурационные BPDU содержат несколько полей, самые важные из которых:

- идентификатор отправителя (Bridge ID);
- идентификатор корневого коммутатора (Root Bridge ID)
- идентификатор порта, из которого отправлен данный пакет (Port ID);
- стоимость маршрута («расстояние») до корневого коммутатора (Root Path Cost).

При этом STP не предполагает установления каких-либо специальных соединений между соседними коммутаторами (никаких отношений смежности/соседства и т.п.). Коммутаторы рассылают BPDU со всех работающих портов, используя в качестве адреса назначения multicast MAC-адрес 01-80-c2-00-00-00 (по умолчанию каждые 2 секунды), который прослушивают все коммутаторы с включенным STP.

Для измерения «расстояния» используется специальная величина STP Cost, связанная с битовой скоростью сегмента обратно пропорциональной зависимостью. Каждый порт коммутатора имеет свою стоимость соединения, установленную либо на заводе-изготовителе (по умолчанию), либо вручную при настройке. Из этих стоимостей нарастающим итогом складывается расстояние до корня от любого порта любого коммутатора.

5.2.3. Этапы работы STP

STP работает поэтапно, количество этапов – три.

1. В сети выбирается один корневой коммутатор – root bridge. Слово bridge (мост) используется, т.к. первоначально протокол создавался для устройств, называвшихся мостами (фактически они представляли собой двухпортовые коммутаторы).

После включения коммутаторов в сеть, по умолчанию каждый (!) коммутатор считает себя корневым. Корневой коммутатор (изначально все) посылает по всем портам служебные кадры BPDU (bridge protocol data unit) каждые 2 секунды. Получив чужой BPDU, коммутатор сравнивает значения bridge ID и, при необходимости, прекращает рассылку собственных BPDU, ретранслирую только те, которые исходят от корневого коммутатора.

Исходя из данных BPDU пакетов, тот или иной коммутатор приобретает статус root, то есть корня, когда в сети не остается иных BPDU. Корневым коммутатором назначается коммутатор с **самым низким** значением bridge ID. Этот числовой идентификатор длиной 8 байт формируется из приоритета (старшие 4 байта) и значения MAC-адреса служебного блока коммутатора (младшие 4 байта), что позволяет управлять процессом выбора путем изменения приоритета (по умолчанию значения приоритета равны (32768) и для выбора используются только MAC-адреса).

Такой подход таит в себе серьезную проблему. Дело в том, что, при равных значениях Priority (по умолчанию это так) корневым выбирается самый старый коммутатор, так как мак адреса прописываются на производстве последовательно, соответственно, чем мак меньше, тем устройство старше (естественно, если в сети все оборудование одного вендора). Это ведет к падению производительности сети, так как старое устройство, как правило, имеет худшие характеристики. Еще большие проблемы может вызвать неоптимальное расположение корневого коммутатора, что может привести к отключению быстрых линков вместо медленных. Подобное поведение протокола следует пресекать, выставляя пониженное значение приоритета на корневом коммутаторе вручную.

2. Каждый коммутатор, кроме корневого, просчитывает кратчайший путь к корневому коммутатору. Соответствующий порт, через который это кратчайшее расстояние достигается, называется корневым портом (англ. root port).

Это «расстояние» определяется суммой стоимостей всех каналов связи, через которые нужно пройти кадру, чтобы достичь корневого коммутатора. В

свою очередь, стоимость канала определяется просто – по его скорости (чем выше скорость, тем меньше стоимость). Процесс определения стоимости маршрута связан с полем BPDU «Root Path Cost» и происходит так:

- Корневой коммутатор посылает BPDU, где Root Path Cost равно нулю;
- Ближайший коммутатор смотрит на скорость своего порта, куда BPDU пришел, и добавляет стоимость согласно таблице 5.1.
- Далее этот второй коммутатор посылает этот BPDU нижестоящим коммутаторам, но уже с новым значением Root Path Cost, и т.д.

Если значения STP Cost совпадают (как в нашем примере с двумя коммутаторами и двумя проводами между ними из раздела 5.2.1 – у каждого пути будет стоимость 19) – корневым выбирается порт с меньшим порядковым номером (в примере это fa0/1).

У любого некорневого коммутатора всегда ровно один корневой порт.

Таблица 5.1. Соответствие скорости канала и STP Cost (по IEEE 802.1d)

Скорость порта	**Стоимость STP (802.1d)**
10 Mbps	100
100 Mbps	19
1 Gbps	4
10 Gbps	2

3. Для каждого сегмента сети также просчитывается кратчайший путь к корневому коммутатору. Коммутатор, через который проходит этот путь, становится назначенным для этой сети (англ. designated bridge), а его непосредственно подключенный к этой сети интерфейс – назначенным портом (англ. designated port). В каждом сегменте выбирается ровно один назначенный порт из всех портов всех коммутаторов сегмента, иначе это петля (loop). В данном случае имеется в виду физический сегмент сети, в современных сетях без концентраторов (хабов) это просто кабель, соединяющий два порта двух коммутаторов. Назначенным портом выбирается тот, который имеет лучшую стоимость в данном сегменте.

Отметим, что все порты, к которым подключены конечные устройства (компьютеры и др.) всегда будут назначенными, т.к. они представляют собой единственный порт, принадлежащий коммутатору, в своем сегменте. В других сегментах назначенным всегда будет выбран порт того коммутатора, для которого меньше расстояние до корня на соответствующем корневом порту. У корневого коммутатора все порты назначенные.

Алгоритмически это происходит так. Каждый коммутатор для всех своих портов (кроме корневого) выполняет сравнение принятых по ним Root Path Cost (до наращивания). Если все принятые на порт расстояния больше расстояния от собственного корневого порта, то этот порт – назначенный для данного сегмента.

Коммутатор вынужден принимать решение, не зная топологии сети и опираясь только на полученные кадры BPDU. Прогнозируя же результаты работы STP в реальной сети с известной топологией, можно руководствоваться иной логикой, а именно: из всех портов всех коммутаторов сегмента назначенный порт будет принадлежать тому коммутатору, у которого значение STP Cost на собственном корневом порту меньше.

В завершение работы протокола на всех коммутаторах блокируются все порты, не являющиеся корневыми или назначенными. В итоге получается древовидная структура (математический граф) с вершиной на корневом коммутаторе. Корневой коммутатор продолжает рассылать конфигурационные BPDU каждые 2 секунды для поддержания древовидной топологии сети в актуальном состоянии.

5.2.4. Состояния портов

По итогам работы STP каждый порт может оказаться в одном из 5 состояний (рис.5.2):

– блокировка (blocking): блокированный порт не отправляет никаких кадров. Это состояние предназначено, как говорилось выше, для предотвращения петель в сети. Блокированный порт, тем не менее, слушает BPDU (для поддержания графа в актуальном состоянии; это позволяет ему при необходимости перейти в другое состояние);

– прослушивание (listening): порт слушает и отправляет BPDU, кадры с данными не отправляет;

– обучение (learning): порт слушает и отправляет BPDU, а также вносит изменения в MAC-таблицу, но данные не перенаправляет;

– продвижение (forwarding): этот порт выполняет все функции: принимает и отправляет пакеты BPDU и пакеты данных; участвует в составлении таблицы MAC-адресов – это обычное состояние рабочего порта;

– отключен (disabled): состояние administratively down, порт отключен командой shutdown; такой порт не делает ничего вообще, пока не будет обратно включен вручную.

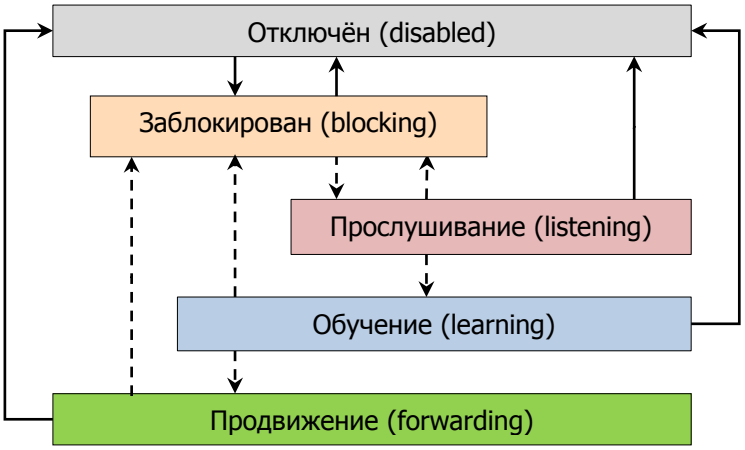


Рис. 5.2. Диаграмма переходов состояний портов

Порядок перечисления состояний не случаен: при включении коммутатора (а также при подключении нового кабеля), все порты на устройстве с STP проходят перечисленные выше состояния именно в таком порядке (за исключением отключенных (disabled) портов). Причина в том, что коммутатор не может знать, какое устройство на другом конце кабеля: компьютер (тупиковый сегмент, риска петли нет) или коммутатор (потенциальная петля). Поэтому в течение первых 15 секунд (по умолчанию) после инициализации порт находится в состоянии прослушивания (listening) – он изучает входящие BPDU и выясняет свое положение в сети. Затем переходит к состоянию обучения (listening) еще на 15 секунд для составления таблицы MAC-адресов для порта. Только после этого порт переходит в рабочее состояние продвижения (forwarding).

В итоге набегает 30 секунд простоя, прежде чем подключенное устройство сможет обмениваться информацией со своими соседями. Загрузка современного компьютера занимает в среднем менее 30 секунд, что приводит к проблемам в работе других протоколов, например, DHCP.

В таких случаях может быть использован особый режим порта – portfast. При подключении устройства к такому порту, он, минуя промежуточные стадии, сразу переходит к состоянию forwarding. Очевидно, portfast следует включать только на интерфейсах, ведущих к конечным устройствам, и никогда – к другим коммутаторам.

5.3. Разбор задач

Задача 5.3.1.

Дана топология сети на основе коммутаторов (рис. 5.3). Рассчитайте результаты работы протокола STP. Определите корневой коммутатор, корневые и назначенные порты на каждом коммутаторе.

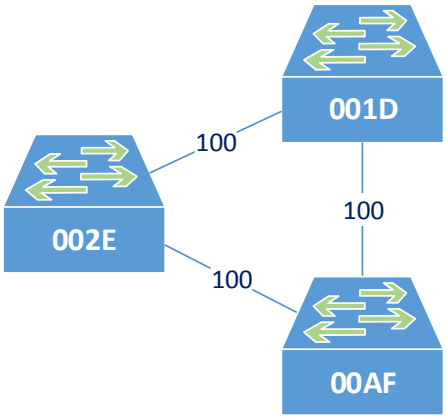


Рис. 5.3. Топология сети к задаче 5.3.1

Решение.

Рассмотрим работу протокола STP поэтапно.

Первый этап – выбор корневого коммутатора. Для выбора сравниваем Bridge ID коммутаторов сети, состоящие из приоритета (в старших битах) и MAC-адреса (в младших). Приоритеты коммутаторов в задаче не указаны, следовательно, они одинаковые и не влияют на выбор корня. Сравниваем MAC-адреса: 001D, 002E, 00AF. Очевидно, адрес 001D наименьший, этот коммутатор и становится корневым (рис. 5.4).

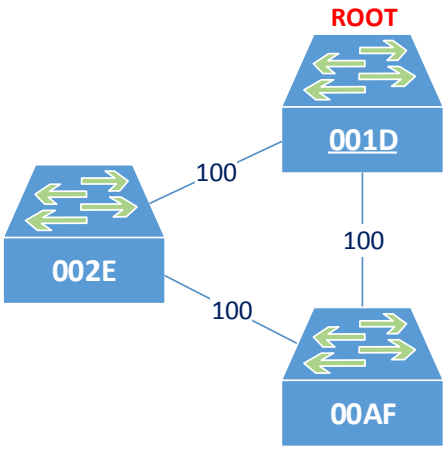


Рис. 5.4. Задача 5.3.1 – выбор корневого коммутатора

Второй этап – выбор корневых портов (рис. 5.5). Необходимо выбрать по одному корневому порту на всех коммутаторах, кроме корневого. Для выбора используем величину STP Cost из входящих пакетов BPDU – она должна быть наименьшей. STP Cost зависит от битовой скорости канала, можно считать ее обратно пропорциональной битовой скорости. Получаем STP Cost на интерфейсах коммутаторов 002E и 00AF равную 1/100 и 2/100.

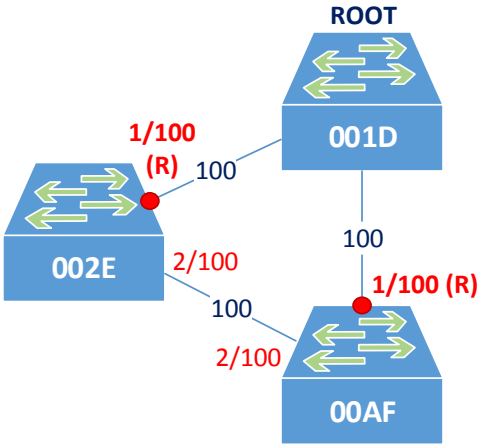


Рис. 5.5. Задача 5.3.1 – выбор корневых портов

Третий этап – выбор назначенных (designated, D) портов (рис. 5.6). Необходимо выбрать по одному назначенному порту в каждом сегменте сети, среди всех портов всех коммутаторов, образующих этот сегмент. Сложность заключается в том, что коммутаторы принимают такое (одинаковое) решение самостоятельно и независимо.

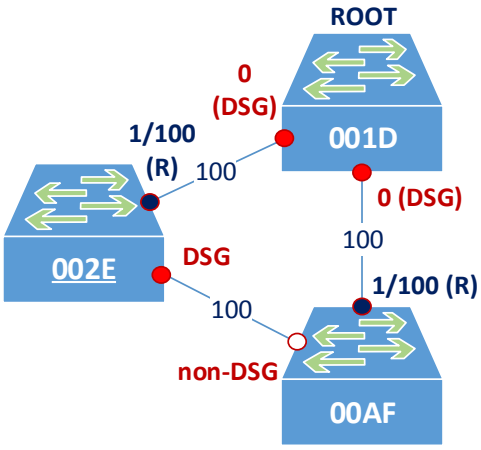


Рис. 5.6. Задача 5.3.1 – выбор назначенных портов

В сети три сегмента. Каждый из них образован двумя портами двух коммутаторов. Сравниваем их расстояния на корневых портах. На корневом

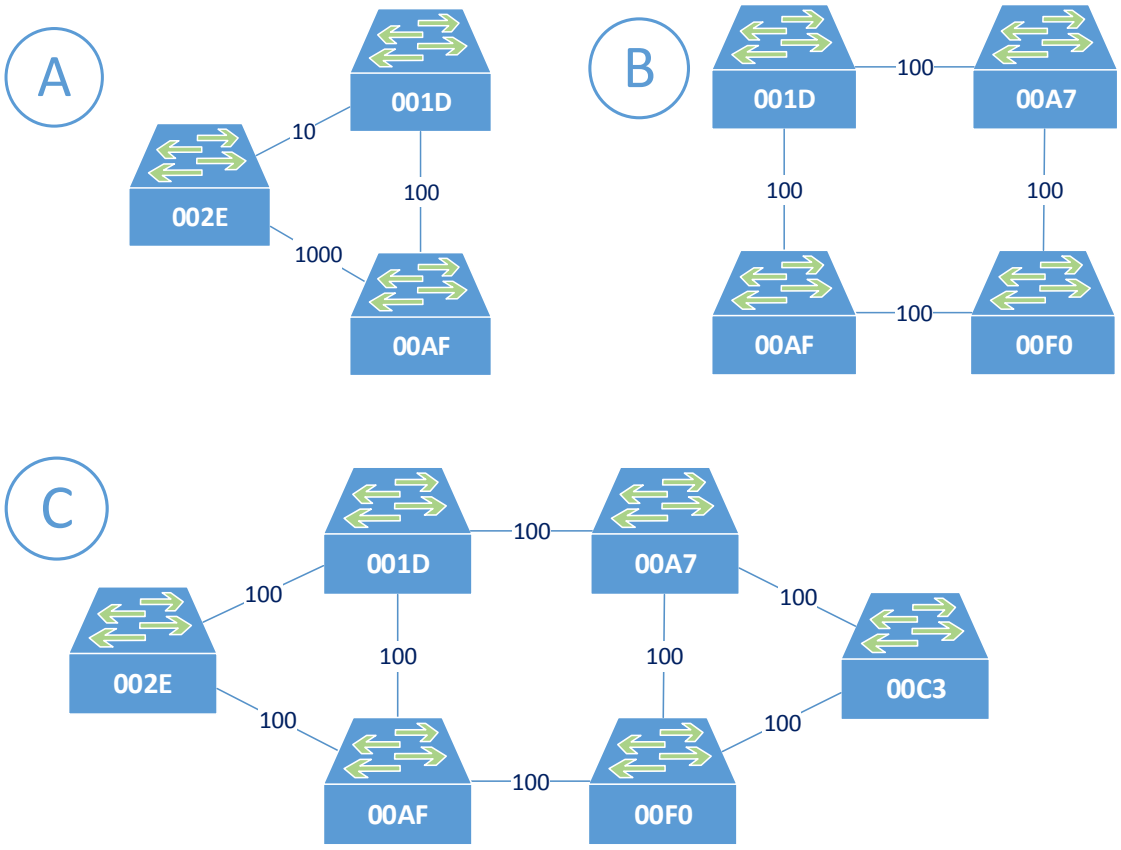
коммутаторе расстояние до корня, очевидно, равно нулю, следовательно, назначенными в своих сегментах станут порты корневого коммутатора. Отметим, что так будет всегда, т.к. на корневом коммутаторе расстояние всегда равно 0.

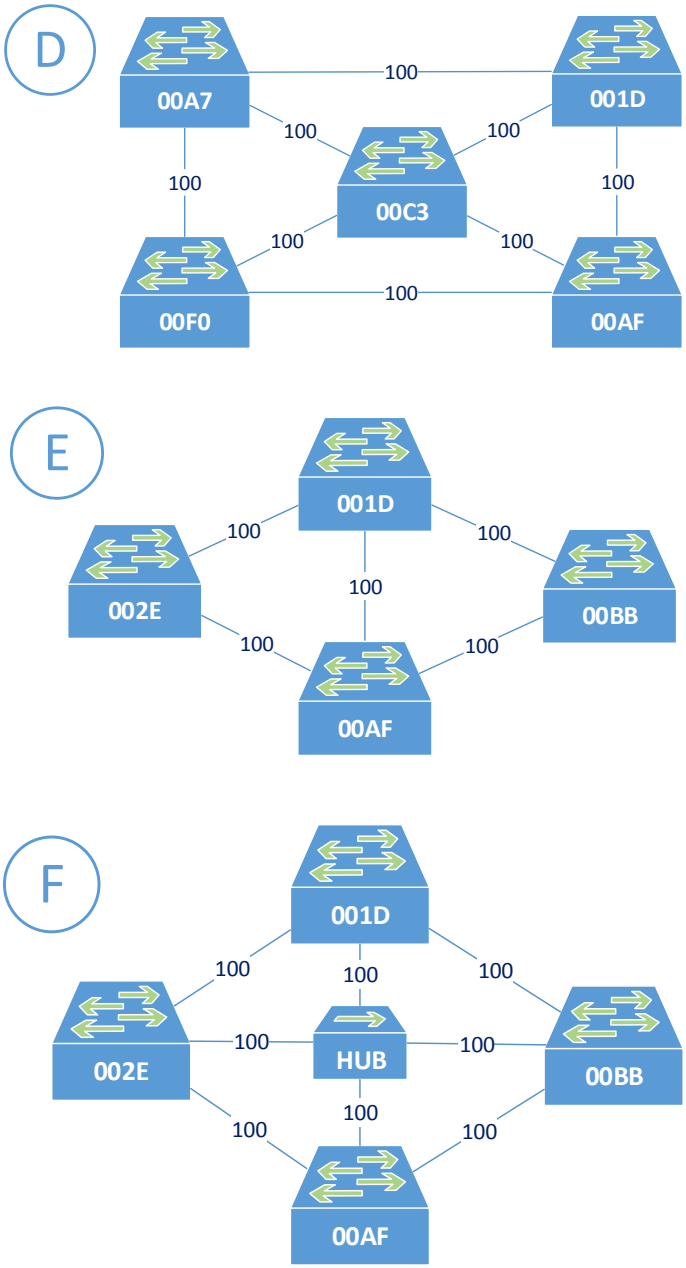
Остается последний сегмент: 002E – 00AF. Сравниваем расстояния на корневых портах, они совпадают (1/100). В таком случае для выбора используем MAC-адреса. 002E < 00AF, поэтому порт именно этого коммутатора станет назначенным.

Оставшийся порт, не отнесенный ни к корневым, ни к назначенным, становится неназначенным (non-designated, ND) и переводится в резервное состояние. Активная фаза работы протокола на этом завершена. Задача решена.

5.4. Тренировочные задания

Дана топология сети на основе коммутаторов. Рассчитайте результаты работы протокола STP. Определите корневой коммутатор, корневые и назначенные порты на каждом коммутаторе.





5.5. Рекомендуемая литература и Интернет-ресурсы

1. Сети для самых маленьких. Часть четвертая. STP [электронный ресурс]. –Режим доступа: https://habr.com/post/143768/ – Заглавие с экрана. Дата обращения: 10.10.2018.

2. Основы компьютерных сетей. Тема № 7. Протокол связующего дерева: STP [электронный ресурс]. – Режим доступа: https://habr.com/post/321132/ – Заглавие с экрана. Дата обращения: 10.10.2018.

3. Принцип работы протокола STP [электронный ресурс]. – Режим доступа: https://habr.com/post/419491/ – Заглавие с экрана. Дата обращения: 10.10.2018.

4. Закольцованные сети, или зачем нам STP [электронный ресурс]. – Режим доступа: https://habr.com/post/129559/ – Заглавие с экрана. Дата обращения: 10.10.2018.

Семинар 6.
Маршрутизация

6.1. Цель и задачи семинара

На семинаре закрепляются знания по сетевой маршрутизации: принципы маршрутизации, критерии выбора маршрута, методы и протоколы маршрутизации. Проводится сравнительный анализ различных подходов к маршрутизации. Рассматривается назначение о общее устройство маршрутизатора, основные структуры данных: таблица маршрутов, топологическая база данных и др. Проводится разбор и решение типовых задач на составление таблиц маршрутов по известной топологии и на восстановление топологии по известным таблицам маршрутов. По итогам семинара проводится контрольная работа на решение задач рассмотренного типа. Примеры задач приведены в фонде оценочных средств дисциплины.

6.2. Теоретическая часть

6.2.1. Таблица маршрутизации

Маршрутизация – процесс определения оптимального пути, по которому пакет может быть доставлен в сеть назначения. Возможные пути передачи пакетов называются маршрутами. Оптимальные маршруты до известных сетей назначения хранятся в таблице маршрутизации.

В зависимости от способа заполнения таблицы маршрутизатора, различают два вида маршрутизации:

- Статическая – удалённые сети вручную вводятся в таблицу маршрутизации с помощью статических маршрутов;
- Адаптивная (динамическая) – удалённые маршруты автоматически добавляются в таблицу с помощью протокола маршрутизации.

Таблица маршрутизации (Рис 6.1) – электронная таблица или база данных, хранимая в памяти маршрутизатора, которая описывает соответствие между адресами сетей назначения и маршрутами, которые следует использовать для отправки пакетов в указанные сети.

Каждая запись таблицы маршрутизации включает следующие сведения:

- Источник записи (англ. source, SRC) – показывает, кем добавлен в таблицу данный маршрут; разные источники характеризуются разной степенью достоверности, см. административная дистанция;
- IP-адрес и маска сети назначения;
- Метрика (англ. metric) – аддитивная характеристика протяженности маршрута (чем меньше метрика, тем короче и, следовательно, лучше, маршрут); используется для сравнения маршрутов, полученных из одного источника;
- Административная дистанция (англ. administrative distance, AD) – критерий достоверности маршрута; используется для сравнения маршрутов, полученных из разных источников, т.к. такие маршруты имеют разные алгоритмы расчета метрики;
- Направление (собственный интерфейс или IP следующего маршрутизатора – nexthop).

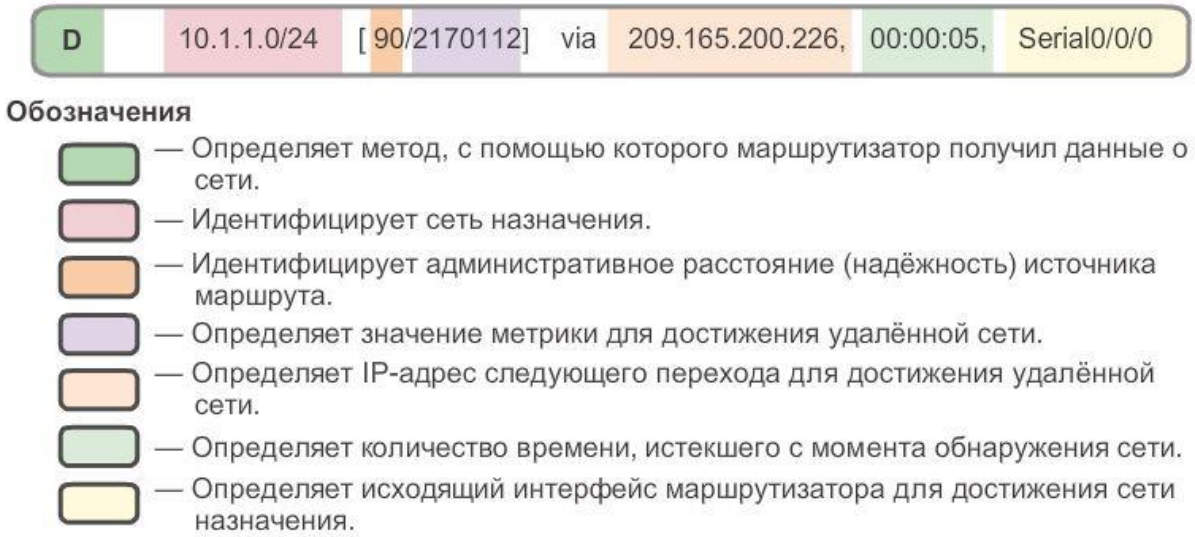


Рисунок 6.1. Маршрут в таблице маршрутизации

6.2.2. Выбор маршрута

Всякий раз при получении пакета маршрутизатор начинает его обработку с декапсуляции, извлекая пакет из кадра данный той сети, в которой находится интерфейс, на который этот пакет получен. Отбросив заголовок и концевик канального уровня, маршрутизатор анализирует пакет, извлекая из него IP-адрес и маску назначения. По адресу и маске производится расчет сети назначения. Именно этот адрес будет использован в процессе маршрутизации данного пакета.

Рассчитав адрес сети назначения, маршрутизатор производит выбор оптимального маршрута. Процедура выбора состоит из нескольких этапов.

1. Среди всех маршрутов, содержащихся в таблице, выбираются те, которые позволят достичь сети назначения. Это условие выполняется, если сеть назначения для пакета совпадает с сетью назначения в маршруте или входит в состав сети назначения, указанной в маршруте. *Например, сеть 192.168.1.0/24 входит в состав сети 192.168.0.0/23, поэтому маршрут до последней пригоден для отправки пакета в первую.* Отобранных маршрутов может быть 0, 1 или несколько. Если их 0, процесс маршрутизации завершается с ошибкой, о чем маршрутизатор уведомляет узел – источник пакета. Если найден единственный маршрут, процесс маршрутизации завершается, не переходя к следующему этапу, и найденный маршрут используется для продвижения пакета. Если найденных маршрутов несколько, процесс переходит ко второму этапу.

2. Среди всех маршрутов, отобранных на первом этапе, выбирается оптимальный по критериям административная дистанция (AD) и метрика. При этом сначала происходит отбор по минимальной величине AD, затем – из отобранных – по метрике. Отобранных маршрутов может 1 или несколько. Как и на первом этапе, отбор единственного маршрута позволяет сразу завершить процесс. Если же вновь найдено несколько маршрутов, необходимо продолжить отбор для нахождения оптимального – переходим к третьему этапу.

3. Переход к третьему этапу возможен в том случае, если в таблице найдено несколько маршрутов до сети назначения с одинаковыми AD и метрикой. В этом случае последний и решающий выбор производится по критерию маски: чем меньше размер сети назначения, тем более точным считается попадание в нее, следовательно, такой маршрут лучше. Он и будет использован для продвижения. Если и этот параметр совпадает, маршруты будут использоваться попеременно, обеспечивая балансировку трафика.

6.2.3. Статическая маршрутизация

Статическая маршрутизация – вид маршрутизации, при котором маршруты вручную указываются администратором при настройке маршрутизатора.

Статическая маршрутизация имеет три основных назначения:

– Обеспечение упрощённого обслуживания таблицы маршрутизации в небольших сетях, которые не планируется существенно расширять.

- Маршрутизация к тупиковым сетям и от них. Тупиковая сеть представляет собой сеть, доступ к которой осуществляется через один маршрут, и маршрутизатор имеет только одно соседнее устройство.
- Использование маршрута по умолчанию для представления пути к любой сети, не имеющего более точного совпадения с другим маршрутом в таблице маршрутизации. Маршруты по умолчанию используются для отправки трафика в любой пункт назначения за пределами следующего маршрутизатора в восходящем направлении.

Достоинства	Недостатки
– Простота настройки (в небольших сетях); – Отсутствие дополнительной нагрузки на сеть (в отличии от динамических протоколов маршрутизации); – Путь, используемый статическим маршрутом для отправки данных, известен; – Статические маршруты не объявляются по сети, поэтому, они более безопасны. – Не потребляет ресурсов маршрутизатора.	– Для внесения изменений в данные маршрута требуется вмешательство администратора. – Недостаточные возможности масштабирования для растущих сетей, обслуживание при этом становится довольно трудоёмким. – Для качественного внедрения требуется доскональное знание всей сети.

6.2.4. Типы статических маршрутов

Статический маршрут по умолчанию – это маршрут, которому соответствуют все пакеты. Маршрут по умолчанию идентифицирует IP-адрес шлюза, на который маршрутизатор отправляет все IP-пакеты, для которых у него нет известного полученного или статического маршрута. Статический маршрут по умолчанию – это статический маршрут с IPv4-адресов назначения равным 0.0.0.0/0. При настройке статического маршрута по умолчанию создаётся «шлюз последней надежды».

Суммарный статический маршрут (Рис 6.2). Для уменьшения числа записей в таблице маршрутизации можно объединить несколько статических маршрутов в один. Это возможно при следующих условиях:

- Сети назначения являются смежными и могут быть объединены в один сетевой адрес.

– Все статические маршруты используют один и тот же выходной интерфейс или один IP-адрес следующего перехода.

Рис. 6.2. Порядок расчета суммарного маршрута

6.2.5. Адаптивная маршрутизации

Адаптивная (динамическая) маршрутизация – вид маршрутизации, при котором формирование таблиц маршрутизации автоматизировано на основе протоколов маршрутизации.

Все протоколы маршрутизации разработаны для получения данных об удалённых сетях и быстрой адаптации к любым изменениям в топологии. Метод, используемый протоколом маршрутизации для выполнения этих задачи, зависит от выбранного протокола и его эксплуатационных характеристик.

В целом, работу протокола динамической маршрутизации можно описать следующим образом:

1. Маршрутизатор отправляет и принимает сообщения маршрутизации с помощью своих интерфейсов.
2. Маршрутизатор предоставляет общий доступ к сообщениям маршрутизации и данным о маршрутах для других маршрутизаторов, использующих тот же протокол маршрутизации.
3. Маршрутизаторы осуществляют обмен данными маршрутизации для получения информации об удалённых сетях.
4. При обнаружении изменений в топологии, маршрутизатор использует протокол маршрутизации для извещения других маршрутизаторов об этом изменении.

Протоколы маршрутизации можно классифицировать по различным группам в соответствии с их назначением и характеристиками (Рис 6.3).

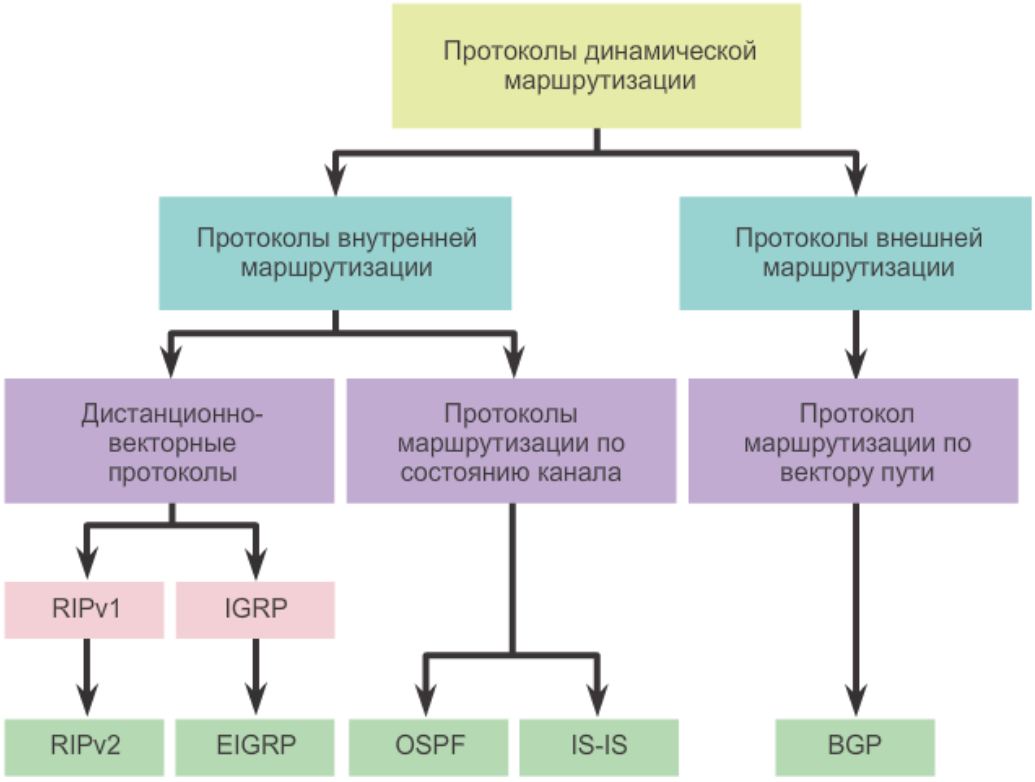


Рис. 6.3. Классификация протоколов маршрутизации

Бывают ситуации, когда протокол маршрутизации получает более одного маршрута до одной сети назначения. Для выбора оптимального маршрута протокол маршрутизации должен уметь оценивать и различать возможные

пути. Эта задача выполняется посредством использования метрик маршрутизации.

Метрика (metric) – аддитивная характеристика протяженности маршрута (напр., количество хопов, битовая скорость, задержки) – критерий выбора маршрута.

Если на маршрутизаторе одновременно работает несколько протоколов динамической маршрутизации, то для выбора лучшего маршрута, маршрутизатор использует другую характеристику – административную дистанцию (AD). AD – первый критерий, который используется маршрутизатором для выбора из протоколов, предоставляющих информацию о маршруте до одной и той же сети. AD – это мера надежности источника информации о маршруте. AD имеет локальное значение (в пределах данного маршрутизатора), информация о ней не включается в рассылку маршрутной информации.

Значения AD по умолчанию для различных источников маршрутной информации приведены в таблице 6.1.

Таблица 6.1 Значения AD для различных источников маршрутов

Источник маршрута	Административная дистанция
Прямой (Connected, C)	0
Статический (Static, S)	1
Суммарный маршрут EIGRP	5
Внешний BGP	20
Внутренний EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
Внешний EIGRP	170
Внешний BGP	200

6.2.6 Routing Information Protocol (RIP)

Routing information protocol – динамический протокол маршрутизации дистанционно-векторного типа. Протокол RIP использует алгоритм Беллмана-Форда в качестве алгоритма маршрутизации. Он основан на двух алгоритмах, разработанных в 1958 и 1956 гг. Ричардом Беллманом (Richard Bellman) и Лестером Фордом-мл. (Lester Ford, Jr). Есть три версии протокола:

– RIPv1 – динамический протокол классовой маршрутизации;

– RIPv2 – динамический протокол бесклассовой маршрутизации;

– RIPng – динамический протокол бесклассовой маршрутизации с поддержкой IPv6.

«Дистанционно-векторный» означает, что маршруты объявляются путём указания двух характеристик:

– Расстояние – определяет удалённость сети назначения;

– Вектор – определяет направление маршрутизатора следующего перехода или выходного интерфейса маршрута для доступа к адресу назначения.

Протокол RIP использует простейшую метрику – количество хопов, т.е количество промежуточных маршрутизаторов, которые нужно преодолеть пакету до сети назначения.

Рассмотрим процесс построения таблицы маршрутизации с помощью протокола RIP. Ранее было показано, что непосредственно подключенные сети появляются в таблице маршрутов сразу после инициализации интерфейсов маршрутизатора IP-адресами в этих сетях. Начнем рассмотрение процесса с момента, когда минимальные таблицы заполнены. Тогда работа протокола RIP может быть описана как бесконечный цикл повторения двух действий:

1. Маршрутизатор выполняет рассылку всем своим соседям специального служебного сообщения протокола RIP, в котором содержатся сведения обо всех известных ему сетях и лучших маршрутах к ним.

2. Маршрутизатор получает аналогичные сообщения от соседних маршрутизаторов, также использующих протокол RIP. Получив такое сообщение, маршрутизатор для всех маршрутов, содержащихся в указанном сообщении, увеличивает метрику на единицу и запоминает через какой порт и от какого маршрутизатора получена новая информация (адрес этого маршрутизатора станет адресом следующего маршрутизатора, если эта запись будет внесена в таблицу маршрутизации). После маршрутизатор сравнивает новую информацию с той, которая хранится в его таблице. Протокол RIP замещает запись о какой-либо сети только в том случае, если новая информация имеет лучшую метрику (с меньшим расстоянием в хопах), чем имеющаяся. В результате в таблице маршрутизации о каждой сети остается только одна запись (за исключением случаев, когда несколько маршрутов имеют одинаковые метрики). Для этого правила существует исключение – если худшая информация о какой-либо сети пришла от того же маршрутизатора, на основании сообщения, которого была создана данная запись, то худшая информация замещает лучшую.

6.2.7. RIP: Обработка изменений в топологии

В протоколе RIP период рассылки обновлений выбран равным 30 секундам, а в качестве тайм-аута выбрано шестикратное значение периода рассылки, то есть 180 секунд. Шестикратный запас времени нужен для уверенности в том, что сеть действительно стала недоступной, а не просто произошли потери RIP-сообщений (а это возможно, так как протокол RIP использует транспортный протокол UDP, который не обеспечивает надежной доставки сообщений). Если какой-либо маршрутизатор отказывает, переставая слать своим соседям сообщения о сетях, которые можно достичь через него, то через 180 секунд все записи, отправленные этим маршрутизатором, у его ближайших соседей станут недействительными. После этого процесс повторится уже для ближайших соседей – они вычеркнут подобные записи уже через 360 секунд.

6.2.8. RIP: Маршрутные петли

В силу недостатков, изначально заложенных в дистанционно-векторных протоколах, протокол RIP подвержен возникновению маршрутных петель – ситуаций, когда два маршрутизатора добавляют в таблицу маршруты до одной сети, направленные друг на друга. Это приводит к зацикливанию пакетов на данном участке сети и имеет крайне негативные последствия для работы сети в целом. Рассмотрим последовательность возникновения петли и зацикливания пакетов на примере (Рис 6.4).

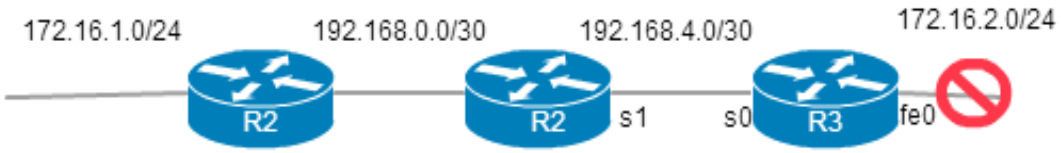


Рисунок 6.4. Маршрутная петля

1. Сеть 172.16.2.0/24 стала недоступной (например, вследствие обрыва кабеля). Интрерфейс fe0 маршрутизатора R3 переходит в состояние down, и маршрутизатор R3 в таблице маршрутизации устанавливает для этой непосредственно подключенной сети метрику 16.
2. Маршрутизатор R3 не успевает отослать соседям обновление о том, что у сети 172.16.2.0/24 теперь метрика 16 и она не доступна. В этот момент R2 посылает обновление свой таблицы маршрутизатору R3 в которой есть путь до сети 172.16.2.0/24 с метрикой 1;

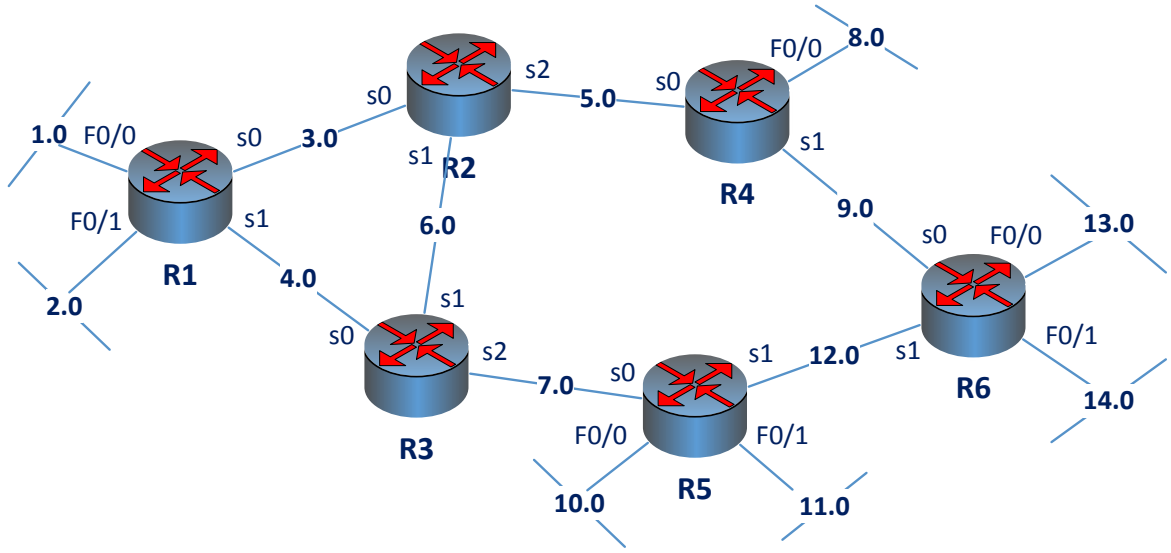
3. Маршрутизатор R3, получив обновление видит, что в таблице маршрутизации R2 есть путь до сети 172.16.2.0/24 с метрикой = 1. R3 записывает новый маршрут до сети 172.16.2.0/24 через R2, увеличивая метрику до 2. Теперь любой пакет, идущий в сеть 172.16.2.0/24 будет зациклен между R3 и R2.

Однако описанный пример носит условный характер, т.к. в протоколе RIP предусмотрен ряд алгоритмов, предотвращающих возникновение петель. Среди них:

- **Triggered updates** – триггерные обновления. Маршрутизатор получив данные об изменение метрики до какой-либо сети, не ждет истечения 30-секундного периода передачи таблицы маршрутизации, а передаёт обновление немедленно.
- **Holddown timer** – таймер заморозки. Маршрутизатор блокирует все изменения маршрутов, связанные с потенциально недоступной сетью, причем величина (240 секунд) таймера превышает стандартные 180 секунд таймаута обновления маршрутной записи.
- **Split horizon** – разделенный горизонт. Маршрутизатор не передает информации о сети на тот интерфейс, через который эта информация была получена. Есть также усовершенствованный алгоритм Split horizon with Poison reverse. При его использовании маршрутизатор передаёт обратно на интерфейс, через который получена информация о сети, маршрут с бесконечной метрикой (16).

6.3. Тренировочные задания

1. Дана топология составной сети. Все сети имеют адреса вида 172.16.X.X/24, значения двух младший байтов указаны на схеме. Составить таблицы маршрутизации для маршрутизатора R1 (варианты: R2, R3 и т.п.).



2. Даны таблицы маршрутизации 5 маршрутизаторов: A, B, C, D и E. Восстановите топологию сети (изобразите графически).

ROUTER A	**ROUTER B**
C 172.16.20.0 C 172.16.28.0 C 172.16.30.0 R 172.16.25.0 [120/1] 172.16.28.2 R 172.16.23.0 [120/1] 172.16.28.2 R 172.16.27.0 [120/1] 172.16.28.2 [120/1] 172.16.30.2 R 172.16.31.0 [120/1] 172.16.28.2 R 172.16.21.0 [120/2] 172.16.28.2 R 172.16.29.0 [120/1] 172.16.30.2 R 172.16.26.0 [120/2] 172.16.28.2 [120/2] 172.16.30.2 R 172.16.22.0 [120/2] 172.16.30.2 R 172.16.24.0 [120/2] 172.16.30.2	C 172.16.27.0 C 172.16.29.0 C 172.16.30.0 R 172.16.20.0 [120/1] 172.16.30.1 R 172.16.28.0 [120/1] 172.16.30.1 [120/1] 172.16.27.2 R 172.16.25.0 [120/1] 172.16.27.2 R 172.16.23.0 [120/1] 172.16.27.2 R 172.16.22.0 [120/1] 172.16.29.2 R 172.16.24.0 [120/1] 172.16.29.2 R 172.16.26.0 [120/1] 172.16.29.2 R 172.16.31.0 [120/1] 172.16.27.2 R 172.16.20.0 [120/1] 172.16.30.1
ROUTER C	**ROUTER D**
C 172.16.23.0 C 172.16.25.0 C 172.16.27.0 C 172.16.28.0 C 172.16.31.0 R 172.16.20.0 [120/1] 172.16.28.1 R 172.16.30.0 [120/1] 172.16.28.1 [120/1] 172.16.27.1 R 172.16.29.0 [120/1] 172.16.27.1 R 172.16.26.0 [120/1] 172.16.31.2 R 172.16.21.0 [120/1] 172.16.31.2 R 172.16.22.0 [120/2] 172.16.31.2 [120/2] 172.16.27.1 R 172.16.24.0 [120/2] 172.16.31.2 [120/2] 172.16.27.1	C 172.16.21.0 C 172.16.26.0 C 172.16.31.0 R 172.16.22.0 [120/1] 172.16.26.1 R 172.16.24.0 [120/1] 172.16.26.1 R 172.16.29.0 [120/1] 172.16.26.1 R 172.16.27.0 [120/1] 172.16.31.1 R 172.16.23.0 [120/1] 172.16.31.1 R 172.16.25.0 [120/1] 172.16.31.1 R 172.16.28.0 [120/1] 172.16.31.1 R 172.16.30.0 [120/2] 172.16.31.1 [120/2] 172.16.26.1 R 172.16.20.0 [120/2] 172.16.31.1

ROUTER E	R 172.16.23.0 [120/2] 172.16.26.2
C 172.16.22.0	[120/2] 172.16.29.1
C 172.16.24.0	R 172.16.25.0 [120/2] 172.16.26.2
C 172.16.26.0	[120/2] 172.16.29.1
C 172.16.29.0	R 172.16.30.0 [120/1] 172.16.29.1
R 172.16.21.0 [120/1] 172.16.26.2	R 172.16.28.0 [120/2] 172.16.26.2
R 172.16.27.0 [120/1] 172.16.29.1	[120/2] 172.16.29.1
R 172.16.31.0 [120/1] 172.16.26.2	R 172.16.20.0 [120/2] 172.16.29.1

Семинар 7.
Фильтрация сетевого трафика

7.1. Цель и задачи семинара

На семинаре закрепляются знания по сетевой маршрутизации: принципы маршрутизации, критерии выбора маршрута, методы и протоколы маршрутизации. Проводится сравнительный анализ различных подходов к маршрутизации. Рассматривается назначение о общее устройство маршрутизатора, основные структуры данных: таблица маршрутов, топологическая база данных и др. Проводится разбор и решение типовых задач на составление таблиц маршрутов по известной топологии и на восстановление топологии по известным таблицам маршрутов. По итогам семинара проводится контрольная работа на решение задач рассмотренного типа. Примеры задач приведены в фонде оценочных средств дисциплины.

7.2. Теоретическая часть

7.2.1. Списки контроля доступа – назначение

Список контроля доступа (англ. access control list, ACL) – это последовательность команд IOS, определяющих порядок обработки пакетов маршрутизатором, а именно: должен ли маршрутизатор пропустить или заблокировать тот или иной пакет, исходя из информации в заголовке пакета. Списки контроля доступа являются основой обеспечения информационной безопасности на границе сетей и одной из наиболее используемых функций операционной системы Cisco IOS.

ACL предназначены для решения следующих классов задач:

– Ограничение сетевого трафика для повышения производительности сети. Например, если корпоративная политика запрещает видеотрафик в сети,

необходимо настроить и применить ACL, блокирующие данный тип трафика. Подобные меры значительно снижают нагрузку на сеть и повышают её производительность.

– Ограничение служебного трафика. Например, ACL могут ограничивать доставку обновлений маршрутизации на отдельных участках сети, что позволяет сократить излишний служебный трафик.

– Ограничение доступа извне к отдельным сетевым расположениям. ACL обеспечивают базовый уровень информационной безопасности периметра сети, выборочно разрешая и запрещая внешний доступ к отдельным устройствам, сетевым ресурсам или участкам сети.

– Ограничение доступа по отдельным протоколам. ACL позволяют осуществлять фильтрацию трафика на основе типа трафика, например, разрешить трафик http или электронной почты, но при этом блокировать весь трафик протоколов telnet и ssh.

– Ограничение доступа отдельным пользователям к внешним ресурсам. ACL допускают сортировку пользователей и устройств в целях разрешения или запрета доступа к отдельным сетевым службам. Например, можно разрешить или запретить доступ из различных VLAN к определённым ресурсам или сетевым службам.

Маршрутизатор (или иное сетевое устройство, поддерживающее ACL) работает как фильтр пакетов, пропускает или отбрасывает пакеты на основе правил фильтрации. Для этого из заголовков каждого пакета, поступившего на интерфейсы маршрутизатора, извлекается служебная информация, необходимая для принятия решения о продвижении или блокировке пакета.

Из заголовков третьего (сетевого) уровня:

– IP-адрес источника;

– IP-адрес назначения.

Дополнительно может извлекаться из заголовков четвертого (транспортного) уровня:

– порт источника TCP/UDP;

– порт назначения TCP/UDP.

7.2.2. Порядок применения ACL

Создание ACL не дает команды маршрутизатору на его использование. Созданный ACL – это просто некий список в глобальной конфигурации маршрутизатора. Для того, чтобы ACL начал работать, его необходимо применить к определенном типу трафика. Например, ACL можно прикрепить:

- к интерфейсу (пакетная фильтрация);
- к линии telnet (ограничения доступа к маршрутизатору);
- к каналу VPN (какой трафик нужно шифровать);
- к сортировщику QoS (какой трафик обрабатывать приоритетнее);
- к преобразованию NAT (какие адреса транслировать).

Применительно к пакетной фильтрации, ACL размещаются на интерфейсах. При прикреплении ACL к интерфейсу маршрутизатор начинает просматривать входящий (поступающий на маршрутизатор) и исходящий (покидающий маршрутизатор) трафик. Соответственно ACL могут быть размещены на входящем или на исходящем направлении.

Рассмотрим пример. Пусть из внутренней сети поступает пакет на интерфейс маршрутизатора fa0/1. Маршрутизатор проверяет наличие ACL на интерфейсе. Если он есть, дальнейшая обработка ведется по правилам списка строго в том порядке, в котором записаны правила. Если ACL разрешает продвижение, пакет передается на выходной интерфейс (допустим, fa0/0); если ACL запрещает продвижение пакета, пакет уничтожается. Если ACL на интерфейсе нет, пакет проходит без всяких ограничений. Перед отправкой пакета с выходного интерфейса, маршрутизатор проверяет интерфейс fa0/0 на наличие исходящего ACL (ACL может быть прикреплен на интерфейсе как входящий или исходящий).

При применении ACL к интерфейсам следует руководствоваться следующим правилом: **расширенные ACL следует размещать как можно ближе к источнику, стандартные же как можно ближе к получателю**. Это обеспечивает максимальную эффективность ACL. Вернемся к рассмотренному примеру. Прикрепление исходящего ACL на интерфейс fa0/1 будет неэффективным, хотя и ACL работать будет. Допустим, на маршрутизатор приходит tcp echo запрос (ping) для какого-то узла во внутренней сети. Маршрутизатор проверяет наличие ACL на внешнем интерфейсе fa0/0 – его нет, пакет проходит; далее проверяется внутренний интерфейс fa0/1, на данном интерфейсе есть ACL, настроенный как исходящий. Такая конфигурация не позволяет запрещенному пакету проникнуть во внутреннюю сеть, но прикрепление этого ACL к интерфейсу fa0/0 как входящего позволило бы отбросить запрещенный пакет сразу при поступлении на маршрутизатор, сэкономив его вычислительные ресурсы.

При этом **нельзя разместить более 1 ACL на интерфейс, на протокол, на направление**. Другими словами, на одном интерфейсе одного

маршрутизатора на входящем направлении для IP-протокола может быть назначен только один ACL.

Ещё одно правило, касающееся самих маршрутизаторов, ACL не действуют на трафик, сгенерированный самим маршрутизатором.

7.2.3. Структура ACL

ACL представляет собой последовательность текстовых команд, начинающихся с ключевого слова permit (разрешить) либо deny (запретить). Чтение правил происходит строго в том порядке в котором они записаны в конфигурационном файле. Соответственно, когда пакет поступает на интерфейс, проверяется первое условие. Если пакет удовлетворяет первому условию, дальнейшая его обработка прекращается: он либо проходит через ACL дальше, либо отбрасывается. В противном случае проверяется второе условие, и т.д. пока не будут проверены все условия. Если пакет не удовлетворяет ни одному условию, он будет отброшен, т.к. **в конце каждого списка стоит неявное правило deny any** (запретить всё). Это правило не вводится вручную и не отображается при просмотре списка, но срабатывает последним при просмотре каждого ACL по умолчанию, т.е. если ни одно из других правил не сработало.

Рассмотрим пример (рис. 7.1). У пакетов, поступающих на маршрутизатор через интерфейс G0/0, проверяются адреса их источника на основе следующих записей стандартного ACL:

```
access-list 2 deny 192.168.10.10
access-list 2 permit 192.168.10.0 0.0.0.255
access-list 2 deny 192.168.0.0 0.0.255.255
access-list 2 permit 192.0.0.0 0.255.255.255
```

Напомним: когда трафик поступает на маршрутизатор, он сравнивается с записями в порядке, заданном в ACL-списке. Маршрутизатор продолжает обработку пакетов, пока не обнаружит совпадение. Маршрутизатор обрабатывает пакет на основе первого найденного совпадения, остальные записи маршрутизатором не учитываются.

Если к концу списка совпадения не найдены, маршрутизатор отклоняет трафик. Это объясняется тем, что по умолчанию в конце каждого ACL-списка содержится команда запрета для трафика, который не совпали ни с одной записью списка.

При создании ACL каждая его строка явно или неявно обозначается порядковым номером, по умолчанию в рамках десяти (10, 20, 30 и т.д). Такая

нумерация облегчает удаление конкретных записей, в т.ч. из середины списка, и на её место вставить другую.

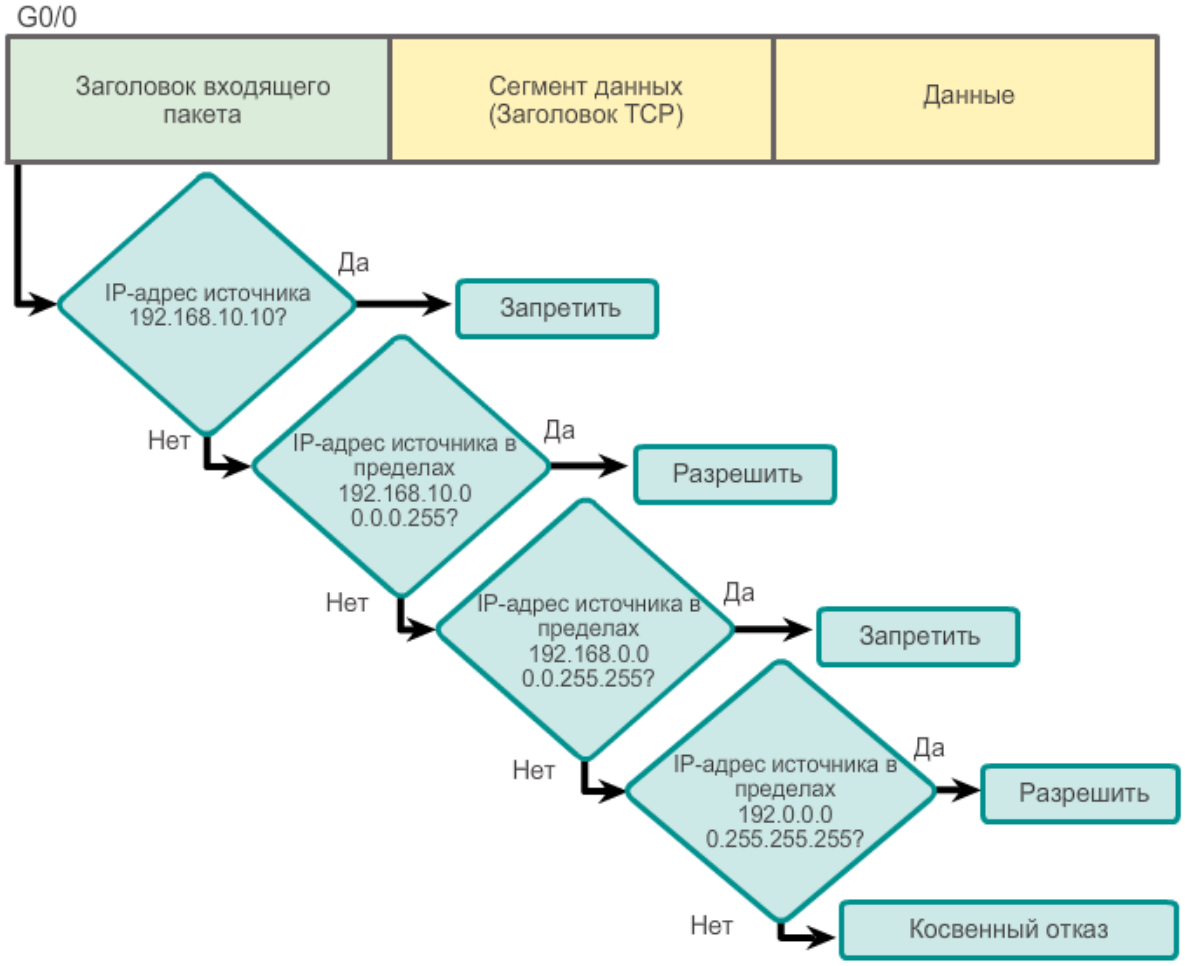


Рисунок 7.1. Логика работы ACL

7.2.3. Стандартные и расширенные ACL

В Cisco IOS предусмотрено два типа ACL для IPv4: стандартные и расширенные.

Стандартные (англ. standard) ACL используются для принятия решения о разрешении или блокировке пакета на основе единственного параметра: IPv4-адреса источника, например:

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

Данный список разрешает весь трафик из сети 192.168.30.0/24. Из-за неявного правила «deny any (запретить все, что не разрешено)» в конце списка данный ACL блокирует весь остальной трафик.

Расширенные (англ. extended) ACL фильтруют IPv4-пакеты, руководствуясь несколькими признаками:

- тип протокола;
- IPv4-адрес источника;
- IPv4-адрес назначения;
- TCP или UDP порт источника;
- TCP или UDP порт назначения.

Пример расширенного ACL:

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

Данный ACL разрешает трафику с любого адреса сети 192.168.30.0/24 идти в любую IPv4-сеть, если порт назначения – 80 (протокол http).

Стандартные и расширенные списки контроля доступа могут быть созданы с помощью номера или имени. Номер присваивается в зависимости то того, какой протокол будет фильтроваться:

- (от 1 до 99) и (от 1300 до 1999) – стандартные ACL протокола IP;
- (от 100 до 199) и (от 2000 до 2699) – расширенные ACL протокола IP.

Создание именованных ACL гораздо более удобно. При работе с именованными ACL используется собственный конфигурационный режим, упрощающий просмотр и редактирование ACL. Кроме этого, присвоение ACL имён упрощает понимание функции того или иного списка. Так, ACL, настроенному для запрета протокола ftp, рекомендуется присвоить имя, по смыслу которого будет понятно его назначение, например, «no_ftp». Именованные ACL также разделяются на стандартные и расширенные.

Расширенные ACL могут проверять гораздо больше параметров, нежели стандартные, но и работают они медленнее, так как маршрутизатору приходится проводить более глубокий анализ пакета, в отличие от стандартных ACL, где проверяется единственный параметр – IP-адрес отправителя.

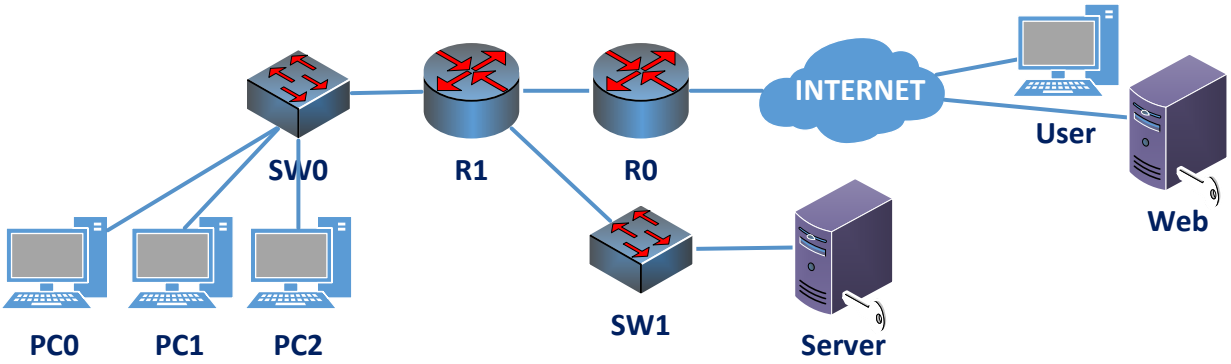
Кроме указанных типов ACL в Cisco IOS предусмотрены временные (англ. time-based), «зеркальные» (англ. reflexive) и динамические (англ. dynamic, lock-and-key) ACL. Временные ACL позволяют применять различные правила в различные дни и часы, например, ограничивать доступ к определенным ресурсам в рабочее время. Зеркальные ACL позволяют отслеживать состояние сессий, открытых из внутренней сети компании, и создавать соответствующие возвратные правила. Динамические ACL применяют те или иные правила в

зависимости от набора прав, полученных пользователем в результате аутентификации. Отметим, что последние три типа списков контроля доступа не будут рассмотрены в настоящем курсе.

7.3. Тренировочные задания

Составьте списки контроля доступа, выборочно разрешающий доступ компьютерам сети к серверам и к сети Интернет:

- Разрешить пользователям PC0, PC1 доступ к Web;
- Разрешить пользователям PC1, PC2 доступ к Server;
- Разрешить пользователям User (адрес неизвестен) доступ к Server;
- Запретить весь оставшийся трафик;



7.4. Рекомендуемая литература и Интернет-ресурсы

1. Cisco IOS ACLs [электронный ресурс]. – Режим доступа: https://habr.com/post/154879/ – Заглавие с экрана. Дата обращения: 17.10.2018.

2. ACL: списки контроля доступа в Cisco IOS [электронный ресурс]. – Режим доступа: https://habr.com/post/121806/ – Заглавие с экрана. Дата обращения: 17.10.2018.

3. Сети для самых маленьких. Часть пятая. ACL и NAT [электронный ресурс]. – Режим доступа: https://habr.com/post/147996/ – Заглавие с экрана. Дата обращения: 17.10.2018.

Семинар 8.
Адресация IPv6

8.1. Цель и задачи семинара

Семинар посвящен обзору особенностей пространства адресов IPv6 и формированию навыков расчета основных параметров IPv6 адресации. На семинаре проводится разбор решения простейших типов задач на IPv6 адресацию, аналогичных рассмотренным ранее задачам по IPv4: определение количества доступных адресов, расчет subnet и broadcast, подбор маски и др. По итогам обсуждения проводится контрольная работа на решение типовых задач. Примеры задач приведены в фонде оценочных средств дисциплины.

8.2. Теоретическая часть

8.2.1. Структура адреса IPv6

Адрес IPv6 представляет собой 128-битное двоичное число, то есть, он в 4 раза длиннее привычного нам 32-битного IPv4 адреса. Как и в случае IPv4, в нем выделяют две части: фиксированные биты (сеть) и свободные (узел). Свободные биты также называют идентификатором интерфейса (interface id). В отличие от IPv4, в IPv6 не применяются маски подсети, так как они получились бы очень длинными. Вместо масок используется префикс – число, обозначающее количество бит в фиксированной части (записывается через слеш после адреса). Например, префикс /64 означает, что из 128 бит адреса первые 64 являются фиксированными (идентификатор сети), а оставшаяся часть (в данном случае вторые 64 бита) – свободными (идентификатор интерфейса). Префикс описывает, сколько бит в адресе используется для хранения информации о сети.

Адреса IPv6 обычно записываются не в десятичном, а в шестнадцатеричном коде, что позволяет несколько сократить длину записи. Адрес разбивается на 8 групп по 16 бит, и каждая группа представляется четырьмя шестнадцатеричными цифрами. Такая группа называется хекстетом (от англ. hexadecimal – шестнадцатеричный). Хекстеты разделяются знаком двоеточия. Таким образом, адрес состоит из 8 хекстетов по 16 бит, что в сумме составляет ровно 128 бит.

8.2.2. Сокращение адресов IPv6

Пример адреса IPv6: 2001:0DB0:0000:123A:0000:0000:0000:0030. С таким длинным адресом работать (записывать, хранить, запоминать) не слишком удобно, поэтому часто применяется сокращённая форма записи.

Для получения сокращенной формы записи адреса IPv6 следует последовательно применить два правила.

Правило 1. В каждом хекстете (группе из 4 шестнадцатеричных цифр) ведущие нули удаляются. Например, во втором хекстете значение 0DB0 заменяется на DB0. Если хекстет состоит из одних нулей, то он заменяется на один нуль. Таким образом адрес 2001:0DB0:0000:123A:0000:0000:0000:0030 преобразуется в 2001:DB0:0:123A:0:0:0:30. А, например, адрес loopback (или localhost) 0000:0000:0000:0000:0000:0000:0000:0001 заменяется на 0:0:0:0:0:0:0:1.

Правило 2. Это правило применяется *строго после* первого. В адресе выбирается одна самая длинная группа, состоящая из полностью нулевых хекстетов, то есть самая длинная последовательность «:0:0:0:» и заменяется на два двоеточия «::» (выкидывается самая длинная последовательность подряд идущих нулевых хекстетов). Эту замену можно произвести только один раз и только с самой длинной последовательностью, в противном случае будет невозможно восстановить исходное значение адреса, т.к. будет невозможно установить, сколько именно хекстетов было заменено в первом и во втором случае. Важный момент: нельзя заменять двумя двоеточиями один нулевой хекстет.

Для примера возьмём адрес из предыдущей замены 2001:DB0:0:123A:0:0:0:30. Самая длинная последовательность из нулевых хекстетов – это «:0:0:0:», она начинается сразу после хекстета «123A». Есть ещё последовательность из одного пустого хекстета (между «DB0» и «123A»), но первая длиннее, поэтому заменить следует её. Длина адреса значительно уменьшится: 2001:DB0:0:123A::30. Это, конечно же, длиннее IPv4 адреса, но гораздо короче исходного.

8.2.3. Восстановление исходного адреса по сокращённой записи

Эта процедура достаточно тривиальна, т.к. повторяет процесс сокращения в обратном порядке. Рассмотрим ее на примере сокращенного адреса, рассмотренного в п.8.2.2: 2001:DB0:0:123A::30.

Вначале следует посчитать количество явно записанных хекстетов в адресе. В нашем случае, в адресе осталось 5 хекстетов. Известно, что полный адрес состоит из восьми хекстетов, следовательно, вместо «::» возвращаем три недостающих полных нулевых хекстета – получаем 2001:DB0:0:123A:0:0:0:30. Теперь в каждой группе, где меньше четырёх шестнадцатеричных цифр, дописываем слева нужное количество нулей. В результате такого преобразования получаем исходный адрес 2001:0DB0:0000:123A:0000:0000:0000:0030.

8.2.4. Виды адресов IPv6

Выделяется несколько типов адресов.

Глобальный юникаст (Global unicast) – это аналог публичных адресов в IPv4. Большая часть всех адресов относятся именно к этому классу. Эти адреса должны быть уникальными в пределах всей сети Интернет, они выдаются IANA региональным регистраторам, те выдают их провайдерам, а провайдеры – клиентам. Клиенту, как правило, выдаётся огромная сеть с префиксом длиной 64 бита (т.е. первые 64 бита – это идентификатор сети). Сама эта сеть тоже имеет иерархическую структуру. Как правило, региональный регистратор предоставляет провайдерам сети с префиксом 48 бит, а провайдеры добавляют ещё 16 бит и получают 65536 сетей с префиксом /64, которые затем предосталяются клиентам.

Диапазон таких адресов – это все адреса, у которых первые три бита равны «001», что означает все адреса, у которых первый хекстет лежит в диапазоне от 2000 до 3FFF. Из этой группы отдельно выделяется сеть 2001:0DB8::/32, которая, согласно спецификации, используется для примеров и документации.

Локальные адреса (Link-local) – адреса, использующиеся для взаимодействия с другими устройствами в той же локальной сети. Отличительной особенностью этих адресов является то, что трафик «с» или «на» эти адреса не маршрутизируется и в принципе не может выйти за пределы той сети, в которой он был создан. Уникальность от этих адресов не требуется – в каждой сети они могут быть одними и теми же. Адреса применяются для разных специальных целей, например, для процедуры обнаружения соседей (аналог ARP в IPv6). Диапазон таких адресов FE80::/10 – что означает все адреса у которых первый хекстет в диапазоне от FE80 до FEBF.

Мультикаст-адреса (Multicast) – адреса, использующийся для мультикастовой рассылки. Все эти адреса находятся в диапазоне FF00::/8, т.е.

все, которые начинаются с FF. Мультикаст в IPv6 выполняет важную роль, так как IPv6 не использует широковещательные адреса и для всех рассылок используется именно мультикаст.

Loopback – специальный адрес вида ::1. Все пакеты, идущие на него, не выходят за пределы устройства, а попадают обратно на уровень IP. Таким образом этот адрес является аналогом адресов из сети 127.0.0.0/8 в IPv4. Командой ping ::1 можно проверить, верно ли сконфигурирован на компьютере стек протоколов TCP/IP и IPv6 в частности.

Неопределённый адрес (Unspecified address) – адрес, состоящий из одних нулей. Записывается в сокращённой форме как «::». Такой адрес не может быть назначен интерфейсу, но может использоваться в некоторых пакетах в качестве адреса отправителя. Например, когда устройство ещё не получило IP адрес с помощью автоматической конфигурации (аналог DHCP для IPv6).

Уникальные локальные адреса (Unique local) – аналог приватных адресов в IPv4, то есть они могут маршрутизироваться в пределах нашей внутренней сети, но в открытых сегментах Интернета их анонсировать запрещено. Вообще, IPv6 подразумевает отказ от приватных адресов в том смысле, в котором они использовались до этого. В IPv4 приватные адреса применяются в основном из-за нехватки публичных и только иногда из соображений безопасности. В IPv6 использовать локальные адреса надо только в том случае, если по соображениям безопасности трафик из данной сети и в неё не должен уходить за пределы нашей зоны ответственности. Во всех остальных случаях следует использовать глобальные юникаст адреса.

Адреса IPv4, отображенные в IPv6 (IPv4 embedded) – это адреса вида ::ffff:xxxx:xxxx, где xxxx:xxxx – это некоторый IPv4 адрес, записанный в шестнадцатеричном виде. Эти адреса используются для устройств, не поддерживающих IPv6, и обеспечивают способ отображения адресного пространства старой версии протокола в адресное пространство новой.

8.3. Тренировочные задания

8.3.1. Упростить IPv6-адреса:

1. 2340:0000:0010:0100:1000:ABCD:0101:1010
2. 30A0:ABCD:EF12:3456:0ABC:B1B2:7563:4705
3. 2156:0214:7845:1234:0000:0000:6410:0AD4
4. 003A:0000:0000:1254:0000:0000:0000:0ADF

5. 210F:0000:0000:0000:CCCC:0000:0000:0000
6. 34BA:000B:00AC:0000:0000:0000:0000:00DB
7. FE80:0000:0000:0000:123E:00AF:ABCD:0123
8. FF80:0000:0000:0000:0123:1234:ABCD:EF12

9. FF02:0000:0000:0000:0000:0001:FF00:0300
10. 2001:0DB8:0000:1111:0000:0000:0000:0200
11. 0000:0000:0000:0000:0000:0000:0000:0001
12. 0000:0000:0000:0000:0000:0000:0000:0000

8.3.2. Развернуть адреса IPv6:

1. AC:0:0:AD::1
2. ::1
3. 3452:EDA:0:1::A12
4. AED:0:23::A
5. ::2AC:0:BAE

6. 23:0:1::1234:564
7. 45::23:0
8. 0:0:0:1::
9. AC:0:0:AE12::1
10. 12:0:0:ABCD::234

8.3.3. Просуммировать сети IPv6:

Пример 1.
1. FE80:1234:00AC:1:1234:AB12:0123:00AA/64
2. FE80:1234:00AC:2:1234:AB12:0123:00AA/64
3. FE80:1234:00AC:3:1234:AB12:0123:00AA/64
4. FE80:1234:00AC:4:1234:AB12:0123:00AA/64

Пример 2.
1. 2001:DB81:0012:ACD3:ABC2:002A:0:123/80
2. 2001:DB81:0012:ACD3:ABC2:002A:0:123/80
3. 2001:DB81:0012:ACD3:ABC2:002A:0:123/80
4. 2001:DB81:0012:ACD3:ABC2:002A:0:123/80

Пример 3.

```
1.  2001:DB81:123:1111::/64
2.  2001:DB81:123:2222::/64
3.  2001:DB81:123:3333::/64
4.  2001:DB81:123:4444::/64
```

8.4. Рекомендуемая литература и Интернет-ресурсы

1. IPv6 теория и практика: введение в IPv6 [электронный ресурс]. – Режим доступа: https://habr.com/post/210100/ – Заглавие с экрана. Дата обращения: 26.10.2018.

2. IPv6 теория и практика: виды пакетов и автоконфигурация [электронный ресурс]. – Режим доступа: https://habr.com/post/210224/ – Заглавие с экрана. Дата обращения: 26.10.2018.

Библиографический список

1. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов. – С-Пб.: Питер, 2015. – 944 с.

2. Сосенушкин С.Е., Левин М.В. Информационно-телекоммуникационные сети. Учебное пособие – М.: ФГБОУ ВО МГТУ «СТАНКИН», 2017. – 115 с.: ил.

3. Лэммл Т. Cisco Certified Network Associate. 640-407. Учебное Руководство. – М.: ООО ИД Вильямс, 2011. – 576 с.

4. С.Е. Сосенушкин, Д.Т. Назаров Сети ЭВМ и телекоммуникации. Моделирование компьютерных сетей: лабораторный практикум. – М.: ФГБОУ ВПО МГТУ «СТАНКИН», 2012 год.– 86 с.

5. Уэнделл Одом. Официальное руководство по подготовке к сертификационным экзаменам Cisco CCENT/CCNA ICND1 100-101. пер. с англ. В. А. Коваленко Академ. изд., 2012. – 912 с.

6. Уэнделл Одом. Официальное руководство по подготовке к сертификационным экзаменам Cisco CCENT/CCNA ICND2, 2-е изд. пер. с англ. – М.: ООО ИД Вильямс, 2013 – 736 с.

7. Э. Таненбаум, Д. Уэзеролл. Компьютерные сети Компьютерные сети. 5-е изд. – СПб.: Питер, 2012. – 960 с.

8. А.Н. Андрончик, А.С. Коллеров. Сетевая защита на базе технологий фирмы CiscoSystem. – Екатеринбург : Изд-во Урал. ун-та,2014. – 180 с.

9. Документация по RFC (Request for Comments) [Электронный ресурс]. – Режим доступа https://www.ietf.org/rfc.html, свободный (дата обращения: 20.12.2018).

10. Русские переводы RFC (Request for Comments) [Электронный ресурс]. – Режим доступа https://rfc2.ru, свободный (дата обращения: 20.12.2018).

11. Учебник «Networking Cisco Topics» [Электронный ресурс]. – Режим доступа https://networklessons.com/cisco, свободный (дата обращения: 20.12.2018).

12. Лаборатория сетевой безопасности «Your Private Network» [Электронный ресурс]. – Режим доступа http://ypn.ru, свободный (дата обращения: 20.12.2018).

13. Документация по технологиям компании Cisco [Электронный ресурс]. – Режим доступа http://xgu.ru/wiki/Категория:Cisco, свободный (дата обращения: 20.12.2018).

14. Блог «сетевые заморочки простыми словами о сложных вещах» [Электронный ресурс]. – Режим доступа http://www.netza.ru, свободный (дата обращения: 20.12.2018).

15. Блоги по технологиям и оборудованию Cisco от инструкторов «antiCisco blogs» [Электронный ресурс]. – Режим доступа http://www.anticisco.ru/blogs, свободный (дата обращения: 20.12.2018).

16. Блоги по системному администрированию [Электронный ресурс]. – Режим доступа http://www.k-max.name, свободный (дата обращения: 20.12.2018).

17. Технический блог специалистов ООО «Интерфейс» [Электронный ресурс]. – Режим доступа https://interface31.ru/tech_it, свободный (дата обращения: 02.01.2017).

18. Блог по сетевым технологиям «Ntwrk Notes» [Электронный ресурс]. – Режим доступа http://sk1f3r.ru, свободный (дата обращения: 20.12.2018).

19. Официальная документация по продукции компании Cisco [Электронный ресурс]. – Режим доступа http://www.cisco.com/c/en/us/support/index.html, свободный (дата обращения: 20.12.2018).

20. ISO/IEC 27000. Словарь и определения.

21. ISO/IEC 27001. Информационные технологии. Методы обеспечения безопасности. Системы управления информационной безопасностью. Требования.

22. ISO/IEC 17799:2005. Информационные технологии. Технологии безопасности. Практические правила менеджмента информационной безопасности.

23. Прончев Г. Б., Фесенко В. В., Брутов В. В., Михасев В. Г., Воробьев С. Г. Об актуальности локальных вычислительных сетей в настоящее время // Молодой ученый. – 2010. – № 12. Т.1. – С. 53–56.

24. Таненбаум Э., Уэзеролл Д. Компьютерные сети. – Питер, 2012. – 960 с.

25. Степанов А.Н. Архитектура вычислительных систем и компьютерных сетей – СПб.: Питер, 2007. – 509 с.: ISBN 978-5-469-01451-5 5-469-01451-7.

26. Новиков Ю. В., Кондратенко С. В. Основы локальных сетей. Курс лекций. – М.: Интернет-университет информационных технологий, 2005. – ISBN 5-9556-0032-9.

27. Самойленко В.В. Локальные сети. Полное руководство. – К., 2002. – ISBN 966-7140-28-8.

28. Локальные вычислительные сети: Справочник. В 3-х кн / Под.ред. С.В. Назарова. – М.: Финансы и статистика, 1994. – Т. Кн.1. Принципы построения, архитектура, коммуникационные средства. – 208 с. – ISBN 5-279-01171-1.

29. Cisco Systems, Inc. Программа сетевой академии Cisco CCNA 3 и 4. Вспомогательное руководство. – М.: «Вильямс», 2006. – С. 944. – ISBN 1-58713-113-7.

30. Groth, David and Skandler, Toby. Network+ Study Guide, Fourth Edition. Sybex, Inc. 2005. ISBN 0-7821-4406-3.

31. Forouzan, Behrouz. Data Communications and Networking. McGraw-Hill. p. 14. ISBN 9780073376226.

32. Сети и телекоммуникации: учеб. пособие / Б.В. Соболь, А.А. Манин, М.С. Герасименко. – Ростов н/Д.: Феникс, 2015. – 191 с.: ил. – (Высшее образование).

33. Самоучитель системного администратора / А.М. Кенин, Д.Н. Колисниченко. – 4-е изд., перераб. и доп. – СПб.: БХВ-Петербрг, 2016. – 528 с.: ил. – (Системный администратор).

34. Компьютерные сети for dummies [электронный ресурс]. – Режим доступа: https://habr.com/post/335816/ – Заглавие с экрана. Дата обращения: 09.10.2018.

35. Основы компьютерных сетей. Тема № 1. Основные сетевые термины и сетевые модели [электронный ресурс]. – Режим доступа: https://habr.com/post/307252/ – Заглавие с экрана. Дата обращения: 09.10.2018.

36. Основы компьютерных сетей. Тема № 3. Протоколы нижних уровней (транспортного, сетевого и канального) [электронный ресурс]. –Режим доступа: https://habr.com/post/308636/ – Заглавие с экрана. Дата обращения: 09.10.2018.

37. Основы компьютерных сетей. Тема № 4. Сетевые устройства и виды применяемых кабелей [электронный ресурс]. – Режим доступа: https://habr.com/post/312340/ – Заглавие с экрана. Дата обращения: 09.10.2018.

38. Основы компьютерных сетей. Тема № 5. Понятие IP адресации, масок подсетей и их расчет) [электронный ресурс]. – Режим доступа: https://habr.com/post/314484/ – Заглавие с экрана. Дата обращения: 09.10.2018.

39. Компьютерные сети for dummies [электронный ресурс]. – Режим доступа: https://habr.com/post/335816/ – Заглавие с экрана. Дата обращения: 09.10.2018.

40. Основы компьютерных сетей. Тема № 7. Протокол связующего дерева: STP [электронный ресурс]. – Режим доступа: https://habr.com/post/321132/ – Заглавие с экрана. Дата обращения: 10.10.2018.

41. Сети для самых маленьких. Часть четвертая. STP [электронный ресурс]. –Режим доступа: https://habr.com/post/143768/ – Заглавие с экрана. Дата обращения: 10.10.2018.

42. Принцип работы протокола STP [электронный ресурс]. – Режим доступа: https://habr.com/post/419491/ – Заглавие с экрана. Дата обращения: 10.10.2018.

43. Закольцованные сети, или зачем нам STP [электронный ресурс]. – Режим доступа: https://habr.com/post/129559/ – Заглавие с экрана. Дата обращения: 10.10.2018.

44. Cisco IOS ACLs [электронный ресурс]. – Режим доступа: https://habr.com/post/154879/ – Заглавие с экрана. Дата обращения: 17.10.2018.

45. ACL: списки контроля доступа в Cisco IOS [электронный ресурс]. – Режим доступа: https://habr.com/post/121806/ – Заглавие с экрана. Дата обращения: 17.10.2018.

46. Сети для самых маленьких. Часть пятая. ACL и NAT [электронный ресурс]. – Режим доступа: https://habr.com/post/147996/ – Заглавие с экрана. Дата обращения: 17.10.2018.

47. IPv6 теория и практика: введение в IPv6 [электронный ресурс]. – Режим доступа: https://habr.com/post/210100/ – Заглавие с экрана. Дата обращения: 26.10.2018.

48. IPv6 теория и практика: виды пакетов и автоконфигурация [электронный ресурс]. – Режим доступа: https://habr.com/post/210224/ – Заглавие с экрана. Дата обращения: 26.10.2018.