

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
"ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ"
Инженерно-технологическая академия

Е.С. АБРАМОВ, О.Ю. ПЕСКОВА, М.В.ТОКАРЕВ

**Практические задачи по администрированию
компьютерных сетей**

Учебно-методическое пособие

Ростов-на-Дону - Таганрог
Издательство Южного федерального университета
2018

УДК 004.451(075.8)

ББК 32.973я73

П 281

Рецензенты:

к.т.н., начальник отдела подготовки и обеспечения проведения
НИОКР АО "ТНИИС", г. Таганрог *М.И. Дулин*
кандидат технических наук *М.Н. Казарин*

Абрамов, Е.С.

П 281 Практические задачи по администрированию компьютерных сетей /
Е.С. Абрамов, О.Ю. Пескова, М.В. Токарев; – Таганрог: Издательство
Южного федерального университета, 2018. – 86 с.

Предназначено для практической подготовки и выполнения лабораторных работ по курсам «Мультисервисные сети» и «Компьютерные сети», а также может быть использовано при проведении других курсов, связанных с сетевым администрированием.

Содержит теоретический материал и практические примеры, описывающие решение различных практических задач, возникающих при администрировании компьютерных сетей.

Пособие ориентировано на студентов Южного федерального университета, обучающихся по специальностям 10.05.03 и 10.05.05, но может быть также полезно студентам других специальностей, предусматривающих изучение курсов по построению и администрированию компьютерных сетей, а также может быть применено для организации дополнительного профессионального обучения.

УДК 004.451(075.8)

ББК 32.973я73

П 281

© Южный федеральный университет, 2018
© Е.С. Абрамов, О.Ю. Пескова, М.В. Токарев, 2018
© Оформление. Макет. Издательство
Южного федерального университета, 2018

ВВЕДЕНИЕ

Цели освоения дисциплины «Мультисервисные сети» - формирование у студентов базовых знаний и компетенций в сфере организации и функционирования локальных информационно-вычислительных сетей и глобальной сети Интернет, необходимых для решения профессиональных задач, а также получение студентами практических умений и навыков построения, установки, конфигурирования, настройки, защиты, использования и сопровождения сетей в различных режимах функционирования.

Задачи дисциплины:

- ознакомить студентов с назначением, функциями, видами, классификацией, принципами построения и режимами функционирования, аппаратурой, компонентами сетей, способами построения сетей;
- ознакомить студентов с подходами к созданию уровня доступа и уровня распределения в сетях, планированию сетевой структуры;
- научить студентов понимать и учитывать многоуровневый подход к разработке средств сетевого взаимодействия;
- научить студентов разбираться в базовых сетевых технологиях, принципах и протоколах маршрутизации, стеке TCP/IP, адресации в IP-сетях;
- привить студентам умения и практические навыки эффективного проектирования, развертывания и обслуживания сетей различного масштаба и назначения.

В программу курса включены разделы, посвященные основным понятиям и структуре мультисервисных сетей, настройке сетевых операционных систем, изучению сетевых протоколов и коммуникаций, принципов IP-адресации, особенностям разделения сетей на подсети, изучению различных уровней сетевой модели OSI, а также концепции различных типов маршрутизации и коммутации.

Процесс изучения дисциплины направлен на формирование элементов общепрофессиональной компетенции в соответствии с собственным образовательным стандартом ЮФУ по специальности 10.05.03 «Информационная безопасность автоматизированных систем» и других собственных образовательных стандартов ЮФУ:

- ОПК-5 способность эффективно применять технические и программные средства и технологии в профессиональной деятельности.

Данное учебное пособие предназначено для практической подготовки и выполнения лабораторных работ по курсам «Мультисервисные сети» и «Компьютерные сети», а также может быть использовано при проведении других курсов, связанных с сетевым администрированием.

1. Работа с IP-адресами

1.1. Теоретическое введение

1.1.1. Двоичная и десятичная форма записи адресов

Адресация является важнейшей функцией протоколов сетевого уровня, которая обеспечивает обмен данными между узлами вне зависимости от того, находятся ли они в одной сети или в разных сетях. Протоколы IPv4 и IPv6 осуществляют иерархическую адресацию пакетов данных.

IPv4-адрес является иерархическим адресом, который состоит из двух частей: адреса сети и адреса узла. Биты в сетевой части адреса должны быть одинаковыми у всех устройств, находящихся в одной сети. Биты в узловой части адреса должны быть уникальными для каждого узла в этой сети. Если два узла имеют одинаковую битовую комбинацию в сетевой части адреса, то эти два узла находятся в одной и той же сети.

IP-адрес имеет длину 4 байта и обычно записывается в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками, например:

128.10.2.30 - традиционная десятичная форма записи IP-адреса,

10000000 00001010 00000010 00011110 - двоичная форма записи этого же адреса.

Эти 8-битные блоки, разделенные точками, называются октетами.

Чтобы переводить числа из двоичной в десятичную систему счисления, нужно понимать позиционную систему счисления. Принцип позиционной системы счисления заключается в том, что значение цифры определяется ее «позицией» в последовательности цифр. Наиболее распространенная система счисления — десятичная (с основанием 10). Для преобразования двоичного IPv4-адреса в десятичный эквивалент с точкой-разделителем сначала следует разделить IPv4-адрес на четыре 8-битных октета. Затем нужно внести двоичное позиционное значение в качестве двоичного числа первого октета и выполнить соответствующее вычисление.

Например, предположим, что IPv4-адрес узла — 11000000.10101000.00001011.00001010. Для преобразования двоичного адреса в десятичный формат, начнем с первого октета.

Введем 8-битное двоичное число 11000000 в качестве позиционного значения строки 1, как показано в таблице 1.1.

Таблица 1.1 - Преобразование двоичного значения первого октета в десятичное

Позиционное значение	$2^7=128$	$2^6=64$	$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
Значение бита из октета	1	1	0	0	0	0	0	0
Вычисления	$1*128$	$1*64$	$0*32$	$0*16$	$0*8$	$0*4$	$0*2$	$0*1$
Суммирование	128	64	0	0	0	0	0	0

В результате получим значение 192. Это число составит первый октет десятичной записи с точкой-разделителем.

Затем аналогично преобразуем второй октет, как показано в таблице 1.2:

Таблица 1.2 - Преобразование двоичного значения второго октета в десятичное

Позиционное значение	$2^7=128$	$2^6=64$	$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
Значение бита из октета	1	0	1	0	1	0	0	0
Вычисления	$1*128$	$0*64$	$1*32$	$0*16$	$1*8$	$0*4$	$0*2$	$0*1$
Суммирование	128	0	32	0	8	0	0	0

Преобразуем третий октет, как показано в таблице 1.3, и получим значение 11:

Таблица 1.3 - Преобразование двоичного значения третьего октета в десятичное

Позиционное значение	$2^7=128$	$2^6=64$	$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
Значение бита из октета	0	0	0	0	1	0	1	1
Вычисления	$0*128$	$0*64$	$0*32$	$0*16$	$1*8$	$0*4$	$1*2$	$1*1$
Суммирование	0	0	0	0	8	0	2	1

И, наконец, рассчитаем последний четвертый октет IP-адреса, как показано в таблице 1.4, и получим значение 10:

Таблица 1.4 - Преобразование двоичного значения третьего октета в десятичное

Позиционное значение	$2^7=128$	$2^6=64$	$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
Значение бита из октета	0	0	0	0	1	0	1	1
Вычисления	$0*128$	$0*64$	$0*32$	$0*16$	$1*8$	$0*4$	$1*2$	$0*1$
Суммирование	0	0	0	0	8	0	2	0

Конечный результат: 192.168.11.10.

Для обратного преобразования может использоваться следующий алгоритм ([6]):

1. Задается вопрос: больше ли или равно десятичное число в октете (n) самому старшему биту (128). Если нет, введите двоичный 0 в качестве позиционного значения числа 128. Если да, введите двоичную 1 в качестве позиционного значения числа 128 и вычтите 128 из десятичного числа.
2. Задается вопрос: больше ли или равен остаток (n) следующему по старшинству биту (64). Если нет, введите двоичный 0 в качестве позиционного значения числа 64; в противном случае введите двоичную 1 и вычтите 64 из десятичного числа.
3. Задается вопрос: больше ли или равен остаток (n) следующему по старшинству биту (32). Если нет, введите двоичный 0 в качестве позиционного значения числа 32; в противном случае введите двоичную 1 и вычтите 32 из десятичного числа.

Далее аналогичным способом продолжается вычисление десятичных значений до ввода всех позиционных значений для получения эквивалента в двоичном формате.

В качестве примера рассмотрим полученный нами адрес IP-адрес 192.168.11.10.

Воспользовавшись описанной выше процедурой, начнем с таблицы двоичных позиционных значений и первого десятичного числа 192.

Поскольку 192 больше 128, добавим 1 в качестве старшего позиционного значения, что соответствует числу 128. Затем вычтем 128 из 192; получаем разницу (остаток) 64.

Выполним сравнение числа 64 со следующим по старшинству битом 64. Поскольку они равны, добавим 1 в качестве следующего по старшинству позиционного значения. Введем двоичный 0 в оставшиеся позиции и получим двоичное значение первого октета — 11000000.

Следующий октет — 168. Выполним сравнение числа 168 со старшим битом 128. Поскольку 168 больше 128, укажем 1 в качестве старшего позиционного значения. Затем вычтем 128 из 168; получаем разницу (остаток) 40.

Дальше выполним сравнение числа 40 со следующим по старшинству битом 64. Поскольку 40 меньше 64, введем 0 в качестве следующего по старшинству позиционного значения.

Сравним 40 со следующим по старшинству битом 32. Поскольку 40 больше 32, укажем 1 в качестве позиционного значения и вычтем 32 из 40; получаем остаток 8. Число 8 соответствует конкретному позиционному значению. Поэтому введем 0 в качестве позиционного значения числа 16 и укажем 1 в качестве позиционного значения числа 8. Введем нули во все остальные позиции. Получим двоичное значение третьего октета — 10101000.

Третий октет — 11. В случае простых или небольших десятичных чисел процедуру вычитания можно пропустить. Это число можно довольно легко получить без вычитания ($8 + 2 + 1 = 11$). Двоичное значение второго октета — 00001011.

Четвертый октет — 10 ($8 + 2$). Таким образом, двоичное значение четвертого октета — 00001010.

1.1.2. Формат IP-адреса

Для определения сетевой и узловой частей адреса используется маска подсети. Единицы в маске подсети определяют сетевую часть, а нули — узловую часть.

Чтобы определить сетевой адрес IPv4-узла, к IPv4-адресу и маске подсети побитово применяется логическая операция И.

Логическое И — это сравнение двух битов:

$$1 \text{ И } 1 = 1$$

$$1 \text{ И } 0 = 0$$

$$0 \text{ И } 1 = 0$$

$$0 \text{ И } 0 = 0$$

Применение логической операции И к адресу и маске подсети в результате дает сетевой адрес.

В качестве примера использования операции И для определения сетевого адреса рассмотрим узел с IPv4-адресом 192.168.10.10 и маской подсети 255.255.255.0.

$$192.168.10.10_{10} = 11000000.10101000.00001010.00001010$$

$$255.255.255.0_{10} = 11111111.11111111.11111111.00000000$$

```

11000000.10101000.00001010.00001010
И
11111111.11111111.11111111.00000000
-----
11000000.10101000.00001010.00000000

```

Адрес сети:
11000000.10101000.00001010.00000000 = 192.168.10.0

Таким образом, узел 192.168.10.10 находится в сети 192.168.10.0 255.255.255.0.

Представление сетевых адресов и адресов узлов путем в виде маски подсети в десятичном формате с точкой-разделителем может быть очень громоздким. Сейчас обычно используется альтернативный, более простой, способ определения маски подсети, называемый длиной префикса.

Длина префикса означает количество бит в маске подсети, расположенных подряд и равных единице, отсчитываемое от начала адреса и до первого нуля. Она обозначается наклонной чертой вправо («/»), после которой идет подсчитанное количество единиц, например, для маски подсети 255.0.0.0 = 11111111.00000000.00000000.00000000 длина префикса будет обозначаться /8. А для маски подсети 255.255.255.252=11111111.11111111.11111111.11111100 длина префикса будет равна /30.

На рисунке 1.1 в первом столбце перечислены различные маски подсети, которые могут использоваться с адресом узла. Во втором столбце указан полученный 32-битный двоичный адрес. В последнем столбце указана полученная длина префикса.

Маска подсети	32-битный адрес	Длина префикса
255.0.0.0	11111111.00000000.00000000.00000000	/8
255.255.0.0	11111111.11111111.00000000.00000000	/16
255.255.255.0	11111111.11111111.11111111.00000000	/24
255.255.255.128	11111111.11111111.11111111.10000000	/25
255.255.255.192	11111111.11111111.11111111.11000000	/26
255.255.255.224	11111111.11111111.11111111.11100000	/27
255.255.255.240	11111111.11111111.11111111.11110000	/28
255.255.255.248	11111111.11111111.11111111.11111000	/29
255.255.255.252	11111111.11111111.11111111.11111100	/30

Рис. 1.1 Представление маски подсети через длину префикса [6]

1.1.3. Типы IP-адресов

Адрес и маска подсети ссылаются на конкретную сеть. Все узлы в сети имеют один сетевой адрес. В узловой части сетевого адреса представлены только нули.

Каждому сетевому адресу соответствуют набор адресов узлов, а также широковещательный адрес.

Адреса узлов – уникальные IP-адреса, назначаемые узлам и устройствам. В узловой части могут быть нули и единицы, но не могут быть только нули или только единицы.

Адрес первого узла – IP-адрес первого доступного узла в сети. Узловая часть всегда содержит одни нули и заканчивается на 1.

Адрес последнего узла – IP-адрес последнего доступного узла в сети. Узловая часть всегда содержит одни единицы и заканчивается на 0.

Широковещательный адрес – специальный адрес, обменивающийся данными со всеми узлами в сети. Например, если узел отправляет пакет на сетевой IPv4-адрес, пакет получают все другие узлы в этой сети. Для широковещательной рассылки используется верхний адрес диапазона сети. В узловой части — одни единицы.

Например, для сети 192.168.10.0 /24 (все нули в узловой части – в последнем октете) адрес первого узла - 192.168.10.1 /24 (последний октет содержит все 0 и заканчивается на 1), адрес последнего узла - 192.168.10.254 /24 (последний октет содержит все 1 и заканчивается на 0), широковещательный адрес - 192.168.10.255 (последний октет содержит все 1)

Узел, успешно подключенный к сети, может обмениваться данными с другими устройствами одним из трех способов.

- Одноадресная рассылка — процесс отправки пакета с одного узла на другой конкретный узел.
- Широковещательная рассылка — процесс отправки пакета с одного узла на все узлы в сети.
- Многоадресная рассылка — процесс отправки пакета с одного узла выбранной группе узлов, возможно, в различных сетях.

Одноадресная рассылка используется для обычного обмена данными между узлами как в сети типа «клиент/сервер», так и в одноранговой сети. Для одноадресной рассылки пакетов в качестве адреса назначения используются адреса устройства назначения.

Широковещательная передача используется для отправки пакетов всем узлам в сети через широковещательный сетевой адрес. Пакет широковещательной рассылки содержит IPv4-адрес назначения, в узловой части которого присутствуют только единицы. Это означает, что пакет получают и обрабатывают все узлы в локальной сети (домене широковещательной рассылки). Широковещательные рассылки предусмотрены во многих сетевых протоколах, например DHCP. Когда узел получает пакет, отправленный на широковещательный сетевой адрес, узел

обрабатывает пакет так же, как и пакет, отправленный на адрес одноадресной рассылки.

Многоадресная рассылка уменьшает трафик, позволяя узлу отправлять один пакет выбранной группе узлов, которые подписаны на группу многоадресной рассылки. Для многоадресной рассылки в протоколе IPv4 зарезервированы адреса от 224.0.0.0 до 239.255.255.255. Групповые IPv4-адреса от 224.0.0.0 до 224.0.0.255 зарезервированы для многоадресной рассылки в пределах локальной сети. Эти адреса используются для групп многоадресной рассылки в локальной сети. Обычно зарезервированные локальные адреса применяются в протоколах маршрутизации с помощью многоадресной передачи для обмена данными маршрутизации. Например, адрес 224.0.0.9 зарезервирован для протокола маршрутизации (Routing Information Protocol, RIP) версии 2 для обмена данными с другими маршрутизаторами RIPv2.

Эти три типа связи используются в сетях передачи данных для различных целей. Во всех трех типах IPv4-адрес узла источника размещен в заголовке пакета в качестве адреса источника.

Некоторые адреса (например, сетевые и широковещательные) нельзя назначать узлам. Также есть особые адреса, которые можно назначать узлам, но с ограничениями способов взаимодействия этих узлов в сети.

- Адреса loopback (127.0.0.0 /8 или от 127.0.0.1 до 127.255.255.254): чаще определяются как только один адрес 127.0.0.1 — это особые адреса, которые используют узлы, чтобы направлять трафик самим себе. Например, они могут использоваться узлом, чтобы проверить работоспособность конфигурации TCP/IP.
- Локальные адреса канала (169.254.0.0 /16 или от 169.254.0.1 до 169.254.255.254) более известны как адреса, назначаемые посредством автоматического назначения частных IP-адресов (Automatic Private IP Addressing, APIPA). Они используются клиентом Windows DHCP для самостоятельной конфигурации в случае, если ни один DHCP-сервер не доступен. Подходят для небольшой одноранговой сети.
- Адреса TEST-NET (192.0.2.0/24 или от 192.0.2.0 до 192.0.2.255) используются исключительно в целях обучения и могут использоваться в качестве примера для документирования при создании сетей.

Публичные IPv4-адреса представляют собой адреса, на глобальном уровне маршрутизируемые между маршрутизаторами интернет-провайдеров (Internet Service Provider, ISP), т.е. они должны быть уникальны. Они выдаются провайдерами, их количество строго ограничено. Для того, чтобы уменьшить дефицит адресного пространства IPv4, в середине 1990-х были введены частные IPv4-адреса, которые не являются уникальными и могут использоваться только для настройки адресации во внутренней сети. Частные адреса определены в RFC 1918.

В частности, блоками частных адресов являются:

10.0.0.0 /8 или от 10.0.0.0 до 10.255.255.255

172.16.0.0 /12 или от 172.16.0.0 до 172.31.255.255

192.168.0.0 /16 или от 192.168.0.0 до 192.168.255.255

1.1.4. Классы IP-адресов

В 1981 г. IPv4-адреса в сети Интернет назначались с помощью классовой адресации согласно RFC 790 (Назначенные адреса). Если рассматривать IP-адрес в двоичной форме записи, то выделяют следующие классы IP-адресов в зависимости от начальных бит адреса.



Рис.1.2

Заказчикам был назначен сетевой адрес на основе одного из трех классов, А, В или С. Согласно стандарту RFC, диапазоны индивидуальных адресов делятся на следующие классы:

- Класс А (от 0.0.0.0/8 до 127.0.0.0/8, адрес в двоичной форме начинается с 0) разработан для очень крупных сетей с более чем 16 млн адресов узлов (формально номера сетей **0** и **127** принадлежат к сети класса А, но на практике номер 0 имеет специальное употребление - для указания маршрута по умолчанию, а номер 127 зарезервирован для специальных целей). Для обозначения сетевого адреса IPv4-адреса класса А использовали фиксированный префикс /8 с первым октетом. Остальные три октета использовались для адресов узлов. Все адреса класса А требуют, чтобы самый старший разряд старшего октета был равен нулю. Это означает, что существовало только 128 возможных сетей класса А. Класс А показан на рис. 1.3.

Специфика класса А	
Блок адресов	0.0.0.0 – 127.0.0.0*
Маска подсети по умолчанию	/8 (255.0.0.0)
Максимальное количество сетей	128
Количество узлов в сети	16,777,214
Старший бит	0xxxxxxx.____.____.____

Рис.1.3. Класс А (*адреса 0.0.0.0 и 127.0.0.0 зарезервированы) [6]

- Класс В (128.0.0.0 /16 – 191.255.0.0 /16, первые два бита адреса равны 10) разработан для поддержки потребностей небольших и крупных сетей, содержащих приблизительно 65 000 узлов. Адрес класса В использовал фиксированный префикс /16, два старших октета для обозначения сетевого адреса. Оставшиеся два октета определяли адреса узлов. Для адресов класса В два самых старших разряда старшего октета равны 10, что обеспечивает возможность создания более 16 000 сетей. Класс В показан на рис. 1.4.

Специфика класса В	
Блок адресов	128.0.0.0 – 191.255.0.0
Маска подсети по умолчанию	/16 (255.255.0.0)
Максимальное количество сетей	16,384
Количество узлов в сети	65,534
Старший бит	10xxxxxx.____.____.____

Рис.1.4. Класс В [6]

- Класс С (192.0.0.0/24 – 223.255.255.0/24, адрес начинается с последовательности 110) предназначен для небольших сетей с количеством узлов не более 254. Блоки адресов класса С использовали префикс /24 для трех старших октетов для указания адреса сети и последний октет — для указания адресов узлов. Три старших бита старшего октета равны 110, что обеспечивает возможность создания более 2 млн сетей. Класс С показан на рис. 1.5.

Специфика класса С	
Блок адресов	192.0.0.0 - 223.255.255.0
Маска подсети по умолчанию	/24 (255.255.255.0)
Максимальное количество сетей	2,097,152
Количество узлов в сети	254
Старший бит	110xxxxx.____.____.____

Рис.1.5. Класс С [6]

Также имеется блок одноадресной передачи класса D (от 224.0.0.0 до 239.0.0.0, адрес начинается с последовательности 1110) и обозначает особый, групповой адрес - multicast. Если в пакете в качестве адреса назначения указан адрес класса D, то такой пакет должны получить все узлы, которые подписались на групповую рассылку по данному адресу.

Блок экспериментальных адресов класса E (от 240.0.0.0 до 255.0.0.0, адрес начинается с последовательности 11110) зарезервирован для будущих применений.

В таблице 1.5 приведены диапазоны номеров сетей, соответствующих каждому классу сетей.

Таблица 1.5 – Классовая адресация

Класс	Начальный адрес	Конечный адрес	Число сетей	Число узлов
A	0.0.0.0	127.255.255.255	128 (2^7)	16,777,214 ($2^{24}-2$)
B	128.0.0.0	191.255.255.255	16,384 (2^{14})	65,534 ($2^{16}-2$)
C	192.0.0.0	223.255.255.255	2,097,152 (2^{21})	254 (2^8-2)
D	224.0.0.0	239.255.255.255	Групповой адрес	
E	240.0.0.0	255.255.255.255	Зарезервировано	

В сетях класса А, В и С адрес начального диапазона внутри сети (т.е. для сети класса В 150.14.0.0, начальным адресом будет 150.14.0.0) называется адресом сети и используется для указания направления маршрутизации пакетов.

В сетях класса А, В и С адрес конечного диапазона внутри сети (т.е. для сети класса В 150.14.0.0, конечный адрес равен 150.14.255.255) имеет специальное назначение и называется "широковещательный" (broadcast) адрес. Данные, отправленные на этот адрес, дойдут до всех компьютеров, имеющих адреса внутри указанной сети (т.е. сеть 150.14.0.0).

В каждом из классов А, В и С выделяют диапазон адресов для нужд локальной сети. Администраторы сети вольны распоряжаться назначением этих адресов компьютерам своей сети.

- в сети класса А зарезервирована сеть 10.х.х.х
- в сети класса В зарезервирован диапазон сетей от 172.16.х.х до 172.31.х.х
- в сети класса С зарезервирован диапазон сетей от 192.168.0.х до 192.168.255.0

По классовой адресации 50 % доступных IPv4-адресов выделялось 128 сетям класса А, 25 % адресов — сетям класса В, и оставшиеся 25 % — сетям классов С, D и E.

1.1.5. Бесклассовая адресация

Первоначальная схема разбиения на классы сетей с течением времени показала свою практическую непригодность. Это было обусловлено тем, что для некоторых организаций сеть класса С (т.е. фактически 255 адресов) была слишком большой и реально из нее им требовалось всего лишь 4-10 адресов, а для других сеть С была маленькой, но сеть класса В опять уже слишком большой (так как там было уже 65535 адресов). конце 1990-х классовая адресация была заменена более новой и актуальной бесклассовой системой адресации (Classless InterDomain Routing - CIDR).

Было введено понятие длины маски сети, которая определяла количество бит в IP-адресе, отведенных под адрес сети, и, соответственно, число равное (32 – «длина маски») определяло число бит, отведенное под количество компьютеров, которое можно в этой сети использовать. Бесклассовая адресация явилась обобщением классовой адресации сетей.

Маска сети – это битовая маска, которая в двоичном коде представляема как последовательность 1 от старшего разряда в количестве равном длине маски сети и далее 0 до общей длины последовательности 32 бит.

К примеру, если у нас длина маски сети равна 19 бит, то маска сети будет иметь следующий вид:

Маска подсети: 11111111 11111111 11100000 00000000 (255.255.224.0)

Число компьютеров в такой сети $2^{32-19} = 2^{13} = 8192$.

Допустим, у нас есть IP-адрес 12.34.56.78 и маска сети /19. Чтобы получить адрес сети, зная IP-адрес и маску подсети, необходимо применить к ним операцию логическое И. Для получения широковещательного адреса внутри сети необходимо выполнить операцию логическое И адреса сети (или любой IP-адрес внутри этой сети) и инверсии маски сети:

IP-адрес: 00001100 00100010 00111000 01001110 (12.34.56.78)

Маска подсети: 11111111 11111111 11100000 00000000 (255.255.224.0)

Адрес сети: 00001100 00100010 00100000 00000000 (12.34.32.0)

Инверсия маски: 00000000 00000000 00011111 11111111 (0.0.31.255)

Широков. адрес: 00001100 00100010 00111111 11111111 (12.34.63.255)

1.2. Примеры решения задач

Задание 1.

По данным IP-адресам определить к сети какого класса они принадлежат, получить IP-адрес сети, маску сети и IP-адрес широковещательной рассылки в данной сети: 110.157.233.184

Решение:

первый октет = 110, поэтому это адрес класса А

адрес сети 110.0.0.0

маска сети 255.0.0.0

адрес шир. расс. 110.255.255.255

Задание 2.

Используйте IP-адреса из задания 1 и соответствующую длину маски сети, чтобы получить IP-адрес сети, маску сети и IP-адрес широковещательной рассылки в данной сети: 110.157.233.184/12

Сначала необходимо получить маску сети в явном виде:

/12 — это 12 единичных бит от 31 бита направо

11111111.11110000.00000000.00000000 или в десятичном виде 255.240.0.0

Так как результат логического И/ИЛИ байтового значения с 0 и 255 очевиден, то нам необходимо получить представление в двоичном виде лишь байта 157 нашего IP-адреса.

Чтобы получить адрес сети, нам необходимо выполнить операцию логического И между IP-адресом и маской сети:

```
110.100111012.233.184 И
255.111100002. 0. 0
-----
110.100100002. 0. 0 = 110.144.0.0 — адрес сети
```

Чтобы получить адрес широковещательной рассылки, необходимо выполнить операцию логического ИЛИ между IP-адресом и инверсией маски сети.

Получим инверсию маски сети:
000000002.000011112.111111112.111111112
или в десятичном виде
0.15.255.255

Тогда:
110.100111012.233.184 ИЛИ
0.000011112.255.255

110.100111112.255.255 = 110.159.255.255 — адрес широковещательной рассылки

Задание 3.

Является ли данная маска сети правильной, и какова ее длина в битах:
255.254.0.0

По определению маска сети является непрерывной последовательностью битов 1 от старшего разряда, после которых идут только биты 0. Поэтому необходимо перевести в двоичное представление указанные маски и проверить этот факт.

В двоичном виде 255.254.0.0 представимо как:

```
111111112.111111102.000000002.000000002
```

Как мы видим последовательность единиц идет от старшего бита IP-адреса и является непрерывной, следовательно эта маска является правильной и имеет длину 15 бит.

Задание 4.

Является ли данный IP-адрес адресом сети с указанной длиной маски сети: 228.0.0.0/3

Получим маску сети в явном виде:

/3 — это 3 единичных бит от 31 бита направо

$$111000002.0.0.0 = 224.0.0.0$$

Нам необходимо получить адрес сети по данному IP-адресу.

$$111001002.0.0.0 \text{ И}$$

$$111000002.0.0.0$$

$$111000002.0.0.0 = 224.0.0.0 \text{ — адрес сети}$$

Так как 224.0.0.0 не равен 228.0.0.0, то 228.0.0.0 не может выступать в качестве адреса сети с маской /3.

Задание 5.

Принадлежат ли указанные IP-адреса к одной подсети: 135.95.4.150 - 135.96.221.49/15

Чтобы узнать принадлежат ли адреса к одной подсети, необходимо получить адрес сети для каждого из адресов и сравнить адреса сетей.

Получим маску сети в явном виде:

/15 — это 15 единичных бит от 31 бита направо

$$111111112.111111102.0.0 = 255.254.0.0$$

Так как в нашей маске отличным от 0 и 255 является второй байт, то при выполнении операции логического И нам необходимо расписывать в двоичном виде только второй байт IP-адресов.

$$135.010111112. \quad 4.150 \text{ И}$$

$$255.111111102. 0. 0$$

$$135.010111102. 0. 0 = 135.94.0.0 \text{ — адрес сети для 1-ого IP-адреса}$$

135.011000002.221. 49 И
255.111111102. 0. 0

135.011000002. 0. 0 = 135.96.0.0 — адрес сети для 2-ого IP-адреса

Адреса сетей не совпадают, значит указанные в задании IP-адреса не могут лежать в одной подсети с длиной маски 15 бит.

Задание 6.

Определить максимальную длину маски сети, чтобы указанные IP-адреса находились в одной сети: 24.177.20.45 - 24.177.23.169

Чтобы определить максимальную длину маски сети необходимо перевести в двоичное представление оба адреса и посчитать число совпадающих бит, начиная со старшего бита, до первого различия.

В нашем задании первые два байта IP-адресов совпадают, и поэтому их не нужно переводить в двоичное представление. Так как каждый байт — это 8 бит, то мы уже имеем $8 * 2 = 16$ совпадающих бит.

Рассмотрим третий байт IP-адресов. В двоичном виде (не забываем про незначащие разряды, которые равны 0!):

20 = 0 0 0 1 0 1 0 0 2

23 = 0 0 0 1 0 1 1 1 2

В третьем байте совпадают 6 бит. Таким образом, всего совпадает $16 + 6 = 22$ бит. Поэтому максимальная длина маски сети, при которой оба указанных IP-адреса будут лежать в одной подсети — это 22 бит.

1.3. Варианты

1. По данным IP-адресам определить, к сети какого класса они принадлежат, получить IP-адрес сети, маску сети и IP-адрес широковещательной рассылки в данной сети:

Вариант				
1	36.24.212.27	151.204.234.208	167.143.166.151	81.207.5.124
2	187.196.89.86	37.38.56.94	194.3.50.241	35.42.64.114
3	42.160.157.215	75.59.233.215	163.143.246.230	218.161.0.172
4	45.45.183.158	10.128.217.44	56.86.29.157	186.113.68.173
5	65.72.172.57	191.194.186.67	117.39.255.239	203.80.81.87
6	98.152.43.182	19.160.138.248	78.123.49.191	205.44.61.253
7	182.76.142.213	80.117.227.93	137.225.232.195	160.22.40.236
8	168.173.44.192	37.73.200.123	213.180.159.172	20.55.186.108
9	56.99.61.195	49.229.236.82	55.23.59.226	4.6.214.143
10	110.157.233.184	159.57.141.205	195.137.48.42	190.30.134.79
11	209.91.67.50	158.133.84.236	168.168.105.250	37.108.141.213
12	7.138.74.144	59.27.242.99	132.219.211.86	54.157.52.232
13	136.203.39.139	3.155.81.90	213.255.238.108	105.243.46.212
14	103.250.75.224	83.252.152.35	208.90.192.85	18.245.178.92
15	167.212.40.42	116.199.97.6	144.104.247.170	1.160.40.122
16	23.98.154.27	184.88.219.125	181.64.49.214	179.9.247.251
17	164.238.74.151	99.18.173.124	24.179.162.91	211.153.106.68
18	180.188.147.97	33.186.227.159	13.90.160.97	191.82.177.209
19	189.199.185.101	164.150.57.99	158.46.195.89	116.195.98.49
20	24.48.130.213	100.225.123.180	62.110.158.124	141.162.24.144
21	3.52.113.141	78.177.231.132	123.231.71.121	103.40.12.25
22	32.201.59.140	125.126.183.49	174.224.51.153	223.177.188.195
23	96.51.61.102	173.196.70.227	133.182.215.218	15.49.14.69
24	98.64.253.7	113.130.115.57	44.66.25.36	84.132.112.84
25	221.244.6.39	204.140.56.227	99.223.163.193	180.177.238.93
26	101.208.168.64	58.245.154.7	119.225.239.162	79.154.67.97

2. Используйте IP-адреса из задания I и соответствующую длину маски сети, чтобы получить IP-адрес сети, маску сети и IP-адрес широковещательной рассылки в данной сети:

Вариант				
1	/30	/18	/20	/28
2	/6	/21	/26	/10
3	/12	/7	/17	/15
4	/24	/3	/23	/8
5	/26	/13	/20	/27
6	/4	/10	/25	/28
7	/28	/24	/18	/3
8	/10	/14	/20	/9
9	/11	/4	/23	/14
10	/17	/25	/26	/20
11	/10	/27	/29	/11
12	/27	/14	/21	/15
13	/15	/29	/14	/19
14	/17	/10	/21	/13
15	/13	/30	/27	/7
16	/21	/21	/19	/12
17	/27	/27	/18	/23
18	/23	/16	/26	/25
19	/5	/22	/13	/17
20	/8	/11	/20	/20
21	/4	/18	/22	/8
22	/18	/10	/23	/11
23	/26	/20	/13	/18
24	/9	/23	/12	/19
25	/11	/30	/18	/21
26	/14	/28	/21	/6

3. Является ли данная маска сети правильной и какова ее длина в битах:

(По определению маска сети является непрерывной последовательностью битов 1 от старшего разряда после которых идут только биты 0. Поэтому необходимо перевести в двоичное представление указанные маски и проверить этот факт).

Вариант				
1	255.254.0.0	255.255.255.214	255.255.255.248	255.255.248.0
2	255.255.255.0	255.255.255.240	255.253.0.0	255.255.252.0
3	255.255.252.0	255.255.255.192	255.7.0.0	248.0.0.0
4	255.254.0.0	255.255.248.0	240.0.3.0	255.255.255.248
5	248.0.0.0	255.249.0.0	255.255.255.240	224.0.0.0
6	255.255.0.0	255.253.0.0	255.255.0.0	255.255.0.0
7	255.248.0.0	255.255.240.0	255.255.254.0	255.255.255.254
8	255.224.0.0	252.2.0.0	255.240.0.0	255.255.255.240
9	255.255.255.248	255.255.255.252	255.255.248.0	192.0.0.0
10	255.248.9.0	255.255.255.0	255.248.0.0	254.0.0.0
11	255.255.225.255	255.255.193.0	255.255.0.0	255.255.255.128
12	255.255.255.252	255.255.255.128	255.255.255.248	255.192.0.0
13	255.224.0.0	250.0.0.0	255.255.254.0	192.0.0.0
14	255.240.0.0	255.255.192.04	255.255.255.252	255.240.0.0
15	255.255.255.128	255.240.0.0	224.0.0.0	255.224.224.0
16	224.0.0.255	255.192.0.0	255.255.255.240	255.252.0.0
17	255.129.0.0	255.255.248.0	255.255.192.0	254.0.0.0
18	248.0.0.0	255.128.8.0	192.0.0.0	255.128.0.0
19	255.255.255.128	255.255.250.254	255.255.255.192	248.0.0.0
20	255.192.254.0	255.255.255.192	255.128.0.0	255.255.252.0
21	255.0.0.0	255.224.10.0	252.0.0.0	255.255.224.0
22	255.252.11.0	248.0.0.0	255.255.248.0	255.255.255.240
23	255.155.255.255	240.0.0.0	254.0.0.0	255.252.0.0
24	255.255.248.0	255.255.254.0	255.255.224.0	255.125.128.0
25	255.205.255.0	255.255.255.252	255.255.255.0	255.224.0.0
26	224.0.0.0	255.255.255.0	240.255.0.0	224.0.0.0

4. Является ли данный IP-адрес адресом сети с указанной длиной маски сети:

(необходимо вычислить по данному IP-адресу адрес сети и сравнить с исходным адресом, указанным в задании)

Вариант				
1	185.129.0.0/9	80.0.0.0/5	100.241.96.0/22	129.199.93.82/31
2	185.214.114.0/22	85.0.0.0/7	157.143.151.177/29	58.189.128.0/17
3	128.0.0.0/2	1.193.76.0/24	127.12.0.0/14	134.0.0.0/6
4	120.118.0.0/12	195.165.102.0/18	184.98.36.0/24	200.0.0.0/5
5	32.0.0.0/3	15.53.210.202/30	240.97.66.0/18	189.66.194.64/26
6	152.228.0.0/14	229.0.0.0/3	126.17.238.0/23	66.37.0.0/16
7	146.0.0.0/11	88.142.0.0/14	107.212.0.0/14	202.58.239.204/3
8	65.0.0.0/7	73.100.0.0/17	105.213.190.0/23	169.22.0.0/15
9	80.243.8.200/31	7.81.247.0/21	40.127.40.54/31	222.117.148.0/22
10	32.10.0.0/9	95.81.1-8.0/18	68.111.8.0/22	52.96.0.0/11
11	43.51.83.162/27	21.96.100.0/11	105.49.54.226/31	164.0.0.0/7
12	122.0.0.0/5	67.109.141.105/30	161.249.88.0/25	104.184.0.0/13
13	33.245.254.0/22	152.0.0.0/6	46.126.200.209/30	155.80.0.0/18
14	147.0.0.0/8	138.182.0.0/14	7.117.120.60/32	112.0.0.0/6
15	127.160.0.0/11	27.100.136.87/29	17.91.200.10/21	166.51.64.0/19
16	236.181.31.134/31	108.21.68.0/23	159.0.0.0/7	178.190.114.180/3
17	6.30.97.0/28	87.104.0.0/14	153.11.102.90/29	96.0.0.0/4
18	182.0.0.0/5	55.204.36.75/30	116.200.156.0/24	128.0.0.0/5
19	104.14.0.0/16	81.0.0.0/10	192.76.12.0/25	135.87.12.0/22
20	157.207.130.0/25	127.3.108.0/23	96.30.0.0/5	128.0.0.0/5
21	121.156.142.0/22	139.128.0.0/9	213.195.0.0/13	144.0.0.0/5
22	48.85.174.0/20	135.128.0.0/10	207.0.0.0/4	4.121.231.192/26
23	196.118.169.133/30	188.128.0.0/10	32.20.0.0/6	128.0.0.0/2
24	112.98.0.0/16	232.159.229.89/29	33.64.0.0/14	87.180.176.0/23
25	106.212.235.0/25	104.200.76.0/31	10.200.0.0/8	117.60.0.0/14
26	194.0.0.0/7	105.227.0.0/11	191.134.130.192/28	239.134.0.0/13

5. Принадлежат ли указанные IP-адреса к одной подсети:

(чтобы узнать принадлежат ли адреса к одной подсети, необходимо получить адрес сети для каждого из адресов и сравнить адреса сетей)

Вариант		
1	229.52.17.190 - 229.50.17.191/30	226.144.183.64 - 226.128.186.152/9
2	223.62.19.244 - 223.67.176.98/14	67.50.242.243 - 67.50.200.172/18
3	127.73.18.240 - 137.114.177.17/9	195.94.59.188 - 195.94.59.191/30
4	185.63.56.182 - 85.63.239.16/16	199.57.36.63 - 199.57.5.169/15
5	136.61.83.119 - 111.181.218.52/5	125.60.255.103 - 125.34.169.199/9
6	133.206.62.249 - 133.105.92.88/11	192.243.42.162 - 192.243.42.246/25
7	94.176.91.111 - 94.176.92.80/20	4.244.159.102 - 4.246.125.165/12
8	47.88.172.145 - 47.88.178.192/21	203.40.171.158 - 203.40.141.180/18
9	244.23.38.153 - 244.23.78.154/29	28.3.34.25 - 19.109.158.253/4
10	123.65.168.74 - 123.65.164.72/27	110.71.140.119 - 110.67.85.239/9
11	116.75.124.87 - 116.75.124.85/20	135.143.91.179 - 135.143.87.229/20
12	253.130.198.145 - 253.130.198.145/22	37.125.13.168 - 37.125.15.13/21
13	108.11.214.167 - 108.11.223.5/19	246.235.45.207 - 246.235.45.215/29
14	74.28.237.200 - 74.28.237.203/25	181.84.249.67 - 181.65.130.204/9
15	199.123.3.50 - 199.123.3.101/23	100.101.216.145 - 100.182.234.25/5
16	24.52.254.96 - 24.52.252.93/21	206.240.138.123 - 206.242.138.65/26
17	125.160.27.126 - 125.160.27.104/29	90.11.41.223 - 90.11.36.71/20
18	245.147.217.10 - 245.137.208.239/20	8.215.223.7 - 8.215.221.121/22
19	203.229.237.163 - 203.229.236.44/24	50.140.6.93 - 50.137.106.16/12
20	138.38.89.122 - 138.38.89.102/27	33.57.125.225 - 33.105.28.206/10
21	1.155.84.168 - 1.155.87.159/25	218.21.244.169 - 218.21.247.183/21
22	107.105.106.169 - 107.121.225.62/12	150.135.197.141 - 150.175.141.163/6
23	219.115.4.199 - 219.113.224.101/14	194.104.201.41 - 194.112.152.83/14
24	128.77.223.26 - 128.77.220.172/18	136.95.4.150 - 136.96.221.49/15
25	111.44.22.209 - 111.231.92.245/8	50.22.147.220 - 50.22.147.221/21
26	243.212.122.21 - 243.204.143.79/10	242.251.231.41 - 242.251.231.42/19

6. Определить максимальную длину маски сети, чтобы указанные IP-адреса находились в одной сети:

(чтобы определить максимальную длину маски сети необходимо перевести в двоичное представление оба адреса и посчитать число совпадающих бит, начиная со старшего бита до первого различия)

Вариант		
1	221.220.88.73 - 223.222.74.206	32.102.0.46 - 32.102.0.47
2	102.244.10.49 - 102.244.10.26	235.41.199.239 - 235.41.41.139
3	251.252.230.152 - 251.250.29.97	54.134.17.147 - 54.10.33.193
4	162.235.231.229 - 160.93.14.253	18.10.124.128 - 18.10.124.169
5	99.149.26.16 - 99.149.26.16	199.225.66.216 - 199.225.66.247
6	250.54.84.49 - 214.7.75.249	149.182.180.56 - 151.66.167.26
7	231.81.216.237 - 231.81.212.30	177.77.34.213 - 191.35.196.43
8	115.115.32.253 - 114.14.56.227	62.225.77.124 - 62.225.76.103
9	184.155.179.54 - 184.155.66.71	251.106.185.206 - 251.126.234.156
10	246.168.67.154 - 246.169.9.220	48.107.202.223 - 48.107.203.56
11	23.115.247.150 - 23.48.37.248	95.129.111.1 - 95.129.111.3
12	207.234.120.181 - 207.234.120.181	38.23.81.102 - 38.127.45.239
13	150.27.130.246 - 150.18.140.87	166.220.34.180 - 166.220.34.183
14	51.79.155.111 - 51.75.182.175	112.56.206.224 - 112.56.202.104
15	236.74.83.193 - 236.75.195.217	12.95.127.35 - 12.131.135.175
16	123.157.136.13 - 123.165.203.131	196.200.12.115 - 196.200.12.116
17	91.1.129.158 - 91.1.172.242	220.225.247.23 - 220.225.71.91
18	5.35.95.106 - 9.58.248.150	226.4.22.186 - 226.163.205.38
19	159.218.202.36 - 159.218.156.20	141.85.107.17 - 141.85.107.97
20	247.242.52.247 - 247.66.88.19	2.57.42.80 - 2.56.92.124
21	120.149.163.181 - 120.186.35.7	41.0.254.221 - 47.86.238.81
22	179.76.216.76 - 179.76.216.76	0.42.239.218 - 19.83.23.66
23	182.133.171.215 - 182.133.221.50	122.186.87.171 - 122.186.87.170
24	11.204.240.150 - 11.204.240.222	225.185.154.217 - 225.185.154.208
25	226.61.98.224 - 226.61.18.215	24.173.207.45 - 24.177.233.169
26	35.115.185.74 - 35.113.230.137	208.114.254.251 - 208.114.254.203

2. Построение таблиц маршрутизации

2.1. Теоретическое введение

Маршрутизация — это процесс определения наилучшего пути к узлу назначения.

Определение оптимального пути подразумевает оценку нескольких путей в одну и ту же сеть назначения и выбор оптимального или кратчайшего пути для прохождения этого маршрута. Когда существует несколько путей до одной сети, каждый путь использует различный выходной интерфейс маршрутизатора для достижения сети.

Протокол маршрутизации выбирает наилучший путь, исходя из значения или метрики, используемых для определения расстояния до сети. Метрика — это числовое значение, используемое для измерения расстояния до заданной сети. Наиболее оптимальным путем к сети является путь с наименьшей метрикой.

Маршрутизатор, подключенный к сегменту локальной сети и направляющий трафик в другие сети, называется шлюзом по умолчанию.

Когда узел отправляет пакет другому узлу, он использует свою таблицу маршрутизации, чтобы определить место отправки пакета. Если узел назначения находится в удаленной сети, пакет пересылается на шлюз по умолчанию. Когда пакет прибывает на шлюз по умолчанию, то маршрутизатор определяет вариант пересылки пакета по своей таблице маршрутизации.

Таблица маршрутизации предоставляет информацию о маршрутизации для сетей с прямым подключением и удаленных сетей, а также о порядке определения маршрута, его достоверности и рейтинге, когда маршрут был последний раз обновлен и какой интерфейс следует использовать, чтобы достичь запрашиваемого назначения.

Когда на интерфейс маршрутизатора поступает пакет, маршрутизатор анализирует его заголовок, чтобы определить сеть назначения. Здесь возможны три варианта:

- Сеть с прямым подключением — если IP-адрес назначения пакета принадлежит устройству в сети с прямым подключением к одному из интерфейсов маршрутизатора, то этот пакет пересылается напрямую в устройство назначения. Это означает, что IP-адрес назначения пакета — это узловой адрес в той же подсети, что и интерфейс маршрутизатора.
- Удаленная сеть — если IP-адрес назначения пакета принадлежит удаленной сети, пакет пересылается на другой маршрутизатор. Отправить пакет в удаленные сети можно только с помощью пересылки на другой маршрутизатор.
- Маршрут не определен — если IP-адрес назначения пакета не принадлежит подключенной или удаленной сети,

маршрутизатору нужно определить, доступен ли «шлюз последней надежды». «Шлюз последней надежды» задается, когда на маршрутизаторе настроен или известен маршрут по умолчанию. Если есть маршрут по умолчанию, то пакет пересылается на «шлюз последней надежды». Если маршрутизатор не располагает маршрутом по умолчанию, то пакет отбрасывается.

Если сеть назначения совпадает с маршрутом в таблице маршрутизации, маршрутизатор пересылает пакет, используя информацию в таблице маршрутизации. Если существуют два и более вероятных маршрута к одному пункту назначения для определения маршрута, который появится в таблице маршрутизации, используется метрика.

В таблице маршрутизации маршрутизатора может храниться следующая информация:

- Маршруты с прямым подключением. Эти маршруты предоставляются активными интерфейсами маршрутизаторов. Маршрутизаторы добавляют маршрут с прямым подключением, когда интерфейс настроен с IP-адресом и активирован. Каждый из интерфейсов маршрутизатора подключен к разному сегменту сети.
- Удаленные маршруты. Эти маршруты предоставляются удаленными сетями, подключенными к другим маршрутизаторам. Маршруты к этим сетям могут быть настроены на локальном маршрутизаторе вручную сетевым администратором или назначены динамически с помощью локального маршрутизатора, который обменивается данными маршрутизации с другими маршрутизаторами, используя для этого протоколы динамической маршрутизации.
- Маршрут по умолчанию. Подобно узлу, маршрутизаторы также используют маршрут по умолчанию в качестве последнего средства, если иного маршрута до нужной сети в таблице маршрутизации нет.

При активировании интерфейса маршрутизатора, настроенного с помощью IPv4-адреса и маски подсети, автоматически создаются следующие два элемента таблицы маршрутизации.

- С означает сеть с прямым подключением. Сети с прямым подключением создаются автоматически, когда интерфейс настраивается с помощью IP-адреса и активируется.
- L означает, что это локальный интерфейс. Это IPv4-адрес интерфейса на маршрутизаторе.

Также в процессе работы добавляются следующие виды маршрутов для удаленных сетей:

- Статические маршруты — добавляются, когда маршрут настроен вручную и активен выходной интерфейс.
- Протокол динамической маршрутизации — добавляется, когда определены сети и реализуются протоколы маршрутизации, которые получают информацию о сети динамически, например EIGRP или OSPF.

В таблице можно получить информацию о следующих параметрах маршрута:

1. Источник маршрута — определяет, каким образом маршрутизатор получил сведения о сети, наиболее распространенными следующие варианты:
 - L — указывает адрес, назначенный интерфейсу маршрутизатора. Данный код позволяет маршрутизатору быстро определить, что полученный пакет предназначен для интерфейса, а не для пересылки.
 - C — определяет сеть с прямым подключением.
 - S — определяет статический маршрут,
 - D — определяет сеть, динамически полученную от другого маршрутизатора с помощью улучшенного протокола внутренней маршрутизации между шлюзами — Enhanced Interior Gateway Routing Protocol (EIGRP)
 - O — определяет сеть, динамически полученную от другого маршрутизатора с помощью открытого протокола предпочтения кратчайшего пути — Open Shortest Path First (OSPF).
2. Сеть назначения — IP-адрес сети назначения
3. Административное расстояние — определяет административное расстояние (достоверность) источника маршрута. Низкие значения означают высокую достоверность источника маршрута.
4. Метрика — указывает стоимость для достижения удаленной сети. Предпочтительные маршруты имеют низкие значения.
5. Следующий переход — определяет IP-адрес следующего маршрутизатора для пересылки пакета.
6. Временная метка маршрута — определяет последнюю активность маршрутизатора.
7. Исходящий интерфейс — определяет выходной интерфейс для его использования при передаче пакета к месту назначения.

Когда на интерфейс маршрутизатора поступает пакет, маршрутизатор анализирует его заголовок, чтобы определить сеть назначения. Если сеть назначения совпадает с маршрутом в таблице маршрутизации, маршрутизатор пересылает пакет, используя информацию в таблице маршрутизации. Если существуют два и более вероятных маршрута к одному

пункту назначения для определения маршрута, который появится в таблице маршрутизации, используется метрика.

Маршрутизатор не может пересылать пакеты, если в таблице маршрутизации отсутствует маршрут для сети назначения. Если маршрут, обозначающий сеть назначения, в таблице не указан, пакет отбрасывается (то есть не пересылается). Тем не менее, поскольку узел может использовать шлюз по умолчанию для пересылки пакета неизвестному адресату, маршрутизатор также может использовать маршрут по умолчанию, чтобы создавать шлюз «последней надежды». Маршрут по умолчанию может быть настроен вручную или получен динамически.

2.2. Методические указания и пример выполнения работы

В задании приведен набор локальных сетей, соединенных маршрутизаторами. В каждой локальной сети назначена своя подсеть IP-адресов с маской указанной длины. В случае, если подсеть для локального сегмента не указана, необходимо выбрать подсеть произвольным образом (но при этом не совпадающей с имеющимся уже набором сетей). Порты маршрутизаторов пронумерованы.

Необходимо назначить IP-адреса портам маршрутизатора. В составленной таблице маршрутизации число записей должно быть минимально, т.е. если определенная подсеть может быть достигнута через уже имеющийся маршрут (как правило – это шлюз по умолчанию), то такую запись добавлять не надо.

Дана следующая схема:

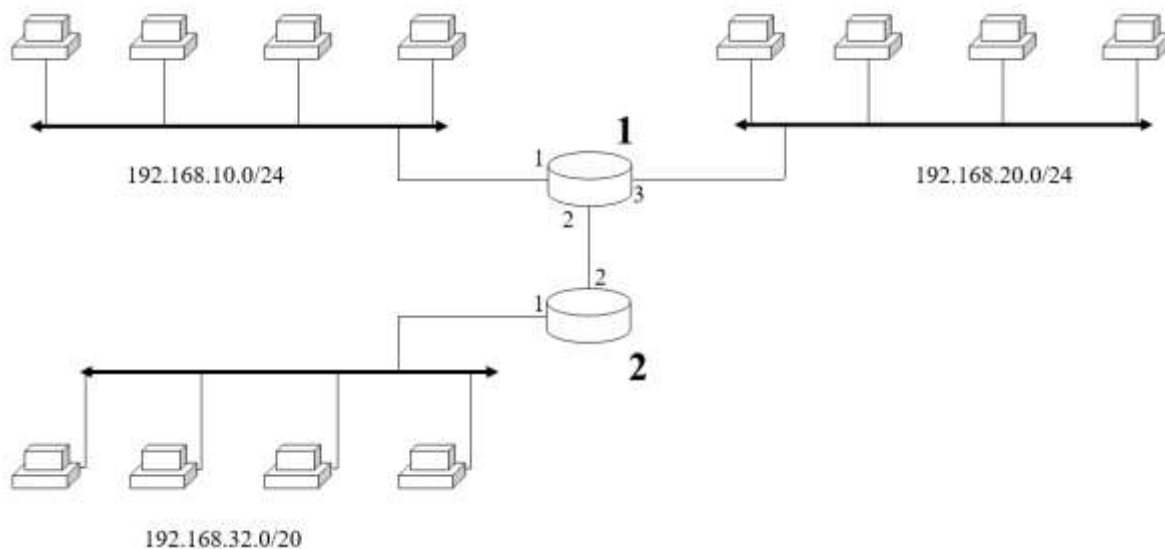


Рис.2.1

1. На приведенной схеме помимо обозначенных трех локальных сегментов (192.168.10.0/24, 192.168.20.0/24, 192.168.32.0/20) имеется еще один локальный сегмент между маршрутизаторами 1 и 2 — назовем для этого сегмента подсеть 192.168.100.0/30.

2. Назначим адреса интерфейсам маршрутизаторов:

Маршрутизатор 1: интерфейс 1 подключен к сети 192.168.10.0/24, следовательно, адрес интерфейса должен быть из этой сети — допустим, 192.168.10.1. Аналогично для интерфейса 2 назначим адрес 192.168.100.1, а для интерфейса 3 адрес 192.168.20.1

Маршрутизатор 2: интерфейс 1 — адрес 192.168.32.1, интерфейс 2 — адрес 192.168.100.2

3. Составим таблицы маршрутизации для каждого из маршрутизаторов.

Маршрутизатор 1

Подсеть	Маска	Шлюз	Интерфейс
192.168.100.0	255.255.255.252	0.0.0.0	2
192.168.10.0	255.255.255.0	0.0.0.0	1
192.168.20.0	255.255.255.0	0.0.0.0	3
192.168.32.0	255.255.240.0	192.168.100.2	2
0.0.0.0	0.0.0.0	192.168.100.2	2

Запись для маршрута на подсеть 192.168.32.0/20 может быть исключена, т.к. она может быть достигнута через имеющийся маршрут - шлюз по умолчанию, следовательно конечный вид таблицы маршрутизации будет таким:

Подсеть	Маска	Шлюз	Интерфейс
192.168.100.0	255.255.255.252	0.0.0.0	2
192.168.10.0	255.255.255.0	0.0.0.0	1
192.168.20.0	255.255.255.0	0.0.0.0	3
0.0.0.0	0.0.0.0	192.168.100.2	2

Маршрутизатор 2

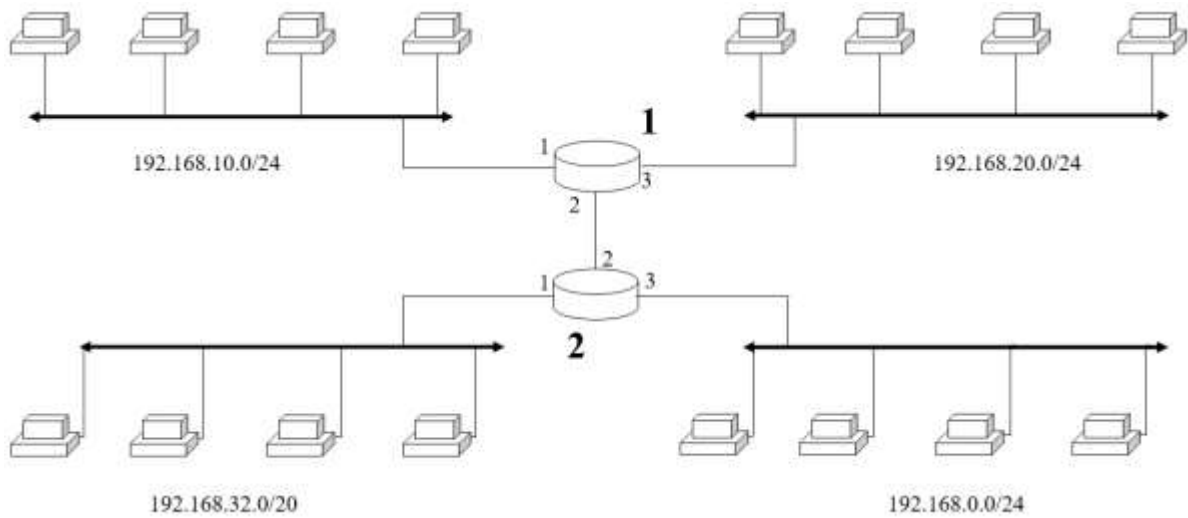
Подсеть	Маска	Шлюз	Интерфейс
192.168.100.0	255.255.255.252	0.0.0.0	2
192.168.32.0	255.255.240.0	0.0.0.0	1
192.168.10.0	255.255.255.0	192.168.100.1	2
192.168.20.0	255.255.255.0	192.168.100.1	2
0.0.0.0	0.0.0.0	192.168.100.1	2

Записи маршрута на подсети 192.168.10.0/24 и 192.168.20.0/24 могут быть исключены, т.к. они могут быть достигнуты через шлюз по умолчанию, следовательно, конечный вид таблицы маршрутизации будет таким:

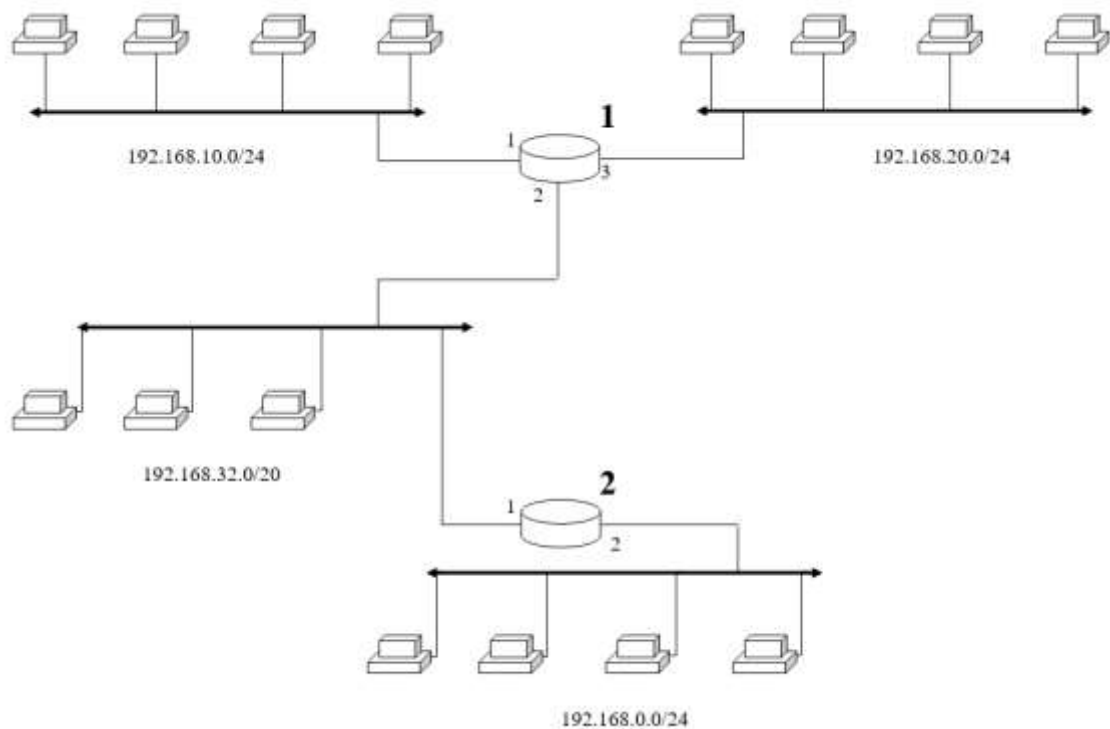
Подсеть	Маска	Шлюз	Интерфейс
192.168.100.0	255.255.255.252	0.0.0.0	2
192.168.32.0	255.255.240.0	0.0.0.0	1
0.0.0.0	0.0.0.0	192.168.100.1	2

2.3. Варианты заданий

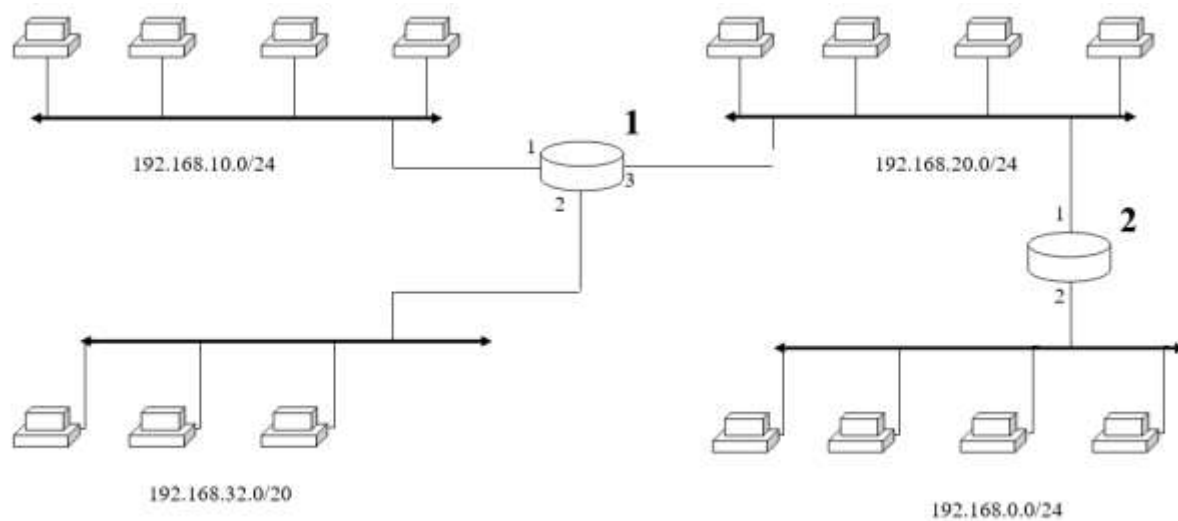
Вариант 1



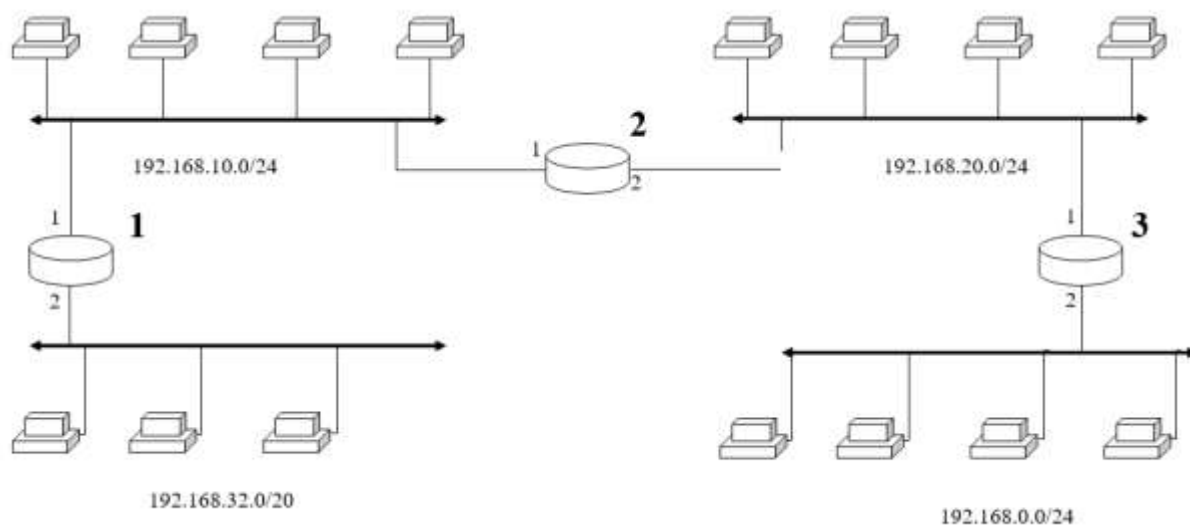
Вариант 2



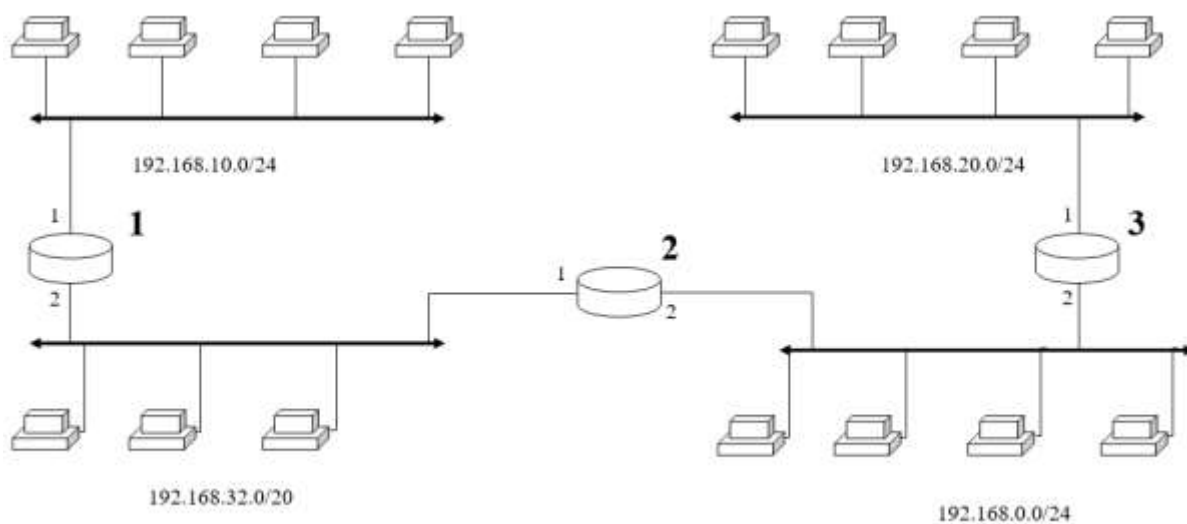
Вариант 3



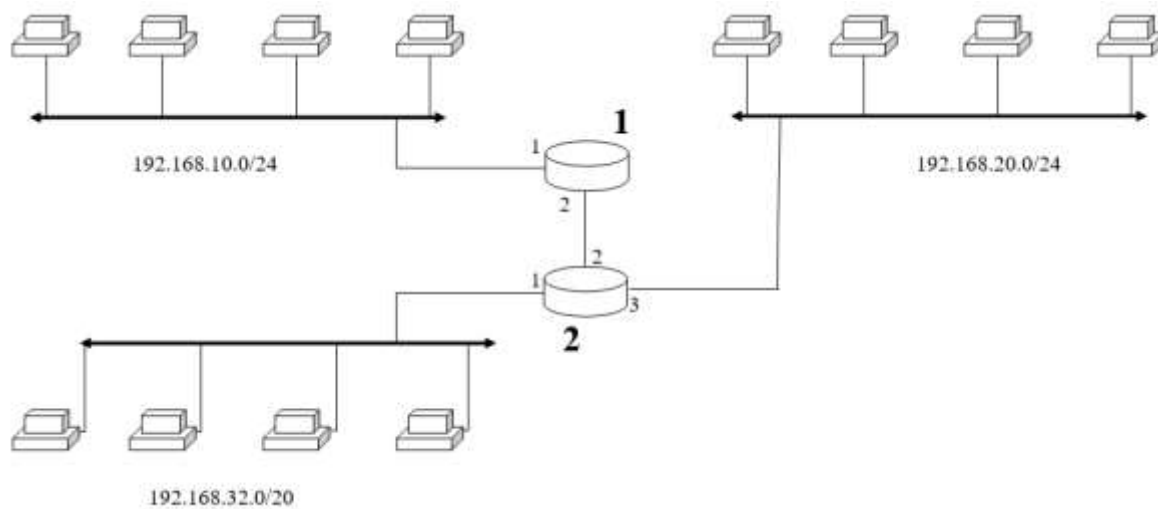
Вариант 4



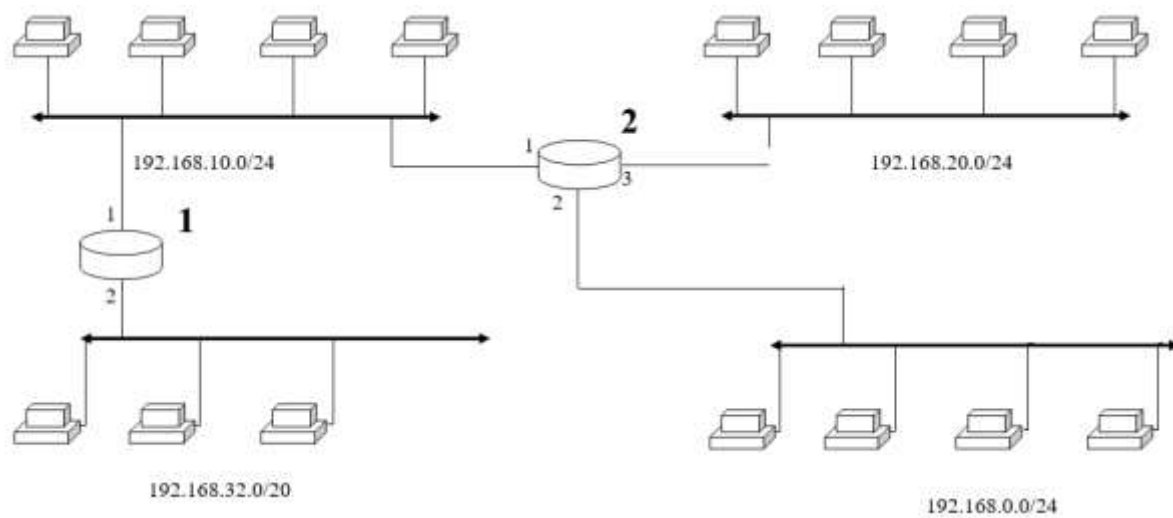
Вариант 5



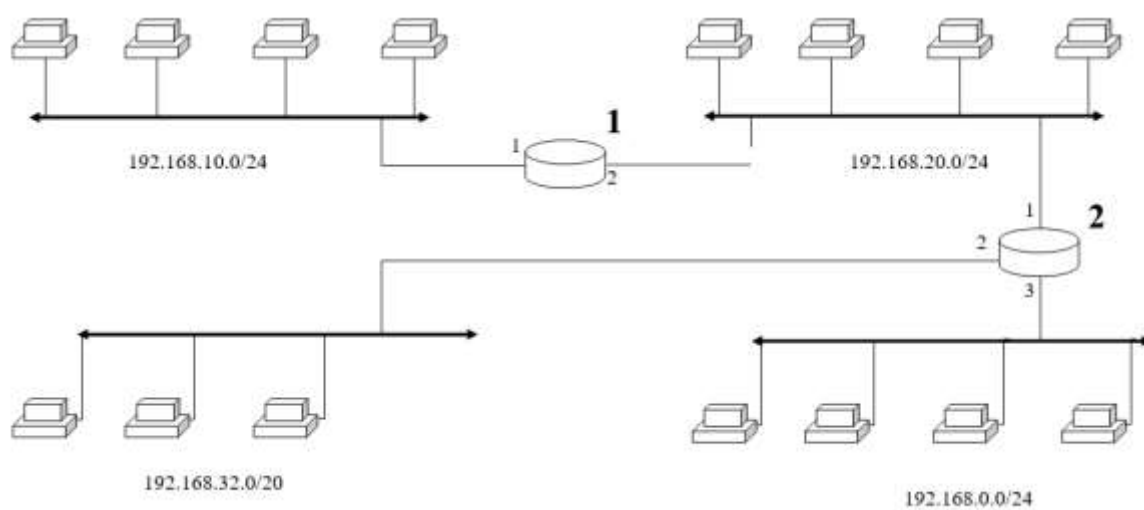
Вариант 6



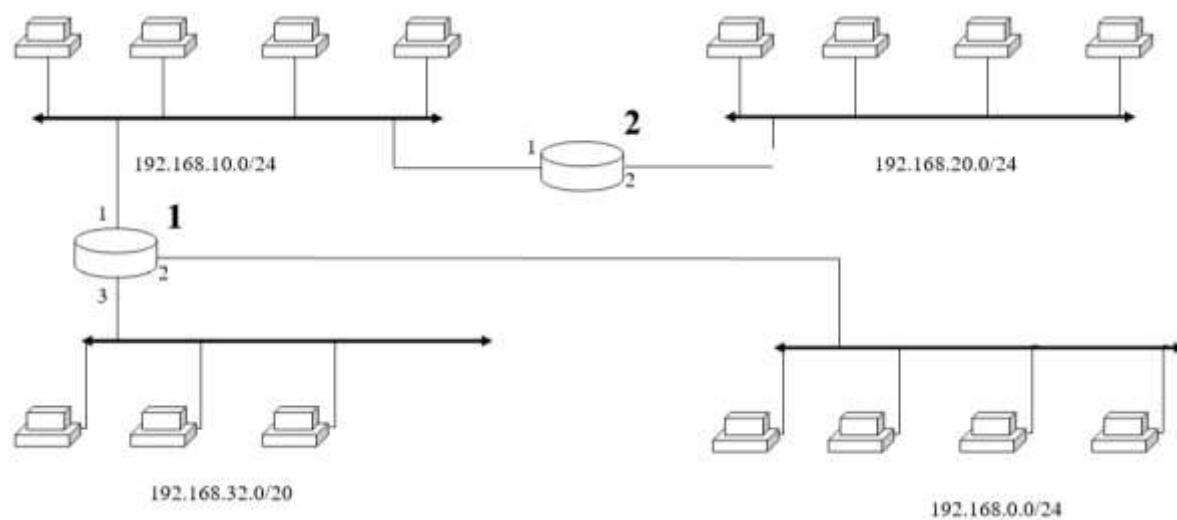
Вариант 7



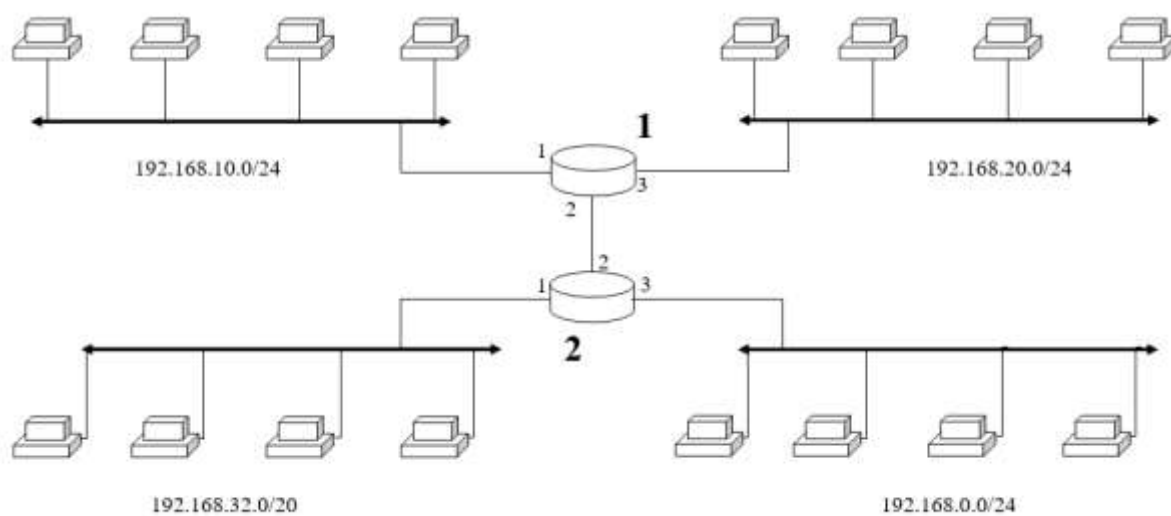
Вариант 8



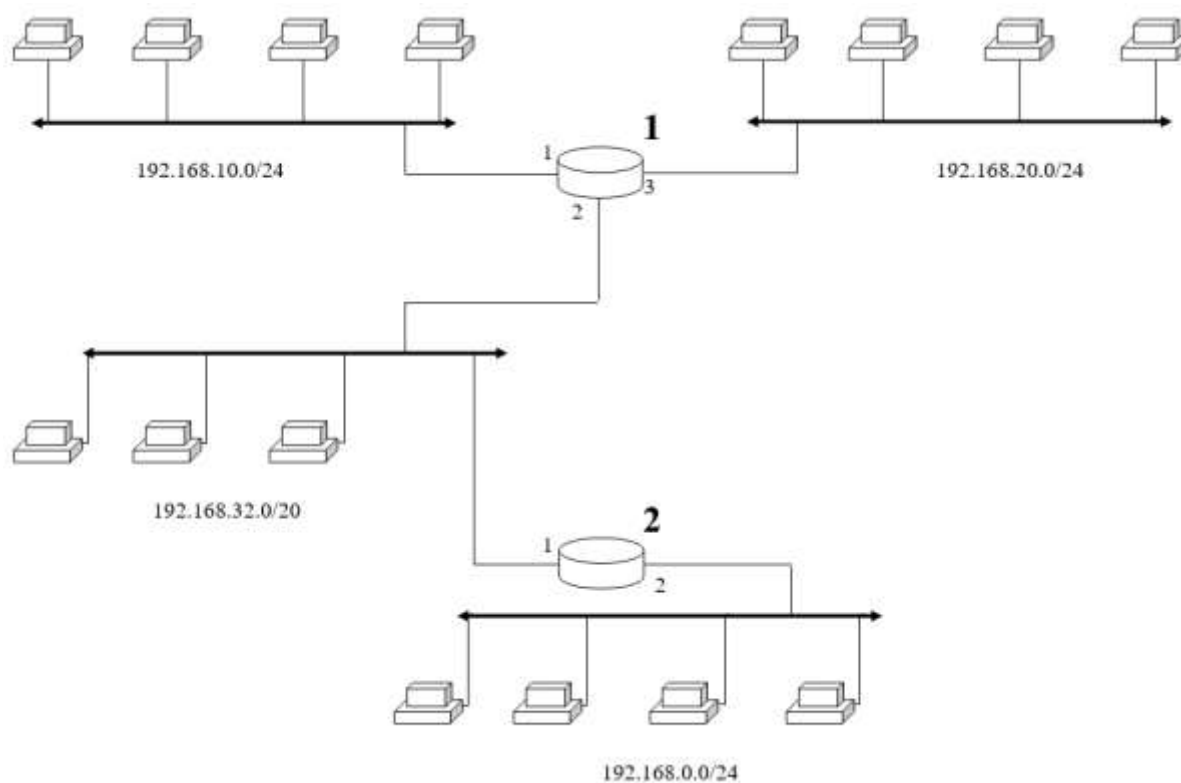
Вариант 9



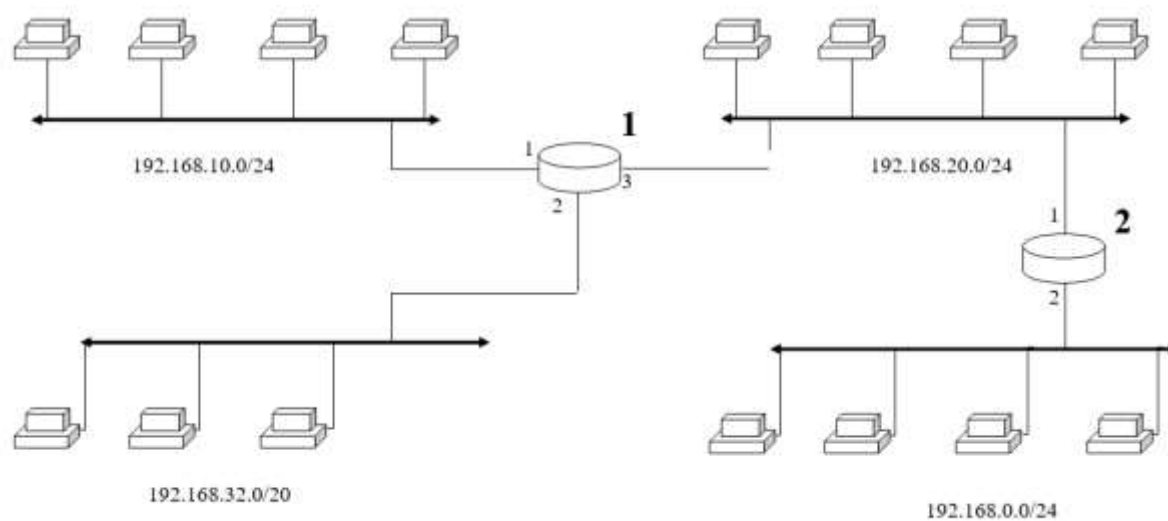
Вариант 10



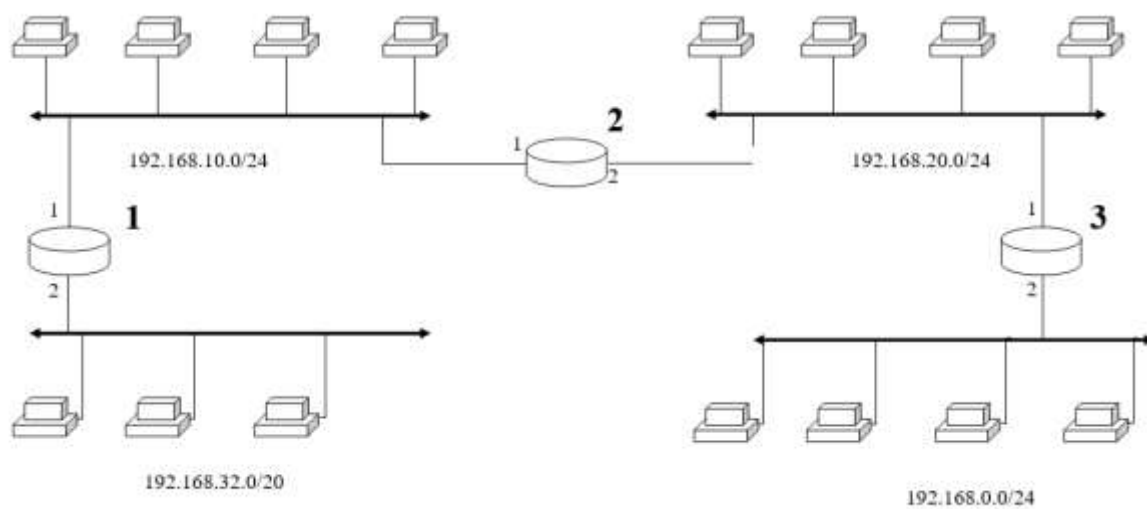
Вариант 11



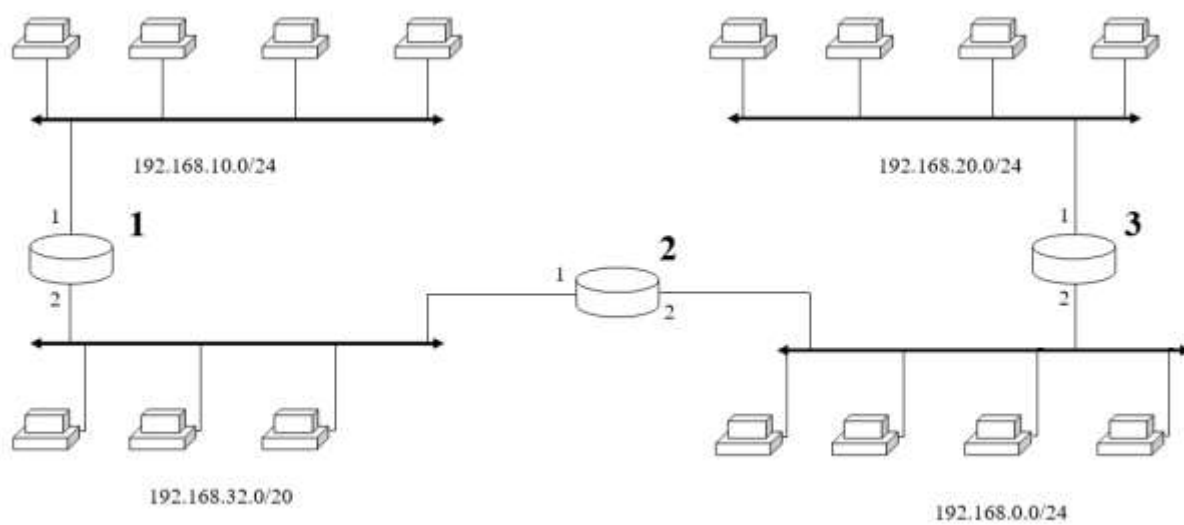
Вариант 12



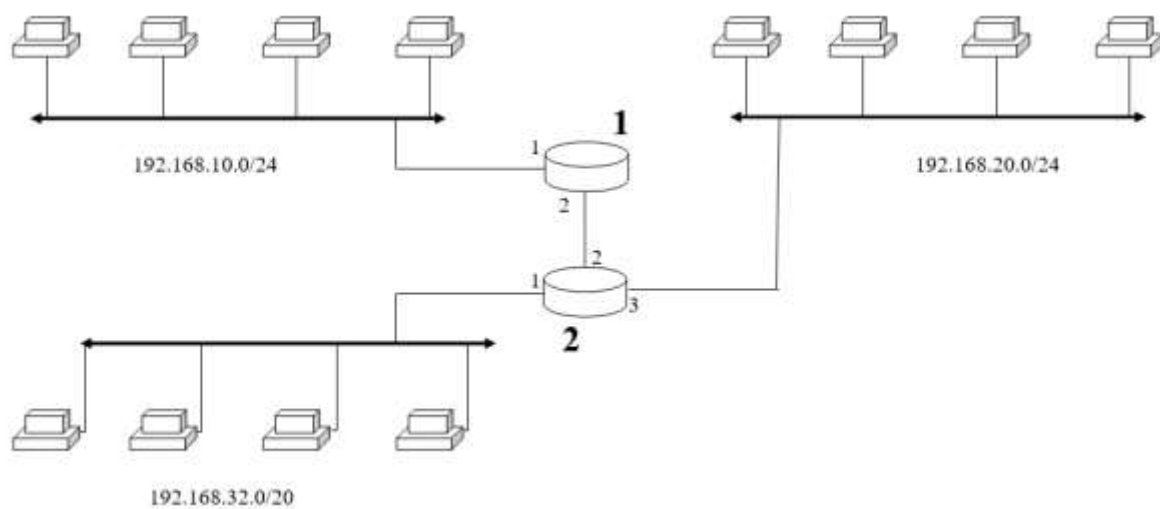
Вариант 13



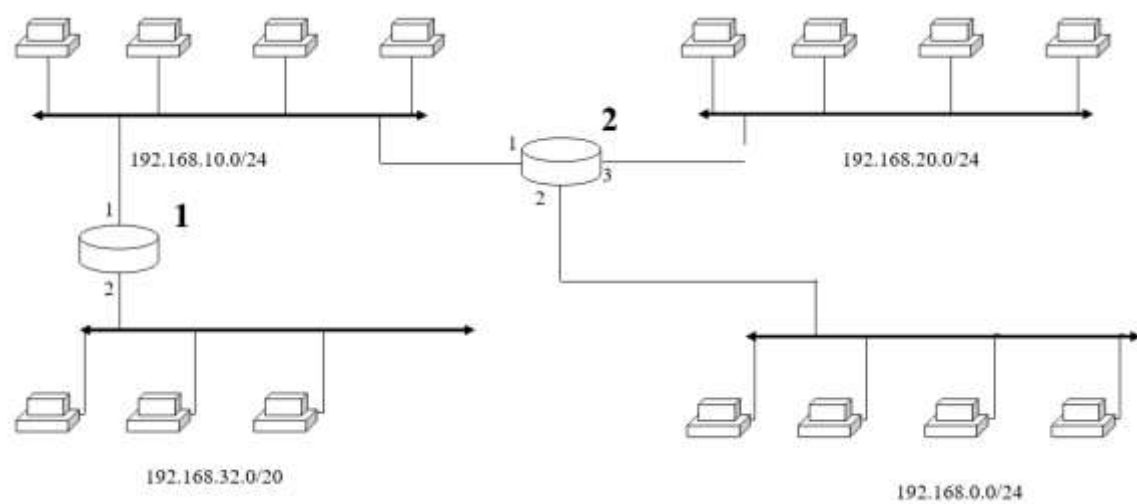
Вариант 14



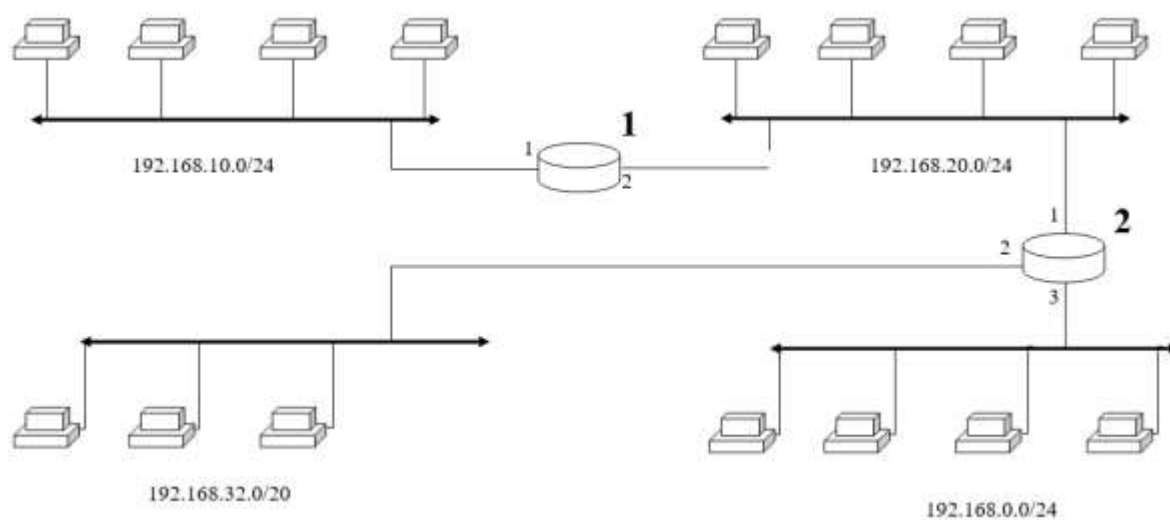
Вариант 15



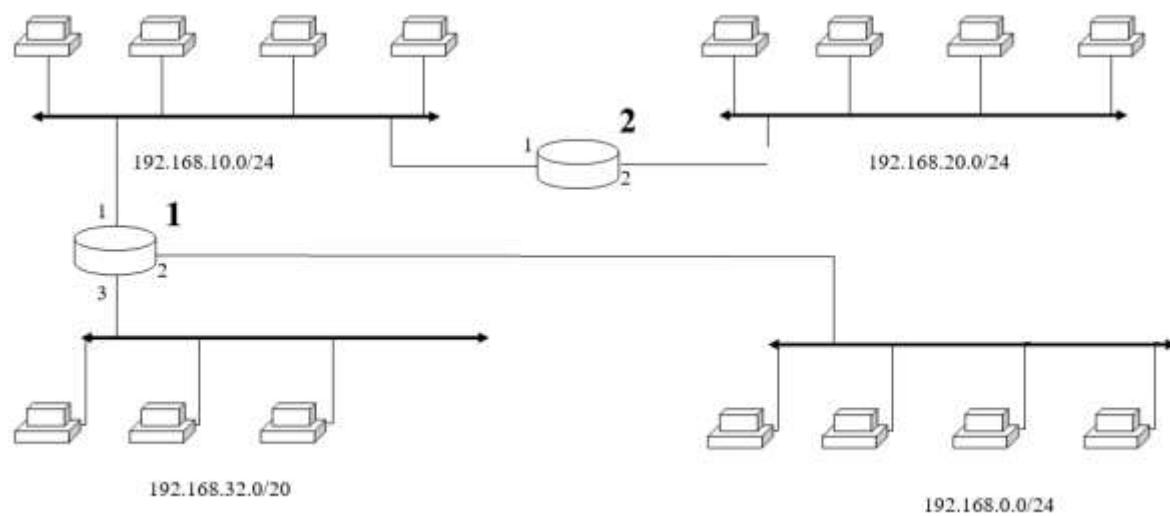
Вариант 16



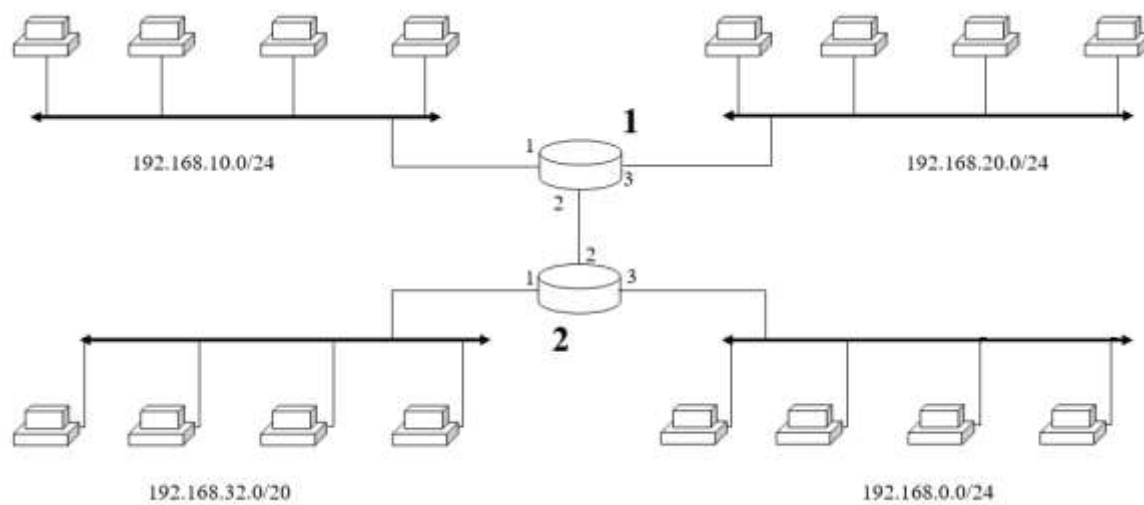
Вариант 17



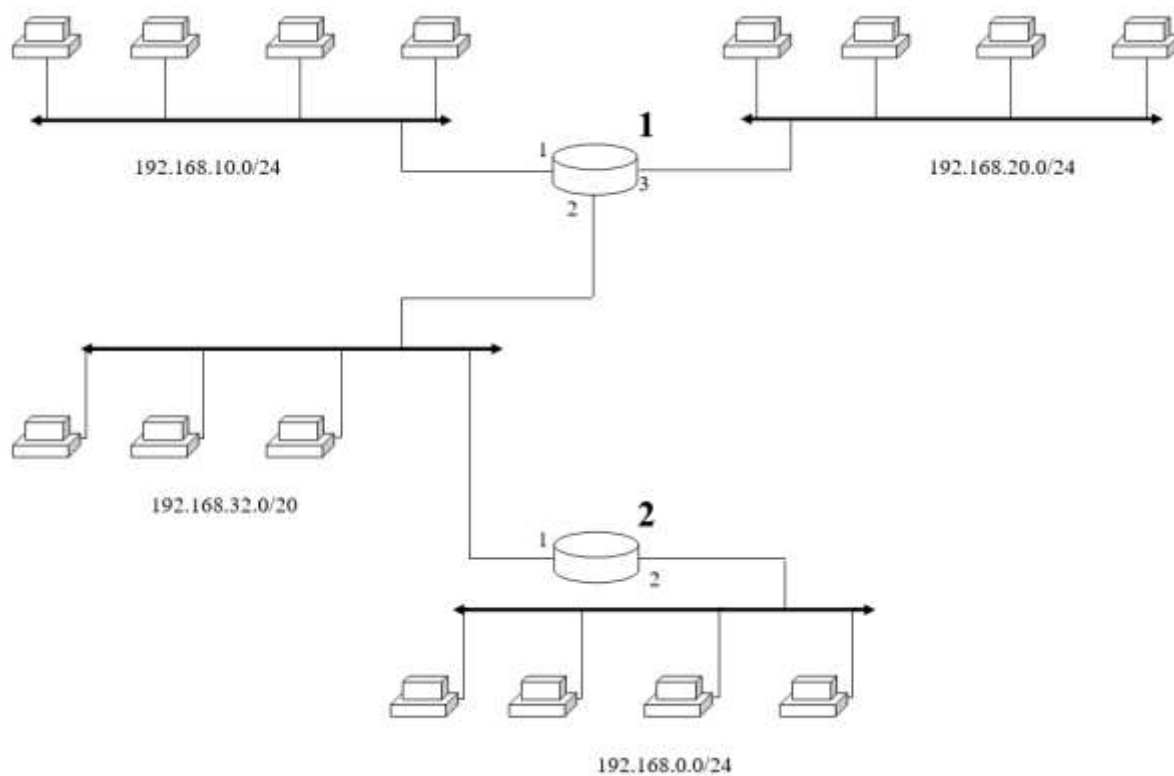
Вариант 18



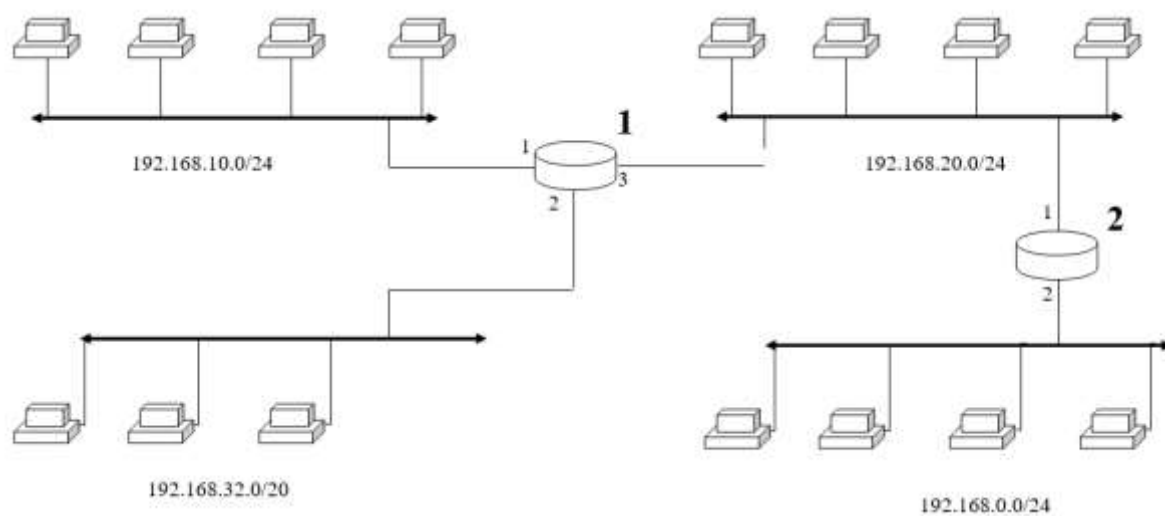
Вариант 19



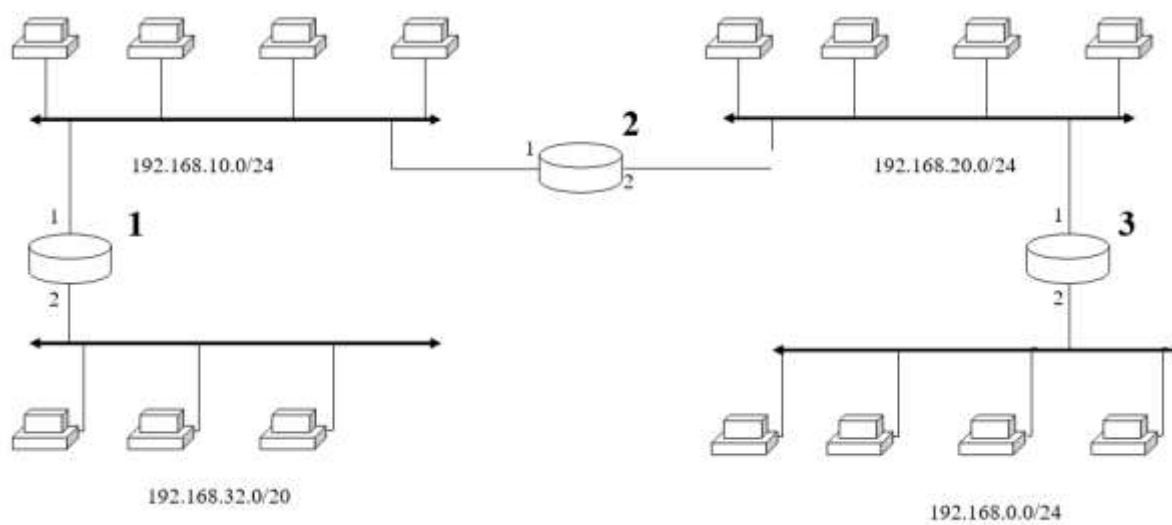
Вариант 20



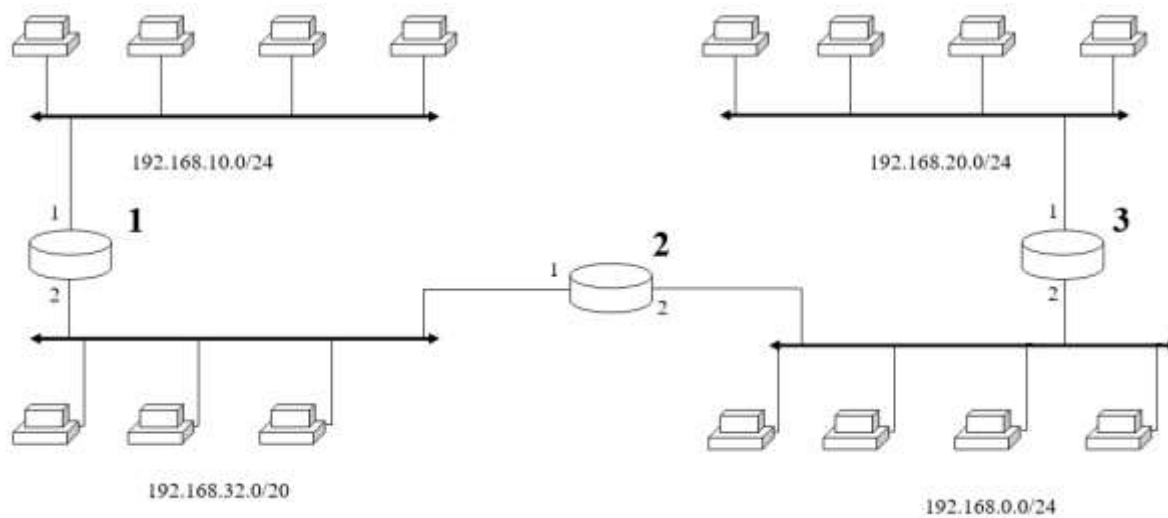
Вариант 21



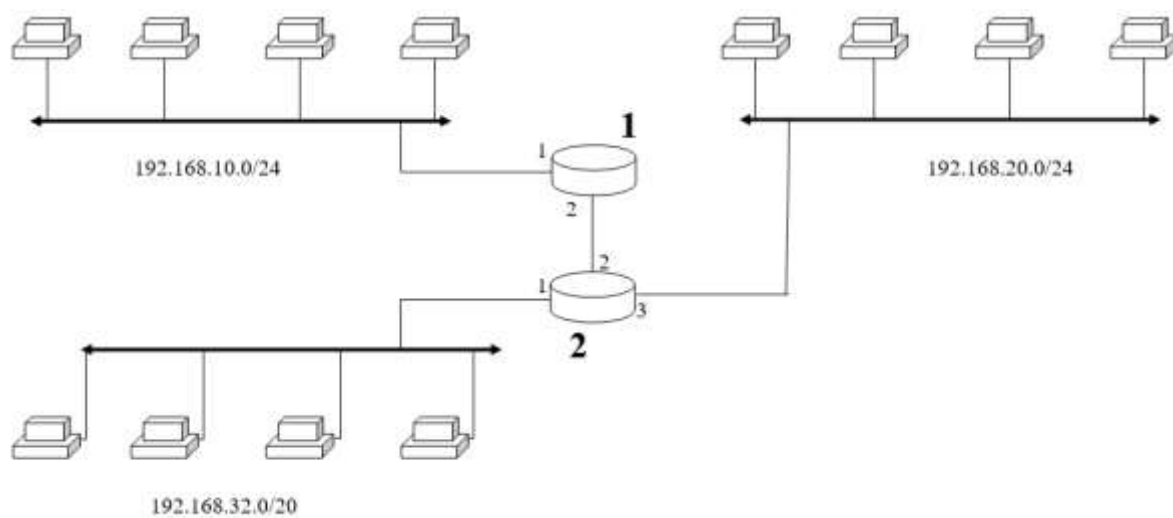
Вариант 22



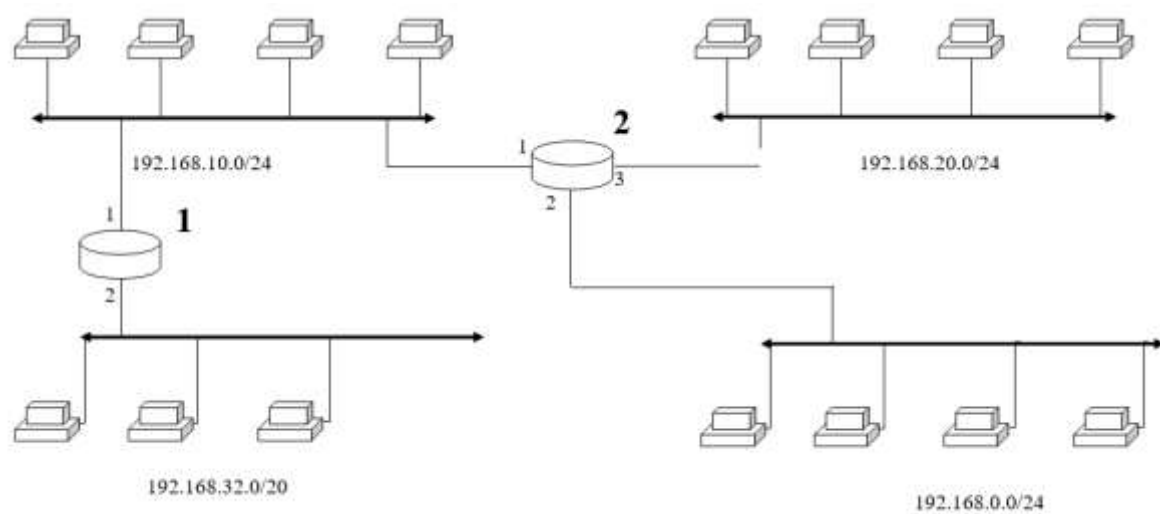
Вариант 23



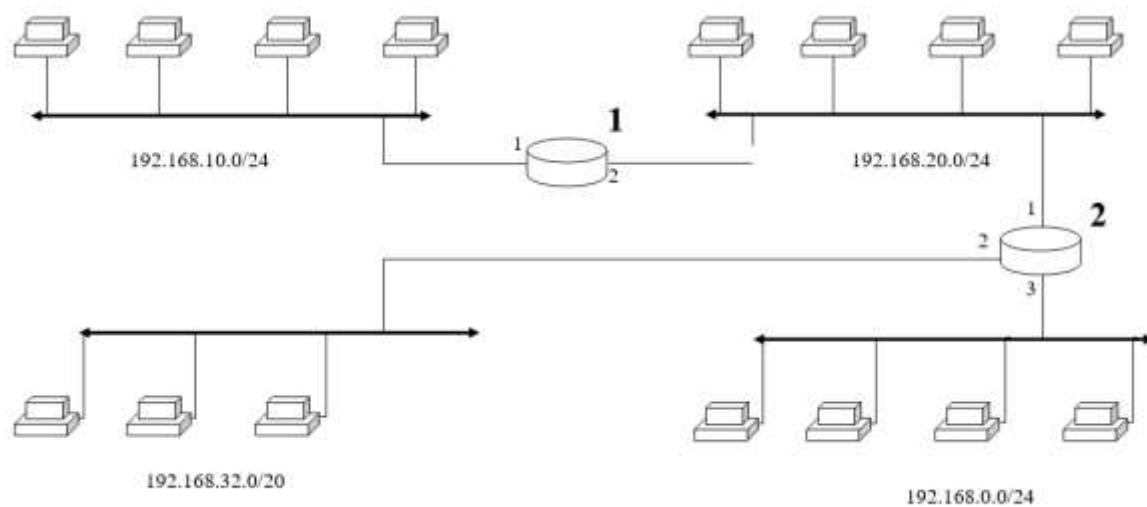
Вариант 24



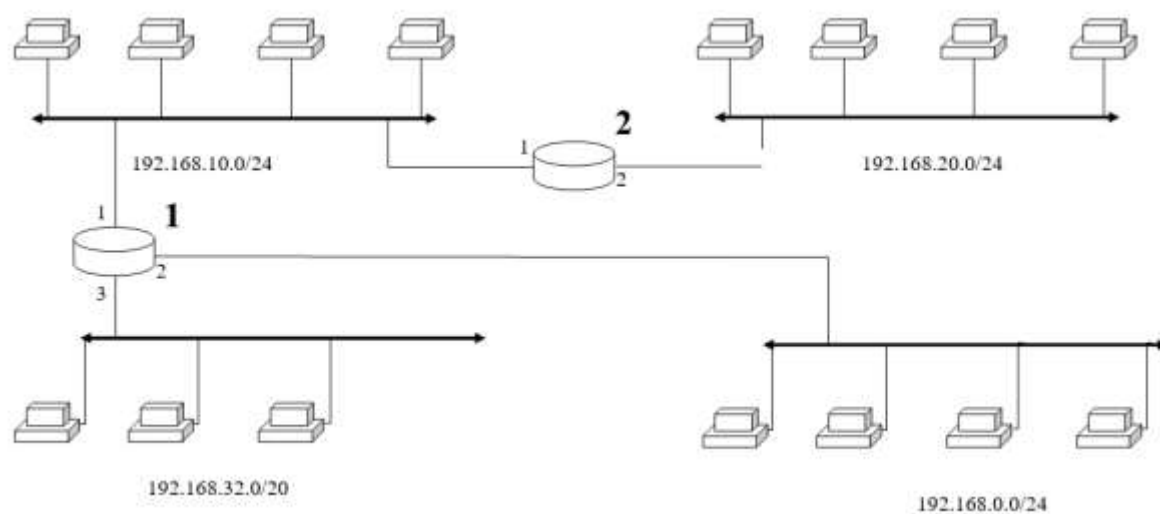
Вариант 25



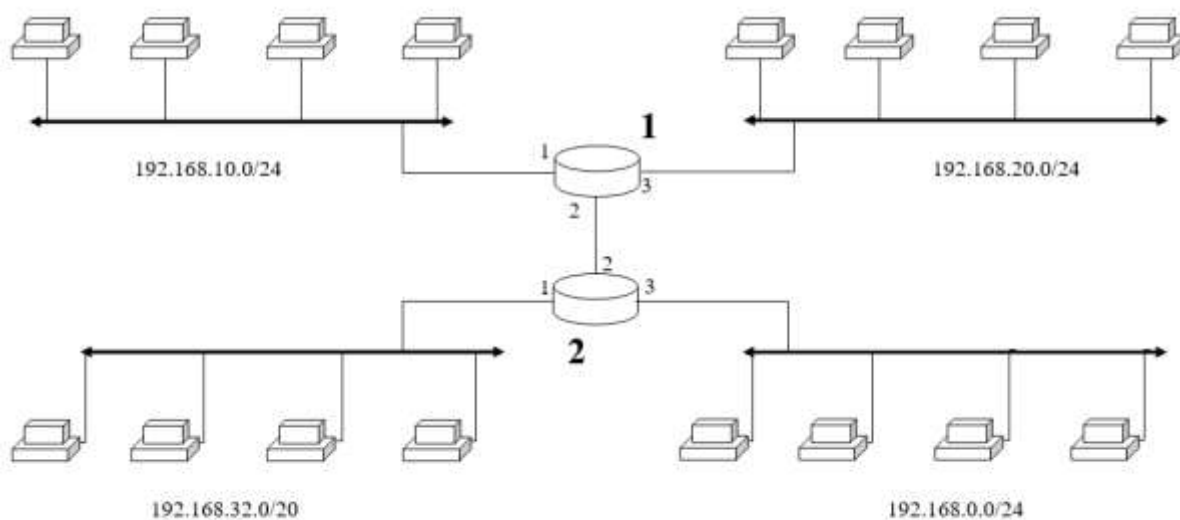
Вариант 26



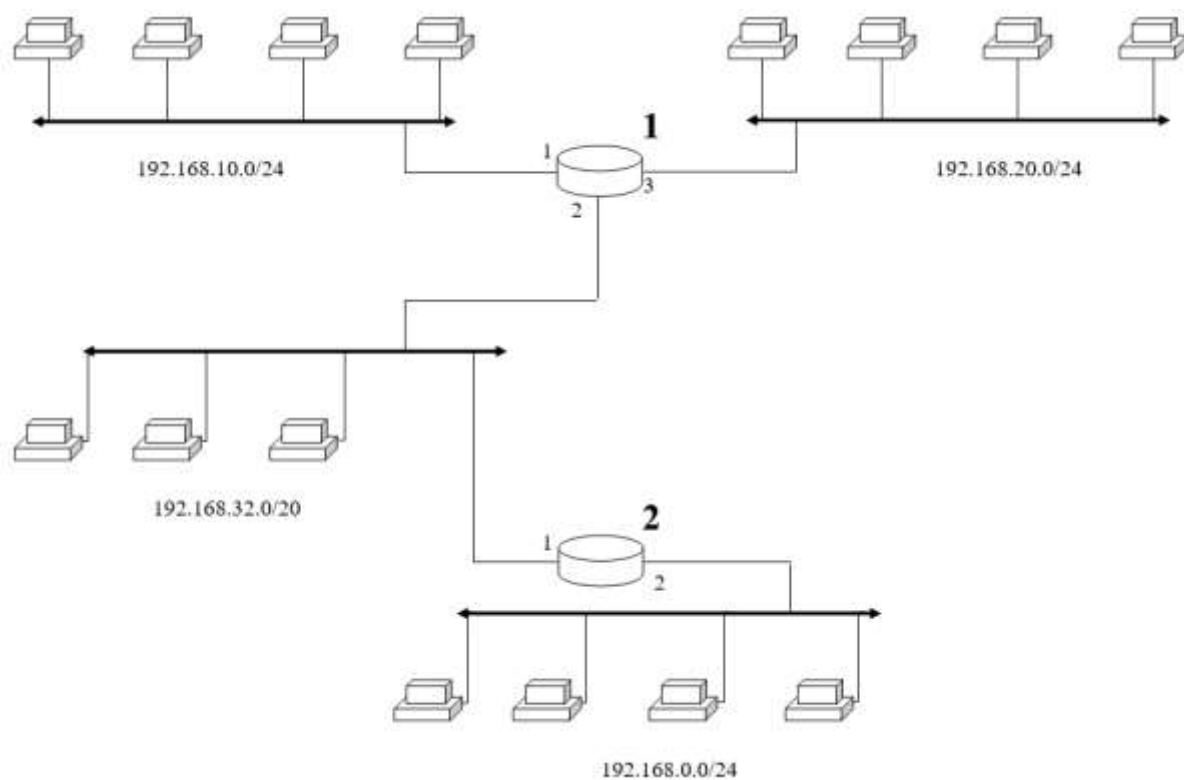
Вариант 27



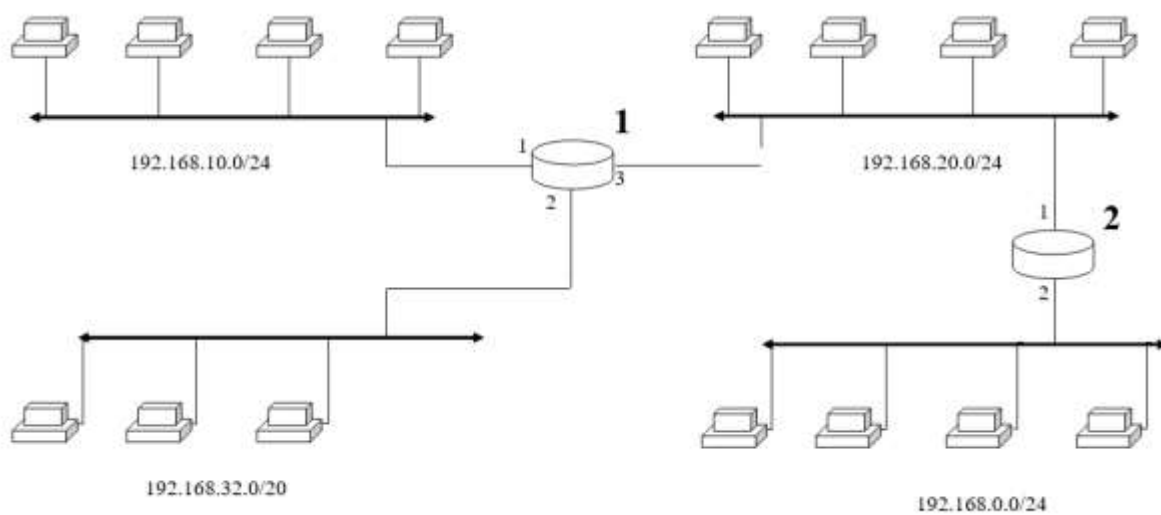
Вариант 28



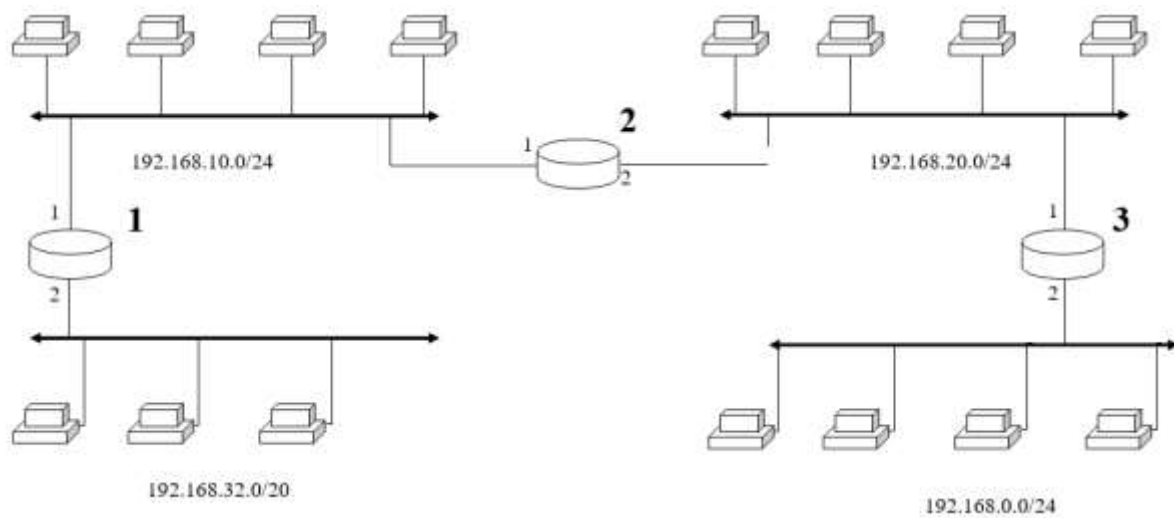
Вариант 29



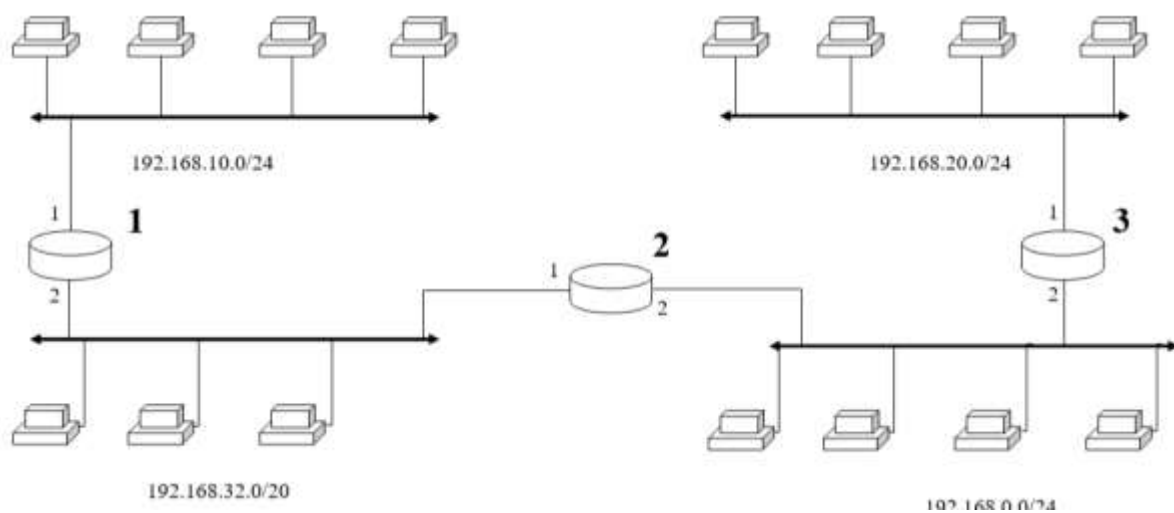
Вариант 30



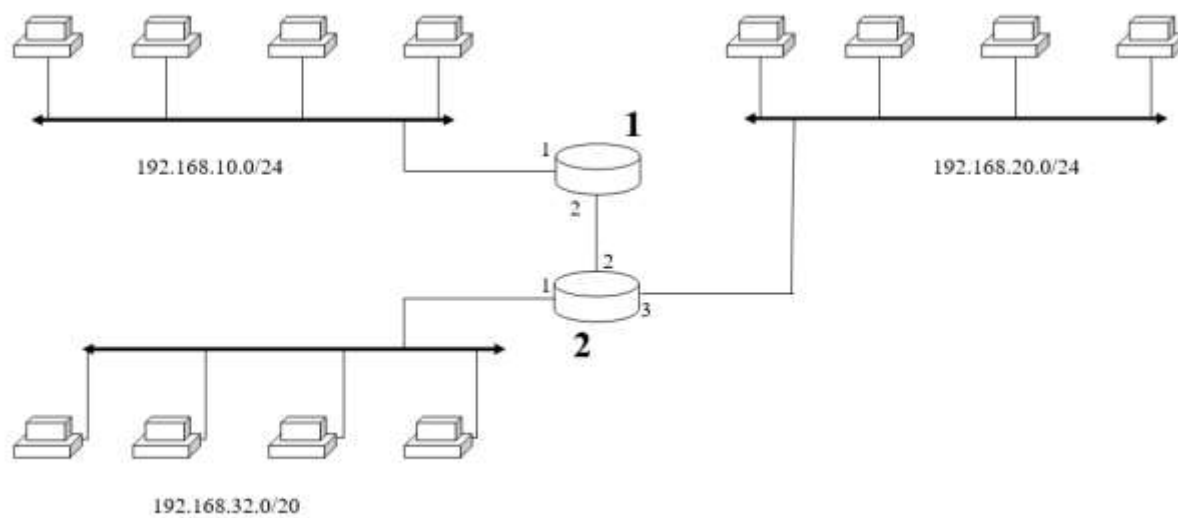
Вариант 31



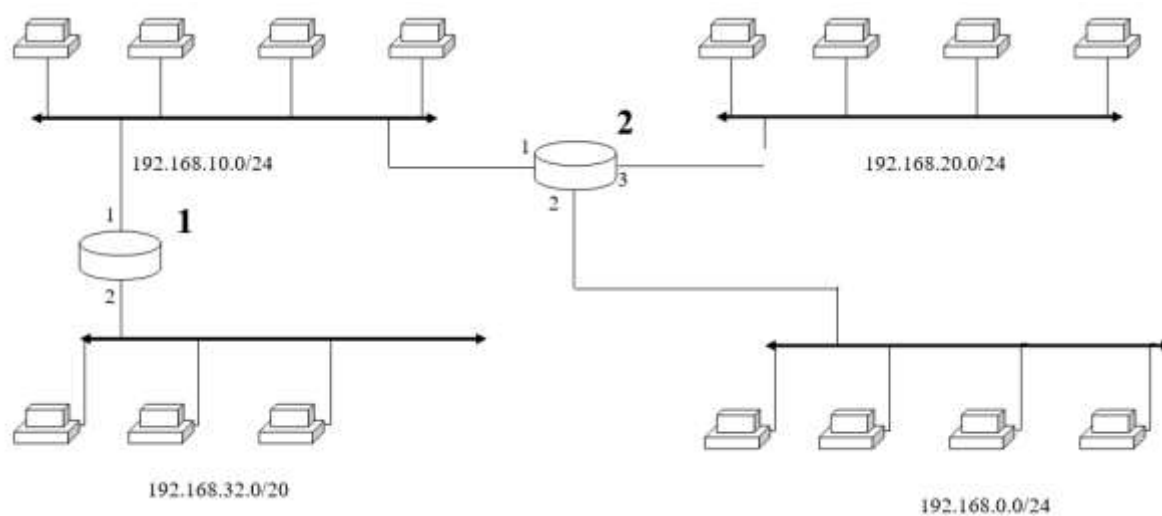
Вариант 32



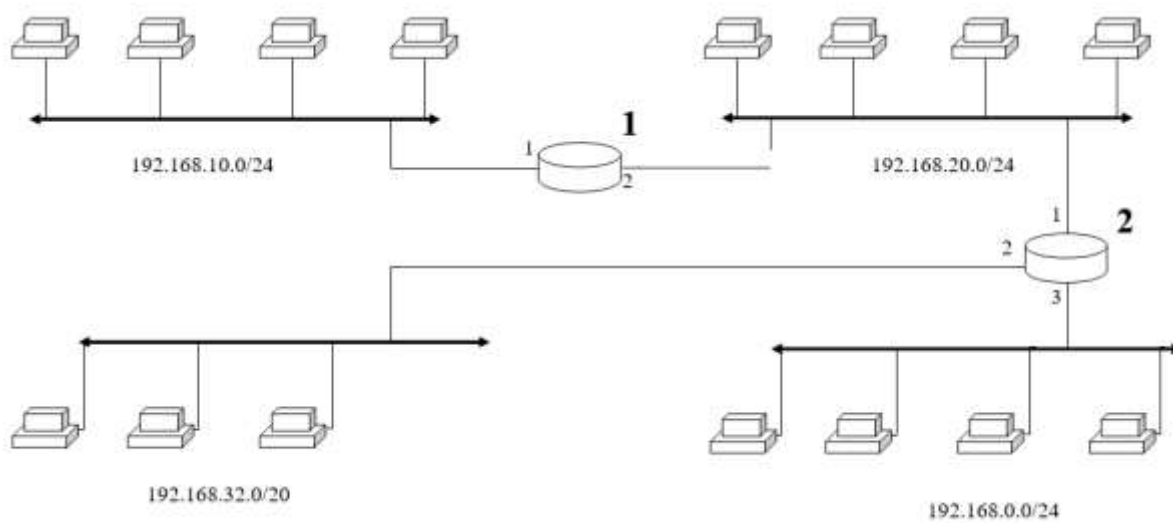
Вариант 33



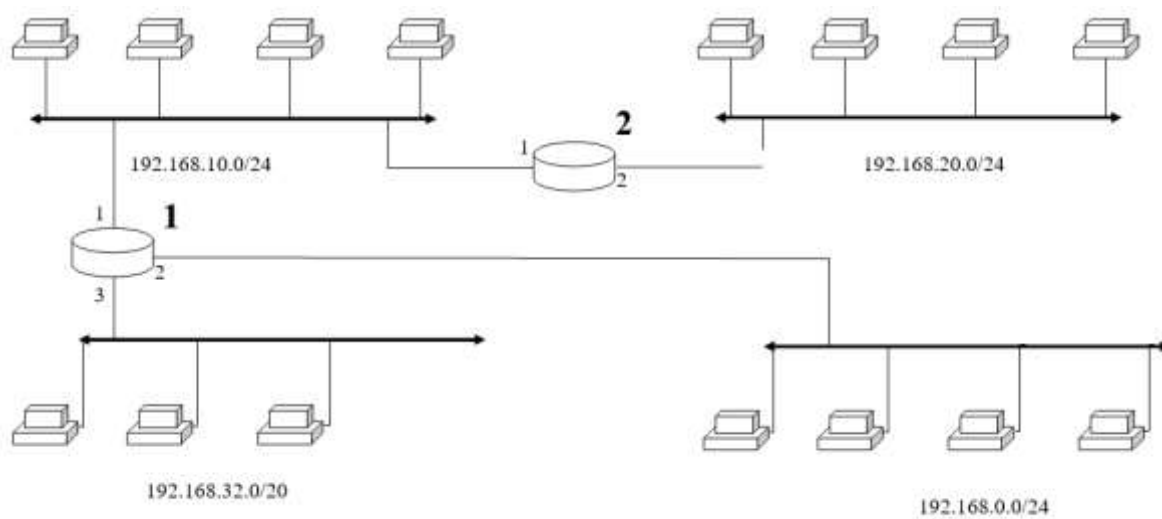
Вариант 34



Вариант 35



Вариант 36



3. Утилиты командной строки Windows для работы с сетью

3.1. Подготовительная часть

Для проведения данной лабораторной работы необходим компьютер под управлением любой операционной системы семейства Microsoft Windows 2000, XP или более старшей версии.

Все команды будут выполняться в командном интерпретаторе. Для его запуска необходимо нажать кнопку «Пуск» и выбрать раздел «Выполнить...». В строке ввода указать имя команды:

cmd

и нажать кнопку “Ok”.

Откроется окно интерпретатора.

Команды вводятся с клавиатуры, завершаются вводом “Enter”. Предыдущие команды можно вызвать для редактирования и последующего выполнения с помощью курсорной клавиши «Вверх».

3.2. Утилита ipconfig

Данная программа предназначена для получения информации о настройках протокола TCP/IP сетевых интерфейсов ОС Windows.

Для получения краткой информации о настройках необходимо выполнить команду ipconfig без параметров.

```
Настройка протокола IP для Windows

Адаптер беспроводной локальной сети Подключение по локальной сети* 2:

    Состояние среды. . . . . : Среда передачи недоступна.
    DNS-соединение подключения . . . . .

Адаптер беспроводной локальной сети Подключение по локальной сети* 14:

    Состояние среды. . . . . : Среда передачи недоступна.
    DNS-соединение подключения . . . . .

Адаптер Ethernet Ethernet 2:

    Состояние среды. . . . . : Среда передачи недоступна.
    DNS-соединение подключения . . . . .

Адаптер беспроводной локальной сети Беспроводная сеть:

    DNS-соединение подключения . . . . . : lan
    IPv6-адрес. . . . . : fd75:6f4e:74c8::855
    IPv6-адрес. . . . . : fd75:6f4e:74c8:0:ec65:f1e7:138a:d5c7
    Временный IPv6-адрес. . . . . : fd75:6f4e:74c8:0:6c13:857:51a6:38e8
    Временный IPv6-адрес. . . . . : fd75:6f4e:74c8:0:adc1:6e4e:1192:2530
    Временный IPv6-адрес. . . . . : fd75:6f4e:74c8:0:c8d5:3230:20dc:50f9
    Временный IPv6-адрес. . . . . : fd75:6f4e:74c8:0:e904:dce0:cee6:96e8
    Локальный IPv6-адрес канала . . . . . : fe80::ec65:f1e7:138a:d5c7%8
    IPv4-адрес. . . . . : 192.168.1.172
    Маска подсети . . . . . : 255.255.255.0
    Основной шлюз. . . . . : 192.168.1.1

Адаптер Ethernet Сетевое подключение Bluetooth:

    Состояние среды. . . . . : Среда передачи недоступна.
    DNS-соединение подключения . . . . .
```

Рис.3.1 Результат выполнения команды ipconfig

Задание 1.

Выполните команду **ipconfig** и запишите информацию об IP-адресе, маске сети и шлюзе по умолчанию для сетевого адаптера.

Для получения подробной информации о настройках TCP/IP необходимо выполнить команду **ipconfig** с ключом **/all**:

```
ipconfig /all
```

С помощью команды **ipconfig /all** можно узнать MAC-адрес компьютера, а также ряд сведений об адресации уровня 3 для устройства.

```
Адаптер беспроводной локальной сети Беспроводная сеть:

DNS-суффикс подключения . . . . . : lan
Описание. . . . . : Intel(R) Dual Band Wireless-AC 7265
Физический адрес. . . . . : 10-02-B5-F3-39-A0
DHCP включен. . . . . : Да
Автонастройка включена. . . . . : Да
IPv6-адрес. . . . . : fd75:6f4e:74c0::855(Основной)
Аренда получена. . . . . : 6 ноября 2018 г. 10:08:14
Срок аренды истекает. . . . . : 14 декабря 2154 г. 21:30:36
IPv6-адрес. . . . . : fd75:6f4e:74c0:0:ec65:f1e7:138a:d5c7(Основной)
Временный IPv6-адрес. . . . . : fd75:6f4e:74c0:0:6c13:857:51a6:38e8(Основной)
Временный IPv6-адрес. . . . . : fd75:6f4e:74c0:0:adc1:6e4a:1192:2530(Устаревший)
Временный IPv6-адрес. . . . . : fd75:6f4e:74c0:0:c0d5:3230:20dc:50f9(Устаревший)
Временный IPv6-адрес. . . . . : fd75:6f4e:74c0:0:e904:dce0:cce6:96e8(Устаревший)
Локальный IPv6-адрес канала . . . . : fe80::ec65:f1e7:138a:d5c7%8(Основной)
IPv4-адрес. . . . . : 192.168.1.172(Основной)
Маска подсети . . . . . : 255.255.255.0
Аренда получена. . . . . : 7 ноября 2018 г. 13:03:52
Срок аренды истекает. . . . . : 8 ноября 2018 г. 1:03:51
Основной шлюз. . . . . : 192.168.1.1
DHCP-сервер. . . . . : 192.168.1.1
IAID DHCPv6 . . . . . : 68158133
DUID клиента DHCPv6 . . . . . : 00-01-00-01-1F-3B-CB-DA-10-02-B5-F3-39-A0
DNS-серверы. . . . . : fe80::c24a:ff:fea4:7b6e%8
                        192.168.1.1
NetBios через TCP/IP. . . . . : Включен
Список поиска DNS-суффиксов подключения :
                        lan
```

Рис.3.2 Результат выполнения команды **ipconfig /all** для одного из интерфейсов

Задание 2.

Выполните команду **ipconfig /all** и запишите информацию об аппаратном адресе сетевой карты, списке DNS-серверов сетевого подключения.

3.3. Утилита `ping`

Данная программа предназначена для проверки доступности удаленного узла по сети. Для этого используется служебный протокол ICMP.

С локальной машины на удаленный узел посылается запрос с кодом “echo-request” (эхо-запрос) и в течение некоторого времени локальная машина ожидает ответа “echo-reply” (эхо-ответ). После получения каждого эхо-ответа служба эхо-тестирования предоставляет данные о времени, прошедшем между отправкой запроса и получением ответа. Это позволяет измерить производительность сети.

После отправки всех запросов утилита `ping` выдает отчет, содержащий уровень успешности запросов и среднее суммарное время доставки запросов и получения ответов.

У команды `ping` предусмотрен интервал ожидания ответа. Если в течение этого интервала ответ не получен, команда `ping` выдает сообщение об отсутствии ответа.

Причин может быть несколько:

- если получено диагностическое сообщение по протоколу ICMP, то необходимо проанализировать это сообщение (например, требуется фрагментация пакета);
- если ничего не получено, то удаленный узел не отвечает на запрос (не включен или на узле ответ блокирует брандмауэр), либо время прохождения пакетов по линии связи больше чем время ожидания ответа – в этом случае следует увеличить время ожидания.

В целом же результатом выполнения команды `ping` является одно из четырех возможных событий.

Во-первых, указанный узел может сгенерировать все четыре отклика. Это означает, что с указанным узлом возможно полноценное взаимодействие на уровне TCP/IP.

Второй вариант – для всех четырех запросов превышен интервал ожидания. Если время ожидания для всех четырех запросов превышает, это значит, что время TTL закончилось до получения ответа. Это может означать один вариант из трех возможных:

- Проблемы с соединением, которые не дают возможности передачи пакетов между двумя узлами и возникают из-за отключения кабеля, ошибок в таблице маршрутизации и тому подобных проблем.
- Передача информации есть, но она слишком медленная для получения ответа по команде `ping`. Это может происходить из-за перегрузки сети, неисправного сетевого оборудования или проводки.
- Передача информации идет, но брандмауэр блокирует ICMP трафик. В таких ситуациях опрос не работает, пока на

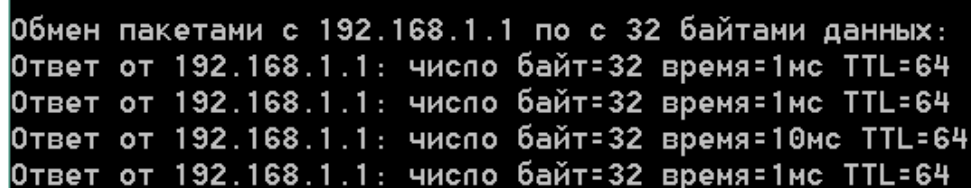
брандмауэре на целевой машине (а также на всех брандмауэрах на пути к ней) не будут разрешены ICMP эхо-сигналы

Третье, что может произойти при выполнении команды `ping`, - ситуация, когда некоторые отклики получены, а некоторые – нет. Это может указывать на неисправности в сетевых кабелях, сетевом оборудовании или на чрезмерную нагрузку сети.

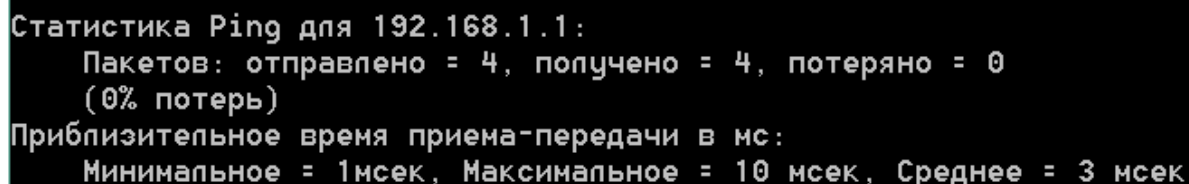
И последний четвертый вариант: ошибка `PING: Transmit Failed` (передача не удалась) указывает на то, что TCP/IP неверно настроен на той машине, на которой вы пытаетесь выполнить команду `ping`. Эта ошибка специфична для Vista или более поздних версий. В более старых версиях Windows при неверной настройке TCP/IP ошибка также происходит, но сообщение в таком случае выглядит так: «Destination Host Unreachable» (Заданный узел недоступен)

В простейшем случае в качестве аргумента команде необходимо указать имя узла (DNS-имя или NetBIOS-имя) или IP-адрес узла, например:

```
ping 10.20.30.40
```



```
Обмен пакетами с 192.168.1.1 по 32 байтами данных:  
Ответ от 192.168.1.1: число байт=32 время=1мс TTL=64  
Ответ от 192.168.1.1: число байт=32 время=1мс TTL=64  
Ответ от 192.168.1.1: число байт=32 время=10мс TTL=64  
Ответ от 192.168.1.1: число байт=32 время=1мс TTL=64
```



```
Статистика Ping для 192.168.1.1:  
Пакетов: отправлено = 4, получено = 4, потеряно = 0  
(0% потерь)  
Приблизительное время приема-передачи в мс:  
Минимальное = 1мсек, Максимальное = 10 мсек, Среднее = 3 мсек
```

Рис.3.3 Результат выполнения команды `ping` с параметрами по умолчанию

Задание 3.

Проверьте доступность по сети шлюза по умолчанию и любого соседнего компьютера.

Среди дополнительных опций команда `ping` принимает флаг `-f`, который запрещает фрагментацию IP-пакетов. Так как сетевой уровень абстрагируется от используемой технологии канального уровня, то необходим механизм, с помощью которого можно передавать блоки данных произвольной длины через различные транспортные сети с их собственными технологиями канального уровня, которые имеют разные ограничения на размер кадра (MTU). В случае если пакет данных плюс служебные заголовки

превышает размер кадра, то пакет разбивается на фрагменты, которые уже могут быть переданы в кадрах канального уровня. На конечном узле фрагменты собираются в единый пакет данных.

Вторая опция команды – это флаг **-l**, после которого через пробел указывается цифра – размер буфера, который будет посылаться на удаленный узел в пакете ICMP. Используя совместно ключи **-l** и **-f** можно выяснить максимальный размер блока данных, помещенного в IP-пакет, который еще иначе называется MSS (максимальный размер сегмента). MSS будет равен длине блока данных + длина ICMP заголовка, который равен 8 байт в случае команды ping. Размер стандартного заголовка IP-пакета равен 20 байт. Таким образом MTU = “размер буфера команды ping” + 8 + 20.

Задание 4.

Экспериментально выясните максимальный размер кадра канального уровня (MTU) в сети. Для этого необходимо посылать пакеты различной длины при установленном флаге запрета фрагментации. В качестве удаленного узла можно выбрать адрес шлюза по умолчанию или адрес любого соседнего компьютера. Начните с начального значения размера буфера 1500.

Еще один флаг команды – это флаг **-a**, который позволяет получить имя DNS по адресу компьютера (как правило, IP-адресу).

```
C:\Users\opesk>ping -a 192.168.1.1

Обмен пакетами с OpenWrt.lan [192.168.1.1] с 32 байтами данных:
Ответ от 192.168.1.1: число байт=32 время=1мс TTL=64
Ответ от 192.168.1.1: число байт=32 время<1мс TTL=64
Ответ от 192.168.1.1: число байт=32 время=1мс TTL=64
Ответ от 192.168.1.1: число байт=32 время=1мс TTL=64

Статистика Ping для 192.168.1.1:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 1 мсек, Среднее = 0 мсек
```

Рис.3.4 Результат выполнения команды ping с параметром -a

Задание 5.

Определите DNS-имя любого соседнего компьютера по его IP-адресу.

Следующий флаг команды – это флаг **-i**, который позволяет задать TTL – «время жизни» посылаемых пакетов. Для решения проблем наличия петель в маршрутизации и, соответственно, бесконечной циркуляции IP-пакетов в сети, каждый посылаемый пакет имеет поле TTL, которое содержит некоторое целое число. Это число уменьшается на единицу при каждом прохождении пакета маршрутизаторов. В тот момент, когда значение TTL станет равным 0, такой пакет отбрасывается, а отправителю по протоколу ICMP отправляется уведомление о том, что пакет отброшен из-за нулевого TTL. По рекомендации RFC 1700 начальное значение TTL должно быть 64. Таким образом, это обеспечивает прохождение пакетом до 63 промежуточных маршрутизаторов. Различные операционные системы могут выбирать начальное значение TTL по-разному.

Приходящее уведомление о нулевом значении TTL, помимо всего прочего, содержит и IP-адрес маршрутизатора, на котором произошло обнуление TTL. Таким образом, посылая пакеты с начальным TTL = 1 и увеличивая TTL на единицу, можно получить список всех маршрутизаторов на пути следования пакета от отправителя к получателю.

```
C:\Users\opesk>ping -i 1 www.ya.ru

Обмен пакетами с ya.ru [87.250.250.242] с 32 байтами данных:
Ответ от 192.168.1.1: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 192.168.1.1: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 192.168.1.1: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 192.168.1.1: Превышен срок жизни (TTL) при передаче пакета.

Статистика Ping для 87.250.250.242:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)

C:\Users\opesk>ping -i 2 www.ya.ru

Обмен пакетами с ya.ru [87.250.250.242] с 32 байтами данных:
Ответ от 10.10.133.1: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 10.10.133.1: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 10.10.133.1: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 10.10.133.1: Превышен срок жизни (TTL) при передаче пакета.

Статистика Ping для 87.250.250.242:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)

C:\Users\opesk>ping -i 3 www.ya.ru

Обмен пакетами с ya.ru [87.250.250.242] с 32 байтами данных:
Ответ от 10.131.92.1: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 10.131.92.1: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 10.131.92.1: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 10.131.92.1: Превышен срок жизни (TTL) при передаче пакета.

Статистика Ping для 87.250.250.242:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
```

Рис.3.5 Результат выполнения команды `ping` с целью получения списка маршрутизаторов по пути следования

Задание 6.

Определите список маршрутизаторов на пути следования пакетов от локального компьютера до адреса 192.168.100.1 и до адреса www.yandex.ru.

3.4. Утилита `tracert`

Команда `tracert` (`tracert`) — это утилита, позволяющая составить список переходов, по которым успешно проходит эхо-запрос на пути к узлу назначения.

Если запрос доходит до узла назначения, утилита `tracert` заносит в список интерфейс каждого маршрутизатора на пути между узлами. Если на каком-то переходе на маршруте происходит сбой передачи данных, то адрес последнего маршрутизатора, ответившего на трассировку, может указать на место нахождения проблемы или ограничения системы безопасности.

Утилита `tracert` определяет суммарное время прохождения сигнала в прямом и обратном направлениях (RTT) для каждого перехода на маршруте и сообщает о возможном отсутствии ответа на одном из переходов. RTT — это время, которое требуется на доставку пакета на удаленный узел и получения ответа от этого узла. Символ звездочки (*) используется для обозначения потерянного пакета или отсутствия ответа на пакет.

Данная утилита автоматизирует рассмотренный ранее процесс получения промежуточных маршрутизаторов с помощью утилиты `ping`. В простейшем случае в качестве аргумента команде необходимо указать имя узла (DNS-имя или NetBIOS-имя) или IP-адрес узла, например:

```
tracert 10.20.30.40
```

```
C:\Users\opesk>tracert ictis.sfedu.ru

Трассировка маршрута к ictis.sfedu.ru [195.208.245.251]
с максимальным числом прыжков 30:

 1      1 ms      1 ms      2 ms  OpenWrt.lan [192.168.1.1]
 2      1 ms      1 ms     <1 мс  10.10.133.1
 3      1 ms      1 ms      1 ms  10.131.92.1
 4      6 ms      3 ms      4 ms  uginfo-c1.r61.net [195.208.245.225]
 5      7 ms      6 ms      7 ms  hosting.r61.net [195.208.245.251]

Трассировка завершена.
```

Рис.3.6 Результат выполнения команды `tracert`

Задание 7.

Определите список маршрутизаторов на пути следования пакетов от локального компьютера до адреса www.sfedu.ru и до адреса www.yandex.ru.

По умолчанию `tracert` выполняет преобразование полученных IP-адресов маршрутизаторов в символьные имена DNS. Это замедляет работу `tracert`, поэтому, если преобразование не требуется, то можно указать ключ `-d`

```
C:\Users\opesk>tracert -d ictis.sfedu.ru

Трассировка маршрута к ictis.sfedu.ru [195.208.245.251]
с максимальным числом прыжков 30:

  1      1 ms      5 ms      8 ms  192.168.1.1
  2      1 ms     <1 ms    <1 ms  10.10.133.1
  3      1 ms      1 ms      1 ms  10.131.92.1
  4      4 ms      3 ms      3 ms  195.208.245.225
  5     14 ms      4 ms      4 ms  195.208.245.251

Трассировка завершена.
```

Рис.3.7 Результат выполнения команды `tracert` с параметром `-d`

Задание 8.

Определите список маршрутизаторов на пути следования пакетов от локального компьютера до адреса 192.168.47.1 и до адреса www.sfedu.ru без преобразования IP-адресов в имена DNS.

3.5. Утилита route

Утилита route позволяет получить/изменить таблицу маршрутизации локального компьютера. Для того чтобы получить таблицу маршрутизации, необходимо выполнить команду route с параметром print:

```
route print
```

Будут отображены следующие три раздела, относящиеся к текущим сетевым подключениям TCP/IP:

- Список интерфейса. Содержит адрес управления доступом к среде (MAC-адрес) и присвоенный номер интерфейса с поддержкой сети на узле, включая адаптеры Ethernet, Wi-Fi и Bluetooth.
- Таблица маршрутизации IPv4. Содержит все известные маршруты IPv4, включая прямые подключения, локальные сети и локальные маршруты, используемые по умолчанию.
- Таблица маршрутизации IPv6. Содержит все известные маршруты IPv6, включая прямые подключения, локальные сети и локальные маршруты, используемые по умолчанию.

```
=====
Список интерфейсов
17...10 02 b5 f3 39 a1 .....Microsoft Wi-Fi Direct Virtual Adapter
14...12 02 b5 f3 39 a0 .....Microsoft Wi-Fi Direct Virtual Adapter #3
 5...00 ff 13 44 e8 3f .....TAP-Windows Adapter V9
 8...10 02 b5 f3 39 a0 .....Intel(R) Dual Band Wireless-AC 7265
 2...10 02 b5 f3 39 a4 .....Bluetooth Device (Personal Area Network)
 1.....Software Loopback Interface 1
=====

IPv4 таблица маршрута
=====
Активные маршруты:
Сетевой адрес      Маска сети      Адрес шлюза      Интерфейс      Метрика
0.0.0.0            0.0.0.0        192.168.1.1      192.168.1.172  50
127.0.0.0          255.0.0.0      On-link          127.0.0.1      331
127.0.0.1          255.255.255.255 On-link          127.0.0.1      331
127.255.255.255    255.255.255.255 On-link          127.0.0.1      331
192.168.1.0        255.255.255.0  On-link          192.168.1.172  306
192.168.1.172      255.255.255.255 On-link          192.168.1.172  306
192.168.1.255      255.255.255.255 On-link          192.168.1.172  306
224.0.0.0          240.0.0.0      On-link          127.0.0.1      331
224.0.0.0          240.0.0.0      On-link          192.168.1.172  306
255.255.255.255    255.255.255.255 On-link          127.0.0.1      331
255.255.255.255    255.255.255.255 On-link          192.168.1.172  306
=====
Постоянные маршруты:
Отсутствует
```

Рис.3.8. Часть вывода таблицы маршрутизации

Задание 9.

Получите таблицу маршрутизации локального компьютера.

Для внесения изменений в таблицу маршрутизации используются параметры **add** и **delete**.

3.6. Утилита **arp**

Данная утилита позволяет получить таблицу соответствия IP-адресов и MAC-адресов. В связи с тем, что сетевой уровень вводит свою систему адресов, уникальных в пределах всей составной сети, то необходим механизм, с помощью которого можно преобразовывать IP-адреса в аппаратные адреса канального уровня, используемой транспортной сети.

В случае, если IP-адрес назначения находится в подсети, подключенной напрямую к одному из сетевых интерфейсов компьютера (т.е. не используя шлюз), то отправитель может отправить пакет данных «напрямую. Для этого отправитель посылает в соответствующий сетевой интерфейс (согласно таблице маршрутизации) широковещательный запрос по протоколу ARP, содержащий следующие данные:

- MAC-адрес источника
- IP-адрес источника
- искомый IP-адрес

Тот компьютер, который владеет искомым IP-адресом, отвечает на запрос. При этом результат опроса, т.е. MAC-адрес конечного компьютера, сохраняется в таблице ARP отправителя в течение некоторого времени, после которого запись удаляется. Конечный компьютер так же сохраняет в своей таблице ARP соответствие IP-адреса и MAC-адресе отправителя.

Если же удаленный узел достижим через шлюз, то пакет передается ему, и он принимает решение о методе доставки конечному узлу. В этом случае ARP запрос будет послан для выяснения аппаратного адреса шлюза.

Для получения таблицы ARP, необходимо запустить команду **arp** с ключом **-a**

```
arp -a
```

Команда **arp -a** позволяет получить список всех устройств, которые в данный момент представлены в ARP-кэше узла, а также IPv4-адрес, физический адрес и тип адресации (статическая/динамическая) для каждого из устройств.

Интерфейс: 192.168.1.172 --- 0x8		
адрес в Интернете	Физический адрес	Тип
192.168.1.1	c0-4a-00-a4-7b-6e	динамический
192.168.1.127	ac-e0-10-2d-ff-9f	динамический
192.168.1.201	4c-bb-58-cd-9d-0f	динамический
192.168.1.255	ff-ff-ff-ff-ff-ff	статический
224.0.0.22	01-00-5e-00-00-16	статический
224.0.0.251	01-00-5e-00-00-fb	статический
224.0.0.252	01-00-5e-00-00-fc	статический
231.0.0.3	01-00-5e-00-00-03	статический
239.255.255.250	01-00-5e-7f-ff-fa	статический
255.255.255.255	ff-ff-ff-ff-ff-ff	статический

Рис.3.9 Результат выполнения команды `arp -a`

Задание 10.

Получите таблицу ARP локального компьютера.

Команда `arp` также позволяет выполнять модификацию таблицы маршрутизации с помощью ключей `-s` и `-d` (добавление и удаление соответственно).

В частности, если администратору сети необходимо повторно заполнить кэш обновленными данными, можно выполнить команду `arp -d*` для очистки кэша.

Кэш ARP содержит данные, полученные только от тех устройств, к которым недавно осуществлялся доступ. Чтобы проверить заполнение кэша ARP, следует выполнить команду `ping` для проверки связи с устройством, чтобы для него была создана запись в таблице ARP.

3.7. Утилита netstat

Команда `netstat` позволяет получить список используемых протоколов, локальных адресов и номеров портов, адрес и номер порта на удаленном узле, а также сообщает состояние соединений.

Если запустить команду `netstat` без параметров, то можно получить список активных TCP соединений между локальным и удаленными компьютерами. В колонке "состояние" отображается статус TCP-соединения.

TCP	127.0.0.1:61643	DESKTOP-N0118HE:61643	ESTABLISHED
TCP	127.0.0.1:61643	DESKTOP-N0118HE:61642	TIME_WAIT
TCP	127.0.0.1:61646	DESKTOP-N0118HE:61644	TIME_WAIT
TCP	127.0.0.1:61892	DESKTOP-N0118HE:61893	ESTABLISHED
TCP	127.0.0.1:61893	DESKTOP-N0118HE:61892	ESTABLISHED
TCP	127.0.0.1:61973	DESKTOP-N0118HE:61974	ESTABLISHED
TCP	127.0.0.1:61974	DESKTOP-N0118HE:61973	ESTABLISHED
TCP	127.0.0.1:62012	DESKTOP-N0118HE:62013	ESTABLISHED
TCP	127.0.0.1:62013	DESKTOP-N0118HE:62012	ESTABLISHED
TCP	127.0.0.1:62113	DESKTOP-N0118HE:62114	ESTABLISHED
TCP	127.0.0.1:62114	DESKTOP-N0118HE:62113	ESTABLISHED
TCP	127.0.0.1:64022	DESKTOP-N0118HE:64023	ESTABLISHED
TCP	127.0.0.1:64023	DESKTOP-N0118HE:64022	ESTABLISHED
TCP	127.0.0.1:65381	DESKTOP-N0118HE:65382	ESTABLISHED
TCP	127.0.0.1:65382	DESKTOP-N0118HE:65381	ESTABLISHED
TCP	192.168.1.172:51070	13.94.211.113:https	ESTABLISHED
TCP	192.168.1.172:52913	srv133-129-240-87:https	ESTABLISHED
TCP	192.168.1.172:53005	40.67.252.61:https	ESTABLISHED
TCP	192.168.1.172:57225	162.125.18.133:https	ESTABLISHED

Рис.3.10. Результат работы команды netstat

Задание 11.

Получите список активных TCP-соединений локального компьютера.

По умолчанию netstat выполняет преобразование полученных IP-адресов в символьные имена DNS и номера портов в название сетевых служб. Это замедляет работу netstat, поэтому если преобразование не требуется, то можно указать ключ **-n**

TCP	192.168.1.172:61726	188.165.150.1:1080	TIME_WAIT
TCP	192.168.1.172:61729	188.165.150.1:1080	ESTABLISHED
TCP	192.168.1.172:61738	188.165.150.1:1080	TIME_WAIT
TCP	192.168.1.172:61741	188.165.150.1:1080	TIME_WAIT
TCP	192.168.1.172:61969	194.226.133.163:2193	ESTABLISHED
TCP	192.168.1.172:62115	169.60.79.10:443	ESTABLISHED
TCP	192.168.1.172:64024	31.13.81.9:443	ESTABLISHED
TCP	192.168.1.172:65383	31.13.81.9:443	ESTABLISHED
TCP	:::1:61548	:::1:61549	ESTABLISHED
TCP	:::1:61549	:::1:61548	ESTABLISHED

Рис.3.11. Результат работы команды netstat -n

Задание 12.

Получите список активных TCP-соединений локального компьютера без преобразования IP-адресов в символьные имена DNS.

Если указать ключ **-a**, то в списке соединений будут указаны также и прослушиваемые компьютером порты TCP и UDP.

TCP	127.0.0.1:61816	DESKTOP-N0118HE:61817	TIME_WAIT
TCP	127.0.0.1:61819	DESKTOP-N0118HE:61820	TIME_WAIT
TCP	127.0.0.1:61892	DESKTOP-N0118HE:61893	ESTABLISHED
TCP	127.0.0.1:61893	DESKTOP-N0118HE:61892	ESTABLISHED
TCP	127.0.0.1:61973	DESKTOP-N0118HE:61974	ESTABLISHED
TCP	127.0.0.1:61974	DESKTOP-N0118HE:61973	ESTABLISHED
TCP	127.0.0.1:62012	DESKTOP-N0118HE:62013	ESTABLISHED
TCP	127.0.0.1:62013	DESKTOP-N0118HE:62012	ESTABLISHED
TCP	127.0.0.1:62113	DESKTOP-N0118HE:62114	ESTABLISHED
TCP	127.0.0.1:62114	DESKTOP-N0118HE:62113	ESTABLISHED
TCP	127.0.0.1:64022	DESKTOP-N0118HE:64023	ESTABLISHED
TCP	127.0.0.1:64023	DESKTOP-N0118HE:64022	ESTABLISHED
TCP	127.0.0.1:65381	DESKTOP-N0118HE:65382	ESTABLISHED
TCP	127.0.0.1:65382	DESKTOP-N0118HE:65381	ESTABLISHED
TCP	192.168.1.172:139	DESKTOP-N0118HE:0	LISTENING
TCP	192.168.1.172:51070	13.94.211.113:https	ESTABLISHED
TCP	192.168.1.172:52913	srv133-129-240-87:https	ESTABLISHED
TCP	192.168.1.172:53005	40.67.252.61:https	ESTABLISHED

Рис.3.12. Результат работы команды `netstat -n`

Задание 13.

Получите список прослушиваемых компьютером портов TCP и UDP с и без преобразования IP-адресов в символьные имена DNS.

Утилита `netstat` в операционной системе Windows XP и старше поддерживает ключ **-o**, с помощью которого можно получить название/идентификатор процесса, создавшего/прослушивающего соединение.

TCP	127.0.0.1:65381	DESKTOP-N0118HE:65382	ESTABLISHED	12000
TCP	127.0.0.1:65382	DESKTOP-N0118HE:65381	ESTABLISHED	10296
TCP	192.168.1.172:51070	13.94.211.113:https	ESTABLISHED	10296
TCP	192.168.1.172:52913	srv133-129-240-87:https	ESTABLISHED	10296
TCP	192.168.1.172:53005	40.67.252.61:https	ESTABLISHED	10296
TCP	192.168.1.172:57225	162.125.18.133:https	ESTABLISHED	10296

Рис.3.13. Результат работы команды `netstat -o`

3.8. Утилита telnet

Данная программа изначально была предназначена для реализации сеансов удаленного терминала с компьютером по сети по протоколу telnet. В качестве «побочного» эффекта утилиты можно отметить ее способность проверять прослушиваемые порту протокола TCP.

Формат выполнения команды следующий:

```
telnet хост порт
```

Хост – это удаленный компьютер, порт – это номер TCP-порта.

Значения номеров портов широко известных сетевых служб приведены в таблице 2.1:

Таблица 2.1 - Порты основных сетевых служб

Порт	Служба	Протокол
25	Почтовый сервис	SMTP
110	Почтовый сервис	POP
143	Почтовый сервис	IMAP
80	Веб-сервер	HTTP
443	Веб-сервер поверх SSL	HTTPS
21	Передача файлов	FTP

Если соединение установилось, то, чтобы прервать сессию связи, необходимо перейти в режим команд, для этого нажать **Ctrl +]** и набрать команду quit.

Задание 14.

Получите список активных сетевых служб на удаленных узлах: www.sfedu.ru и любой внешний адрес по вашему выбору.

4. Мониторинг состояния элементов сети с использованием анализаторов сетевого трафика Wireshark

4.1. Цели и задачи работы

Мониторинг и анализ сетевого трафика являются неотъемлемой частью процесса управления компьютерной сетью и используются для диагностики, тестирования и поиска неисправностей, для оптимизации структуры информационных потоков, а также выявления и решения проблем в обеспечении безопасности узлов компьютерной сети и информации, циркулирующей между ними.

Целью данного занятия является приобретение навыков захвата сетевого трафика в сегменте локальной сети и анализа собранной информации с помощью программного анализатора протоколов Wireshark. Для успешного достижения целей занятия слушателям необходимо повторить теоретический материал, касающийся назначения и функционирования протоколов стека TCP/IP.

Для выполнения практических заданий в учебном классе должен быть развернут сегмент локальной вычислительной сети на концентраторе или коммутаторе, включающий в себя рабочие станции с операционной системой семейства Microsoft Windows по количеству слушателей. При выполнении некоторых упражнений понадобится наличие сервера HTTP или подключение к сети Интернет. Для установки необходимого программного обеспечения на рабочих станциях должны быть доступны инсталляционные пакеты библиотеки WinPCap (версия не ниже 2.3) и анализатора Wireshark.

4.2. С니феры, их назначение и применение

Сниффер, или анализатор трафика, (от англ. to sniff — нюхать) — программа или программно-аппаратное устройство, предназначенное для перехвата и / или последующего анализа сетевого трафика определенных протоколов.

Анализаторы сетевых пакетов применяются для:

- анализа имеющихся проблем в сети;
- обнаружения сетевых попыток вторжения;
- определения злоупотребления трафика пользователями (внутри системы так и снаружи нее);
- документирования нормативных требований (возможного периметра входа в систему, конечных точек распространения трафика);
- получения информации о возможностях сетевого вторжения;
- изолирования эксплуатируемых систем;
- мониторинга загрузки каналов глобальной сети;

- использования для отслеживания состояния сети (в том числе деятельность пользователей как в системе, так и за ее пределами);
- мониторинга перемещаемых данных;
- отслеживания WAN и безопасности конечных точек состояния;
- сбора сетевой статистики;
- фильтрации подозрительного контента, идущего от сетевого трафика;
- создания первичного источника данных для отслеживания состояния и управления сети;
- слежения онлайн в качестве шпиона, собирающего конфиденциальную информацию пользователей;
- отладки серверной, клиентской связи;
- проверки эффективности внутреннего контроля (контроля доступа, брандмауэров, фильтров спама и пр.).

Анализаторы трафика могут предоставлять пользователю следующие возможности:

- поддержка протоколов канального уровня, а также физических интерфейсов
- декодирование протоколов
- доступ к статистике, просмотру трафика в реальном времени и др.

Перехват информационных пакетов идет в следующей последовательности.

1. Перехват заголовков или всего содержимого.

Снифферы могут перехватывать или все содержимое пакетов данных, или всего лишь их заголовки. Второй вариант позволяет уменьшить общие требования к хранению информации, а также избежать юридических проблем, связанных с несанкционированным изъятием личной информации пользователей. При этом, история передаваемых заголовков пакетов может иметь достаточный объем информации, для выявления необходимой информации или диагностики неисправностей.

2. Декодирование пакетов.

Перехваченная информация декодируется из цифрового (нечитабельного вида) в удобный для восприятия, чтения тип. Система снифферов позволяет администраторам анализатора протоколов легко просматривать информацию, которая пересылалась или получалась пользователем.

Анализаторы различаются по:

- способности отображения данных (создание временных диаграмм, реконструирование UDP, TCP протоколов данных и пр.);
- типу применения (для обнаружения ошибок, первопричин либо для слежения онлайн за пользователями).

Некоторые снифферы могут генерировать трафик и действовать в качестве исходного устройства, что позволяет им, например, использоваться в качестве тестеров протоколов. Такие системы тест-снифферов позволяют генерировать трафик, необходимый для функционального тестирования. Кроме этого, снифферы могут целенаправленно вводить ошибки для проверки способностей тестируемого устройства.

Анализаторы трафика могут быть реализованы как в программном, так и аппаратном варианте.

Можно выделить следующие наиболее функциональные системные анализаторы для слежения онлайн.

Wireshark. Программа находится в открытом, свободном доступе. Анализатор используют для устранения неполадок в сети, анализа, разработки программного обеспечения и протокольной связи, ее образования. Первоначально программа была названа Ethereal. Однако в мае 2006 была переименована из-за проблем товарного знака. Сниффер работает на базе следующих ОС: Linux и OS X, BSD и Solaris, Microsoft Windows и других Unix систем.

Tcpdump. Анализатор пакетов, который не имеет графического интерфейса (ввод необходимых требований происходит через командную строку). Этот нюанс позволяет администратору перехватывать протоколы управления и IP информационных пакетов. Tcpdump работает на большинстве Unix-подобных ОС: Linux и Solaris, BSD и OS X, HP-UX и AIX, Windows.

Ngrep – сетевой анализатор информационных пакетов, написанный Риттером И. Команды вводятся в командной строке, опираясь на библиотеки PCAP и GNU. Ngrep - приложение с открытым исходным кодом, который доступен для загрузки с сайта Ngrep на SourceForge. Он может быть скомпилирован и перенесен на несколько платформ. Сниффер работает во многих UNIX-подобных ОС: Linux и Solaris, AIX и BSD, Microsoft.

Ettercap. Сниффер применяется для анализа протоколов компьютерной сети и проверки безопасности. Работает на Mac OS X, Linux, Solaris и BSD, Microsoft Windows. Способен перехватывать трафик сегментов сети, считывать пароли, может использоваться для слежения онлайн ряда распространенных протоколов.

Kismet – сетевой анализатор пакетов, способный обнаруживать сетевые вторжения для беспроводных локальных сетей (802.11). Kismet работает с любой беспроводной картой, поддерживающей режим мониторинга и прослушивающий 802.11a и 802.11b, 802.11g, 802.11n трафик. Программа

работает под управлением FreeBSD и Linux, NetBSD и OpenBSD, Mac OS X, Microsoft Windows.

Cain&Abel (Каин и Абель). Сниффер перехватывающий пароли в ОС Microsoft Windows. Помимо алгоритмов анализа трафика может осуществлять криптоанализ, выявлять внешние атаки и различные трещины в безопасности систем.

Capsa – сетевой анализатор, разработанный Colasoft с целью помочь администраторам контролировать, анализировать и быстро устранять неполадки в проводных, беспроводных сетях. Сейчас имеется 3 вида Capsa анализаторов: Enterprise, Free и Professional Edition.

Carnivore (позднее DCS1000). Разработан для Федерального Бюро Расследований как система мониторинга электронной почты, сообщений подозреваемых. Используется и в качестве сетевого анализатора, контролирующего весь интернет-трафик пользователей.

Justniffer. TCP анализатор, который позволяет войти в сетевой трафик во время отклика. Формат вывода трафика можно легко настроить. Перехватывает HTML, JavaScript и CSS, звуки, изображения и текстовые файлы и пр.

Microsoft Network Monitor. Сниффер, позволяющий перехватывать, просматривать и анализировать сетевые данные, расшифровывать протоколы сети. Изначально разработан Raymond Patch (транспортный протокол) и сетевой адаптер для менеджеров Microsoft.

4.3. Общие сведения о программе Wireshark

Существует множество инструментальных средств, предоставляющих необходимые возможности для выполнения мониторинга сети и анализа сетевого трафика. Одним из таких средств является программный пакет Wireshark (также известный как Ethereal), представляющий собой программный анализатор протоколов. Анализатор протоколов переводит сетевой адаптер в режим «беспорядочного» приема кадров, записывает в свой буфер отфильтрованные кадры сетевого трафика, по запросам пользователя выводит на экран те или иные кадры из буфера и посредством декодера протоколов предоставляет пользователю информацию о значениях полей заголовка протокола и содержимое его блока данных.

Как и большинство программ такого класса, Wireshark содержит следующие основные компоненты:

- фильтр захвата,
- буфер кадров,
- декодер протоколов,
- фильтр отображения захваченных кадров
- модуль статистики с элементами экспертной системы.

К несомненным достоинствам Wireshark относятся:

- наличие реализаций для Unix и Windows;

- наличие исходного кода программы;
- возможность захвата трафика в сетевых сегментах различных базовых технологий;
- возможность анализа огромного числа протоколов (более 700);
- возможность экспорта/импорта файлов данных в формат распространенных анализаторов (несколько десятков форматов);
- мощная и удобная система поиска и фильтрации информации в буфере пакетов;
- наличие элементов экспертной системы;
- возможность сохранения на диск выделенного фрагмента пакета;
- наличие полезных утилит командной строки для осуществления захвата трафика и обработки сохраненных файлов.

Wireshark может работать с файлами данных tcpdump, Sniffer Pro, NetXray, MS Network Monitor, Novell's Lanalyzer и т.п. Поддерживает DNS, FDDI, FTP, HTTP, ICQ, IPV6, IPX, IRC, MAPI, MOUNT, NETBIOS, NFS, NNTP, POP, PPP, TCP, TELNET, X25 и т.д.

4.4. Пользовательский интерфейс программы

На экране монитора в программе Wireshark вы увидите несколько панелей с отображением сетевых пакетов, только что записанных в буфер. Общий вид окна приложения представлен на рис. 4.1.

Пользовательский интерфейс программы содержит следующие компоненты:

- меню команд и панель инструментов;
- фильтр отображения пакетов;
- список пакетов в буфере;
- панель отображения декодера протоколов;
- панель отображения пакета в шестнадцатеричном коде и символах ASCII.

Панель со списком пакетов построчно отображает характеристики того или иного пакета (номер по порядку в буфере, время захвата, адреса источника и получателя, тип протокола и общая информация о нем). Перемещение по списку осуществляется с помощью мыши или клавиатуры, причем информация на двух других панелях обновляется автоматически.

На панели декодера протоколов, нажимая указателем мыши на символы «+» или «-», можно отображать информацию о полях заголовков протоколов с требуемым уровнем детализации. При выборе того или иного служебного поля в заголовке оно автоматически выделяется на нижней панели, где отображается текущий пакет в шестнадцатеричном виде.

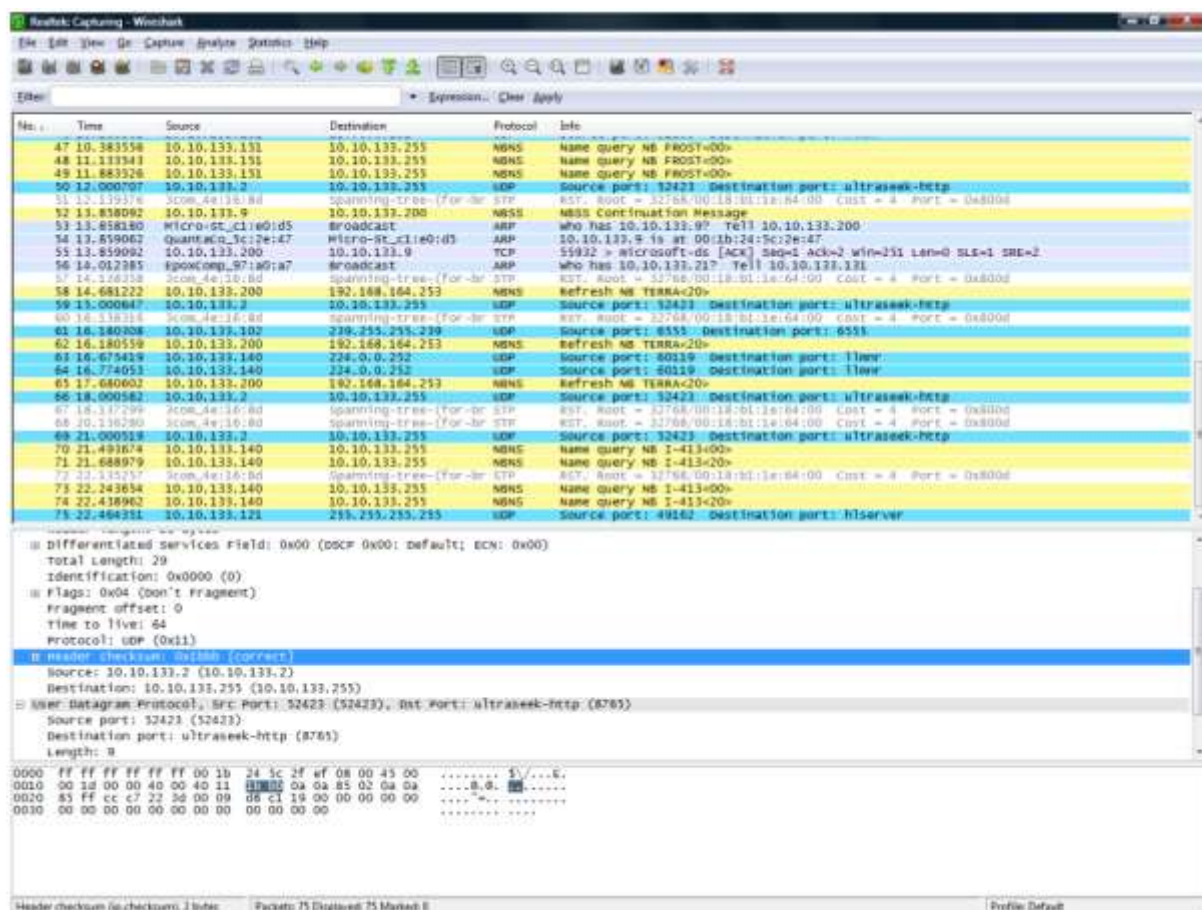


Рис. 4.1. Общий вид приложения Wireshark

4.5. Фильтр отображения пакетов

С помощью фильтра отображения можно быстро убрать «мусор». Выражение фильтрации может представлять собой просто название протокола, который присутствует в пакете на том или ином уровне вложенности. Например: `arp` — для отображения пакетов протокола ARP, `tcp` — для отображения пакетов, в которых присутствует заголовок протокола TCP.

Более сложные выражения фильтрации строятся с помощью зарезервированных слов, обычно представляющих собой названия полей заголовков того или иного протокола, знака операции сравнения и конкретного значения в шестнадцатеричном или десятичном виде. Наиболее часто используемые выражения фильтрации и их значения приведены в табл. 4.1.

Таблица 4.1 Выражения фильтрации и их значения

Выражение	Значение выражения и пример записи
frame.marked	Маркированный кадр frame.marked == true
frame.number	Номер кадра frame.number == 150
frame.time	Время захвата кадра frame.time == "Feb 1, 2006 09:00:00"
frame.pkt_len	Длина кадра frame.pkt_len == 48
eth.dst	Заголовок Ethernet: МАС адрес назначения eth.dst == 01:00:5e:00:00:02
eth.src	Заголовок Ethernet: МАС адрес источника eth.src == 00:a0:cc:30:c8:db
eth.type	Заголовок Ethernet: тип вложенного протокола eth.type == 0x0800
arp.hw.type	Заголовок протокола ARP: тип протокола канального уровня arp.hw.type == 0x0001
arp.proto.type	Заголовок протокола ARP: тип протокола сетевого уровня arp.proto.type == 0x0800
arp.opcode	Заголовок протокола ARP: код операции arp.opcode == 0x0001
arp.src.hw_mac	Заголовок протокола ARP: МАС адрес источника arp.src.hw_mac == 00:10:4b:30:c4:4a
arp.src.proto_ipv4	Заголовок протокола ARP: IP адрес источника arp.src.proto_ipv4 == 10.1.0.1
arp.dst.hw_mac	Заголовок протокола ARP: МАС адрес назначения arp.dst.hw_mac == 00:00:00:00:00:00
arp.dst.proto_ipv4	Заголовок протокола ARP: IP адрес назначения arp.dst.proto_ipv4 == 10.1.0.2
ip.version	Заголовок протокола IP: версия протокола IP ip.version == 4
ip.hdr_len	Заголовок протокола IP: длина заголовка ip.hdr_len == 24

Выражение	Значение выражения и пример записи
ip.flags.df	Заголовок протокола IP: флаг фрагментации ip.flags.df == 0
ip.flags.mf	Заголовок протокола IP: флаг не последнего фрагмента ip.flags.mf == 0
ip.frag_offset	Заголовок протокола IP: смещение фрагмента ip.frag_offset == 0
ip.ttl	Заголовок протокола IP: время жизни пакета ip.ttl == 1
ip.proto	Заголовок протокола IP: протокол вышестоящего уровня ip.proto == 0x01
ip.src	Заголовок протокола IP: IPадрес источника ip.src == 10.0.0.99
ip.dst	Заголовок протокола IP: IPадрес назначения ip.dst == 224.0.0.2
ip.addr	Заголовок протокола IP: IPадрес ip.addr == 10.2.0.0/16
tcp.srcport	Заголовок протокола IP: порт источника tcp.srcport == 1054
tcp.dstport	Заголовок протокола IP: порт назначения tcp.dstport == 21
tcp.seq	Заголовок протокола IP: последовательный номер tcp.seq == 4856133
tcp.ack	Заголовок протокола IP: номер подтверждения tcp.ack == 4856134
tcp.flags.urg	Заголовок протокола IP: бит присутствия срочных данных tcp.flags.urg == 0
tcp.flags.ack	Заголовок протокола IP: бит присутствия подтверждения tcp.flags.ack == 1
tcp.flags.push	Заголовок протокола IP: бит выталкивания данных tcp.flags.push == 0
tcp.flags.reset	Заголовок протокола IP: бит сброса соединения tcp.flags.reset == 0
tcp.flags.syn	Заголовок протокола IP: бит синхронизации сессии tcp.flags.syn == 1
tcp.flags.fin	Заголовок протокола IP: бит завершения сессии tcp.flags.fin == 0
tcp.window_size	Заголовок протокола IP: размер приемного окна tcp.window_size == 8760
udp.srcport	Заголовок протокола UDP: порт источника udp.srcport == 2364

Выражение	Значение выражения и пример записи
udp.dstport	Заголовок протокола UDP: порт назначения <code>udp.dstport == 53</code>
icmp.type	Заголовок протокола ICMP: тип сообщения <code>icmp.type == 8</code>
icmp.code	Заголовок протокола ICMP: уточняющий код сообщения <code>icmp.code == 0x00</code>

В примерах записи выражений таблицы 3.1 приведены выражения с операцией сравнения «Равно», которая записывается с помощью двойного знака равенства «==» (допустимо использование «eq»).

Другие операции сравнения записываются с помощью следующих операторов:

- a. `!=` (ne) — не равно, пример: `eth.type != 0x0800;`
- b. `>` (gt) — больше, пример: `tcp.srcport > 1023;`
- c. `<` (lt) — меньше, пример: `frame.pkt_len lt 60;`
- d. `>=` (ge) — больше или равно, пример: `frame.pkt_len ge 60;`
- e. `<=` (le) — меньше или равно, пример: `tcp.dstport <= 1023.`

Значение любого выражения фильтрации возвращает переменную булевского типа. Таким образом, выражение `udp` означает присутствие в кадре заголовка протокола UDP, по аналогии с этим выражение `tcp.flags.syn` означает присутствие в заголовке протокола TCP бита синхронизации сессии в установленном состоянии (значение 1).

К любому из выражений можно применить операцию логического отрицания, заключив его в скобки и поставив перед ним знак отрицания «NOT» или «!». Например, выражение `!(ip.addr == 10.0.0.1)` означает, что из буфера отображения необходимо убрать все пакеты, в которых встречается IP-адрес 10.0.0.1.

Другой удобный способ ввода выражения фильтрации состоит в следующем. На панели декодера протоколов отображается требуемое поле, в контекстном меню выбирается пункт «Apply as Filter» и далее исполняется либо команда «Selected», либо «Not Selected» в зависимости от задачи фильтрации (рис. 4.2).

При необходимости создания сложного выражения фильтрации в меню «Apply as Filter» выбирайте команды, начинающиеся с многоточия, при этом новое выражение будет добавлено к результирующему выражению фильтрации.

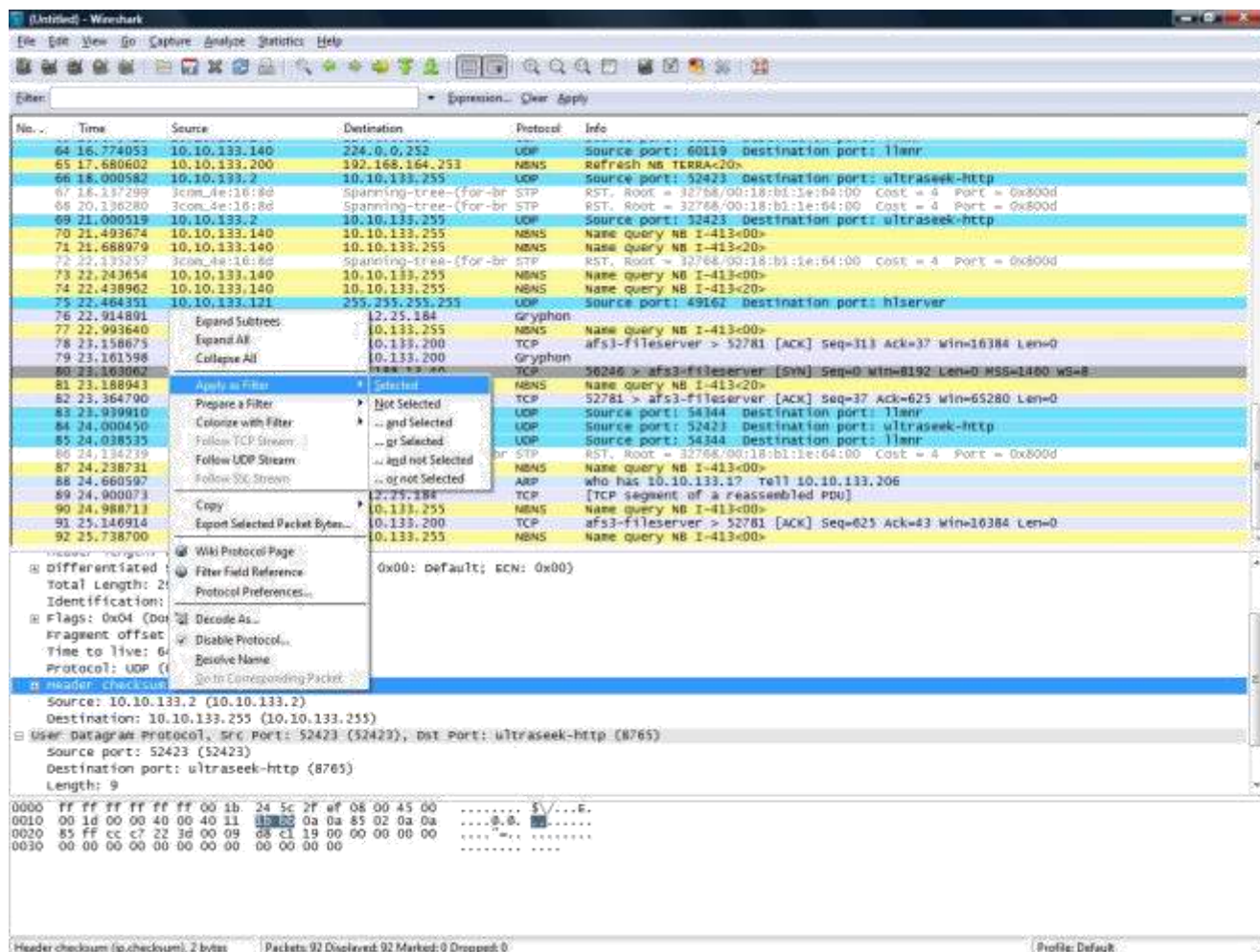


Рис. 4.2. Контекстное меню создания фильтра

При создании выражения фильтрации имейте в виду, что в буфере могут находиться кадры других узлов.

В выражениях фильтрации первый операнд операции сравнения допускает использование указателя диапазона, если второй операнд представляет собой массив байт или строку символов. Указатель диапазона определяется с помощью квадратных скобок и может быть использован как применительно к кадру в целом (frame), так и с любым полем заголовка.

Указатель диапазона допускает следующий синтаксис:

- [i:j] начальное смещение i, длина j;
- [ij] начальное смещение i, конечное смещение j, включительно; с. [i] начальное смещение i, длина 1;
- [:j] начальное смещение 0, длина j;
- [i:] начальное смещение i, до конца поля.

Например, записи `frame[6:3]` и `eth.src[:3]` идентичны и могут быть использованы для указания на код фирмыпроизводителя сетевого адаптера, передавшего кадр. Начальное смещение может иметь

отрицательное значение, в этом случае оно отсчитывается от конца поля, причем последний байт поля имеет смещение, равное -1 , предпоследний -2 и так далее. Например, выражение `frame[5:] == "hello"` определяет кадр, оканчивающийся строкой «hello».

Строка, как видно из предыдущего примера, записывается в кавычках. Запись массива байт осуществляется побайтно в шестнадцатеричном виде с разделителем «.» или «:», например `00.45.f5.2d`.

Используя символ «,» в указателе диапазона, можно перечислить несколько непересекающихся диапазонов, объединив их в одном операнде. Например, выражение `tcp[2,10,1316] == 00.01.c0.f8.01.66` сравнивает в заголовке протокола TCP поле «Тип обслуживания» с «0x00», поле «Протокол» с «0x01» и поле «IPадрес источника» с «0xc0f80166».

Быстро вернуться к тому или иному ранее вводимому выражению фильтрации можно с помощью списка истории ввода, доступ к которому осуществляется нажатием на кнопку с символом «▼», расположенную в строке фильтра (не забывайте нажимать кнопку «Apply» для применения того или иного фильтра к буферу кадров).

4.6. Поиск кадров

Поиск кадров в буфере, удовлетворяющих тем или иным критериям, осуществляется с помощью команды меню **Edit** ⇒ **Find Packet**.

Диалоговое окно определения критериев поиска пакетов изображено на рис. 4.3.

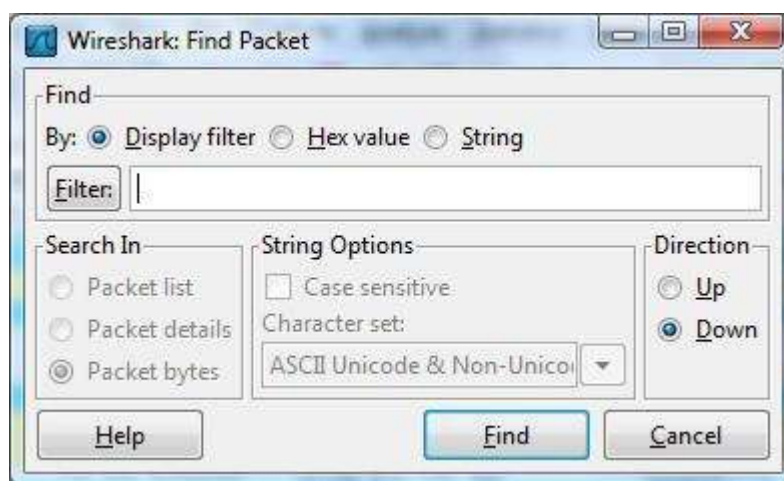


Рис.4.3. Диалоговое окно определения критериев поиска кадров

Критерии поиска можно определять в виде выражения фильтрации (Display filter), шаблона в шестнадцатеричном виде (Hex value) и текстовой строки (String) в кодировке ASCII и (или) Unicode.

В первом случае можно использовать все допустимые выражения фильтрации (таблица 4.1) и их логические комбинации.

Во втором случае указывается шаблон для поиска в шестнадцатеричном коде.

Поиск в строке может осуществляться в области общей информации о пакете (Packet list), в панели декодера протоколов (Packet details) и непосредственно в самом пакете (Packet bytes). Поиск может производиться вверх или вниз по списку пакетов (Direction).

Команды меню **Edit** \Rightarrow **Find Next** и **Edit** \Rightarrow **Find Previous** используются для поиска с заданными критериями следующего или предыдущего пакета соответственно.

4.7. Выделение ключевых кадров

В списке буфера ключевые или наиболее важные для дальнейшего анализа пакеты можно пометить с помощью команды **Edit** \Rightarrow **Mark Packet** (toggle) основного меню или команды **Mark Packet** (toggle) контекстного меню. Эта возможность полезна при дальнейшем поиске таких пакетов в большом буфере, так как они выделяются другим цветом, а также при сохранении, экспортировании и печати пакетов.

Информация о маркированных пакетах нигде не сохраняется, поэтому все маркеры будут потеряны при выгрузке файла данных.

4.8. Сохранение данных захвата

Сохранение данных в файле производится из меню **File** \Rightarrow **Save** или **File** \Rightarrow **Save As**. Диалоговое окно сохранения данных изображено на рис. 4.4.

Обратите внимание, что сохранить можно все пакеты (All packets), только отображаемые (Displayed), выбранный пакет (Selected packet only), ранее маркированные с помощью основного или контекстного меню (Marked packet only и From first to last marked packet) или указанный диапазон пакетов (Specify a packet range).

По умолчанию Wireshark сохраняет данные в файле типа Libpcap, совместимом по формату с файлами программы TcpDump, но путем указания определенного формата в строке ввода «File Type» этого диалогового окна данные захвата можно сохранять для экспорта в другие программы анализа трафика (около двадцати поддерживаемых в настоящее время форматов).

Не забывайте сохранять данные, прежде чем начинать другой сеанс записи.

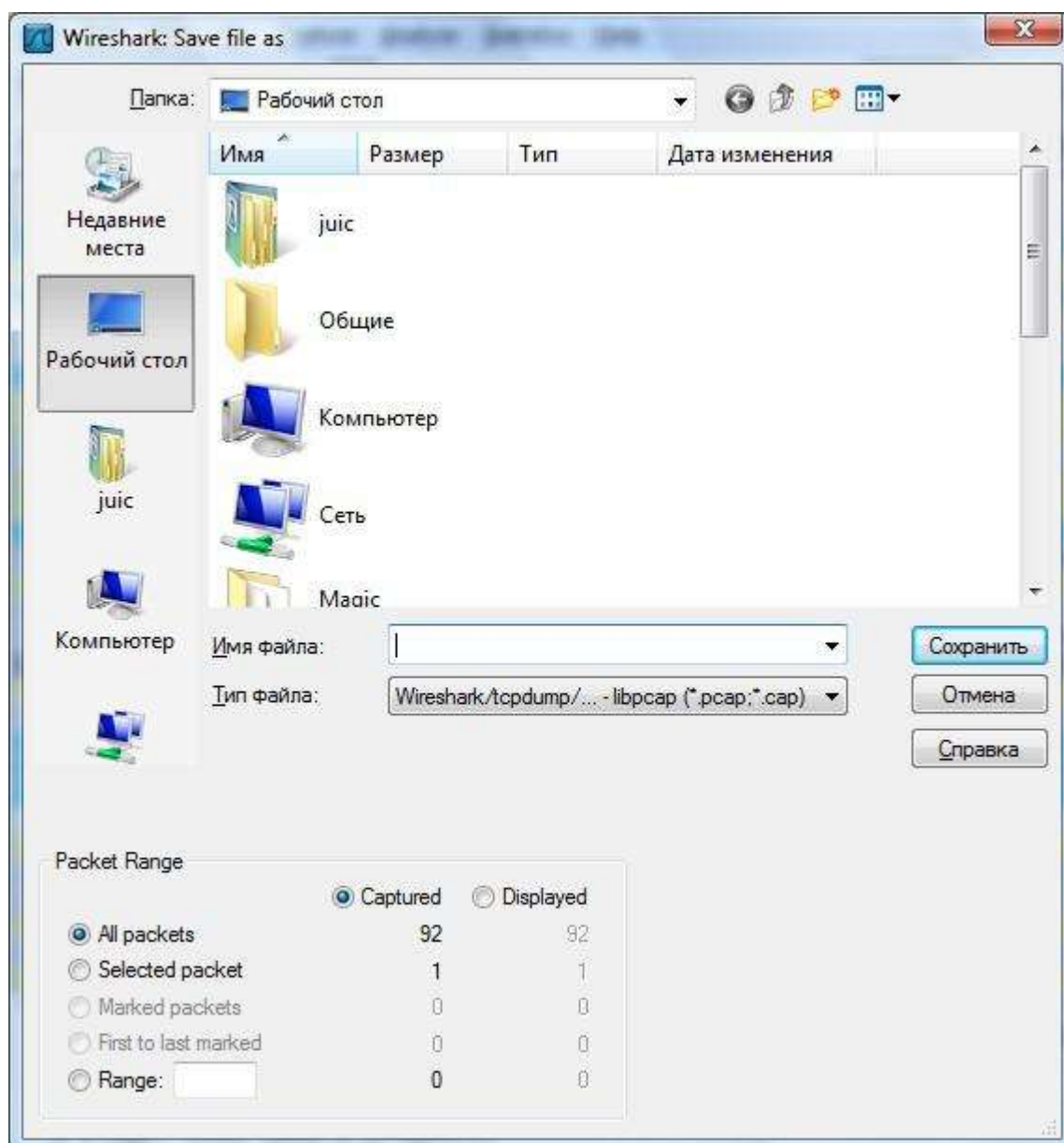


Рис. 4.4. Диалоговое окно сохранения данных

4.9. Печать информации

Распечатка информации о том или ином пакете или их множестве осуществляется посредством выполнения команды Print основного или контекстного меню. Диалоговое окно печати данных изображено на рис. 4.5.

При печати есть возможность осуществить вывод в указанный файл (Output to file) в виде простого текста (Plain text), определив диапазон распечатываемых пакетов (Packet Range) и формат вывода информации (Packet Format). Опции панели «Packet Range» полностью идентичны опциям соответствующей панели диалогового окна сохранения данных. При определении формата вывода в панели «Packet Format» есть возможность включить общую характеристику пакета (информацию верхней панели основного окна — «Packet summary line»), информацию, отображаемую на панели декодера протоколов с той или иной степенью детализации (Packet details) и собственно сам пакет в шестнадцатеричном виде (Packet bytes).



Рис. 4.5. Диалоговое окно печати данных

4.10. Просмотр кадра в отдельном окне

При составлении отчетов с использованием «скриншотов», а иногда и при анализе данных для просмотра двух пакетов одновременно удобно использовать возможность отображения пакета в отдельном окне.

Это реализуется с помощью команды «Show Packet in New Window» контекстного или основного меню программы «View». Окна, отображающие различные пакеты, показаны на рис. 4.6.

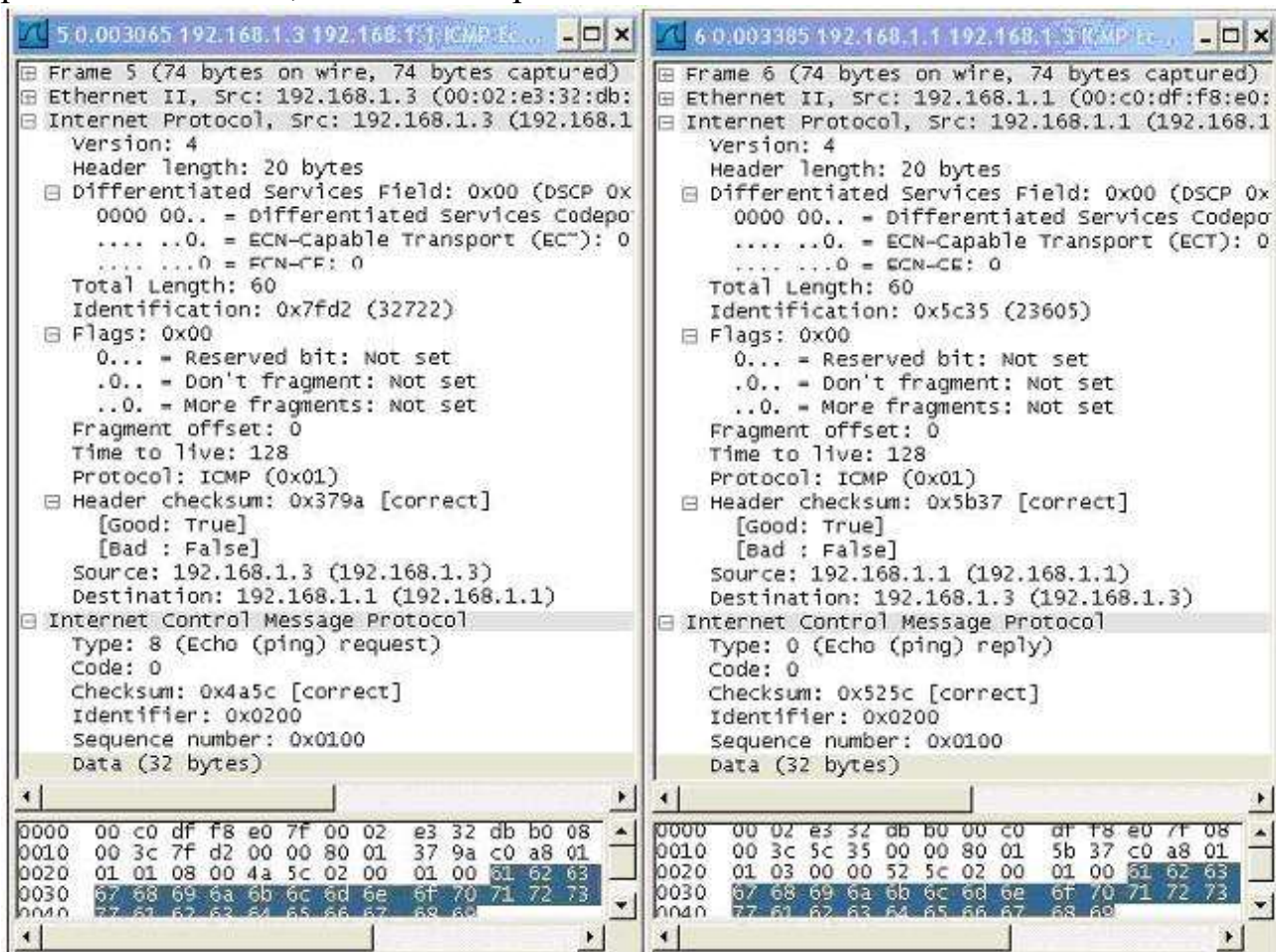


Рис. 4.6. Отображение пакетов в отдельных окнах

4.11. Анализ протоколов Ethernet и ARP

При анализе протоколов Ethernet и ARP, которые находятся в иерархии протоколов ниже IP, для выключения отображения «лишней» информации на панелях программы целесообразно отключить в программе анализ заголовка IP. Это реализуется с помощью команды «Enabled Protocols...» основного меню программы «Analyze». В диалоговом окне данной команды необходимо найти протокол IP, убрать соответствующий маркер, затем последовательно нажать кнопки «Apply» и «ОК» (рис. 4.7).

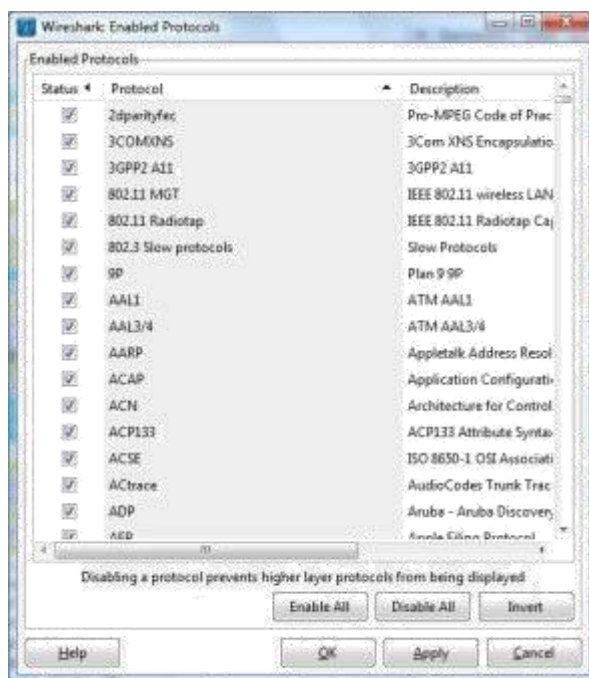


Рис. 4.7. Окно выбора протоколов для анализа

В ряде случаев при отключении анализа заголовка IP отображаемые в списке буфера IP-адреса источника и получателя могут измениться!

4.12. Анализ протокола TCP

В буфере захвата у вас находятся кадры, принадлежащие обмену клиента с сервером по протоколу HTTP, но в рамках текущего упражнения прикладной протокол нас не интересует, поэтому отключите анализ протокола HTTP. Фрагмент панелей со списком кадров после отключения анализа протокола FTP показан на рис. 4.8:

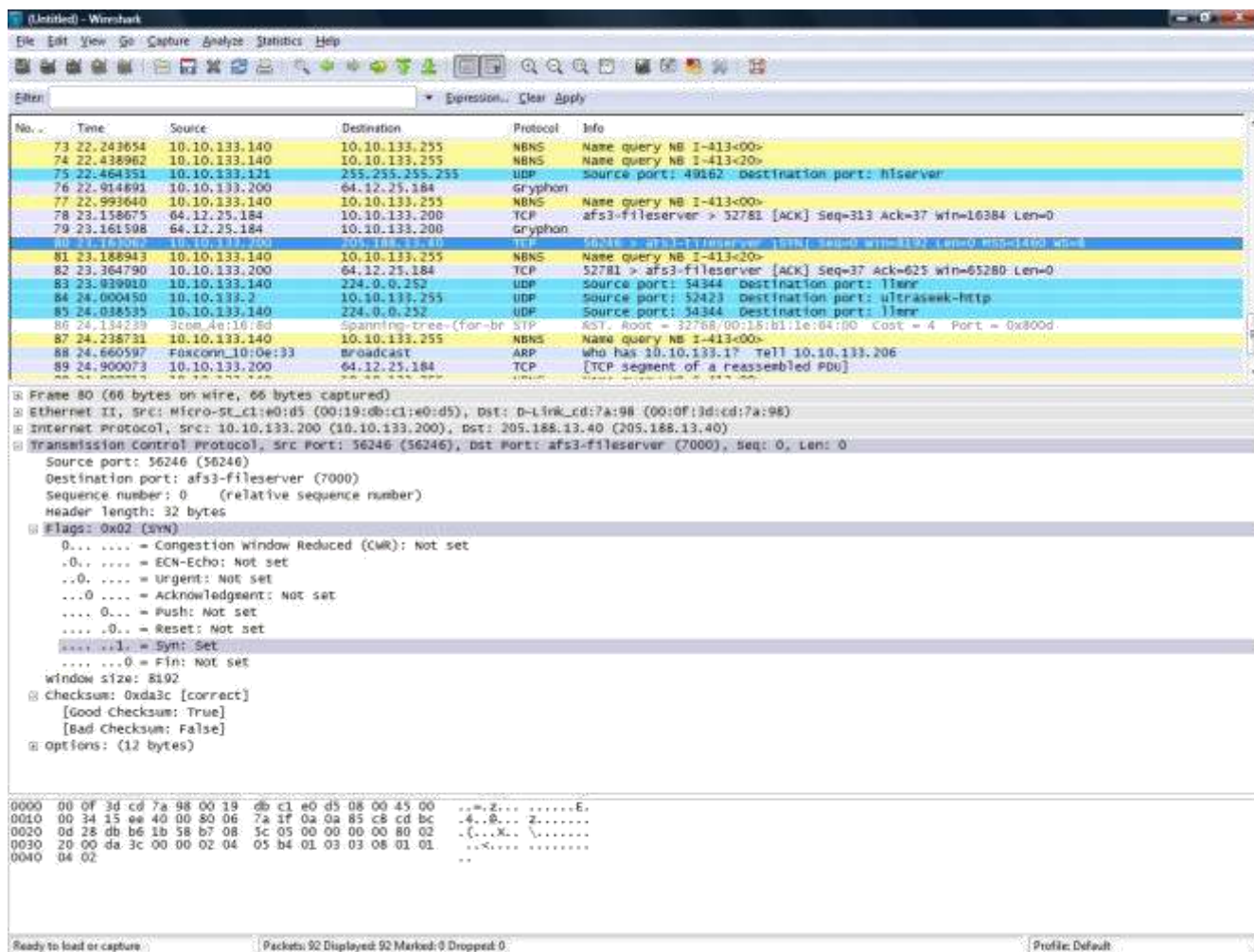


Рис. 4.8. Отображение информации о протоколе TCP

Обратите внимание, что теперь по каждому захваченному кадру приводится информация, касающаяся только протокола TCP. Например, для пакета № 4 (рис. 1.9) запись «1061> ftp» означает порты источника и назначения, «[PSH, ACK]» — установленные биты флагов, «Seq=1» — последовательный номер, «Ack=1» — номер подтверждения, «Win=16560» — размер приемного окна, «Len=398» — размер пересылаемого блока данных.

Каждая TCP-сессия (причем при обращении к одной странице сессий может быть несколько!) начинается с обмена тремя TCP-сегментами с установленными битами SYN, SYNACK и ACK.

Сеансы TCP начинаются с относительных последовательных номеров, равных нулю. Для того чтобы отобразить реальные последовательные номера, выбранные узлами при взаимодействии, необходимо выполнить команду меню **Edit** ⇒ **Preferences**, в появившемся диалоговом окне (фрагмент диалогового окна см. на рис. 4.9) выбрать протокол TCP и убрать маркер в строке параметра «Relative sequence numbers and window scaling».

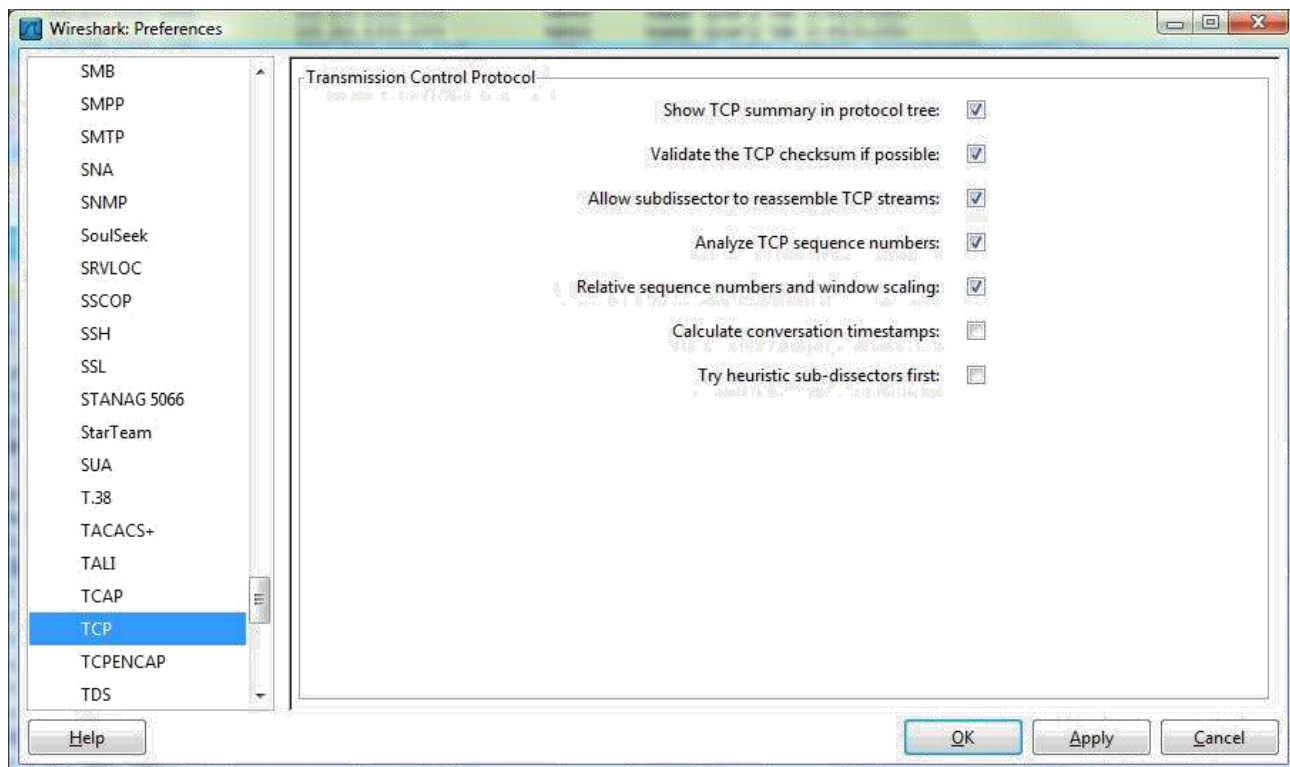


Рис. 4.9. Параметры анализа протокола TCP

Немаловажная возможность программы Wireshark по анализу TCP трафика состоит в том, что с помощью команды меню **Statistics** ⇒ **Conversations** можно быстро определить все сеансы, имеющиеся в буфере. В диалоговом окне для отображения сеансов TCP необходимо выбрать закладку TCP (рис. 4.10).

Conversations: (Untitled)

Ethernet: 14 Fibre Channel FDDI IPv4: 11 IPX JXTA NCP RSVP SCTP **TCP: 4** Token Ring UDP: 14 USB WLAN

TCP Conversations

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B
10.10.133.200	56246	205.188.13.40	afs3-fileserver	1	66	1	66	0	0
10.10.133.200	55932	10.10.133.9	microsoft-ds	2	126	1	66	1	60
10.10.133.200	56245	205.188.13.40	afs3-fileserver	3	194	3	194	0	0
10.10.133.200	52781	64.12.25.184	afs3-fileserver	10	1224	5	312	5	912

☒ Name resolution ☐ Limit to display filter

Help Copy Close

Рис. 4.10. Статистика по сеансам TCP

Для того чтобы быстро просмотреть передаваемые данные в рамках того или иного сеанса, используют команду меню Analyze ⇒ Follow TCP Stream.

После выполнения команды на экране появится диалоговое окно, в котором разными цветами будут отображены как запросы клиента, так и ответы сервера (рис. 4.11).

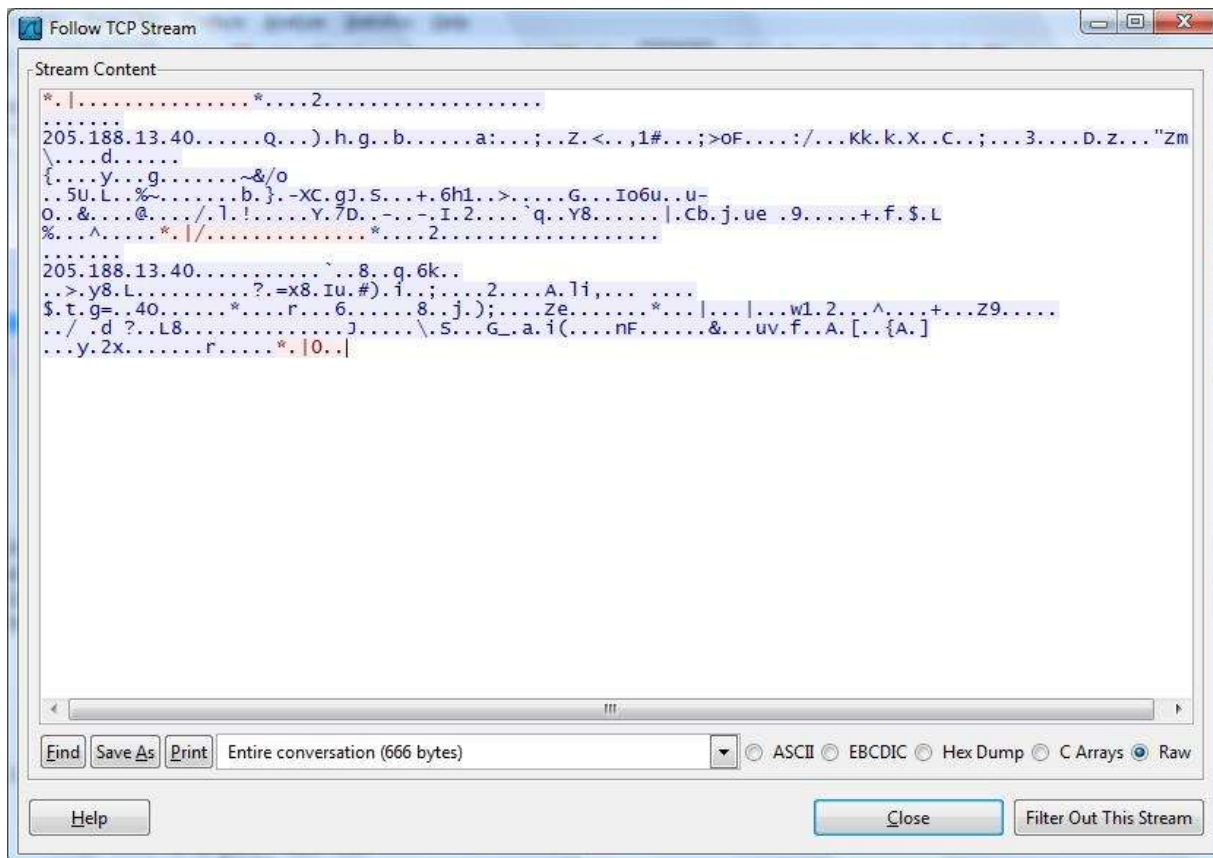


Рис. 4.11. Восстановленный сеанс TCP

Кнопка «Entire conversation» с раскрывающимся списком позволяет отобразить обе стороны, участвующие в обмене, или только одну из них. Диалоговое окно позволяет отобразить данные в различных форматах (ASCII, EBCDIC, Hex Dump, C Arrays, Raw) и сохранить их в файл. При обнаружении в сеансе кадров с каким-либо файлом можно отобразить лишь поток соответствующего направления, выбрать необходимый формат и сохранить его на диск.

Литература

1. *Котов Э.М., Целых А.Н.* Технология разработки и принципы построения вычислительных сетей / Э.М. Котов - Ростов-на-Дону: Изд-во ЮФУ, 2014.-79 с. -
<https://hub.lib.sfedu.ru/repository/material/800820776>
2. *Пуговкин А. В.* Сети передачи данных: учебное пособие / А. В. Пуговкин - Томск: Факультет дистанционного обучения ТУСУРа, 2015. – 138 с. -
http://biblioclub.ru/index.php?page=book_red&id=480793&sr=1
3. Сетевая защита на базе технологий фирмы Cisco Systems. Практический курс: учебное пособие / Под общей редакцией: Синадский Н.И. - Екатеринбург: Издательство Уральского университета, 2014. – 179 с. -
http://biblioclub.ru/index.php?page=book_red&id=275694&sr=1
4. *Гриценко Ю. Б.* Вычислительные системы, сети и телекоммуникации: учебное пособие / Ю. Б. Гриценко - Томск: ТУСУР, 2015. – 134 с. -
http://biblioclub.ru/index.php?page=book_red&id=480639&sr=1
5. *Гладких Т. В., Воронова Е. В.* Информационные системы и сети: учебное пособие / Т. В. Гладких - Воронеж: Воронежский государственный университет инженерных технологий, 2016. – 88 с.- http://biblioclub.ru/index.php?page=book_red&id=481994&sr=1
6. Электронный учебно-методический комплекс Сетевой академии Cisco по модулю «Введение в сетевые технологии».-
www.netacad.ru (доступен при регистрации)
7. Электронный учебно-методический комплекс Сетевой академии Cisco по модулю «Введение в коммутируемые сети».-
www.netacad.ru (доступен при регистрации)

Оглавление

ВВЕДЕНИЕ	3
1. Работа с IP-адресами.....	4
1.1. Теоретическое введение	4
1.1.1. Двоичная и десятичная форма записи адресов	4
1.1.2. Формат	IP-адреса 7
1.1.3. Типы	IP-адресов 9
1.1.4. Классы	IP-адресов 11
1.1.5. Бесклассовая	адресация 14
1.2. Примеры решения задач.....	15
1.3. Варианты	19
2. Построение таблиц маршрутизации.....	25
2.1. Теоретическое введение	25
2.2. Методические указания и пример выполнения работы	28
2.3. Варианты заданий	31
3. Утилиты командной строки Windows для работы с сетью.....	49
3.1. Подготовительная часть	49
3.2. Утилита ipconfig	49
3.3. Утилита ping.....	51
3.4. Утилита tracert	55
3.5. Утилита route	57
3.6. Утилита arp	58
3.7. Утилита netstat	59
3.8. Утилита telnet.....	62
4. Мониторинг состояния элементов сети с использованием анализаторов сетевого трафика Wireshark.....	63
4.1. Цели и задачи работы	63
4.2. Общие сведения о программе	66
4.3. Пользовательский интерфейс программы.....	67
4.4. Фильтр отображения пакетов	68
4.5. Поиск кадров	73

4.6.	Выделение ключевых кадров.....	74
4.7.	Сохранение данных захвата.....	74
4.8.	Печать информации	76
4.9.	Просмотр кадра в отдельном окне	77
4.10.	Анализ протоколов Ethernet и ARP.....	78
4.11.	Анализ протокола TCP	79
Литература		83

Учебное издание

**Абрамов Евгений Сергеевич,
Пескова Ольга Юрьевна,
Токарев Михаил Валерьевич**

**Практические задачи по администрированию
компьютерных сетей**

Редактор *Т.А. Кочергина*
Корректор *Н.И. Селезнева*
Компьютерная верстка *Т.А. Кочергина*

Подписано в печать 15.12.2018 г.

Формат 60×84 1/16 Усл. п.л. 5,25. Уч.-изд. л. 5,00.

Бумага офсетная. Тираж 40 экз Заказ № _____

Издательство Южного федерального университета

Отпечатано в отделе полиграфической, корпоративной и сувенирной продукции
Издательско-полиграфического комплекса КИБИ МЕДИА ЦЕНТРА ЮФУ
344090, г.Ростов-на-Дону. пр. Стачки 200/1 Тел. (863) 247-80-51