Министерство транспорта и связи Украины
Государственный департамент по вопросам связи и информатизации
Одесская национальная академия связи им. А.С. Попова

Кафедра сетей связи

Д.А. Зайцев, Т.Р. Шмелёва, С.П. Шнайдер

Методические указания к практическим занятиям и лабораторным работам по курсу «Сетевые технологии»

Для подготовки бакалавров и магистров по направлению «Телекоммуникации»

Одобрено
на заседании кафедры
«Сети связи»
Протокол № 9 от
05.06.2007 г.

Одеса 2007

УДК 621.39, 004.7, 51.681.3

План НМВ 2006/2007

Рецензент – д.т.н., доцент А.С. Лемешко

Составители: д.т.н., доцент Д.А. Зайцев, асп. Т.Р. Шмелёва, асс. С.П. Шнайдер

Курс посвящён углубленному изучению стандартов современных телекоммуникационных протоколов магистральных сетей. Основные темы курса изучаются на практических занятиях, затем выполняются соответствующие лабораторные работы. По каждой теме приведен перечень необходимого материала из конспекта лекций, представлены примеры решения типовых задач и выполнения основных этапов лабораторных работ. Для выполнения лабораторных работ используются анализаторы трафика (Ethereal) и моделирующие системы (Opnet).

Утверждено
Советом факультета
Информационных сетей
Протокол № 5 от
07.06.2007 г.

Содержание

Введение

Основой конкретной сетевой технологии является протокол, либо семейство протоколов, представленное стандартными спецификациями. Затем протокол реализуется в виде программного обеспечения, либо специализированного сетевого устройства, такого как сетевой адаптер, модем, коммутатор, маршрутизатор, конвертор интерфейсов, из которых строятся сети. Именно поэтому основное внимание уделяется изучению стандартных спецификаций протоколов и вопросам взаимодействия протоколов различных уровней в процессе инкапсуляции информации, а также вопросам доставки информации (пакетов) по назначению.

Основой для углублённого изучения протоколов и их взаимодействия является анализ трафика с полной интерпретацией передаваемых потоков битов (байтов), выделением заголовков пакетов и их полей, установлению взаимосвязей между заголовками различных уровней. Для этих целей использован программный анализатор трафика Ethereal, позволяющий записать передаваемые в сети пакеты и выполнить их автоматизированную интерпретацию. На практических занятиях решаются также задачи построения заголовков пакетов в процессе их инкапсуляции. Таким образом, имитируется работа как передающей, так и принимающей подсистем.

Изучено доминирующее на рынке семейство протоколов сетевого-сеансового уровней TCP/IP и их инкапсуляция в такие протоколы канального уровня как IEEE 802.3* (Ethernet) и PPP. Выбор канальных технологий обусловлен их широким применением в магистралях DWDM.

Вопросы доставки пакетов (кадров) в сетях составляют вторую часть материала настоящих указаний. Изучение организовано по нарастанию сложности технологий: коммутация кадров Ethernet, статическая маршрутизация в IP-сетях, протоколы динамической маршрутизации RIP, OSPF, BGP и, наконец, современная технология коммутации меток MPLS, интегрирующая принципы коммутации пакетов и коммутации каналов. Для изучения вопросов маршрутизации в сетях использована моделирующая система Opnet, позволяющая имитировать процессы создания таблиц маршрутизации (коммутации) для заданной структурной схемы сети.

Практическое занятие №1
Инкапсуляция IP-Ethernet

Подготовка:

- формат заголовков IP-пакета и Ethernet-фрейма;
- протокол ARP: таблицы, запросы.

Задачи:

- построение IP-адресов (хост, сеть, широковещательный);
- построение заголовков пакетов и фреймов;
- построение ARP запросов и таблиц.

Типовые задания:

1. По IP-адресу и маске (количеству битов адреса сети) построить адрес сети и широковещательный адрес
2. Для заданной пары хостов с известными IP и MAC-адресами построить заголовки IP и Ethernet. Обратить внимание на взаимодействие уровней эталонной модели при инкапсуляции с помощью номера типа сетевого (транспортного) уровней.
3. Построить последовательность IP-пакетов при фрагментации дейтаграмм.
4. Для заданной локальной сети построить ARP-запрос и ответ.

Примеры решения задач:

Задача № 1.

По IP-адресу и маске сети построить:

а) адрес сети;

б) широковещательный адрес.

Исходные данные:

а) IP-адрес – 198.87.137.221; маска – 255.255.128.0;

б) 198.87.137.221/14.

Решение

а) Представим IP-адрес в двоичной форме с указанием границы номер сети в соответствии с маской. Для получения адреса IP-сети следует заполнить поле номера хоста двоичными нулями, а для широковещательного адреса единицами. В таблице 1.1 представлена последовательность указанных действий.

Таблица 1.1

	Номер сети								Номер хоста							
Номер бита	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
IP-адрес	1	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	1	1	1	0	1	1	1	0	1
Маска	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Сетевой адрес	1	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Широковещательный	1	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Переведём результаты в десятичную систему и получим:

адрес сети: 198.87.128.0

широковещательный адрес: 198.87.255.255

Следует отметить, что результат может быть получен также с помощью побитовых логических операций:

- адрес сети: A & M;
- широковещательный адрес: A $\oplus$ $\neg$M.

б) Указание маски с помощью длины в битах адреса сети является более компактной формой представления; записанное через слеш число задаёт границу между последовательностью единиц и нулей двоичной маски.

Таблица 1.2

	Номер сети									Номер хоста							
Номер бита	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
IP-адрес	1	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	0	0	0	1	0	0	1	1	1	0	1	1	1	0	1
Маска	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Сетевой адрес	1	1	0	0	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Широковещательный	1	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Из таблицы 1.2 видно, что адрес сети: 198.84.0.0., а широковещательный адрес не изменился: 198.87.255.255.

Варианты заданий для самостоятельных упражнений:

1) 88.37.246.135/2;
2) 88.37.246.135/9;
3) 88.37.246.135/18;
4) 88.37.246.135/25.

Задача № 2.

Для заданной пары хостов с известными IP и MAC-адресами построить заголовки IP и Ethernet:

X: IP=194.125.16.38, MAC=08:00:09:a1:cc:b6, TTL=67, Protocol=TCP, Data_Length=356

Y: IP=224.88.137.15, MAC=02:60:8c:cd:a8:1b

Решение

Значения полей заголовков будем представлять в шестнадцатеричной системе счисления. Вначале построим заголовок Ethernet II, основной тип заголовка для инкапсуляции IP-дейтаграмм. Заметим, что поле типа позволяет выполнить демультиплексирование пакетов различных протоколов, инкапсулированных в кадр; тип протокола IP равен 0x800 (RFC).

```
         6 bytes            6 bytes      2 bytes
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Destination Address | Source Address  | Type  |
|    02608ccda81b     | 0800091acc6b    | 0800  +
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Затем построим заголовок IP в соответствии с RFC 791:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
|   4   |   5   |      00       |             0178              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Identification       |Flags|      Fragment Offset    |
|               12ab            |  0  |           000           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
|       43      |       06      |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Source Address                         |
|       c2              7d              10              26      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Destination Address                       |
|       e0              58              89              0f      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Options                     |    Padding    |
|                    (absent)                     |   (absent)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Длина IP-заголовка (IHL) измеряется в 4 байтовых словах.
- Общая длина (`Total Length`) складывается из длины заголовка (20 байтов) и длины инкапсулированных данных (356 байтов).
- Идентификация (`Identification`) выбирается хостом произвольно и используется для одинаковой пометки всех пакетов фрагментированной дейтаграммы.
- Флаги (`Flags`) и смещение фрагмента (`Fragment Offset`) используются для описания фрагментации; флаг 0x4 запрещает фрагментацию.
- Поле времени жизни (`Time to Live`) предотвращает зацикливание пакетов; оно уменьшается на единицу каждым шлюзом; пакет с нулевым временем жизни отбрасывается.
- Поле протокола (`Protocol`) служит для демультиплексирования пакетов: значение 6 соответствует протоколу TCP, значение 17 – UDP, значение 0 – ICMP.

- Контрольная сумма заголовка (`Header Checksum`) представляет собою двоичное дополнение до всех единиц циклической суммы 16 битовых слов заголовка; при вычислении циклической суммы бит переноса суммируется с младшим битом; первоначально само поле контрольной суммы заполняется нулями.
- В простейшем случае опции IP-дейтаграммы отсутствуют.

Таким образом, полученные заголовки могут быть представлены последовательностью октетов (байтов):

```
02 60 8c cd a8 1b 08 00 09 1a cc 6b 08 00 45 00 01 78 12 ab 00 00 43 06 27 ca
c2 7d 10 26 e0 58 89 0f
```

Варианты заданий для самостоятельных упражнений:
1) X: IP=25.125.225.75, MAC=00:00:0c:aa:bb:cc, TTL=10, Protocol=UDP, Data_Length=100; Y: IP=111.222.133.155, MAC=00:00:0f:1a:2b:3c
2) X: IP=94.194.34.244, MAC=00:00:10:a5:c6:b7, Precedence=5, TTL=100, Protocol=TCP, Data_Length=500; Y: IP=33.131.237.215, MAC=00:00:0d:dd:ee:3a
3) X: IP=143.19.74.138, MAC=00:00:c0:88:a6:be, Precedence=3, TTL=200, Protocol=UDP, Data_Length=300; Y: IP=244.166.242.185, MAC=00:aa:00:da:ba:a7
4) X: IP=14.65.243.138, MAC=08:00:20:10:11:c1, TTL=210, Protocol=TCP, Data_Length=1500; Y: IP=47.87.237.219, MAC=08:00:5a:c6:ae:8b

Задача № 3.

Построить последовательность IP-пакетов при фрагментации дейтаграмм:

X: IP=237.163.83.179, длина исходного пакета – 600, TTL=220, Protocol=TCP
Y: IP=92.157.165.18, MRU=256

Решение

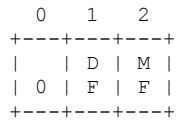
Первоначально построим заголовок нефрагментированной дейтаграммы:

```
45 00 02 6c
1a 2b 00 00
dc 06 xx xx
ed a3 53 b3
5c 9d a5 12
```

Теперь необходимо выбрать длину пакета в пределах MRU, так, чтобы можно было указать смещение фрагмента в 8 байтовых словах. В каждом фрагменте заголовок IP занимает 20 байтов; остаётся 236 байтов; 29*8=232 – ближайшее меньшее число, кратное 8, таким образом, следует фрагментировать на пакеты длиной 252 байта. Всего требуется 3 пакета; из них 2 длиной 252 (232+20) и последний длиной 156 (136+20). Заголовки фрагмента будут различаться лишь полями общей длины, флагов и смещения фрагмента (и контрольной суммы). Укажем лишь первые два 4 байтовых слова заголовков фрагментов:

```
45 00 00 fc
1a 2b 20 00
…
45 00 00 fc
1a 2b 20 1d
…
45 00 00 9c
1a 2b 00 3a
…
```

Заметим, что 252=0xFC, 156=0x9C, 29=0x1D, 29*2=58=0x3A, идентификатор фрагмента выбран произвольно 0x1A2B, он одинаков у всех фрагментов. Значение флага 0x2 указывает наличие следующих фрагментов. Рассмотрим формат поля флагов:



Флаг DF (don't fragment) запрещает фрагментацию; флаг MF (more fragments) указывает наличие следующих фрагментов. Заметим, что поля флагов пересекает границу шестнадцатеричного числа; младший бит шестнадцатеричного числа содержит старший бит смещения. Для небольших смещений он равен нулю, тогда флаг DF представлен числом 0x4, а флаг MF – числом 0x2.

Варианты заданий для самостоятельных упражнений:
1) X: IP=25.125.225.75, TTL=10, Protocol=UDP, Data_Length=1000; Y: IP=111.222.133.155, MRU=400.
2) X: IP=94.194.34.244, Precedence=5, TTL=100, Protocol=TCP, Data_Length=500; Y: IP=33.131.237.215, MRU=300.
3) X: IP=143.19.74.138, Precedence=3, TTL=200, Protocol=UDP, Data_Length=300; Y: IP=244.166.242.185, MRU=100.
4) X: IP=14.65.243.138, TTL=210, Protocol=TCP, Data_Length=1500; Y: IP=47.87.237.219, MRU=500.

Задача № 4.

Для заданной локальной сети построить кадры ARP-запроса и ответа.

X: IP=237.163.83.179, MAC=08:00:09:a1:cc:b6,
Y: IP=92.157.165.18, MAC=02:60:8c:cd:a8:1b

Решение

Заметим, что хосту X первоначально известен лишь IP-адрес хоста Y, а для доставки пакетов, инкапсулированных в кадр Ethernet, требуется указать MAC-адрес назначения. Для определения неизвестного MAC-адреса хост X формирует широковещательный кадр с запросом ARP. Широковещание доставляет кадр хосту Y, который распознаёт в нём свой IP-адрес и посылает прямой ответ с указание собственного MAC-адреса хосту X. Построим ARP-запрос и ARP-ответ в соответствии с форматом, описанном в RFC 826:

Поле	ARP-запрос	ARP-ответ
48.bit: Ethernet address of destination адрес назначения	ffffffffffff	02608ccda81b
48.bit: Ethernet address of sender адрес отправителя	02608ccda81b	080009a1ccb6
16.bit: Protocol type тип	1800	1800
16.bit: (ar$hrd) Hardware address space пространство аппаратных адресов	0001	0001
16.bit: (ar$pro) Protocol address space пространство адресов протокола	0800	0800
8.bit: (ar$hln) byte length hardware address длина аппаратного адреса	06	06
8.bit: (ar$pln) byte length protocol address длина адреса протокола	04	04
16.bit: (ar$op) opcode код операции: 1 – запрос, 2 – ответ	0001	0002

nbytes: (ar$sha) Hardware address of sender аппаратный адрес отправителя	02608ccda81b	080009a1ccb6
mbytes: (ar$spa) Protocol address of sender протокольный адрес отправителя	eda353b3	5c9da512
nbytes: (ar$tha) Hardware address of target аппаратный адрес получателя (0 для запроса)	000000000000	02608ccda81b
mbytes: (ar$tpa) Protocol address of target протокольный адрес получателя	5c9da512	eda353b3

Тогда соответствующие последовательности байтов имеют вид:

ARP-запрос:

```
ff ff ff ff ff ff 0a 0c 0b cd a8 1b 18 00 00 01 08 00 06 04 00 01 0a 0c 0b cd
a8 1b ed a3 53 b3 00 00 00 00 00 00 5c 9d a5 12
```

ARP-ответ:

```
02 60 8c cd a8 1b 08 00 09 a1 cc b6 18 00 00 01 08 00 06 04 00 02 00 0c 00 a1
cc b6 5c 9d a5 12 0a 0c 0b cd a8 1b ed a3 53 b3
```

Варианты заданий для упражнений:
1) X: IP=25.125.225.75, MAC=00:00:77:22:aa:33; Y: IP=111.222.133.155, MAC=08:00:10:ab:cd:ef.
2) X: IP=94.194.34.244, MAC=08:00:69:cc:aa:2b; Y: IP=33.131.237.215, MAC=00:00:1d:ae:ed:e5.
3) X: IP=143.19.74.138, MAC=00:00:6b:9d:a7:c3; Y: IP=244.166.242.185, MAC=08:00:11:ac:77:ed.
4) X: IP=14.65.243.138, MAC=00:00:0f:99:aa:bb; Y: IP=47.87.237.219, MAC=08:00:38:a6:c7:e5.

Контрольные вопросы:
1. Перечислить основные поля заголовка Ethernet.
2. Перечислить основные поля заголовка IP.
3. Перечислить основные поля ARP запроса/ответа.
4. Каким образом ПО стека протоколов определяет, какой пакет инкапсулирован в Ethernet кадр?
5. Каким образом ПО стека протоколов определяет какой протокол транспортного уровня следует использовать при интерпретации IP-дейтаграммы?
6. Для чего необходима фрагментация пакетов?
7. Каким образом задаётся последовательность фрагментов?
8. Как определяется неизвестный MAC-адрес с помощью протокола ARP?

Лабораторная работа №1
Передача IP-трафика в сетях Ethernet

Цель работы: изучить особенности инкапсуляции IP-пакетов в Ethernet-фреймы и отображения IP-адресов на MAC-адреса Ethernet

Подготовка:
- знать структуру заголовка Ethernet-фрейма
- знать структуру заголовка IP-пакета
- знать структуру запросов/ответов протокола ARP
- знать команды работы с ftp-сервером, команды отображения таблиц ARP/RARP

Задание: выполнить трассировку процессов формирования ARP-таблиц и передачи IP-трафика в Ethernet с помощью анализатора трафика

Порядок выполнения работы
1. Запустить анализатор трафика и установить фильтр для отображения ARP-пакетов
2. Включая/выключая другие хосты сети и проверяя их доступность с помощью команды ping выполнить трассировку процессов заполнения APR-таблиц. Отобразить на экране построенные таблицы.
3. Установить фильтр для отображения ftp-трафика.
4. Выполнить трассировку передачи известного файла с ftp-сервера.

Варианты задания: IP, MAC-адреса компьютера, на котором выполняется работа.

Дополнительные требования:
1. Отобразить процесс формирования ARP-таблиц не менее чем для четырёх хостов.
2. Отобразить процесс получения не менее чем четырёх IP-пакетов.
3. Отобразить поля заголовков, задающих взаимодействие протоколов канального - сетевого, сетевого – транспортного уровней при инкапсуляции.
4. Один из фреймов представить в исходной (шестнадцатеричной) форме с интерпретацией всех полей IP и Ethernet-заголовков.

Содержание отчёта:
- структуры заголовков Ethernet-фрейма и IP-пакета
- структура запросов/ответов протокола ARP
- трасса процесса построения ARP таблиц (последовательность запросов/ответов)
- построенные ARP, RARP-таблицы
- трасса процессов передачи IP-пакетов
- полная интерпретация одного из фреймов по шестнадцатеричному представлению

Указания по выполнению работы:
Использовать анализатор пакетов Ethereal либо tcpdump (windump). Выбрать интерфейс (карта Ethernet). Установить требуемый фильтр (arp, ftp) и запустить запись передаваемых фреймов. Остановит процесс записи фреймов. Использовать сохранённые фреймы для написания отчёта.

Команды ARP:

-a: отображает текущие ARP-записи, опрашивая текущие данные протокола
-g: то же что и -a

inet_addr: определяет IP-адрес

-N if_addr: отображает ARP записи для заданного в if_addr сетевого интерфейса
-d: удаляет узел
-s: добавляет узлы и связывает internet адреса с физическими адресами

Задание: Представить не менее 4 пакетов и 1 в hex виде.

Вход в систему:
`login: student04`

```
password: student
startx &
```

В одном окне xterm запускаем команду:

```
sudo ethereal
```

Трафик создается: в другом окне xterm запускаем команду ftp (жирным шрифтом отмечен ввод пользователя. Имя, пароль, и ip-адрес сервера приведены для примера и могут отличаться).

```
ftp 192.168.0.145
Name (192.168.0.145): dmitry
Password: daze
230 User dmitry logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>_
```

После успешной аутентификации на сервере, можно использовать следующие команды:

ftp>**dir** (Просмотр текущего каталога ftp)
ftp>**get *имя_файла*** (Скачать файл с сервера)
ftp>**quit** (Выход из ftp)

Команда оболочки cat выводит содержимое файла указанного в качестве первого аргумента

```
cat rfc903.txt
```

Перед выполнением команды get включаем прослушку Capture → Interface → Capture
После получения файла нажимаем Stop (остановить прослушку и просмотреть результаты)

В hex (16-ричном) виде:

```
0000 00 80 48 67 8d 73 00 16    76 82 3b 3a 00 00 45
0010 00 84 9b 1f 00 00 40 11    37 2b e0 a8 00 0f c0
0020 00 91 03 fc 08 01 00 70    ea 05 00 00 3b 3a 08
0030 00 00 00 00 00 02 00 01    76 82 3b 33 c0 a8
0040 00 04 00 00 00 01 00 00    33 e2 c0 00 00 14
```

В текстовом:

1)

```
0.000000   192.168.0.207 192.168.0.145  TCP 62276>ftp
[syn] seq=0 mss=1460 ws=1 tsv=2395351 tser=0
Frame 1 (78 bytes on wire, 78 bytes captured)
Ethernet II, src: Intel_82:3b:3a(00:16:76:82:3b:3a)
dst: Compex_b7:bd:73(00:80:48:b7:bd:73)
Internet Protocol, Src: 192.168.0.207(192.168.0.207),
Dst: 192.168.0.145 (192.168.0.145)
Transmission Control Protocol, Src Port: 62276 (62276), Dst Port: ftp (21), Seq: 0, Ack: 0, Len:
0
```

2)

```
0.000129  192.168.0.145 192.168.0.207  TCP ftp>62276
[syn, ack] Seq=0 Ack=8 Win=1448 Len=0 TSV=125099355 TSER=72740216
Frame2 (74 bytes on wire, 74 bytes captured)
Ethernet II, Src: Compex_b7:bd:73(00:80:48:b7:bd:73), Dst: Intel_82:3b:3a(00:16:76:82:3b:3a)
Internet Protocol, Src: 192.168.0.145 (192.168.0.145), Dst: 192.168.0.207(192.168.0.207)
Transmission Control Protocol, Src Port: ftp (21), Dst Port: 62276 (62276), Seq: 0, Ack: 8, Len:
0
```

3)

```
Frame 21 (83 bytes on wire, 83 bytes captured)
    Arrival Time: Feb  17, 2007 15:19:50.726006000
    [Time delta from previous packet: 0.005147000 seconds]
    [Time since reference or first frame: 7.438924000 seconds]
    Frame Number: 21
    Packet Length: 83 bytes
    Capture Length: 83 bytes
    [Frame is marked: False]
    [Protocols in frame: eth:ip:tcp:ftp]
    [Coloring Rule Name: TCP]
    [Coloring Rule String: tcp]
Ethernet II, Src: Intel_82:3b:3a(00:16:76:82:3b:3a), Dst: Compex_b7:bd:73(00:80:48:b7:bd:73)
    Destination: Compex_b7:bd:73(00:80:48:b7:bd:73)
        Address: Compex_b7:bd:73(00:80:48:b7:bd:73)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Source: Intel_82:3b:3a(00:16:76:82:3b:3a)
        Address: Intel_82:3b:3a(00:16:76:82:3b:3a)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Type: IP (0x0800)
Internet Protocol, Src: 192.168.0.207(192.168.0.207), Dst: 192.168.0.145 (192.168.0.145)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x10 (DSCP 0x04: Unknown DSCP; ECN: 0x00)
        0001 00.. = Differentiated Services Codepoint: Unknown (0x04)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 69
    Identification: 0x6484 (25729)
    Flags: 0x04 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0x5371 [correct]
    Source: 192.168.0.207(192.168.0.207)
    Destination: 192.168.0.145 (192.168.0.145)
```

4)

```
Frame 22 (76 bytes on wire, 76 bytes captured)
Internet Protocol
    Version: 4
    Header length: 20 bytes
    Total Length: 60
    Flags: 0x04 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0x3e47 [correct]
```

ARP:

```
1. arp -a
   192.168.0.145 at 00:80:48:b7:bd:73 on r10 [ethernet]
   192.168.0.205 at 00:16:76:82:3b:a6 on r10 permanent [ethernet]
2. ping 192.168.0.102
   PING 192.168.0.102 (192.168.0.102) 56(84) bytes of data.
   64 bytes from 192.168.0.102: icmp_seq=1 ttl=63 time=6.45 ms
   64 bytes from 192.168.0.102: icmp_seq=2 ttl=63 time=2.63 ms
   64 bytes from 192.168.0.102: icmp_seq=3 ttl=63 time=2.05 ms
   64 bytes from 192.168.0.102: icmp_seq=4 ttl=63 time=1.93 ms

   --- 192.168.0.102 ping statistics ---
   4 packets transmitted, 4 received, 0% packet loss, time 3004ms
   rtt min/avg/max/mdev = 1.937/3.269/6.459/1.861 ms
```

Выполнение лабораторного задания:

В данной лабораторной работе будем использовать следующие команды.

- arp- a - просмотр имеющихся в таблице IP-адресов.
- ipconfig – определение собственного ip-адреса
- ping – пеленгование («присоединение») чужого ip-адреса.

ARP-запрос/ответ – выглядит следующим образом:

0.000000 Intel _82:3b:85 broadcast
ARP who has 192.168.0.208 Tell 192.168.0.203.

Рассмотрим два фрейма:

1) Frame 1 (*60 bytes*)
Ethernet II: Src:Intel_82:3b:85 (00:16:7b:82:3b:85), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (**REQUEST**)
Type:ARP (0x0806);
Hardware size: 6;
Protocol size: 4;
Opcode: request (0x0001);
Sender MAC-address: Intel_82:3b:85 (00:16:76:82:3b:85);
Sender IP-address: 192.168.0.203 (192.168.0.203);
Target MAC-address: 00:00:00:00:00:00;
Target IP-address: 192.168.0.208 (192.168.0.208);

	Dst	*Src*	*Type ARP*	*Ethernet (0x0001)*
0000	ffffffffffff	001676823b85	0806	0001

	IP (0x0800)	*size (6)*	*size (4)*	*opcode*	*sender MAC-address*	*sender IP-address*
0010	0800	06	04	0001	001676823b85	c0a800cb

	target MAC-address	target IP-address
0020	000000000000	c0a800d0

2) Frame 2 (*60 bytes*)
Ethernet II: Src: Intel_82:3b:85 (00:16:76:82:3b:85), Dst: Intel_82:3b:b4 (00:16:76:82:3b:b4);
Address Resolution Protocol (**REPLY**)
Type:ARP (0x0806);
Hardware size: 6;
Protocol size: 4;
Opcode: reply (0x0002);

Sender MAC-address: Intel_82:3b:b4 (00:16:76:82:3b:b4);
Sender IP-address: 192.168.0.208 (192.168.0.208);
Target MAC-address: Intel_82:3b:85 (00:16:76:82:3b:85);
Target IP-address: 192.168.0.203 (192.168.0.203);

	Dst	*Src*	*Type ARP*	*Ethernet (0x0001)*
0000	001676823b85	001676823bb4	0806	0001

	IP (0x0800)	*size (6)*	*size (4)*	*opcode*	*sender MAC-address*	*sender IP-address*
0010	0800	06	04	0002	001676823bb4	c0a800d0

	target MAC-address	target IP-address
0020	001676823b85	c0a800cb

Практическое занятие № 2
Инкапсуляция IP-PPP (RFC 1548)

Подготовка:

- формат кадра PPP и служебных кадров LCP, IPCP;
- фазы работы протокола PPP, переговоры конфигурирования;
- опции пакетов LCP, аутентификация PAP.

Задачи:

- построение кадров PPP с инкапсуляцией IP дейтаграмм;
- построение последовательностей служебных пакетов LCP, PAP, IPCP;
- анализ дампов работы протокола PPP;
- изучение особенностей конфигурирования интерфейсов PPP в MS Windows и Unix.

Типовые задания:

1. Построить кадр PPP с инкапсуляцией IP дейтаграммы.
2. Построить последовательность LCP пакетов конфигурирования линии.
3. Построить последовательность LCP пакетов завершения связи.
4. Построить последовательность пакетов аутентификации PAP;
5. Построить последовательность IPCP пакетов для получения IP адреса;
6. Выполнить анализ заданного дампа последовательности PPP пакетов.

Примеры решения задач:

Задача № 1.

Построить кадр PPP с инкапсуляцией IP дейтаграммы.

Решение

Protocol	Information	Padding
0x0021	IP-пакет	Заполнитель

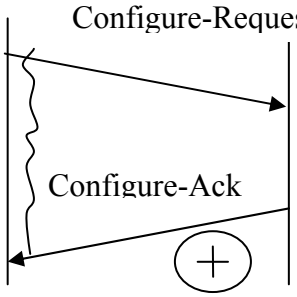
Задача № 2.

Построить последовательность LCP пакетов конфигурирования линии:
- параметры соединения согласованы (Configure–Ack),
- параметры соединения частично согласованы (Configure–Nak),

- параметры соединения не согласованы (Configure–Reject).

Решение

Первое взаимодействующее устройство формирует запрос с параметрами соединения (Configure–Request). В протоколе PPP имеется набор стандартных установок, действующих по умолчанию и учитывающих все стандартные конфигурации. Взаимодействующие устройства могут использовать эти установки или описать свои возможности и требования. На основании этой информации принимаются параметры соединения, устраивающие обе стороны. В этом случае процесс переговоров состоит из:



У них один идентификатор

Запрос с параметрами соединения:

Код LCP	Код Configure-Request	Идентификатор	Длина	Параметрами соединения
C021	01	Identifier	Length	Options

Ответ, принимающий параметры соединения:

Код LCP	Код Configure-Ack	Идентификатор	Длина	Параметрами соединения
C021	02	Identifier	Length	Ack Options

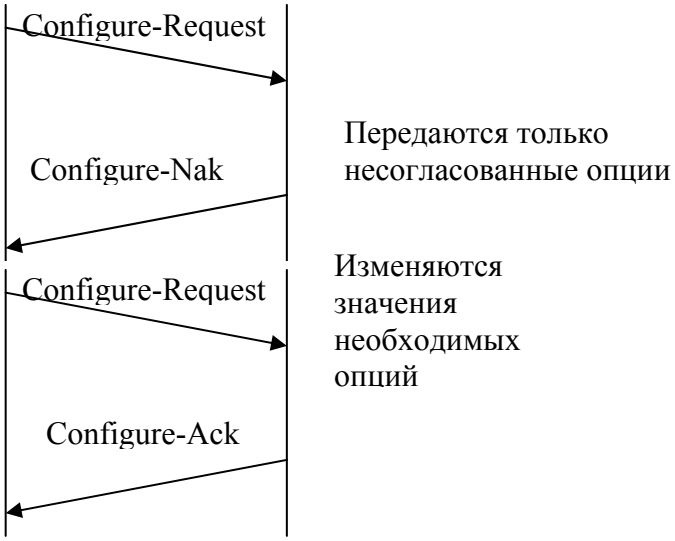
Представление запроса в 16-м коде:
c0 21 01 00 00 17 02 06 00 00 00 00 05 06 34 8f 1f 1a 07 02 08 02 0d 03 06, где
c021 – код LCP
01 – код Configure-Request
00 – идентификатор
0017 – длина
Опции:

Тип	Длина	Данные	
02	06	00 00 00 00	– Async-Control-Character map (операция для передачи асинхронного управления символов)
05	06	34 8f 1f 1a	– Magic-Number (магическое число)
07	02		– Protocol Field - Compression
08	02		– Address-and-control-Field-Compression
0d	03	06	– call-back (независимая опция)

Представление ответа в 16-м коде:
c0 21 02 00 00 14 02 06 00 00 00 00 05 06 34 8f 1f 1a 07 02 08 02
c021 – код LCP
02 – код Configure-Ack
00 – идентификатор
0014 – длина

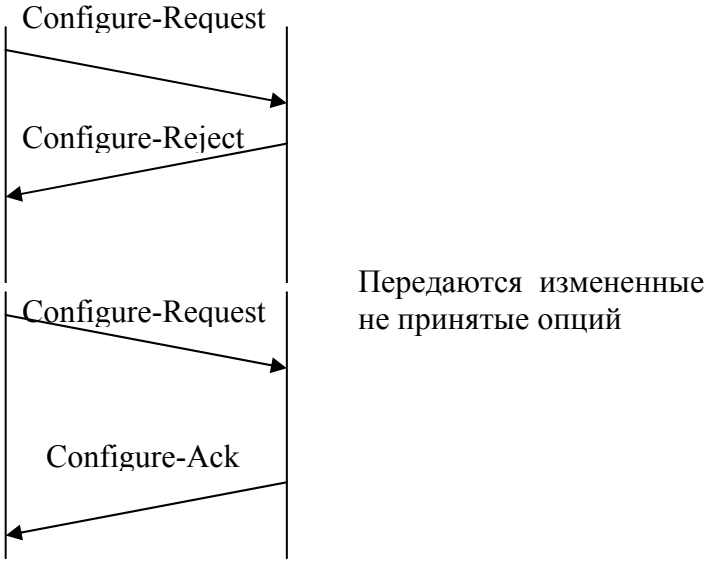
Следующий вариант переговоров происходит при адаптации требований одной стороны к требованиям другой, для этого используется запрос Configure-Nak (код 03). Переговорная

процедура может завершиться согласованием параметров, либо по истечении тайм-аута закончиться безрезультатно.



Представление процедуры согласования в 16-м коде:
c0 21 01 00 00 1b 01 04 05 f4 02 06 00 00 00 00 05 06 34 8f 1f 1a 07 02 08 02 0d 03 06
c0 21 03 00 00 0b 01 04 05 04
c0 21 01 01 00 1b 01 04 05 04 02 06 00 00 00 00 05 06 34 8f 1f 1a 07 02 08 02 0d 03 06
c0 21 02 01 00 1b 01 04 05 04 02 06 00 00 00 00 05 06 34 8f 1f 1a 07 02 08 02 0d 03 06

Рассмотрим вариант переговоров, когда параметры соединения не согласованы, в этом случае используется запрос Configure-Reject (код 04).



Представление процедуры переговоров в 16-м коде:
c0 21 01 00 00 17 02 06 00 00 00 00 05 06 34 8f 1f 1a 07 02 08 02 0d 03 06
c0 21 04 00 07 0d 03 06
c0 21 01 01 00 18 02 06 00 00 00 00 03 04 c0 23 05 06 f6 dc 63 30 07 02 08 02
c0 21 02 01 00 18 02 06 00 00 00 00 03 04 c0 23 05 06 f6 dc 63 30 07 02 08 02

Варианты заданий для самостоятельных упражнений:
1) Составьте запрос с параметром соединения Authentication-Protocol (протокол аутентификации) PAP.
2) Составьте запросы согласования параметра соединения Maximum-Receive-Unit (MRU).
3) Составьте запрос с параметром соединения Quality-Protocol.

Задача № 3.

Построить последовательность LCP пакетов завершения связи.

Решение

Для завершения связи между взаимодействующими устройствами используются два типа запроса: Termination-Request (код 05) – запрос на разрыв связи, Termination-Ack (код 06) – подтверждение на окончание соединения.

Представление процедуры завершения связи в 16-м коде:
c0 21 05 08 00 10 34 8f 1f 1a 00 3c cd 74 00 00 00 00
c0 21 06 08 00 10 34 8f 1f 1a 00 3c cd 74 00 00 00 00

Задача № 4.

Построить последовательность пакетов аутентификации PAP.

Решение

Режим аутентификации устройства (пользователя) в PPP-соединении является одним из важных параметров. Для целей аутентификации по умолчанию используется протокол аутентификации по паролю (PAP, код 0xC023), передающий пароль по линиям связи в открытом виде. Процедура аутентификации с использованием протокола PAP имеет следующую последовательность запросов:
Authenticate-Request (код 0x01), передаются имя пользователя (ID) и длина ID, пароль и длина пароля. В случае идентификации пользователя приходит подтверждающий ответ Authenticate-Ack (код 0x02); частичная идентификация, не принято имя пользователя или пароль, Authenticate-Nak (код 0x03); Authenticate-Reject (код 0x04) все данные не приняты.

Представление режима аутентификации в 16-м коде:
c0 23 01 01 00 13 05 69 67 70 72 73 08 69 6e 74 65 72 6e 65 74 – Authenticate-Request
c0 23 02 01 00 04 – Authenticate-Ack, где
c0 23 – протокол PAP
01 – Authenticate-Request
01 – идентификатор
00 13 – общая длина запроса
05 – длина имени пользователя
69 67 70 72 73 – имя пользователя
08 – длина пароля
69 6e 74 65 72 6e 65 74 – пароль пользователя

Варианты заданий для самостоятельных упражнений:
1) Составьте запрос Authenticate-Request с параметрами: имя_пользователя=car, пароль=bmw, используя протокол аутентификации PAP.
2) Составьте ответ Authenticate-Nak, не принятый параметр пароль_пользователя.
3) Составьте ответ Authenticate-Reject.

Задача № 5.

Построить последовательность IPCP пакетов для получения IP адреса;

Решение

Для получения IP адреса устройства и IP адресов серверов используется протокол IPCP (код 0x8021). Как и в предыдущих процедурах, последовательность пакетов состоит из

запроса Configure-Request и ответа Configure-Ack (или Configure-Nak, Configure- Reject). В запросе используются следующие опции:
0x01 IP адрес,
0x02 IP протокол компрессии,
0x03 IP адрес, состоящий из длины адреса (length) и IP адреса (IP-address),
0x81 IP адрес основного DNS сервера,
0x82 IP адрес основного NBNS сервера,
0x83 IP адрес дополнительного DNS сервера,
0x84 IP адрес дополнительного NBNS сервера.

Представление последовательность IPCP пакетов для получения IP адреса в 16-м коде:
80 21 01 04 00 22 03 06 00 00 00 00 81 06 00 00 00 00 82 06 00 00 00 00 83 06 00 00 00 00 84 06 00 00 00 00
80 21 02 04 00 16 03 06 0a 0a 5b fc 81 06 c1 29 3c 16 83 06 c1 29 3c 12

Варианты заданий для самостоятельных упражнений:
1) Составьте ответ Configure-Nak, в котором передаются опции с кодами 0x03, 0x81, 0x83.
2) Составьте ответ Configure-Reject, в котором не согласованы опции с кодами 0x81, 0x83.

Задача № 6.

Выполнить анализ заданного дампа последовательности PPP пакетов.

Решение

Для описания последовательности PPP пакетов в приведенном дампе используем навыки и знания, полученные при решении задач 1-5.

Представление дампа из четырех запросов в 16-м коде:
1)c0 21 01 02 00 32 02 06 00 00 00 00 05 06 74 fd 35 8b 07 02 08 02 0d 03 06 11 04 06 4e 13 17 01 07 62 cd af a3 ac 40 b2 bb 28 db 1f 4d af 66 d7 00 00 00 00
2)c0 21 04 02 00 08 11 04 06 4e
3)c0 21 01 03 00 2e 02 06 00 00 00 00 05 06 74 fd 35 8b 07 02 08 02 0d 03 06 13 17 01 07 62 cd af a3 ac 40 b2 bb 28 db 1f 4d af 66 d7 00 00 00 00
4)c0 21 02 03 00 2e 02 06 00 00 00 00 05 06 74 fd 35 8b 07 02 08 02 0d 03 06 13 17 01 07 62 cd af a3 ac 40 b2 bb 28 db 1f 4d af 66 d7 00 00 00 00
Первый запрос Configure-Request на установление параметров соединения по протоколу LCP. Второй запрос Configure-Reject, не принятая опция MRRU. Третий запрос вторая попытка на установление параметров соединения по протоколу LCP, при этом параметр MRRU не задается. Четвертый запрос Configure-Ack, параметры соединения согласованы.

Варианты заданий для самостоятельных упражнений:
1) Выполните анализ следующего дампа:
c0 21 0a 04 00 12 74 fd 35 8b 4d 53 52 41 53 56 35 2e 30 30
80 21 01 01 00 10 02 06 00 2d 0f 01 03 06 51 19 e0 29
80 21 02 01 00 10 02 06 00 2d 0f 01 03 06 51 19 e0 29
2) Выполните анализ заданной последовательности PPP пакетов:
80 21 01 08 00 1c 02 06 00 2d 0f 01 03 06 00 00 00 00 81 06 00 00 00 00 83 06 00 00 00 00
80 21 03 08 00 16 03 06 c0 a8 0c 20 81 06 51 19 e0 01 83 06 51 19 e0 12
80 21 01 09 00 1c 02 06 00 2d 0f 01 03 06 c0 a8 0c 20 81 06 51 19 e0 01 83 06 51 19 e0 12
80 21 02 09 00 1c 02 06 00 2d 0f 01 03 06 c0 a8 0c 20 81 06 51 19 e0 01 83 06 51 19 e0 12

Контрольные вопросы:
1. Перечислить основные поля пакета PPP.
2. Перечислить основные фазы PPP-соединения.
3. Перечислить основные запросы для установления соединения.
4. Перечислить основные опции, определяемые для установления соединения.
5. Какие протоколы аутентификации используются в PPP-соединении?
6. Каким образом происходит присвоение IP адреса устройству?
7. Как осуществляется передача информации (данных) в PPP-соединении?
8. Какова процедура завершения в PPP-соединения?

Лабораторная работа № 2
Передача IP-трафика по выделенным линиям

Цель работы: изучить особенности инкапсуляции IP-пакетов в PPP-кадры, работу средств управления линией LCP и конфигурирования сетевых протоколов NCP (IPCP)

Подготовка:
- знать структуру кадра PPP и служебных пакетов LCP, IPCP;
- знать процедуру переговоров LCP для конфигурирования линии;
- знать основные опции пакетов LCP;
- знать основные опции PAP и IPCP.

Задание: выполнить трассировку процессов конфигурирования линии связи, аутентификации, конфигурирования интерфейсов IP, передачи IP-трафика и разъединения протокола PPP с помощью анализатора трафика

Порядок выполнения работы
1. Запустить анализатор трафика, выбрать интерфейс и установить фильтр для отображения PPP-пакетов.
2. Активизировать линию связи (включить модемы выделенной линии либо выполнить дозвон по коммутируемой линии).
3. Выполнить передачу известного файла с ftp-сервера.
4. Остановить запись пакетов и сохранить записанную информацию в файле.
5. Проанализировать последовательность PPP пакетов со схематическим представлением переговоров LCP, PAP, IPCP.

Варианты задания: IP-адрес компьютера, на котором выполняется работа; особенности текущего сеанса протокола PPP (возможно использование собственной трассы).

Дополнительные требования:
1. Отобразить переговоры конфигурирования LCP, PAP, IPCP.
2. Отобразить процесс получения не менее восьми LCP пакетов.
3. Отобразить переговоры завершения связи.
4. Один из LCP пакетов, содержащий не менее 4 опций, представить в исходной (шестнадцатеричной) форме с интерпретацией всех полей.

Содержание отчёта:
- структура кадра PPP, а также использованных в работе LCP, PAP, IPCP пакетов;
- структура использованных в работе опций LCP, PAP, IPCP;
- сохранённая трасса последовательности PPP пакетов;
- схематическое представление переговоров конфигурирования LCP, PAP, IPCP;

- полная интерпретация одного из LCP пакетов по шестнадцатеричному представлению.

Указания по выполнению работы:
Использовать анализатор пакетов Ethereal либо tcpdump (windump). Выбрать интерфейс (PPP interface, dialup adapter). Установить требуемый фильтр (PPP) и запустить запись передаваемых кадров. Остановит процесс записи кадров. Использовать сохранённые кадры для написания отчёта.

Практическое занятие № 3
Особенности работы протоколов транспортного и сеансового уровней

Подготовка:
- формат заголовков TCP и UDP;
- процедура установления TCP соединения ;
- основные опции сегмента TCP для передачи данных;
- процедура завершения TCP соединения.

Задачи:
- построение дейтаграмм UDP;
- построение пакетов TCP с инкапсуляцией IP дейтаграмм;
- построение последовательностей пакетов конфигурирования, передачи и завершения TCP-соединения;
- анализ дампов работы протокола TCP.

Типовые задания:
1. Построить дейтаграмму UDP.
2. Отобразить TCP пакет в шестнадцатеричной форме с интерпретацией всех полей.
3. Построить последовательность TCP пакетов процедуры переговоров конфигурирования соединения.
4. Отобразить процесс передачи TCP пакетов.
5. Построить переговоры завершения связи TCP соединения.

Примеры решения задач:

Задача № 1.

Построить дейтаграмму UDP.

Решение

Дейтаграмма UDP состоит из заголовка и поля данных. Заголовок UDP состоит из четырех двухбайтовых полей: номера порта отправителя (Source Port), номера порта получателя (Destination Port), длины дейтаграммы (Total length), контрольной суммы (Checksum).

Представление дейтаграммы UDP в 16-м коде:
00 35 04 79 00 9c 17 ff 0c e6 85 80 00 01 00 01 00 03 00 03 03 77 77 77 03 73 6b 79 02 64 02 75 61 00 00 01 00 01 c0 0c 00 01 00 01 00 01 51 80 00 04 51 19 e0 04 c0 10 00 02 00 01 01 51 80 00 05 02 6e 73 c0 10 c0 10 00 02 00 01 00 01 51 80 00 06 03 6e 73 31 c0 10 c0 10 00 02 00 01 00 01 51 80 00 06 03 6e 73 32 c0 10 c0 3b 00 01 00 01 00 01 51 80 00 04 51 19 e0 01 c0 4c 00 01 00 01 00 01 51 80 00 04 c3 05 23 b2 c0 5e 00 01 00 01 00 01 51 80 00 04 51 19 e0, где

00 35 – номер порта источника, в данном примере сообщение DNS-сервера,

04 79 – номер порта получателя,
00 9c – общая длина дейтаграммы,
17 ff – контрольная сумма, и далее сообщение DNS-сервера в 16-м коде.

Варианты заданий для самостоятельных упражнений:
1) Выполните анализ следующего дампа:
00 44 00 43 01 34 04 72 01 08 06 00 00 5e f0 06 06 00 00 00 c0 a8 0c 3b …
2) Составьте дейтаграмму UDP, порт получателя которой равен 69 и длина поля данных равна 40 байтам.

Задача № 2.

Отобразить TCP пакет в шестнадцатеричной форме с интерпретацией всех полей.

Решение

Пакет TCP состоит из заголовка и поля данных. Заголовок содержит следующие основные поля:

Порт источника	source port	2 байта
Порт приемника	destination port	2 байта
Последовательный номер	sequence number	4 байта
Подтвержденный номер	acknowledgement number	4 байта
Длина заголовка	hlen	4 бита
Резерв	reserved	6 бит
Кодовые биты	code bits	6 бит
Окно	window	2 байта
Контрольная сумма	checksum	2 байта
Указатель срочности	urgent pointer	2 байта
Параметры	options	переменная длина
Заполнитель	padding	переменная длина

Представление пакета TCP в 16-м коде:
04 7e 00 50 85 c9 61 3a bb 8e 9e d2 50 10 22 38 49 68 00 00, где
04 7e – порт источника (1150)
00 50 – порт приемника (80)
85 c9 61 3a – последовательный номер (329)
bb 8e 9e d2 – подтвержденный номер (5841)
50 – длина заголовка (20 байт)
10 – кодовые биты (квитанция на принятый сегмент Ack=1)
22 38 – размер окна
49 68 – контрольная сумма
00 00 – заполнитель

Варианты заданий для самостоятельных упражнений:
1) Интерпретируйте поля следующего дампа пакета TCP:
04 83 00 50 85 d5 1d 5d 00 00 00 00 70 02 22 38 ba e0 00 00 02 04 05 b4 01 01 04 02.
2) Интерпретируйте поля следующего дампа пакета TCP:
00 50 04 80 bc 26 5f 27 85 ce f4 3c 50 11 19 20 fe 88 00 00.
3) Составьте пакет TCP: порт источника 21, получателя 1045, последовательный номер 45, подтвержденный номер 112, квитанция на принятый сегмент, окно 834, остальные поля произвольные.
4) Составьте дейтаграмму UDP, порт получателя которой равен 69 и длина поля данных равна 40 байтам.

Задача № 3.

Построить последовательность TCP пакетов процедуры переговоров конфигурирования соединения.

Решение

При установлении логического соединения модули TCP договариваются между собой о параметрах процедуры обмена. При конфигурировании соединения стороны обмениваются параметрами, такими как:

- начальный порядковый номер, с которого начинается отсчет передачи данных;
- максимальный объем данных, который разрешается передавать другой стороне, если еще не получена квитанция на предыдущую порцию данных;
- максимальный размер сегмента.

В процедуре установления соединения (three-way handshake или «тройное рукопожатие») используются кодовые биты Syn и Ack. Построим последовательность TCP пакетов логического соединения модулей X и Y.

X -> Y
Src Port: 1035, Dst Port: http (80),
Sequence number (Sn): 0 (relative sequence number)
Header length: 28 bytes
Flags: 0x0002 (Syn)
Window size: 8760
Checksum 0xe330[correct]
Options (4 bytes):
Maximum segment size (MSS): 1460 bytes

X <- Y
Src Port: http (80), Dst Port: 1035,
Sequence number: 0 (relative sequence number)
Acknowledgment number (An): 1 (relative ack number)
Header length: 28 bytes
Flags: 0x0012 (Syn, Ack)
Window size: 5840
Checksum: 0x1c72 [correct]
Options (4 bytes):
Maximum segment size: 576 bytes

X -> Y
Src Port: 1035, Dst Port: http (80),
Sequence number: 1 (relative sequence number)
Acknowledgment number: 1 (relative ack number)
Header length: 20 bytes
Flags: 0x0010 (Ack)
Window size: 8760
Checksum 0x3dce [correct]

Варианты заданий для самостоятельных упражнений:
Постройте последовательность TCP пакетов процедуры соединения:
1) Для X параметры соединения: Sequence number=300, Window size=6556, MSS=1240, для Y из примера.
2) Для Y параметры соединения: Sequence number=100, Window size=5840, MSS=1460, для X из примера.

3) Параметры X -> Y: Src Port: 1047, Dst Port: http (80), Sn: 907, Header length: 28 bytes, Flags: 0x0002 (Syn), Window size: 8760, Checksum 0xe330[correct], Options (4 bytes): Maximum segment size (MSS): 1460 bytes.
Напишите параметры X <- Y.
4) Параметры X <- Y: Src Port: http (80), Dst Port: 1047, Sn: 100, An:100, Header length: 28 bytes, Flags: 0x0012 (Syn, Ack), Window size: 8760, Options (4 bytes): Maximum segment size (MSS): 1790 bytes.
Напишите параметры X -> Y.

Задача № 4.

Отобразить процесс передачи TCP пакетов.

Решение

В рамках установленного соединения происходит процесс передачи данных. Правильность передачи сегмента подтверждается квитанцией от получателя с указанием номера принятого сегмента (подтверждающий номер), на единицу превышающий максимальный номер байта полученного сегмента. В протоколе TCP в одном и том же сегменте могут быть помещены и квитанция, и данные, которые приложение передает другой стороне.
Построим последовательность TCP пакетов процесса передачи данных приложений X и Y.
X -> Y (передача 169 байт)
Src Port: 1035, Dst Port: http (80),
Sequence number: 1 (relative sequence number)
[Next sequence number: 170 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
Header length: 20 bytes
Flags: 0x0018 (Psh, Ack)
Window size: 8760
Checksum 0xa6a0 [correct]
Hypertext Transfer Protocol

X <- Y (квитанция на получение 169 байт)
Src Port: http (80), Dst Port: 1035,
Sequence number: 1 (relative sequence number)
Acknowledgment number: 170 (relative ack number)
Header length: 20 bytes
Flags: 0x0010 (Ack)
Window size: 6432
Checksum: 0x463d [correct]

X <- Y (передача 290 байт)
Src Port: http (80), Dst Port: 1035,
Sequence number: 1 (relative sequence number)
[Next sequence number: 291 (relative sequence number)]
Acknowledgment number: 170 (relative ack number)
Header length: 20 bytes
Flags: 0x0018 (Psh, Ack)
Window size: 6432
Checksum: 0x6304 [correct]
TCP segment data

X -> Y (квитанция на получение 290 байт и передача 190 байт)
Src Port: 1035, Dst Port: http (80),
Sequence number: 170 (relative sequence number)
[Next sequence number: 360 (relative sequence number)]
Acknowledgment number: 291 (relative ack number)
Header length: 20 bytes
Flags: 0x0018 (Psh, Ack)
Window size: 8470
Checksum 0xf4d6 [correct]
Hypertext Transfer Protocol

X <- Y (квитанция на получение 190 байт и передача 291 байт)
Src Port: http (80), Dst Port: 1035,
Sequence number: 291 (relative sequence number)
[Next sequence number: 582 (relative sequence number)]
Acknowledgment number: 360 (relative ack number)
Header length: 20 bytes
Flags: 0x0018 (Psh, Ack)
Window size: 7504
Checksum: 0x312b [correct]
TCP segment data

Варианты заданий для самостоятельных упражнений:
1) Постройте последовательность передачи TCP пакетов: X передает Y 40 байт информации, размер окна 8990; Y передает X квитанцию на получение 40 байт и 120 байт информации, X передает Y квитанцию на получение 120 байт информации, начальные значения Sn и Ackn произвольные.
2) Текущие значения для X Sn=20 и Ackn=60, X передает Y 70 байт, Y передает X 58 байт. Какие значения принимают параметры Sn и Ackn для X и Y.
3) Текущие значения для Y Sn=120 и Ackn=1. Постройте последовательность передачи TCP пакетов: Y передает X 10 байт информации; X передает Y квитанцию на получение 10 байт и 80 байт информации, Y передает X квитанцию на получение 80 байт информации и 8 байт информации, X передает Y квитанцию на получение 8 байт информации.

Задача № 5.

Построить переговоры завершения связи TCP соединения.

Решение

Завершение TCP соединения состоит из последовательности сообщений, которыми обмениваются участники соединения. Признаком завершения соединения (или достижения передающей стороной последнего байта передаваемых данных) является наличие кодового бита FIN. Процесс завершения представлен следующими сообщениями:
X <- Y (квитанция на получение данных и признак завершения передачи данных)
Src Port: http (80), Dst Port: 1042,
Sequence number: 5578 (relative sequence number)
Acknowledgment number: 285 (relative ack number)
Header length: 20 bytes
Flags: 0x0011 (Ack, Fin)
Window size: 65535
Checksum: 0x5838 [correct]

X -> Y (квитанция на получение признака завершения - Acknowledgment number увеличился на единицу)
Src Port: 1042, Dst Port: http (80),
Sequence number: 285 (relative sequence number)
Acknowledgment number: 5579 (relative ack number)
Header length: 20 bytes
Flags: 0x0010 (Ack)
Window size: 7892
Checksum 0x3964 [correct]

X -> Y (передача признака завершения)
Src Port: 1042, Dst Port: http (80),
Sequence number: 285 (relative sequence number)
Acknowledgment number: 5579 (relative ack number)
Header length: 20 bytes
Flags: 0x0011 (Ack, Fin)
Window size: 7892
Checksum 0x3963 [correct]

Варианты заданий для самостоятельных упражнений:
Постройте последовательность передачи TCP пакетов:
1) Для X параметры соединения: Sequence number=300, Window size=6556, MSS=1240.
2) Для X параметры соединения: Sequence number=500, Window size=3000, MSS=500.

Контрольные вопросы:
1. Перечислите основные поля пакета TCP.
2. Перечислите основные поля дейтаграммы UDP.
3. Перечислить основные фазы TCP-соединения.
4. Какие параметры определяются в процессе установления соединения?
5. Чем идентифицируется логическое TCP-соединение?
6. Опишите работу алгоритма скользящего окна.
7. Для чего служит окно приема?

Лабораторная работа № 3
Передача информации посредством протокола TCP

Цель работы: изучить особенности инкапсуляции UDP дейтаграмм и TCP сегментов в IP-пакеты, процедуры установления соединения («тройное рукопожатие») и параметров передачи данных .

Подготовка:
- знать структуру заголовков TCP и UDP;
- знать процедуру установления TCP соединения ;
- знать основные опции сегмента TCP для передачи данных;
- знать процедуру завершения TCP соединения.;

Задание: выполнить трассировку процессов конфигурирования линии связи, передачи IP-трафика и разъединения протокола TCP с помощью анализатора трафика.

Порядок выполнения работы

1. Запустить анализатор трафика, выбрать интерфейс и установить фильтр для отображения TCP пакетов.
2. Активизировать линию связи (включить модемы выделенной линии либо выполнить дозвон по коммутируемой линии).
3. Выполнить передачу известного файла с ftp-сервера.
4. Остановить запись пакетов и сохранить записанную информацию в файле.
5. Проанализировать последовательность TCP пакетов со схематическим представлением переговоров (установления соединения, передача данных, завершение соединения).

Варианты задания: IP-адрес компьютера, на котором выполняется работа; оособенности текущего сеанса протокола TCP (возможно использование собственной трассы).

Дополнительные требования:
1. Отобразить переговоры конфигурирования TCP.
2. Отобразить процесс получения не менее восьми TCP пакетов.
3. Отобразить переговоры завершения связи.
4. Один из TCP пакетов представить в исходной (шестнадцатеричной) форме с интерпретацией всех полей.

Содержание отчёта:
- структура сегмента TCP, перечень обслуживаемых прикладных служб;
- структура сегмента UDP, перечень обслуживаемых прикладных служб;
- описание использованных в работе опций TCP;
- сохранённая трасса последовательности TCP пакетов;
- схематическое представление переговоров конфигурирования TCP (процедур соединения, передачи данных, завершения соединения, с указанием использованных кодовых битов);
- полная интерпретация одного из TCP пакетов по шестнадцатеричному представлению.
- полная интерпретация одной из UDP дейтаграмм по шестнадцатеричному представлению.

Указания по выполнению работы:
Использовать анализатор пакетов Ethereal либо tcpdump (windump). Выбрать интерфейс (TCP interface, dialup adapter). Установить требуемый фильтр (TCP) и запустить запись передаваемых пакетов. Остановит процесс записи пакетов. Использовать сохранённые пакеты для написания отчёта.

Примеры анализа пакетов:

Frame 31.

Transmission control protocol (TCP)

- Src Port: 1158, Dst Port: http (80),
- Sequence number: 0 (relative sequence number)
- Header length: 28 bytes
- Flags:0_x0002 (Syn)
- Window size: 16384
- Checksum 0_x91b6 [correct]
- Options (8 bytes):
    1. Maximum segment size: 1460 bytes
    2. Nop
    3. Sack permitted.

Frame 32.

Transmission control protocol (TCP)

- Src Port: http (80), Dst Port: 1158,
- Sequence number: 0 (relative sequence number)
- Acknowledgment number: 1 (relative ack number)
- Header length: 24 bytes
- Flags:0_x0012 (Syn, Ack)
- Window size: 32768
- Checksum: 0_x21be [correct]
- Options (4 bytes):
    1. Maximum segment size: 576 bytes.

Frame 33.

Transmission control protocol (TCP)

- Src Port: 1158, Dst Port: http (80),
- Sequence number: 1 (relative sequence number)
- Acknowledgment number: 1
- Header length: 20 bytes
- Flags:0_x0010 (Ack)
- Window size: 16704
- Checksum: 0_x74c7 [correct]

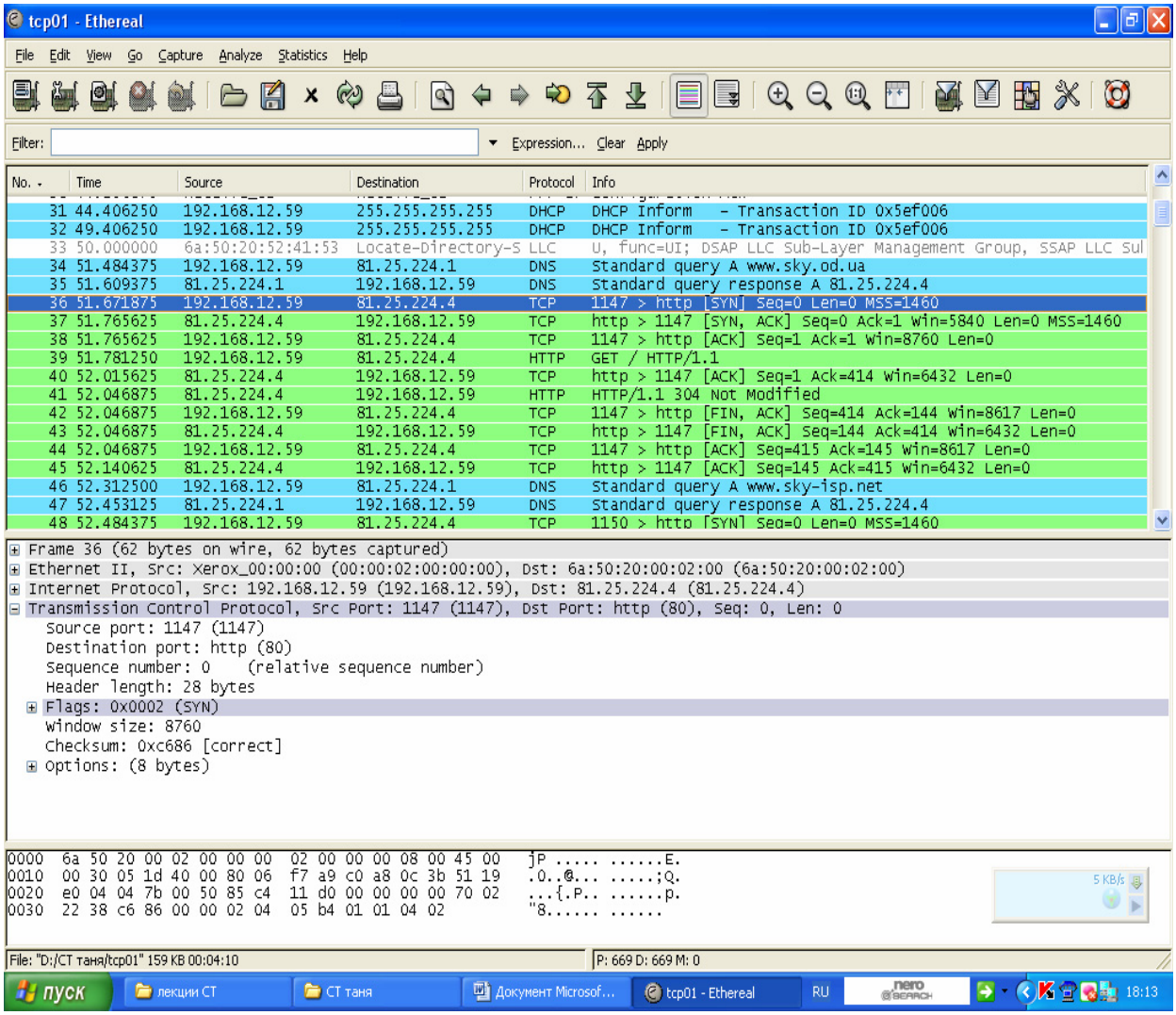
Frame 35.

Transmission control protocol (TCP)

- Src Port: http (80), Dst Port: 1158,
- Sequence number: 1 (relative sequence number)
- Acknowledgment number: 1 (relative ack number)
- Header length: 20 bytes
- Flags:0_x0010 (Ack)
- Window size: 33030
- Checksum: 0_x32db [correct].

03 00 03 00 00 00 Destination Address	56 88 20 00 03 00 Source Address	08 00 Type IP	45 Version, Length	00 Dif. Service Field
00 28 Total Length	Cc 02 Identification	10 Flags	00 Fragment offset	32 TTL
06 Protocol	62 b3 Header Checksum	C243391a Source	0a 0a 54 b3 Destination	00 50 Source Port
04 86 Dst Port	08 9a 3f c9 Sequence number	F11663 88 Ack.number	50 Header length	10 Flags

Пример образа экрана Ethereal:



Практическое занятие № 4
Организация коммутируемых сетей Ethernet

Цель: освоить алгоритмы построения основных структур данных, обеспечивающих функционирование коммутируемых сетей Ethernet: таблиц коммутации, покрывающих деревьев.

Подготовка:

- алгоритм покрывающего дерева;
- алгоритм работы коммутатора;
- алгоритм построения динамических таблиц коммутации.

Задачи:

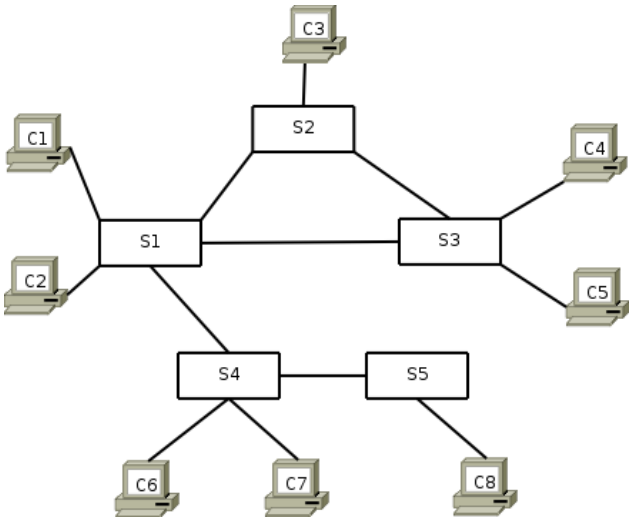
1. Алгоритм покрывающего дерева.
2. Построение статических таблиц коммутации.
3. Трассировка процессов построения динамических таблиц коммутации.
4. Трассировка доставки кадров в сети.

Вариант *N*3

Обозначения:

S – коммутаторы (switch)

C – компьютеры



Задача *N*1

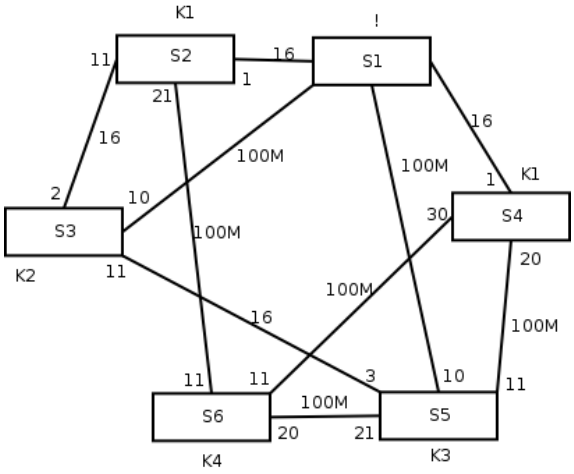
Построить покрывающее дерево STA, STP

а) для микросегментированной Ethernet

Веса:

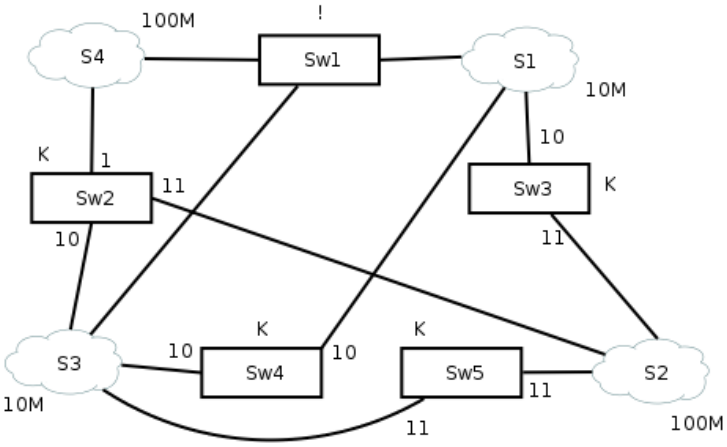
16 – 1

100M – 10

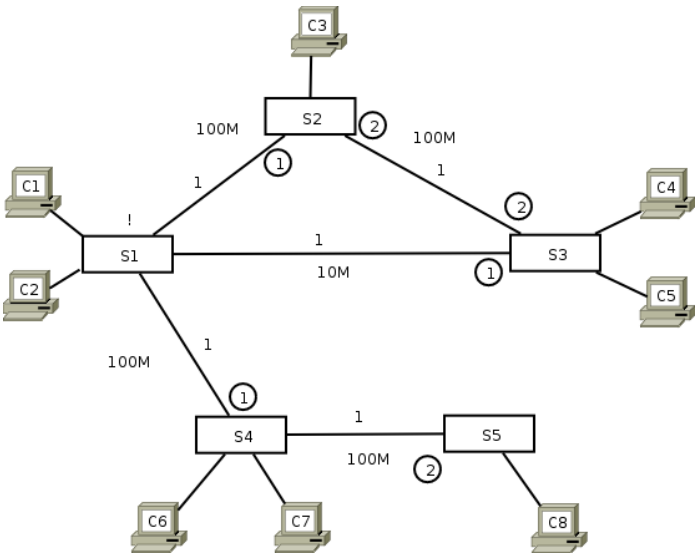


1. Определить корневой коммутатор → S1
   Находим веса (по min)
2. Выбираем корневые порты
3. Генерируем дерево

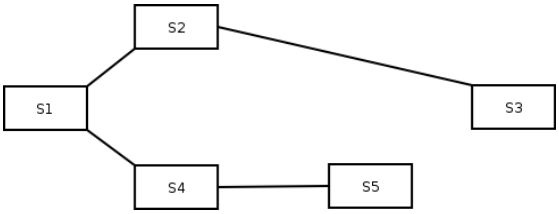
б) Построить покрывающее дерево для сети Ethernet



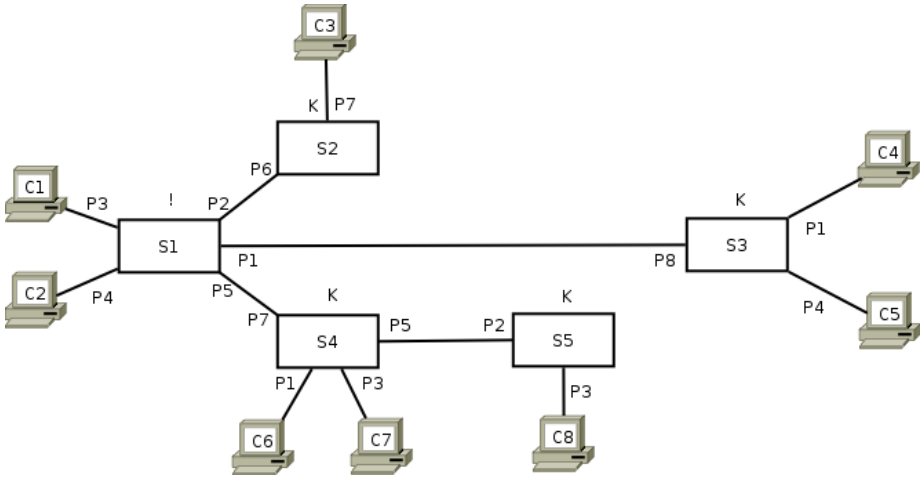
назначим каждому сегменту вид команды и вид порта:



Дерево:



Структурная схема древовидной сети:



S1		S2		S3		S4		S5	
MAC	Port	MAC	Port	MAC	Port	MAC	Port	MAC	Port
1	P3	1	P6	1	P8	1	P7	1	P2
2	P4	2	P6	2	P8	2	P7	2	P2
3	P2	3	P7	3	P8	3	P7	3	P2
4	P1	4	P6	4	P1	4	P7	4	P2
5	P1	5	P6	5	P4	5	P7	5	P2
6	P5	6	P6	6	P8	6	P1	6	P2
7	P5	7	P6	7	P8	7	P3	7	P2
8	P5	8	P6	8	P8	8	P5	8	P3

Выполним трассировку динамического заполнения таблиц коммутации в процессе доставки следующих фреймов:

C1 → C6

	→ P1	→ P1 → принимает
	→ P2	→ P2
	→ P3	→ P3
C1(S1,P3) →	→ P4	→ P4
	→ P5 (S4, P7) →	→ P5
	→ P6	→ P6
	→ P7	→ P7
	→ P8	→ P8

C2 → C3

	→ P1	→ P1
	→ P2 (S2, P6) →	→ P2
C2(S1,P4) →	→ P3	→ P3
	→ P4	→ P4
	→ P5	→ P5
	→ P6	→ P6
	→ P7	→ P7→ принимает
	→ P8	→ P8

C1 → C8

	→ P1	→ P1	→ P1
C1(S1,P3) →	→ P2	→ P2	→ P2
	→ P3	→ P3	→ P3→ принимает
	→ P4	→ P4	→ P4
	→ P5 (S4, P7) →	→ P5 (S5,P2) →	→ P5
	→ P6	→ P6	→ P6
	→ P7	→ P7	→ P7
	→ P8	→ P8	→ P8

C6 → C7

	→ P1
	→ P2
C6(S4,P1) →	→ P3 → принимает
	→ P4
	→ P5
	→ P6
	→ P7
	→ P8

S1		S2		S3		S4		S5	
MAC	Port	MAC	Port	MAC	Port	MAC	Port	MAC	Port
1	3	1	6	1	8	1	7	1	2
2	4	2	6	2	8	2	7	2	2
6	5	6	6	6	8	6	1	6	2

Контрольные вопросы:
1. Перечислить основные поля кадра Ethernet.
2. Описать структуру таблицы коммутации.
3. Каким образом коммутатор определяет порт назначения кадра?
4. В каком случае коммутатор выполняет широковещание кадра?
5. Перечислить основные этапы построения покрывающего дерева.
6. Для чего необходим алгоритм построения покрывающих деревьев?
7. Когда коммутатор добавляет запись в динамическую таблицу коммутации?

Лабораторная работа № 4
Построение таблиц коммутации и покрывающих деревьев

Цель работы: освоить алгоритмы построения таблиц коммутации и покрывающих деревьев.

Подготовка:
- знать структуру кадра Ethernet;
- знать алгоритм работы коммутатора Ethernet;
- знать алгоритм построения покрывающего дерева.

Задание: Выполнить имитацию работы сети с заданными статическими таблицами коммутации в среде моделирующей системы.

Порядок выполнения работы
1. Построить покрывающее дерево.
2. Построить статические таблицы коммутации.
3. Ввести структурную схему сети в моделирующую систему.
4. Ввести таблицы коммутации в моделирующую систему.
5. Выполнить анализ работы сети при заданном трафике; оценить количество потерянных пакетов.

Варианты задания: Структурная схема сети, Приложение 1.

Дополнительные требования:
1. Представить таблицы коммутации всех коммутаторов.
2. Выполнить трассировку доставки не менее чем четырёх кадров.
3. Моделировать трафик между не менее чем четырьмя парами терминальных устройств.
4. Интенсивность трафика 400Кб/с, время работы сети – 10 мин.

Содержание отчёта:
- структурная схема сети
- таблицы коммутации
- трассы доставки пакетов
- описание генераторов трафика
- результаты анализа работы сети – количество потерянных пакетов

Указания по выполнению работы:
Использовать моделирующую систему Opnet. Выбрать коммутаторы Catalyst. Из терминальных устройств: два моделировать как серверы, остальные – как рабочие станции.

Практическое занятие № 5

Маршрутизация в IP-сетях

Подготовка:

- иерархическая бесклассовая система IP-адресов CIDR;
- схема доставки IP-пакетов;
- структура таблиц IP-маршрутизации;
- алгоритм работы IP-маршрутизатора.

Задачи:

- агрегирование IP-адресов сетей (хостов);
- построение статических таблиц IP-маршрутизации;
- трассировка прохождения пакетов в IP-сети.

Типовые задания:

1. Определить, возможно ли агрегирование заданных IP-адресов подсетей под общей маской; выполнить агрегирование адресов.
2. Для заданной структурной схемы IP-сети построить статические таблицы маршрутизации.
3. Выполнить ручную трассировку прохождения IP-пакетов между заданными парами терминальных устройств с использование таблиц маршрутизации.

Примеры решения задач:

Задача № 1.

Выполнить агрегирование заданных IP-адресов:

Пусть заданы адреса:
IP_1=243.44.212.0/20
IP_2=243.44.216.0/20
Рассмотрим двоичное представление третьего байта адреса, по которому проходит граница маски подсети:

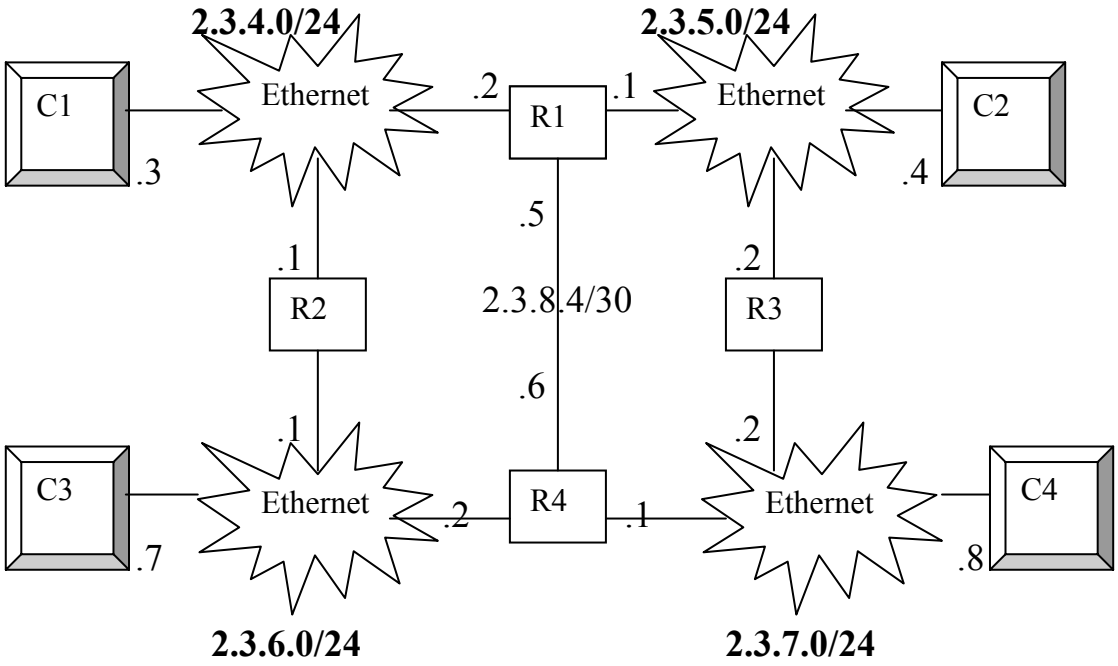
	7	6	5	4	3	2	1	0
IP_1	1	1	0	1	0	1	0	0
IP_2	1	1	0	1	1	0	0	0

Каждая IP-сеть является подсетью 243.44.208.0/18, в которую они могут быть агрегированы в таблицах маршрутизации.

Варианты заданий для самостоятельных упражнений:
1) IP_1=243.44.0.0/16, IP_2=243.48.0.0/16
2) IP_1=194.215.54.169/29, IP_2=194.215.54.170/29, IP_3=194.215.54.171/29
3) IP_1=194.215.54.0/24, IP_2=194.215.54.169/29
4) IP_1=194.215.54.192/26, IP_2=194.216.54.192/26

Задача № 2.

Построить таблицы маршрутизации для заданной сети (интерфейсы обозначены младшей цифрой IP-адреса); в качестве метрики использовать количество хопов:



Построим таблицы:

C1:

Назначение	Шлюз	Интерфейс	Метрика
2.3.4.0/24	-	2.3.4.3	0
2.3.5.0/24	2.3.4.2	2.3.4.3	1
2.3.6.0/24	2.3.4.1	2.3.4.3	1
2.3.7.0/24	2.3.4.1	2.3.4.3	2

C2:

Назначение	Шлюз	Интерфейс	Метрика
2.3.5.0/24	-	2.3.5.4	0
2.3.4.0/24	2.3.5.1	2.3.5.4	1
2.3.7.0/24	2.3.5.2	2.3.5.4	1
2.3.6.0/24	2.3.5.1	2.3.5.4	2

C3:

Назначение	Шлюз	Интерфейс	Метрика
2.3.6.0/24	-	2.3.6.7	0
2.3.4.0/24	2.3.6.1	2.3.6.7	1
2.3.7.0/24	2.3.6.2	2.3.6.7	1
2.3.5.0/24	2.3.6.1	2.3.6.7	2

C4:

Назначение	Шлюз	Интерфейс	Метрика
2.3.7.0/24	-	2.3.7.8	0
2.3.5.0/24	2.3.7.2	2.3.7.8	1
2.3.6.0/24	2.3.7.1	2.3.7.8	1
2.3.4.0/24	2.3.7.1	2.3.7.8	2

R1:

Назначение	Шлюз	Интерфейс	Метрика
2.3.4.0/24	-	2.3.4.2	0
2.3.5.0/24	-	2.3.5.1	0
2.3.8.6	-	2.3.8.5	0
2.3.6.0/24	2.3.8.6	2.3.8.5	1
2.3.7.0/24	2.3.8.6	2.3.8.5	1

R2:

Назначение	Шлюз	Интерфейс	Метрика
2.3.4.0/24	-	2.3.4.1	0
2.3.6.0/24	-	2.3.6.1	0
2.3.5.0/24	2.3.4.2	2.3.4.1	1
2.3.7.0/24	2.3.6.2	2.3.6.1	1

R3:

Назначение	Шлюз	Интерфейс	Метрика
2.3.5.0/24	-	2.3.5.2	0
2.3.7.0/24	-	2.3.7.2	0
2.3.4.0/24	2.3.5.1	2.3.5.2	1
2.3.6.0/24	2.3.7.1	2.3.7.2	1

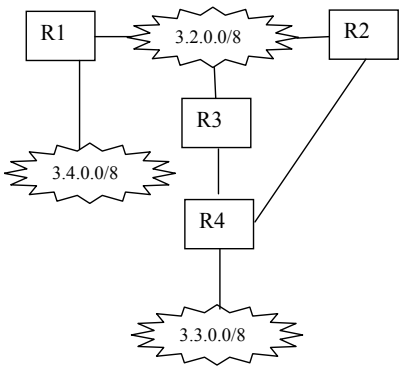
R4:

Назначение	Шлюз	Интерфейс	Метрика
2.3.6.0/24	-	2.3.6.2	0
2.3.7.0/24	-	2.3.7.1	0
2.3.8.5	-	2.3.8.6	0
2.3.4.0/24	2.3.8.5	2.3.8.6	1
2.3.5.0/24	2.3.8.5	2.3.8.6	1

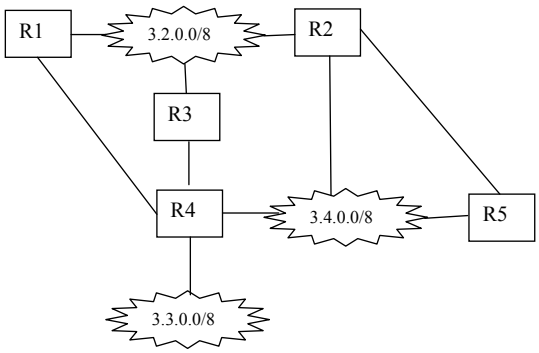
Заметим, что для альтернативных маршрутов равной метрики в таблицах указан произвольный; возможно указание нескольких альтернативных маршрутов из соображений надёжности и равномерной загрузки линий связи. Каждая таблица содержит все известные сети, но в качестве шлюза всегда указан следующий хоп. В общем случае необходимо представить во всех таблицах также и вырожденную сеть 2.3.8.4/30, используемую для указания линии “точка-точка” (например, для удалённого доступа к маршрутизаторам R1, R4). Заметим, что современные маршрутизаторы как правило используют ненумерованные интерфейсы, заданные их локальными обозначениями в целях экономии IP-адресов.

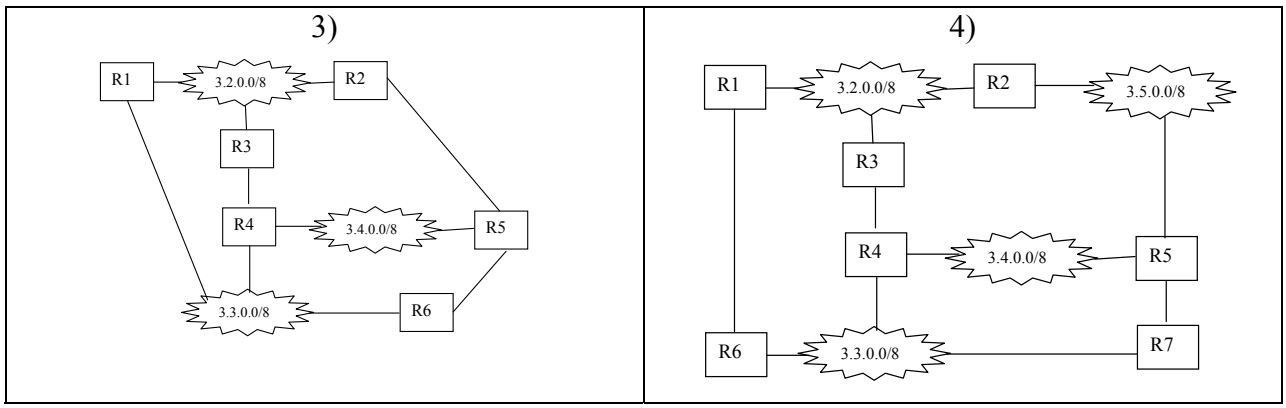
Варианты заданий для самостоятельных упражнений:

1)



2)





Задача № 3.

Выполнить ручную трассировку прохождения IP-пакетов между заданными парами терминальных устройств:

C1→C4:

IP_{C1}=2.3.4.3, IP_{C4}=2.3.7.8

Ключом поиска в таблицах маршрутизации является IP-адрес назначения IP_{C4}:

C1: строка 4: интерфейс=2.3.4.1, шлюз=2.3.4.3 → сеть 2.3.4.0 →
R2: строка 4: интерфейс=2.3.6.1, шлюз=2.3.6.2 → сеть 2.3.6.0 →
R4: строка 2: интерфейс=2.3.7.1 → сеть 2.3.7.0 →
C4

Варианты заданий для самостоятельных упражнений:

1) R1→R6
2) R2→R4
3) R3→R1
4) R4→R2

Контрольные вопросы:

1. Перечислить основные поля заголовка IP-пакета.
2. Описать структуру таблицы маршрутизации.
3. Каким образом маршрутизатор определяет следующий хоп?
4. Что происходит, если запись о сети отсутствует в таблице маршрутизации?
5. Какие типы метрик используются в таблицах маршрутизации?

Лабораторная работа № 5
Построение статических таблиц маршрутизации

Цель работы: изучить особенности построения и использования статических таблиц IP-маршрутизации

Подготовка:

- знать структуру заголовка IP-пакета;
- знать бесклассовую система IP-адресации CIDR;
- знать схему доставки IP-пакетов;
- знать структуру таблиц IP-маршрутизации;
- знать алгоритм работы IP-маршрутизатора.

Задание: Выполнить имитацию работы сети с заданными статическими таблицами маршрутизации в среде моделирующей системы.

Порядок выполнения работы
1. Построить статические таблицы маршрутизации.
2. Ввести структурную схему сети в моделирующую систему.
3. Ввести таблицы маршрутизации в моделирующую систему.
4. Выполнить трассировку передачи пакетов между парами терминальных устройств.
5. Выполнить анализ работы сети при заданном трафике; оценить количество потерянных пакетов.

Варианты задания: Структурная схема сети, Приложение 2.

Дополнительные требования:
1. Представить таблицы маршрутизации всех терминальных и сетевых устройств.
2. Выполнить трассировку доставки не менее чем четырёх IP-пакетов.
3. Моделировать трафик между не менее чем четырьмя парами терминальных устройств.
4. Интенсивность трафика 400Кб/с, время работы сети – 10 мин.

Содержание отчёта:
- структурная схема сети
- таблицы маршрутизации
- трассы доставки пакетов
- описание генераторов трафика
- результаты анализа работы сети – количество потерянных пакетов

Указания по выполнению работы:
Использовать моделирующую систему Opnet. Выбрать маршрутизаторы Cisco 2500. Из терминальных устройств: два моделировать как серверы, остальные – как рабочие станции.

Практическое занятие № 6
Протоколы динамической маршрутизации

Подготовка:
- основы маршрутизации в IP-сетях;
- протокол динамической маршрутизации RIP;
- протокол динамической маршрутизации OSPF;
- протокол динамической маршрутизации магистральных сетей BGP.

Задачи:
- построение таблиц маршрутизации с помощью протокола RIP;
- построение таблиц маршрутизации с помощью протокола OSPF.

Типовые задания:
1. Для заданной структурной схемы IP-сети выполнить ручную трассировку процессов построения таблиц маршрутизации с помощью протокола RIP.
2. Для заданной структурной схемы IP-сети выполнить ручную трассировку процессов построения таблиц маршрутизации с помощью протокола OSPF.

Примеры решения задач:

Задача № 1.

Выполнить построение таблиц маршрутизации с помощью протокола RIP. Структурная схема сети из практического занятия 5.

Этап 1. Заполнение таблиц записями о непосредственно подключенных сетях:

R1:

Назначение	Шлюз	Интерфейс	Метрика
2.3.4.0/24	-	2.3.4.2	0
2.3.5.0/24	-	2.3.5.1	0
2.3.8.6	-	2.3.8.5	0

R2:

Назначение	Шлюз	Интерфейс	Метрика
2.3.4.0/24	-	2.3.4.1	0
2.3.6.0/24	-	2.3.6.1	0

R3:

Назначение	Шлюз	Интерфейс	Метрика
2.3.5.0/24	-	2.3.5.2	0
2.3.7.0/24	-	2.3.7.2	0

R4:

Назначение	Шлюз	Интерфейс	Метрика
2.3.6.0/24	-	2.3.6.2	0
2.3.7.0/24	-	2.3.7.1	0
2.3.8.4/30	-	2.3.8.6	0

Этап 2. Обмен таблицами между соседними маршрутизаторами. Добавление записи, если указана новая сеть. Корректировка записи, если указана лучшая метрика.

Например, маршрутизатор R2 получает таблицы от соседних маршрутизаторов R1 и R4. После получения таблицы маршрутизации R1 таблица маршрутизатора R2 имеет вид:

R2:

Назначение	Шлюз	Интерфейс	Метрика	Источник
2.3.4.0/24	-	2.3.4.1	0	
2.3.6.0/24	-	2.3.6.1	0	
2.3.5.0/24	2.3.4.2	2.3.4.1	1	R1

Заметим, что в качестве интерфейса указан тот интерфейс маршрутизатора R2, который получил таблицу, в качестве шлюза – тот интерфейс маршрутизатора R1, который послал таблицу. Метрика, равная числу хопов, увеличилась на 1.

После получения таблицы маршрутизации R4 таблица маршрутизатора R2 имеет вид:

R2:

Назначение	Шлюз	Интерфейс	Метрика	Источник
2.3.4.0/24	-	2.3.4.1	0	
2.3.6.0/24	-	2.3.6.1	0	
2.3.5.0/24	2.3.4.2	2.3.4.1	1	R1
2.3.7.0/24	2.3.6.2	2.3.6.1	1	R4
2.3.8.4/30	2.3.6.2	2.3.6.1	1	R4

Аналогичным образом корректируются таблицы других маршрутизаторов. Заметим, что:

R1 получает таблицы от R2, R3, R4;

R3 получает таблицы от R1, R4;

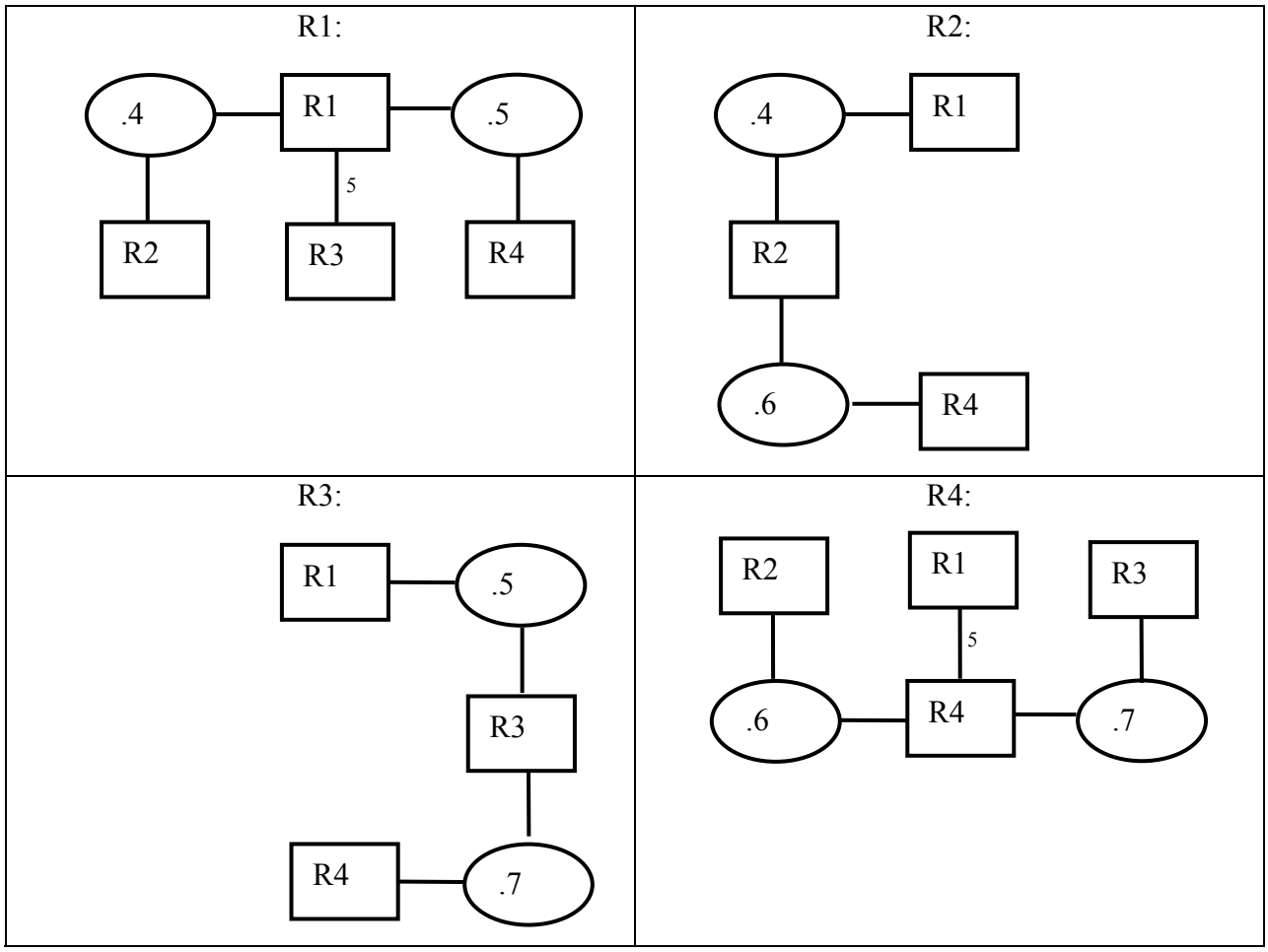
R4 получает таблицы от R1, R2, R3.

Затем повторяется обмен обновлёнными таблицами. При отсутствии изменений (отказов и новых подключений) таблицы стабилизируются после конечного числа шагов. Для нашей сети стабилизация наблюдается после первой итерации, так как максимальная метрика равна 1. Заметим, что протокол RIP работает на метриках до 15.

Задача № 2.

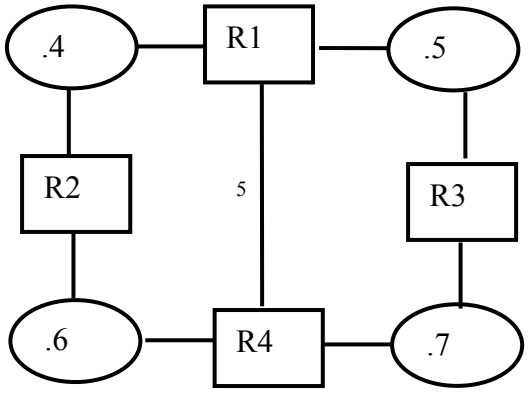
Выполнить построение таблиц маршрутизации с помощью протокола OSPF. Структурная схема сети из практического занятия 5. Метрика – скорость передачи. Считаем, что скорость локальной сети 10Мб/с, скорость двухточечной линии – 2 Мб/с.

Этап 1. Проверка состояний линий связи с помощью сообщений HELLO. Построение графа смежных маршрутизаторов/сетей:



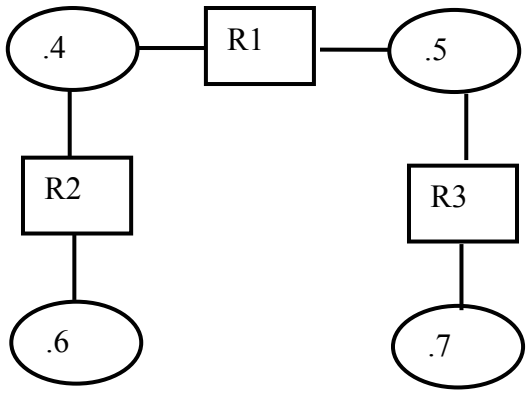
Этап 2. Обмен сообщениями с графами смежных маршрутизаторах между всеми маршрутизаторами, поддерживающими протокол OSPF. То есть: R1→R2,R3,R4; R2→R1,R3,R4; R3→R1,R2,R4; R4→R1,R2,R3. Для этого используется специальный адрес ограниченного широковещания OSPFrouters.

Этап 4. Построение графа структуры сети всеми маршрутизаторами по полученным сообщениям. Заметим, что все маршрутизаторы строят одинаковый граф:

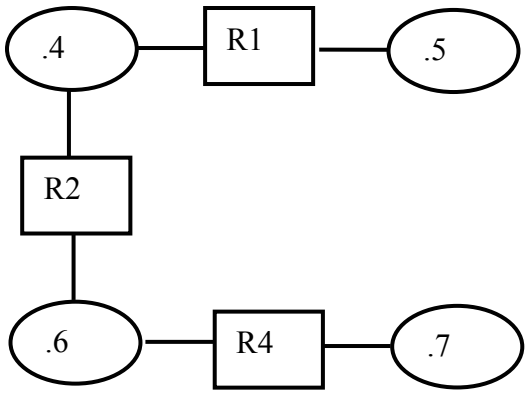


Этап 5. Построение дерева кратчайших маршрутов с помощью алгоритма Дейкстры:

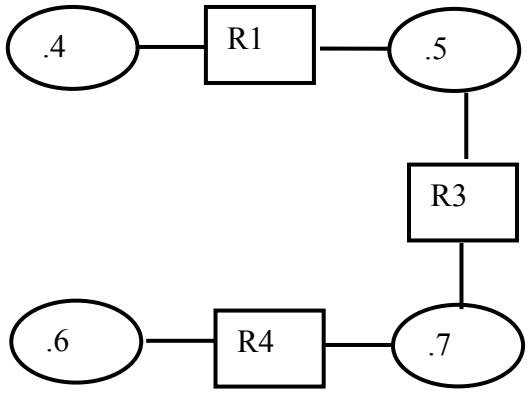
R1:



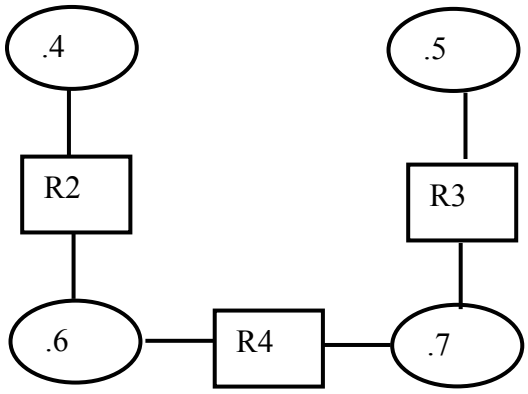
R2:



R3:



R4:



Этап 6. Генерация таблицы маршрутизации по дереву кратчайших маршрутов:

R1:

Назначение	Шлюз	Метрика
2.3.4.0	-	1
2.3.5.0	-	1
2.3.6.0	R2	2
2.3.7.0	R3	2

R2:

Назначение	Шлюз	Метрика
2.3.4.0	-	1
2.3.6.0	-	1
2.3.5.0	R1	2
2.3.7.0	R4	2

R3:

Назначение	Шлюз	Метрика
2.3.5.0	-	1
2.3.7.0	-	1
2.3.4.0	R1	2
2.3.6.0	R4	2

R4:

Назначение	Шлюз	Метрика
2.3.6.0	-	1
2.3.7.0	-	1
2.3.4.0	R2	2
2.3.5.0	R3	2

Замечания:
1. Метрика 1 не подписана на рёбрах; скорость 10M соответствует метрике 1, скорость 2M – метрике 5.
2. Таблицы приведены в упрощённом формате; IP-адреса шлюзов и интерфейсов указать самостоятельно.

Варианты заданий для самостоятельных упражнений (задача 1, 2) – сети из упражнений практического занятия 5.

Контрольные вопросы:
1. Для чего необходимо применение протоколов динамической маршрутизации?
2. Указать основные этапы работы протокола RIP.
3. Каким образом протокол RIP предотвращает использование устаревших записей?
4. Указать основные этапы работы протокола OSPF.
5. Для чего используются области в протоколе OSPF?
6. Что такое автономная система?
7. Указать основные этапы работы протокола BGP.
8. Каким образом организуется совместная работа нескольких протоколов динамической маршрутизации?

Лабораторная работа № 6
Построение таблиц маршрутизации с помощью протоколов RIP и OSPF

Цель работы: изучить особенности построения таблиц IP-маршрутизации с помощью протоколов динамической маршрутизации RIP и OSPF

Подготовка:
- знать структуру таблиц и алгоритм работы IP-маршрутизатора;
- знать протокол динамической маршрутизации RIP;
- знать протокол динамической маршрутизации OSPF.

Задание: Выполнить имитацию работы протоколов динамической маршрутизации и работу сети, использующей построенные таблицы маршрутизации.

Порядок выполнения работы
1. Построить таблицы маршрутизации с помощью протоколов RIP и OSPF.
2. Ввести структурную схему сети в моделирующую систему.
3. Выполнить трассировку процессов построения таблиц маршрутизации с помощью протоколов RIP и OSPF.
4. Сравнить полученные таблицы маршрутизации с построенными вручную.
5. Выполнить анализ работы сети при заданном трафике; оценить количество потерянных пакетов.

Варианты задания: Структурная схема сети, Приложение 2.

Дополнительные требования:
1. Представить таблицы всех маршрутизазаторов, построенные с помощью протоколов RIP и OSPF.
2. Для одного маршрутизатора представить трассировку процесса построения таблиц с помощью протоколов RIP и OSPF.
3. Моделировать трафик между не менее чем четырьмя парами терминальных устройств.
4. Интенсивность трафика 400Кб/с, время работы сети – 10 мин.

Содержание отчёта:
- структурная схема сети
- таблицы маршрутизации, полученные с помощью протоколов RIP и OSPF
- трассы процессов построения таблиц
- результаты анализа работы сети – количество потерянных пакетов

Указания по выполнению работы:
Использовать моделирующую систему Opnet. Выбрать маршрутизаторы Cisco 2500. Из терминальных устройств: два моделировать как серверы, остальные – как рабочие станции.

Практическое занятие № 7
Особенности организации сетей с коммутацией меток

Подготовка:
- формат метки (стека меток) технологии MPLS;
- структура таблиц коммутации меток;
- алгоритм работы LSR/LER маршрутизатора;
- особенности разделения трафика на FEC и построения LSP.

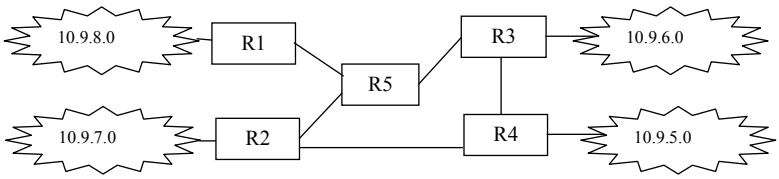
Задачи:
- разделить трафик сети на классы эквивалентности доставки FEC;
- построить пути коммутации меток LSP;
- построить таблицы коммутации меток для LSR/LER.

Типовые задания:
1. Для заданной MPLS сети выполнить разделение трафика на FEC.
2. Для заданной сети и FEC построить пути коммутации меток LSP.
3. Для заданной сети и путей коммутации меток LSP построить таблицы коммутации меток всех LSR/LER.
4. Выполнить трассировку прохождения пакетов, используя построенные таблицы коммутации меток.

Примеры решения задач:

Задача № 1.

Выполнить разделение трафика на FEC:



Если не учитывать возможное разделение трафика по требуемому качеству обслуживания, то при выделении FEC рассматривается только пара IP-адресов источника и приёмника. Тогда можно выделить следующие FEC для представленных маршрутизаторов:
- FEC1 (10.9.8.*→10.9.7.*), FEC2 (10.9.8.*→10.9.6.*), FEC3 (10.9.8.*→10.9.5.*);
- FEC4 (10.9.7.*→10.9.8.*), FEC5 (10.9.7.*→10.9.6.*), FEC6 (10.9.8.*→10.9.5.*) ;
- FEC7 (10.9.6.*→10.9.8.*), FEC8 (10.9.6.*→10.9.7.*), FEC9 (10.9.6.*→10.9.5.*) ;
- FEC10 (10.9.5.*→10.9.8.*), FEC11 (10.9.5.*→10.9.7.*), FEC12 (10.9.5.*→10.9.6.*).

Задача № 2.

Построить пути коммутации меток LSP:

	10.9.5.*	10.9.6.*	10.9.7.*	10.9.8.*
10.9.5.*	-	R4(1)-R3	R4(1)-R2	R4(2)-R2(1)-R5(1)-R1
10.9.6.*	R3(1)-R4	-	R3(2)-R4(3)-R2	R3(2)-R5(2)-R1
10.9.7.*	R2(3)-R4	R2(3)-R5(2)-R3	-	R2(4)-R5(3)-R1
10.9.8.*	R1(5)-R5(3)-R3(4)-R4	R1(6)-R5(4)-R3	R1(7)-R5(4)-R2	-

Заметим, что при назначении меток, указанных в скобках, использован уникальный выбор метки для каждого FEC в пределах маршрутизатора. Количество используемых меток можно сократить, если использовать уникальные метки только в пределах одного и того же интерфейса. Выполнить указанное назначение меток самостоятельно.

Задача № 3.

Построить таблицы коммутации меток для LSR/LER:

R1:

Входной интерфейс	Входная метка	Выходной интерфейс	Выходная метка
i10.9.8(→10.9.5)	-	iR5	5
i10.9.8(→10.9.6)	-	iR5	6
i10.9.8(→10.9.7)	-	iR5	7
iR5	1	i10.9.8	-
iR5	2	i10.9.8	-
iR5	3	i10.9.8	-

R2:

Входной интерфейс	Входная метка	Выходной интерфейс	Выходная метка
i10.9.7(→10.9.5)	-	iR4	3
i10.9.7(→10.9.6)	-	iR5	3
i10.9.7(→10.9.8)	-	iR5	4
iR4	1	i10.9.7	-
iR4	3	i10.9.7	-
iR5	4	i10.9.7	-
iR4	2	iR5	1

R3:

Входной интерфейс	Входная метка	Выходной интерфейс	Выходная метка
i10.9.6(→10.9.5)	-	iR4	1
i10.9.6(→10.9.7)	-	iR4	2
i10.9.6(→10.9.8)	-	iR5	2
iR4	1	i10.9.6	-
iR5	2	i10.9.6	-
iR5	4	i10.9.6	-
iR5	3	iR4	4

R4:

Входной интерфейс	Входная метка	Выходной интерфейс	Выходная метка
i10.9.5(→10.9.6)	-	iR3	1
i10.9.5(→10.9.7)	-	iR2	1
i10.9.5(→10.9.8)	-	iR2	2
iR3	1	i10.9.5	-
iR2	3	i10.9.5	-
iR3	4	i10.9.5	-
iR3	2	iR2	3

R5:

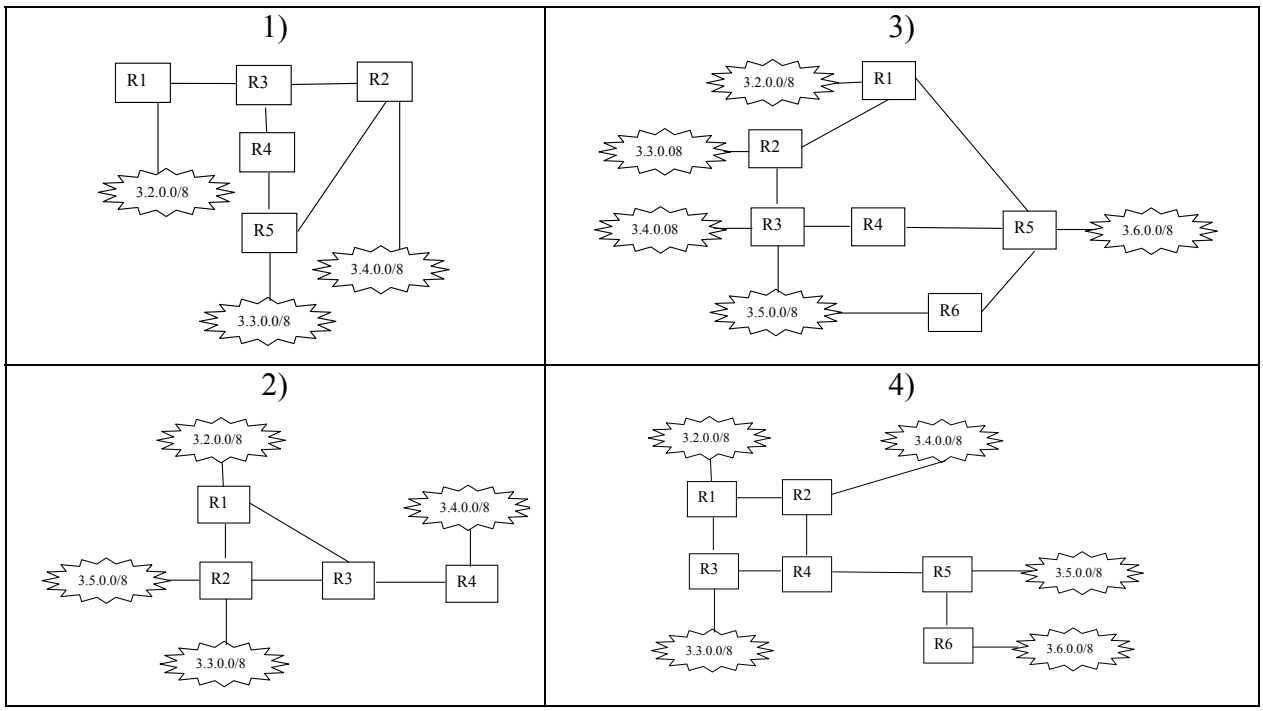
Входной интерфейс	Входная метка	Выходной интерфейс	Выходная метка
iR2	1	iR1	1
iR3	2	iR1	2
iR2	3	iR3	2
iR2	4	iR1	3
iR1	5	iR3	3
iR1	6	iR4	4
iR1	7	iR2	2

Задача № 4.

Выполнить трассировку прохождения пакетов:

10.9.8.115→10.9.5.47:
10.9.8.115→
R1 (строка 1: метка 5, интерфейс iR5) →
R5 (строка 5: метка 3, интерфейс iR3) →
R3 (строка 7: метка 4, интерфейс iR4) →
R4 (строка 6: интерфейс i10.9.5) →10.9.5.0→10.9.5.47

Варианты заданий для самостоятельных упражнений:



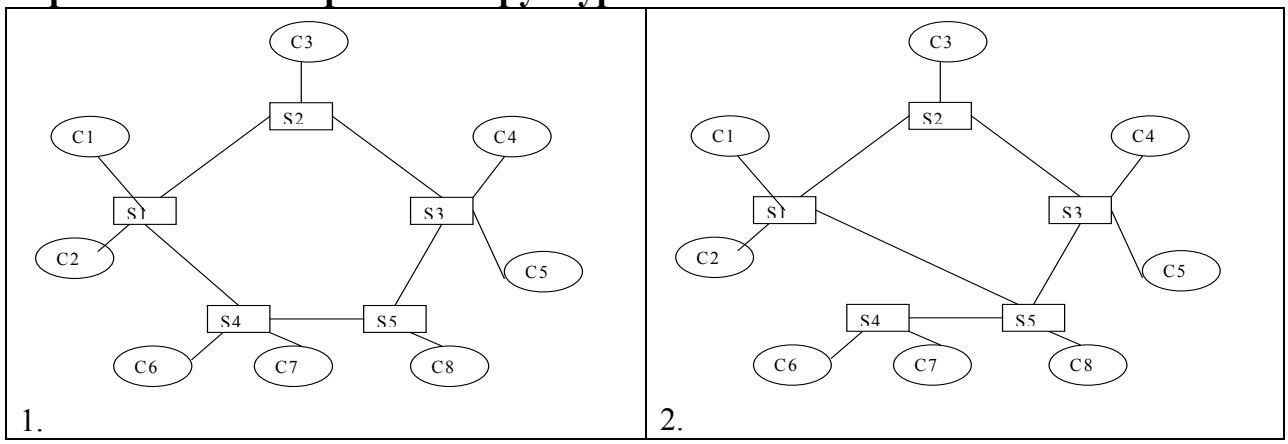
Контрольные вопросы:

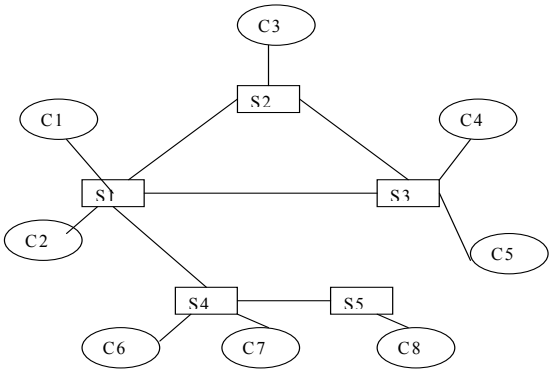
1. Для чего необходимо применение технологии коммутации меток?
2. Описать формат заголовка MPLS.
3. Описать структуру таблицы коммутации меток.
4. Что такое путь коммутации меток LSP?
5. Что такое класс эквивалентности доставки FEC?
6. Каким образом выполняется назначение меток?
7. Для чего применяется стек меток в технологии MPLS?

Литература

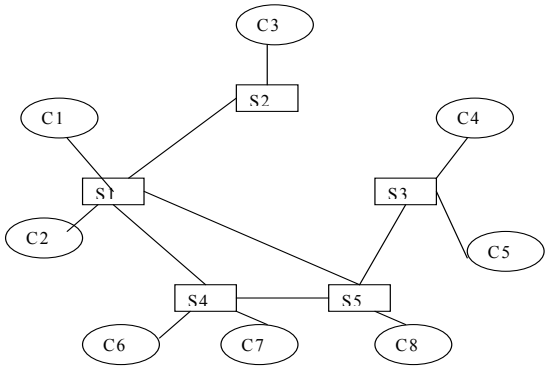
1. В.Г. Олифер, Н.А. Олифер Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 3-е изд. – СПб.: Питер, 2006. – 958 с.
2. IETF References for Comments (RFC): 791, 1548, 826, 903, 793, 1332, 1877
3. IEEE 802.3*, 802.1*.

Приложение 1. Варианты структурных схем сетей Ethernet

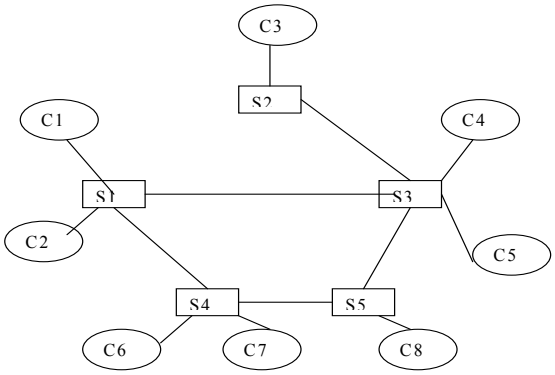



C3
S2
C1
C4
S1
S3
C2
C5
S4
S5
C6
C7
C8

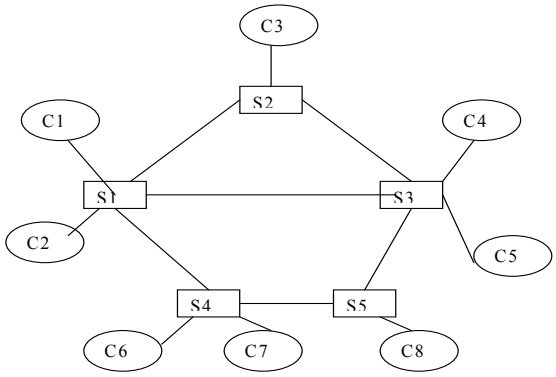
3.


C3
S2
C1
C4
S1
S3
C2
C5
S4
S5
C6
C7
C8

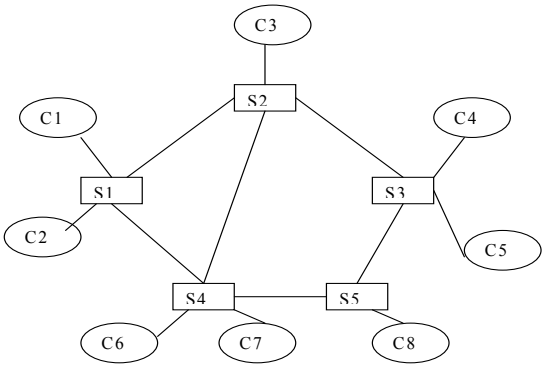
4.


C3
S2
C1
C4
S1
S3
C2
C5
S4
S5
C6
C7
C8

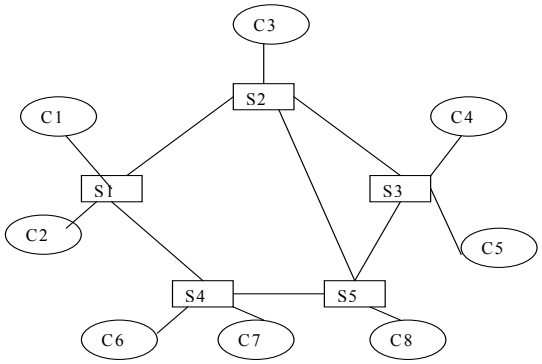
5.


C3
S2
C1
C4
S1
S3
C2
C5
S4
S5
C6
C7
C8

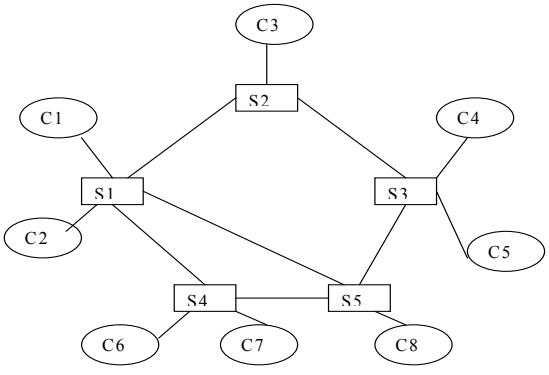
6.


C3
S2
C1
C4
S1
S3
C2
C5
S4
S5
C6
C7
C8

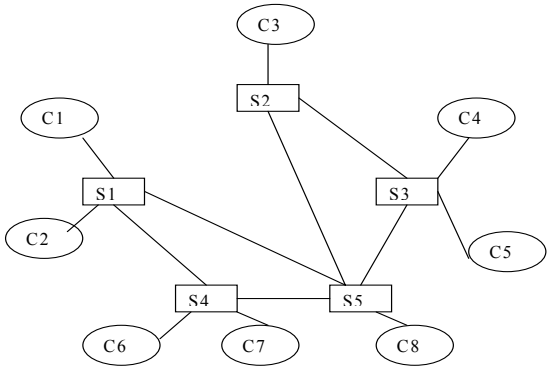
7.


C3
S2
C1
C4
S1
S3
C2
C5
S4
S5
C6
C7
C8

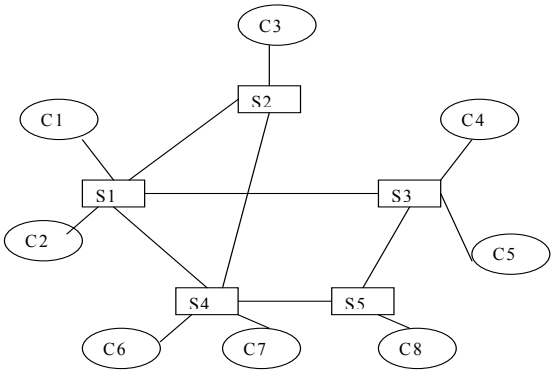
8.


C3
S2
C1
C4
S1
S3
C2
C5
S4
S5
C6
C7
C8

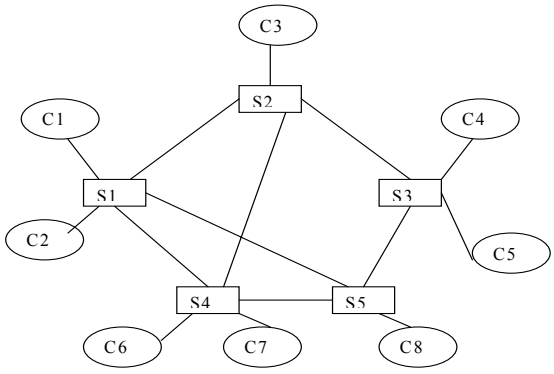
9.


C3
S2
C1
C4
S1
S3
C2
C5
S4
S5
C6
C7
C8

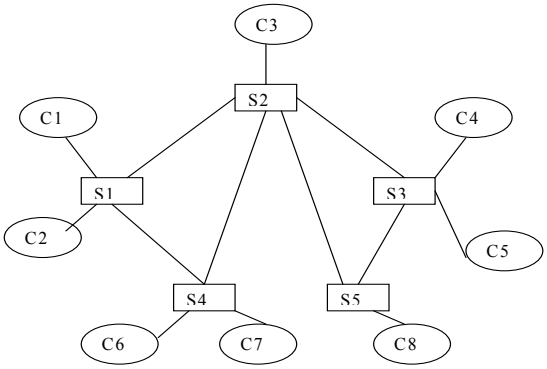
10.



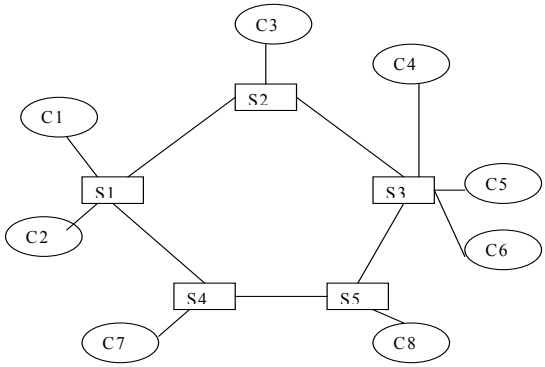
11.



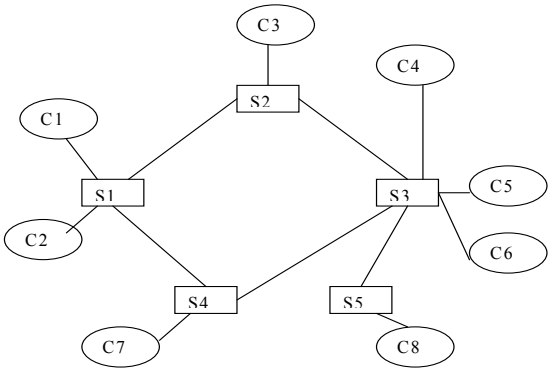
12.



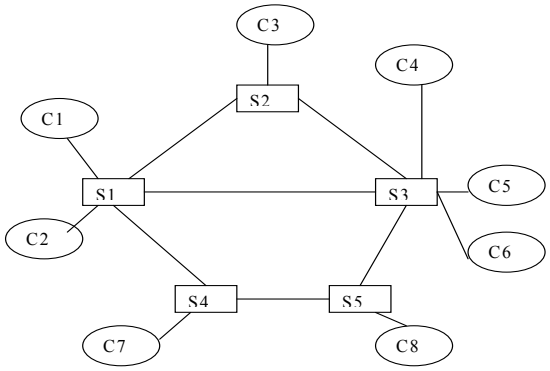
13.



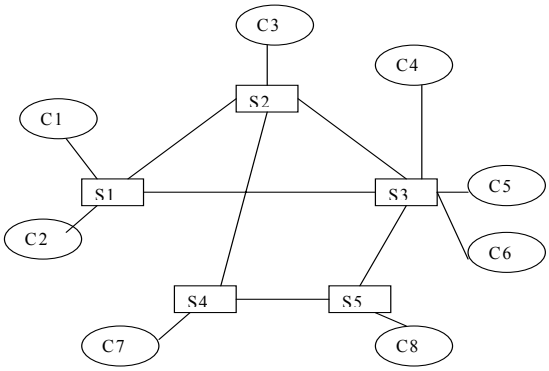
14.



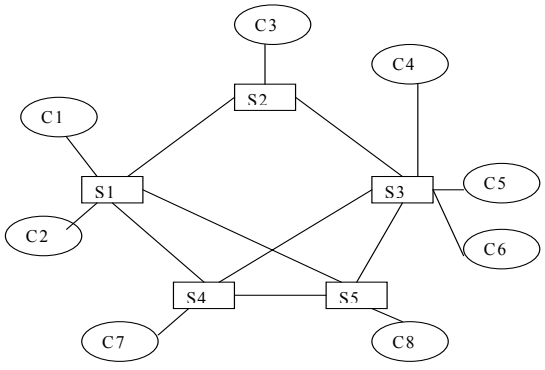
15.



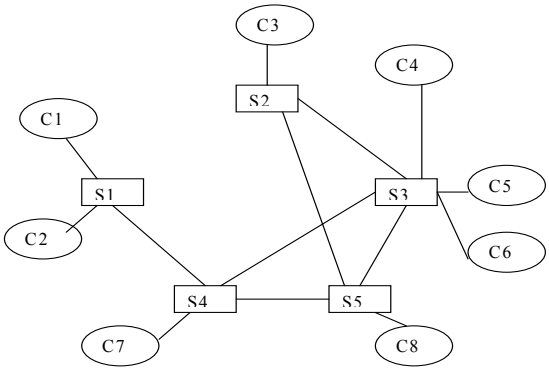
16.



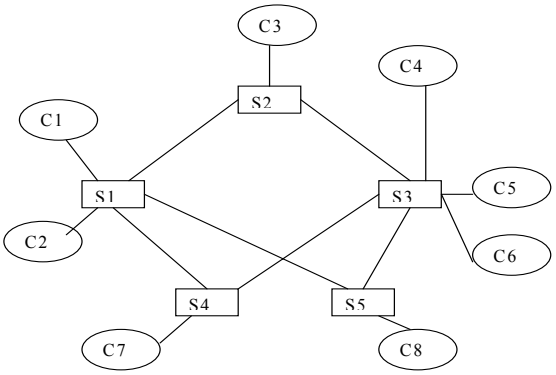
17.



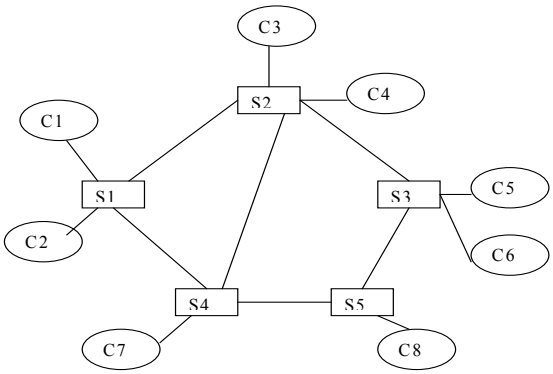
18.



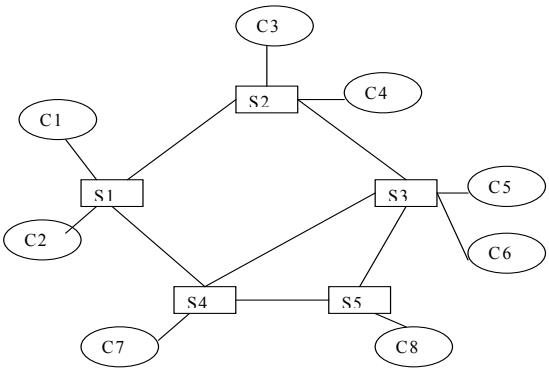
19.



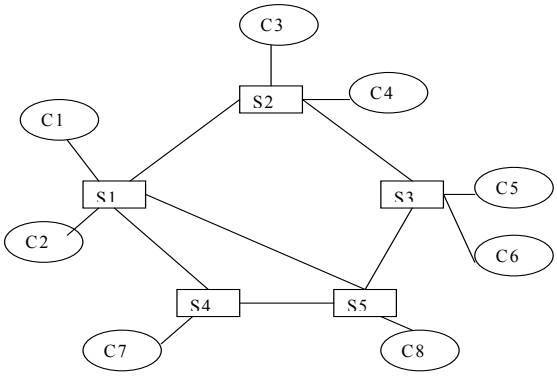
20.



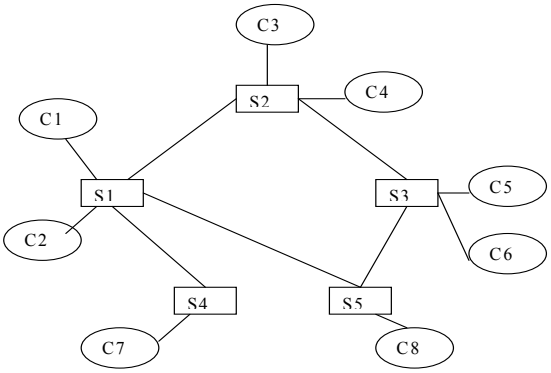
21.



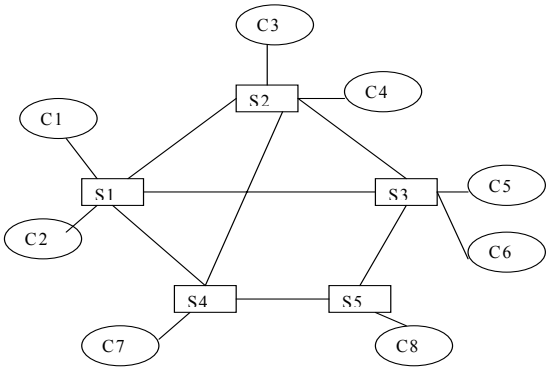
22.



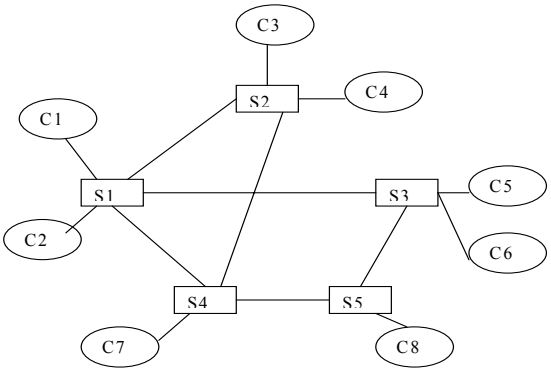
23.



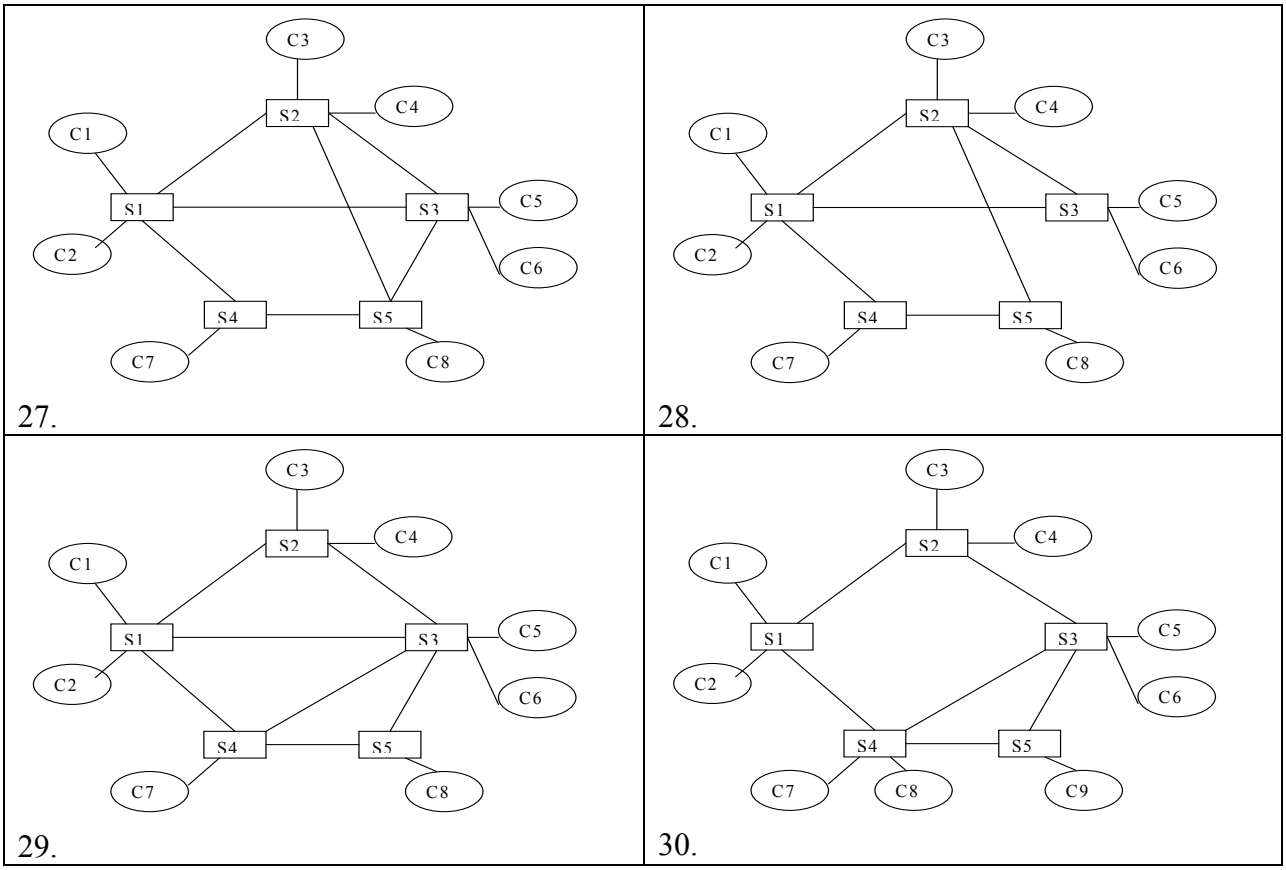
24.



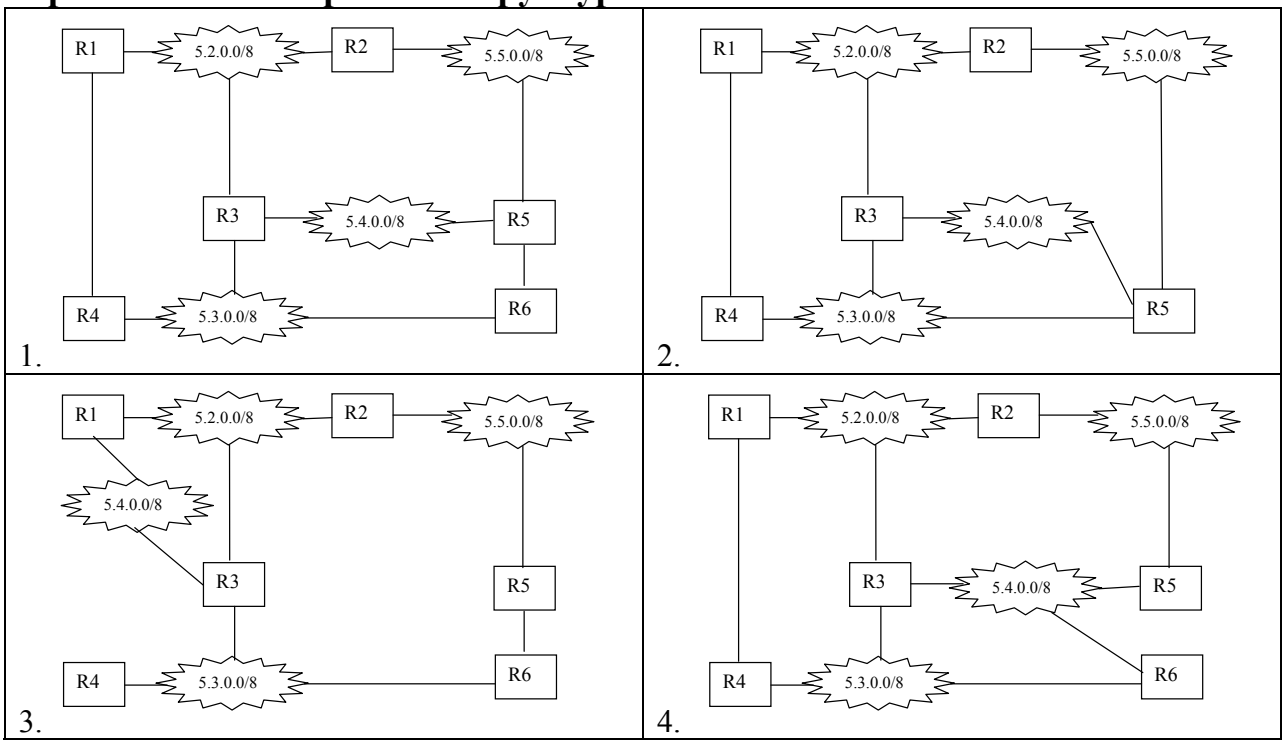
25.

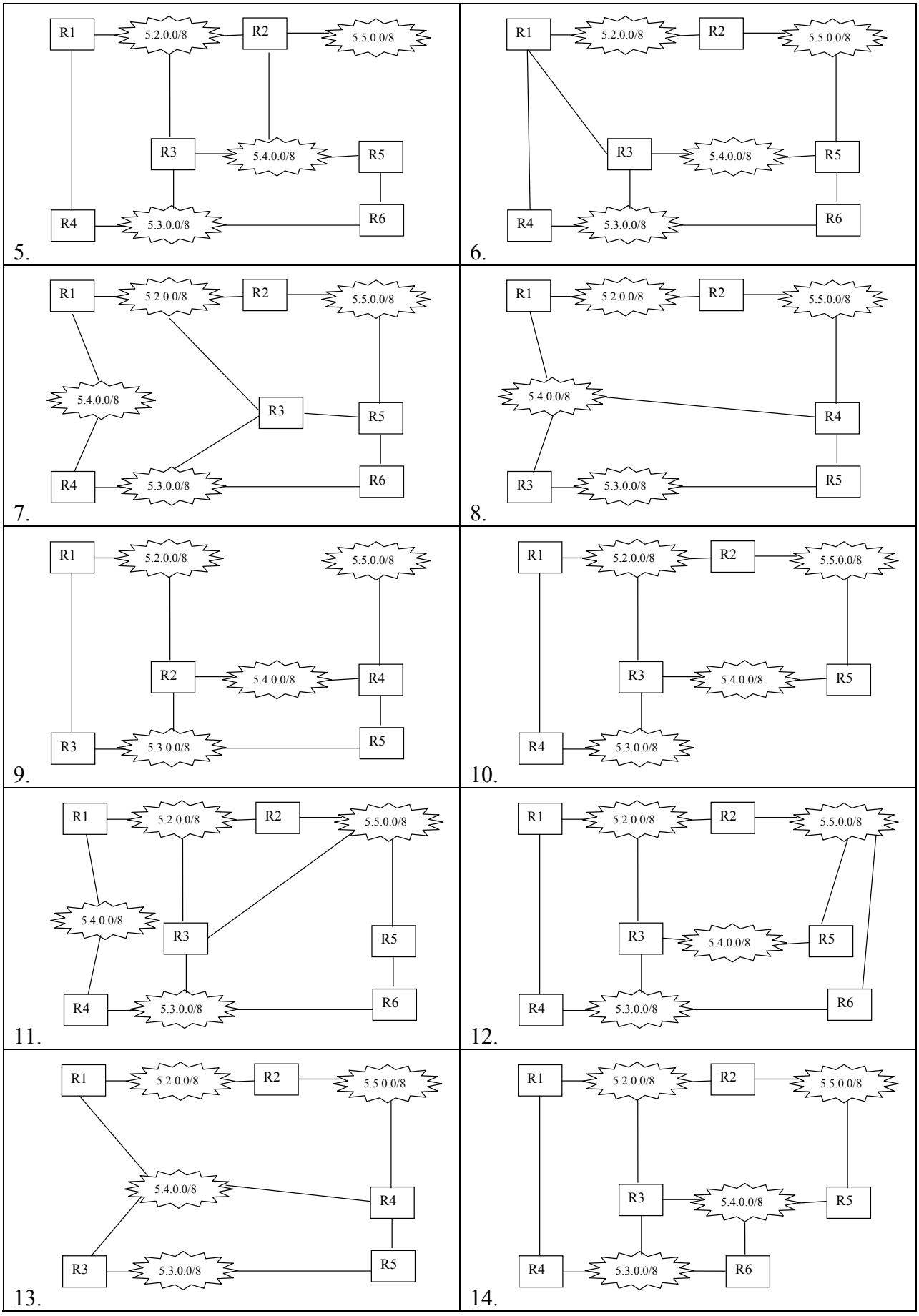


26.

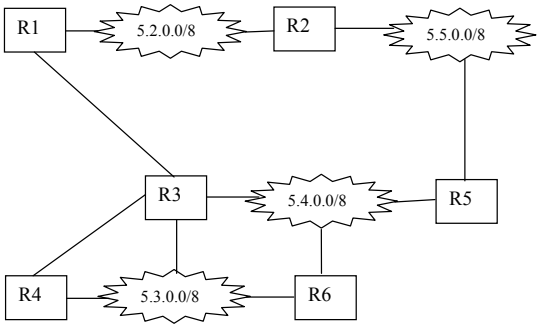


Приложение 2. Варианты структурных схем IP-сетей

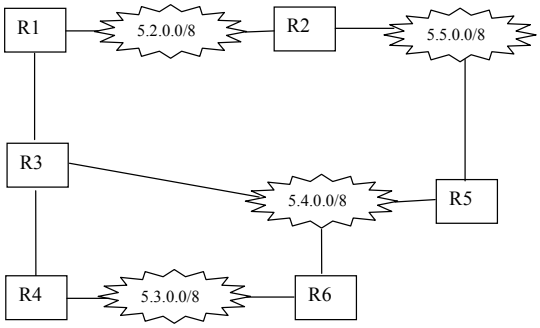



5.
R1
5.2.0.0/8
R2
5.5.0.0/8
R3
5.4.0.0/8
R5
R4
5.3.0.0/8
R6
6.
R1
5.2.0.0/8
R2
5.5.0.0/8
R3
5.4.0.0/8
R5
R4
5.3.0.0/8
R6
7.
R1
5.2.0.0/8
R2
5.5.0.0/8
5.4.0.0/8
R3
R5
R4
5.3.0.0/8
R6
8.
R1
5.2.0.0/8
R2
5.5.0.0/8
5.4.0.0/8
R4
R3
5.3.0.0/8
R5
9.
R1
5.2.0.0/8
5.5.0.0/8
R2
5.4.0.0/8
R4
R3
5.3.0.0/8
R5
10.
R1
5.2.0.0/8
R2
5.5.0.0/8
R3
5.4.0.0/8
R5
R4
5.3.0.0/8
11.
R1
5.2.0.0/8
R2
5.5.0.0/8
5.4.0.0/8
R3
R5
R4
5.3.0.0/8
R6
12.
R1
5.2.0.0/8
R2
5.5.0.0/8
R3
5.4.0.0/8
R5
R4
5.3.0.0/8
R6
13.
R1
5.2.0.0/8
R2
5.5.0.0/8
5.4.0.0/8
R4
R3
5.3.0.0/8
R5
14.
R1
5.2.0.0/8
R2
5.5.0.0/8
R3
5.4.0.0/8
R5
R4
5.3.0.0/8
R6

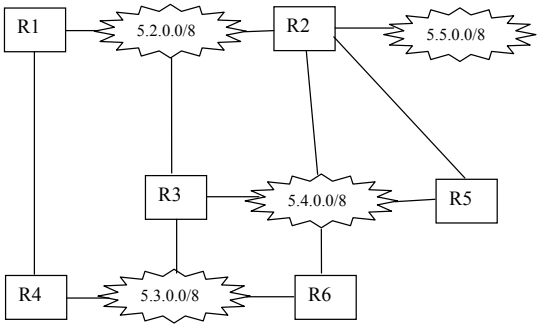
15.



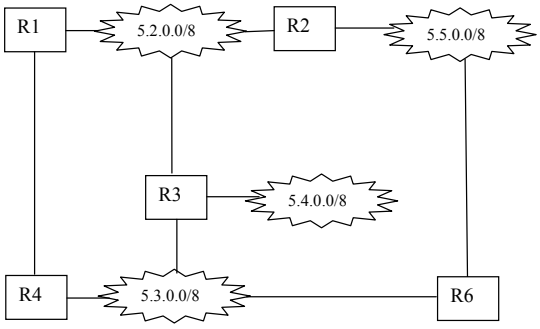
16.



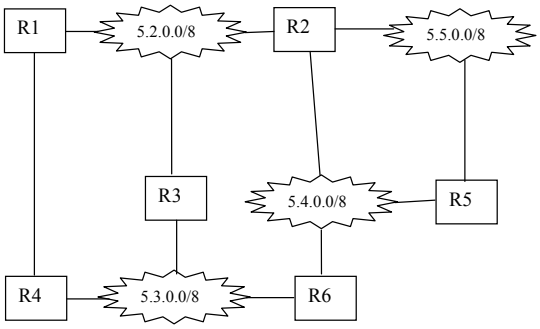
17.



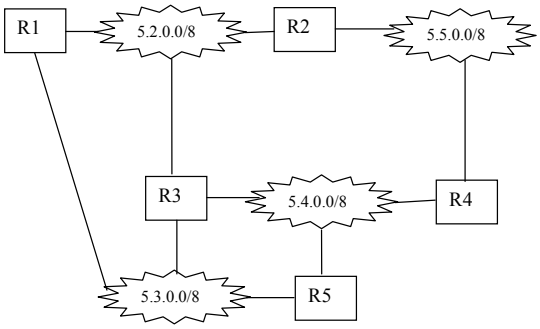
18.



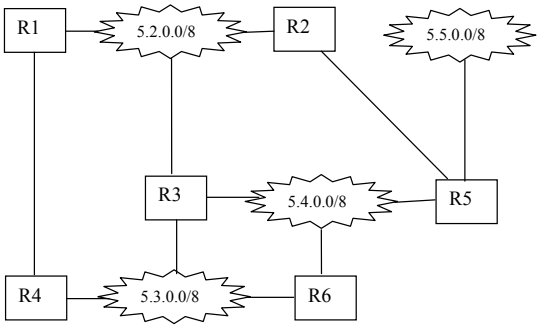
19.



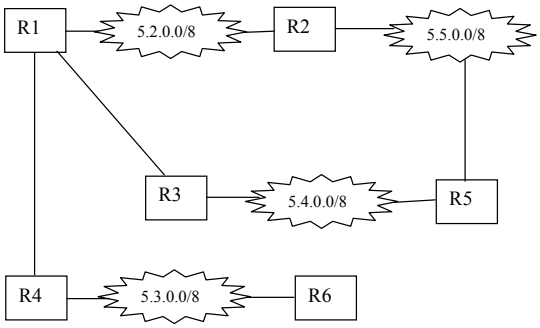
20.



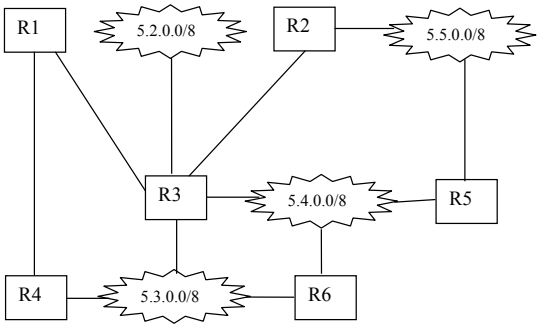
21.



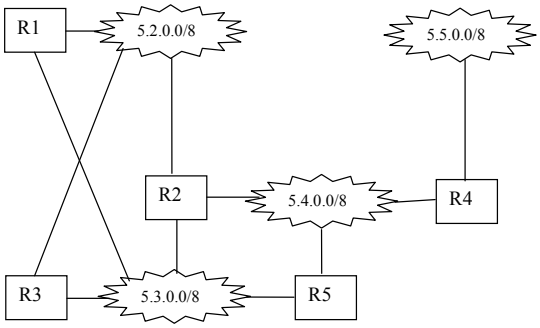
22.



23.

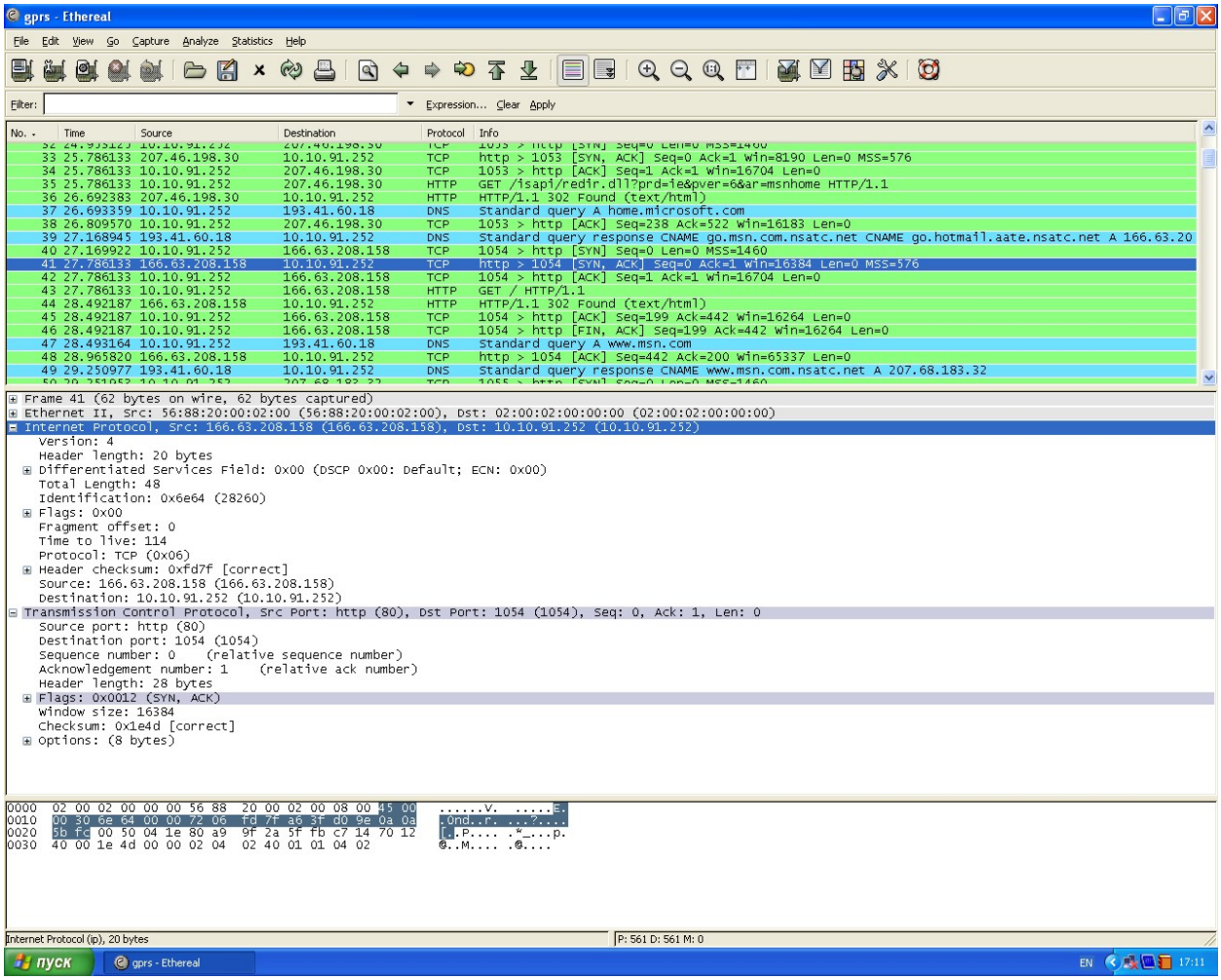


24.



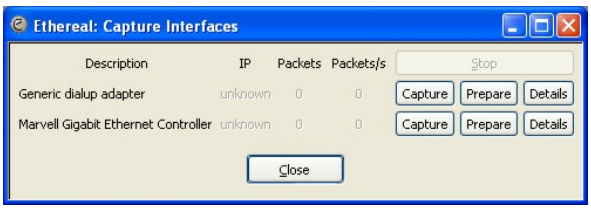
Приложение 3. Краткое описание анализатора трафика Ethereal

Программа Ethereal (http://www.ethereal.com) позволяет прослушивать выбранный сетевой интерфейс с динамическим отображением передаваемых пакетов, записать последовательность пакетов в указанный файл, проанализировать содержимое пактов. Программа обеспечивает подсчет количества переданных пакетов по каждому из указанных протоколов. Возможна фильтрация трафика для отображения и сохранения выбранных типов пакетов, а также анализ ранее сохранённой в файле трассы. Далее приведен пример образа экрана Ethereal:



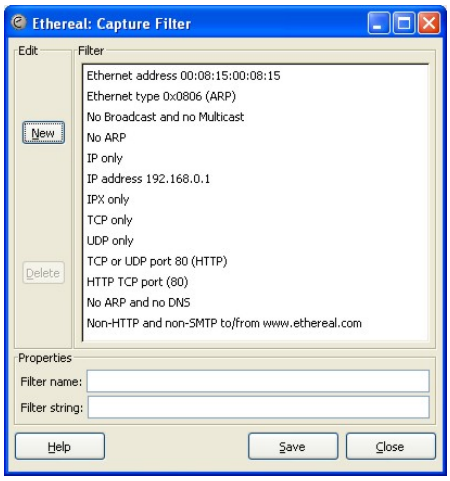
В верхней части экрана отображен фрагмент последовательности пакетов с их кратким описанием: номер (No.), время поступления (Time), источник (Source), назначение (Destination), протокол (Protocol), краткая информация (Info). В средней части экрана представлены допустимые шаблона интерпретации заголовков пакета в соответствии с инкапсулированными протоколами. В настоящем примере интерпретированы IP и TCP заголовки текущего пакета. В нижней части экрана представлен шестнадцатеричный дамп пакета: в первой колонке указаны смещения от начала пакета, во второй – шестнадцатеричный дамп, в третьей – символьная интерпретация.

Для записи передаваемой информации служит раздел меню Захват (Capture). Возможно простое прослушивание указанного сетевого интерфейса с помощью кнопок окна пункта меню Интерфейсы (Interfaces):

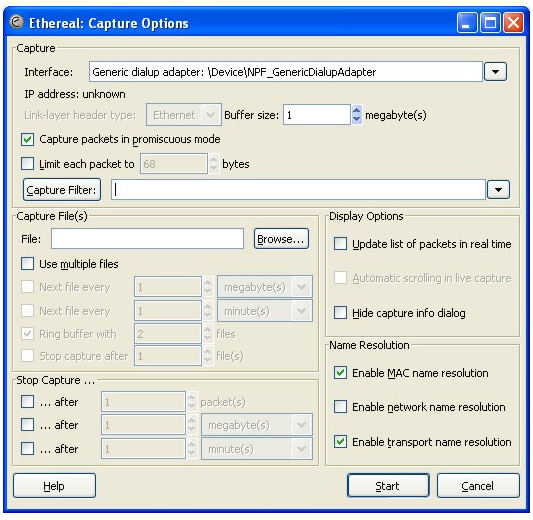


В указанном примере доступны два интерфейса: Адаптер коммутируемой связи (Generic dialup adapter) и Контроллер сети Ethernet (Marvell Gigabit Ethernet Controller). Запуск прослушивания выполняется нажатием соответствующей кнопки Захват (Capture). После этого появляется окно со статистикой полученных пакетов; завершение прослушивания выполняется нажатием кнопки Стоп (Stop). Записанная последовательность пакетов может быть сохранена в файле с помощью пунктов раздела меню Файл (File); этот раздел меню позволяет также загрузить ранее сохранённую последовательность пакетов.

Трафик реальных сетей может быть весьма интенсивным, что приводит к большим объемам сохранённых последовательностей пакетов. В программе Ethereal предусмотрена возможность фильтрации, в этом случае записываются только пакеты, удовлетворяющие указанному фильтру. В простейшем случае фильтр задаёт имя протокола; возможно формирование более сложных фильтров указанием адресов отправителя либо получателя, номеров портов и другой информации. Фильтры создаются в окне пункта меню Фильтры захвата (Capture Filters):



Для совместного указания интерфейса и фильтра служит пункт меню Опции (Options):



Приложение 4. Краткое описание моделирующей системы Opnet

Система Opnet (http://www.opnet.com) позволяет ввести в графическом редакторе структурную схему сети. Графическими элементами являются сетевые устройства (коммутаторы, маршрутизаторы), линии связи, а также терминальные устройства: серверы и рабочие станции. Предусмотрено большое число моделей реальных устройств в библиотеке компонентов моделирующей системы. Кроме того, в дополнительных текстовых окнах вводятся параметры конфигурации устройств. Например, MAC- и IP-адреса, таблицы коммутации и маршрутизации, перечень используемых протоколов.

Для терминальных устройств возможно описание трафика сети. В генераторах трафика указываются прикладные протоколы ftp, http, интенсивность трафика, длины передаваемых пакетов, адреса назначения.

Система Opnet не предоставляет средства непосредственной визуализации телекоммуникационных процессов. Однако, возможна проверка работоспособности сети с помощью имитационного моделирования, представленного итоговыми результатами работы сети за указанный период реального времени, собранными с помощью специальных вычислительных элементов модели.

Целостность моделируемой сети и гарантированная доставка пакетов может быть косвенно оценена нулевым количеством потерянных пакетов. Возможна оценка дополнительных характеристик, таких как трафик сети, время доставки пакета.