Дюмин А.А.

Методические рекомендации
для лабораторного практикума
по курсу

«Протоколы и оборудование корпоративных сетей»

Beta Release 0.7

2016 г.

Лабораторная работа №7.

Основы динамической маршрутизации в IP-сетях. RIP протокол. Основы организации групповой рассылки данных.

В данной лабораторной работе предлагается изучить механизмы динамической маршрутизации в реальных условиях на примере протокола RIP, а также механизмы доставки multicast-трафика в сетях с несколькими маршрутизаторами..

Также в данной лабораторной работе предлагается изучить протокол транспортного уровня UDP, не гарантирующий доставку сообщений.

Протокол UDP позволяет передавать сообщения как в unicast и broadcast, так и в multicast режимах. В данной лабораторной работе рассматривается также способ разработки приложений использующих данный протокол, а также основы организации маршуртизации multicast трафика в корпоративных сетях.

При выполнении данной лабораторной работы студентам доступны:

- персональные компьютеры с одним (двумя) сетевым интерфейсом с клиентской ОС (WXP) – далее **клиенты**;
- персональные компьютеры с несколькими сетевыми интерфейсами (от 2 до 4) с серверной ОС (W2K3/W2K8) – далее **сервера**;
- коммутаторы и концентраторы;
- набор кабелей.

Компьютеры лаборатории должны иметь выход в Интернет или интранет.

Лабораторная работа рассчитана на 3 академических часа.

Примечание 1: задания лабораторной работы связанные с маршрутизацией выполняются фронтально, рекомендуемое количество студентов в группе при выполнении лабораторной работы в данном варианте от 6 до 9 (2-3 студента на группу хостов из сервера и двух клиентов).
Примечание 2: Студенты принимают решение об архитектуре сети сообща под руководством и контролем преподавателя.
Примечание 3: Состав оборудования и конфигурация коммуникационной инфраструктуры могут отличаться от приведенной в описании лабораторной работы, могут использоваться другие операционные системы (Linux, FreeBSD и т.п.), в этом случае требуется корректировка заданий проводящим занятие преподавателем.

1. Обеспечение доступности сетей средствами статической маршрутизации.

В крупных корпоративных сетях (более 8-10 подсетей) применение статической маршрутизации не целесообразно из-за сложности конфигурирования и поддержки (количество статических маршрутов зависит от количества портов маршрутизации). При этом «ситуация усугубляется», если часть интерфейсов доступны непостоянно, либо необходимо учитывать при выборе маршрута динамические характеристики маршрута.

В данной части лабораторной работы предлагается сконфигурировать сеть, состоящую из нескольких подсетей, на которой и будут изучены протоколы динамической

маршрутизации. А также сконфигурировать в ней статические маршруты, обеспечивающие доступность ресурсов любой подсети из любой другой подсети (N*N).

Задание 1. Построение топологии сети для изучения RIP и IGMP.

1. Используя коммутационное оборудование и набор кабелей модифицировать физическую топологию сети лаборатории в соответствии с рисунком 1, организовав 8 отдельных сети (широковещательных домена) A - J, ограниченных маршрутизаторами. Для сетей A, D, E, H использовать неуправляемое сетевое оборудование.
2. Настроить сетевые интерфейсы на хостах следующим образом:
    a. Сеть A: 192.168.4.0/24
    b. Сеть B: 192.168.3.0/24
    c. Сеть C: 192.168.9.0/24
    d. Сеть D: 192.168.5.0/24
    e. Сеть E: 172.17.0.0/16
    f. Сеть F: 172.18.0.0/16
    g. Сеть G: 172.19.0.0/16
    h. Сеть H: 172.20.0.0/16
    i. Сеть I: 10.0.0.0/16
    j. Сеть I: 10.1.0.0/16
3. Адреса интерфейсов назначаются по следующему принципу:
    a. Для серверов (S1-S4) – номер сервера
    b. Для сервера Core – 100

Шлюз по умолчанию на маршрутизаторах не указывать, предполагая, например, что он используется для доступа в другие подсети.

4. Проверить доступность соседних хостов через локальные интерфейсы.

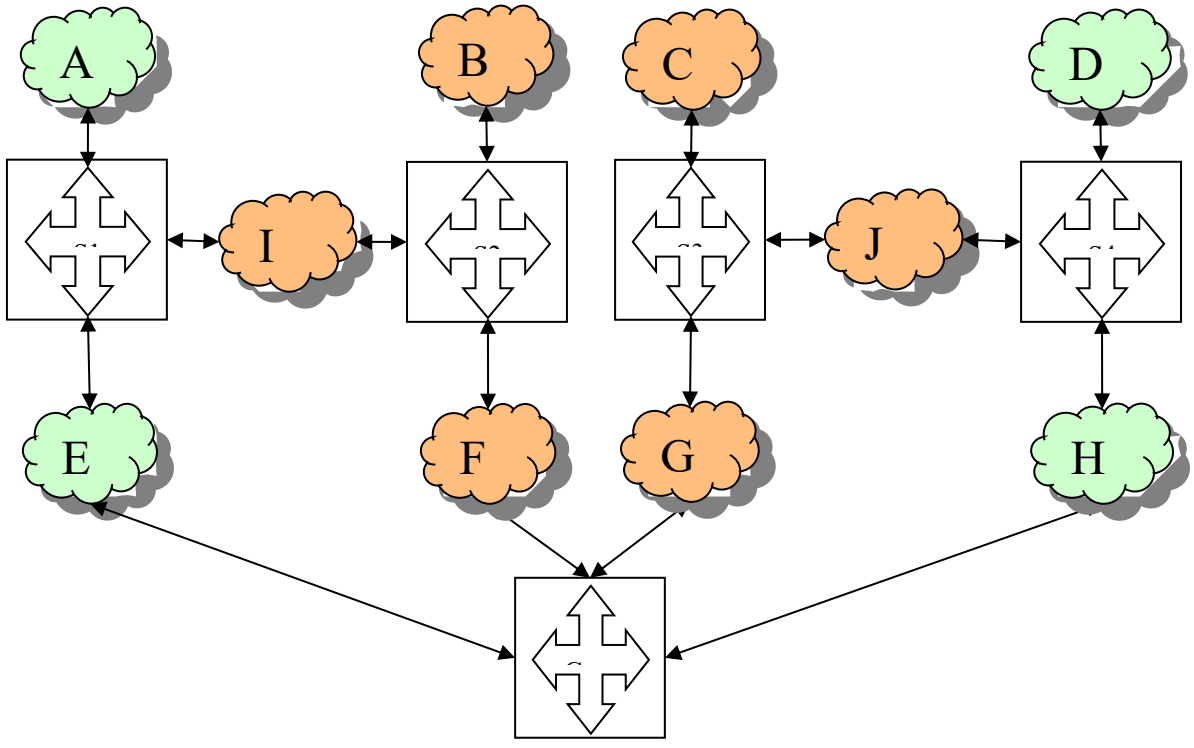

Рисунок 1. Топология сети для первой части лабораторной работы (IGMP, RIP).

5. Включить на серверах службу Routing and Remote Access в режиме маршрутизации (LAN Routing).
6. Настроить статические маршруты на маршрутизаторах (серверах) в симметричном режиме.
7. Подтвердить возможность коммуникаций между сетями.

2. Изучение протоколов динамической маршрутизации: RIP.

Протокол RIP предназначен для организации маршрутизации в сетях сравнительно небольшого размера – радиус сети ограничен максимум 15 маршрутизаторами, при этом маршруты с метрикой 16 считаются недоступными. RIP эксплуатирует достаточно простой принцип формирования маршрутных таблиц «вектор расстояний» (distance vector), которое определяется количеством переходов (иногда с учетом стоимости маршрута / интерфейса – за счет увеличения метрики). Соседние маршрутизаторы обмениваются своими маршрутными таблицами, и если через соседний маршрутизатор существует маршрут более выгодный с точки зрения стоимости (с меньшей метрикой), то маршрутизатор модифицирует свою таблицу маршрутизации.

Существует две версии протокола RIP – версия 1 и версия 2. Основные отличия второй версии от первой заключаются в том, что во второй версии добавлена возможность использования безклассовой адресации (в пакет протокола добавлено поле маски сети), также появилась возможность использования multicast (групповой адрес 224.0.0.9) сообщений для распространения маршрутных таблиц вместо broadcast и unicast.

Кроме интерфейсов с периодической рассылкой сообщений у маршрутизатора могут быть сконфигурированы автостатические интерфейсы (auto-static), в этом случае маршруты, выученные маршрутизатором через этот интерфейс, заносятся в таблицу маршрутизации как статические, через интерфейс не происходит рассылка периодических обновлений таблицы маршрутизации, запрос маршрутов через данный интерфейс происходит по указанию администратора или внешнего по отношению к реализации протокола программного обеспечения. Подобный тип интерфейсов имеет смысл применять в соединениях по требованию (dial-demand), когда взимается плата за трафик и/или время пользования каналом связи.

RIP использует два основных типа пакета – «запрос» (request) и «ответ» (response). «Запросы» применяются при включении маршрутизатора и при использовании автостатических интерфейсов. «Ответы» – как ответ на запрос (unicast), так и как периодические анонсы маршрутов (broadcast или multicast)

С точки зрения безопасности базовая версия протокола RIP имеет серьезные уязвимости. Для повышения защищенности сетевых инфраструктур, использующих данный протокол, в RIP добавлены возможность аутентификации маршрутизаторов при помощи пересылки ключевой информации в открытом виде или с использованиям MD5-хеша (поддерживается не всеми реализациями), также многие реализации поддерживают возможность ограничения диапазона изучаемых от соседних маршрутизаторов маршрутов и ограничение множества соседних маршрутизаторов («белые» и «черные» списки).

На временные характеристики RIP протокола, например, время схождения сети (переход в устойчивое состояние после реконфигурации) влияют следующие временные параметры протокола:

периодичность рассылки таблиц маршрутизации (по умолчанию 30 секунд);

время, через которое неподтвержденный маршрут будет считаться недействительным (по умолчанию 180 секунд);

время, через которое недействительный маршрут будет удален из таблицы маршрутизации (по умолчанию 120 секунд);

время задержки реакции на события изменения таблицы маршрутизации (например, отключение интерфейса), если маршрутизатор поддерживает внеочередную рассылку таблиц маршрутизации при их наступлении (по умолчанию 5 секунд).

Следует отметить, что базовая реализация RIP не лишена недостатка в плане возможности образования колец маршрутизации при изменении коммутации в сети. Для борьбы с кольцами применяется технология «разделения горизонта» (split horizon) – маршрутизатор не посылает маршруты соседнему маршрутизатору, от которого он их выучил, и ее расширение «отравление отправителя» / «обратное давление» (poisoning reverse) – маршрутизатор посылает маршруты соседнему маршрутизатору, от которого он их выучил с максимальной метрикой.

В данной части лабораторной работы предполагается использовать в качестве маршрутизауторов компьютеры с несколькими сетевыми интерфейсами под управление Windows Server 2003 с установленной службой Routing and Remote Access, аналогичные задания можно выполнить и с использованием других операционных систем (например, Linux) и/или аппаратных средств (маршрутизаторов и L3-коммутаторов с поддержкой RIP протокола).

Задание 2. Изучение протокола RIP.

1. Сеть лаборатории для данного задания должна соответствовать изображенной на рисунке 1.
2. На каждом из маршрутизаторов (S1 – S4 и Core) включить поддержку RIP в оснастке RaRA. (General->New Routing Protocol…)
3. Изучить свойства, глобально настраиваемые на уровне протокола (RIP->Properties). Обратить внимание на таймер срабатывания триггера и глобальные параметры безопасности (списки соседних маршрутизаторов от которых можно либо нельзя получать маршруты).
4. Запустить Wireshark на интерфейсах, подключенных к сетям I и J.
5. На маршрутизаторах S1 – S4 передать интерфейсы, подключенные к сетям I и J, под управление RIP протоколом. Обратить внимание на режимы работы протокола на интерфейсе:
    a. на вкладке General (основные): режим работы интерфейса (auto-static / periodic), тип принимаемых и отправляемых пакетов, возможность аутентификации, стоимость интерфейса и т.п.;
    b. на вкладке Security (безопасность): списки маршрутов доступных или запрещенных для изучения через данный интерфейс;
    c. на вкладке Neighborhoods (соседи): списки маршрутизаторов, которым будут отправляться адресные сообщения;
    d. на вкладке Advanced (дополнительные): временные параметры работы RIP, включение дополнительных технологий, таких как разделение горизонта, механизм триггеров и т.п; изменить временные параметры работы RIP:
        i. периодичность рассылки таблиц маршрутизации установить в 15 секунд;
        ii. время, через которое неподтвержденный маршрут будет считаться недействительным, установить в 60 секунд;
        iii. время, через которое недействительный маршрут будет удален из таблицы маршрутизации, установить в 40 секунд.
6. Изучить RIP трафик между интерфейсами в сетях I и J

7. Включить аутентификацию для RIP на данных интерфейсах и изучить изменения в трафике.
8. Выключить аутентификацию и включить использование механизма групповой рассылки вместо широковещательной и изучить изменения в трафике.
9. Включить рассылку пакетов протокола RIP v1 и изучить изменения в трафике.
10. Включить последовательно технологии poisoning reverse и split horizon и изучить изменения в трафике.
11. Вернуть настройки по умолчанию (кроме временных и механизма триггеров), и передать все остальные интерфейсы на всех маршрутизаторах под управление RIP с такими же настройками.
12. Изучить полученные маршруты в сети, убедится в доступности клиентам всех подсетей, изучить таблицы маршрутизации на маршрутизаторах.
13. Повлиять на маршруты в сети изменением метрик интерфейсов.
14. Вернуть метрики в первоначальное состояние.
15. Отключить на всех интерфейсах маршрутизаторов механизм триггеров.
16. Провести эксперименты по времени схождения сети в случае аварийных ситуаций:
    a. На всех интерфейсах всех маршрутизаторов включить перехват пакетов при помощи wireshark.
    b. На клиентах из сетей B и C организовать бесконечный взаимный обмен ICMP эхо-запросами.
    c. Отключить интерфейс от маршрутизатора Core в сеть F и засечь время построения альтернативных маршрутов между B и C.
17. Включить на всех интерфейсах маршрутизаторов механизм триггеров и повторить эксперимент из пункта 16.
18. Повторить эксперименты из пунктов 16 и 17, не отключая интерфейс физически, а блокируя трафик на управляемом коммутаторе.
19. Изучить RIP трафик при отключении службы маршрутизации на маршрутизаторе.
20. (*)Построить модели сети и аварийных ситуаций с образованием колец маршрутизации.

3. Протокол транспортного уровня UDP

UDP (англ. User Datagram Protocol – протокол пользовательских дейтограмм) – это транспортный протокол для передачи данных в сетях IP. Он является одним из самых простых протоколов транспортного уровня модели OSI. UDP не гарантирует доставку пакета и обнаружение недоставки. Это позволяет приложениям, использующим его, гораздо быстрее и эффективнее доставлять данные, если требуется большая пропускная способность линий связи, либо требуется малое время задержки в передачи данных, т.к. в UDP не тратится время на установление соединения и заголовочная часть дейтаграммы содержит всего 8 октетов. Формат заголовка пакета UDP приведен в таблице 2.

Таблица 2. Формат дейтограммы UDP

	Биты 0—15	**16—31**
0	Порт отправителя	Порт получателя
32	Длина пакета	Контрольная сумма
64	Данные (до 65536 байт минус заголовок)	

В качестве **порта отправителя/получателя** используется шестнадцатибитное число, которое определяет приложение, которому следует доставить ту или иную дейтаграмму.

Длина пакета – содержит общую длину дейтаграммы (данные + заголовок)

Контрольная сумма – контрольная сумма при использовании IPv4 вычисляется с использованием, так называемого, псевдозаголовка, в который входят IP адреса отправителя и получателя, комбинации нулей в старших разрядах и номера протокола (17), длина UDP дейтаграммы. Алгоритм вычисления сводятся к получению путем сложения 16 битных слов с переносом разрядов переполнения в младшие разряды по всем словам дейтаграммы и псевдозаголовку. При вычислении сама контрольная сумма принимается равной нулю, при необходимости дейтаграмма выравнивается нулями до длины кратной 2. Поразрядное дополнение полученного числа до 0xffff и является контрольной суммой. Вычисление контрольной суммы необязательно, если контрольная сумма установлена в 0, то считается, что она не вычислялась.

UDP используют следующие протоколы более высокого уровней DNS, DHCP, NTP, TFTP и д.р. UDP в отличие от TCP протокола позволяет производить широковещательную рассылку сообщений, и рассылку по групповым адресам.

Задание 3а. Изучение формата пакета UDP с помощью программы Wireshark.

1. Получить перехваченный трафик UDP-трафик на примере DNS (генерируется командой ping с DNS-именем удаленного компьютера, командой nslookup или любым другим способом).
2. Изучить формат заголовка протокола UDP, изучить возможность применения различных дисплейных фильтров в зависимости от полей заголовка.
3. Изучить реакцию системы в зависимости от настроек брандмауэра (ICMP Outgoing Destination Unreachable), в случае отсутствия прослушивающего сокета на удаленном хосте по данному порту.

4. Разработка приложений использующих протокол UDP

Для обмена данными между процессами, в том числе и по протоколу UDP, можно использовать механизм сокетов (socket). Изначально появившейся в BSD UNIX, данный механизм, в том или ином виде, поддерживается практически всеми современными операционными системами. В операционных системах Microsoft Windows обеспечивается с помощью библиотеки Windows Sockets 2 (winsock2).

Для создания приложения предлагается использовать среду разработки Microsoft Visual Studio 2005 (2008/2010). Работу целесообразно проводить в парах: один студент создает приложение, отправляющее сообщения, другой – принимающее.

Для создания простейшего отправляющего приложения необходимо выполнить следующие шаги.

1. Создайте проект консольного приложения.
2. Подключите заголовочный файл winsock2.h, а в настройках линкера библиотеку ws2_32.lib.
3. Проинициализируйте библиотеку Winsock2

```
WSADATA wsaData;

WSAStartup(MAKEWORD(2,2), &wsaData);
```

4. Создайте сокет для передачи данных по протоколу UDP

```
SendSocket = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
```

5. Проинициализируйте адрес получателя (IP-адрес и порт)

```
sockaddr_in RecvAddr;
RecvAddr.sin_family = AF_INET;
RecvAddr.sin_port = htons(Port);
RecvAddr.sin_addr.s_addr = inet_addr(IPAddr);
```

6. Отправьте заранее подготовленную порцию данных получателю

```
sendto(SendSocket, SendBuf, BufLen, 0, (SOCKADDR *) &RecvAddr, sizeof(RecvAddr));
```

7. Закройте сокет

```
  closesocket(SendSocket);
```

8. Завершите использование ws2_32.dll

```
  WSACleanup();
```

При необходимости для отправки широковещательных сообщений необходимо установить опцию SO_BROADCAST для сокета при помощи метода setsockopt()

```
BOOL OptionValue=true;
setsockopt(SendSocket,SOL_SOCKET,SO_BROADCAST,(char*)& OptionValue,sizeof(OptionValue));
```

Для создания простейшего приложения получателя необходимо выполнить действия аналогичные приведенным выше:

9. Создайте проект консольного приложения.
10. Подключите заголовочный winsock2.h, а в настройках линкера ws2_32.lib.
11. Проинициализируйте библиотеку

```
WSADATA wsaData;
WSAStartup(MAKEWORD(2,2), &wsaData);
```

12. Создайте сокет для получения данных по протоколу UDP

```
RecvSocket = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
```

13. Свяжите сокет с любым адресом и определенным портом

```
  sockaddr_in RecvAddr;
  RecvAddr.sin_family = AF_INET;
  RecvAddr.sin_port = htons(Port);
  RecvAddr.sin_addr.s_addr = htonl(INADDR_ANY);
  bind(RecvSocket, (SOCKADDR *) &RecvAddr, sizeof(RecvAddr));
```

14. Получите данные от отправителя

```
recvfrom(RecvSocket, RecvBuf, BufLen, 0, (SOCKADDR *)&SenderAddr, &SenderAddrSize);
```

15. Закройте сокет

```
closesocket(SendSocket);
```

16. Завершите использование ws2_32.dll

```
WSACleanup();
```

Задание 3б. Создание простейшего приложения для обмена данными по протоколу UDP

1. Создайте простейшее консольные приложения для получения и отправки сообщений из приведенного выше каркаса. Проконтролируйте их работу с помощью программы Wireshark.
2. Определить размер максимальной дейтаграммы, которую можно послать, используя функцию getsockopt() (опция SO_MAX_MSG_SIZE уровня SOL_SOCKET).
3. Создайте простейшее консольное приложение для получения и отправки широковещательных сообщений. Проконтролируйте его работу с помощью программы WireShark.

5. Групповая рассылка данных (Multicasting)

Помимо одноадресной и широковещательной рассылки сообщений IP поддерживает режим групповой рассылки (multicasting). В современных сетях для групповой рассылки в качестве транспортного протокола используется UDP, хотя существует еще ряд протоколов транспортного уровня поддерживающих данный режим работы и обладающих большими возможностями по сравнению с UDP, например PGM, который будет изучен в одной из последующих лабораторных работ.

Для создания UDP-сокета, способного работать в multicast-режиме в winsock версии 2 можно создать его, например, с помощью функции:

```
WSASocket(AF_INET, SOCK_DGRAM, 0, NULL, 0, WSA_FLAG_MULTIPOINT_C_LEAF|
WSA_FLAG_MULTIPOINT_D_LEAF)
```

После этого необходимо привязать сокет с помощью функции bind () к локальному адресу. И присоеденить его к multicast-группе с помощью функции WSAJoinLeaf():

```
struct sockaddr_in  remote;
remote.sin_family      = AF_INET;
remote.sin_port        = htons(12345);
remote.sin_addr.s_addr = inet_addr("234.5.6.7");
WSAJoinLeaf(sock, (SOCKADDR *)&remote, sizeof(remote), NULL, NULL, NULL, NULL, JL_BOTH))
```

После этого можно использовать функции sendto() и recvfrom() для передачи и получения multicast-сообщений.

Для оповещения маршрутизаторов о том, что некоторый хост входит в некоторую multicast-группу применяется протокол Internet Group Management Protocol версий 2 (IGMPv2) и 3 (IGMPv3) описываемых в RFC 2236 [10] и 3376 [11] соответственно. Версия 3 отличается возможностью выбора хостом отправителей, в сообщениях от которых он заинтересован.

Задание 4а. Создание простейшего приложения для обмена данными по протоколу UDP с использование multicast-сообщений.

1. Создайте простейшее консольные приложения для получения и отправки multicast-сообщений. Проконтролируйте их работу с помощью программы WireShark.
2. Обратите внимание на применение пакетов протокола IGMP.
3. Обратите внимание на MAC-адрес multicast IP-адреса. Меняя IP-адрес в приложении проследить принцип его формирования.
4. Сравнить работу приложения в режиме с включенным (по умолчанию) и выключенным замыканием multicast-трафика на себя. (Для отключения использовать функцию setsockopt() (опция IP_MULTICAST_LOOP уровня IPPROTO_IP).
5. Обратить внимание на значение поле TTL в multicast-сообщениях. Сделать соответствующие выводы. Увеличить TTL с помощью функции setsockopt() (опция IP_MULTICAST_TTL уровня IPPROTO_IP). Проследить изменения в IP-дейтаграммах.
6. Изучить использование протокола IGMP при использовании multicast-сообщений. Изучить формат пакетов.
7. Изучить различие в генерируемом multicast-трафике в сети при включенном и выключенном режиме IGMP Snooping на коммутаторах.

6. Изучение механизмов доставки multicast трафика: IGMP Router / Proxy.

Для доставки multicast-трафика в крупных корпоративных сетях необходимо применение 2 протоколов маршрутизации – IGMP, при помощи которого клиенты сообщают маршрутизатору о заинтересованности в том или ином multicast-трафике, и одного из протоколов динамической маршрутизации для multicasting'а – DVMRP, PIM-DM, PIM-SM, MOSPF, и т.п.. IGMP в этом случае ретранслирует информацию о группах протоколам динамической multicast-маршрутизации. В большинстве случаев протоколы динамической маршрутизации для multicast-трафика используют в качестве инфраструктуры для определения маршрутов протоколы маршрутизации для unicast-трафика (RIP, OSPF, и т.п.).

Для простейших сетей, в случае использования одного маршрутизатора в сети, нет необходимости применять на маршрутизаторах для multicasting'а других протоколов, кроме IGMP в режиме Router / Router. Когда используется древовидная коммутация без колец маршрутизации, то для организации multicasting'а между сетями, опять же, возможно использовать маршрутизаторы только с поддержкой IGMP, в этом случае используется механизм IGMP Router / Proxy (Downstream / Upstream). Следует отметить, что данная конфигурация не является рекомендуемой большинством вендоров, но в ряде случаев является вполне работоспособной и позволяет избежать расходов на дорогостоящее оборудование или программное обеспечение.

Служба RaRA в Microsoft Server 2003 / 2008 поддерживает протокол IGMP, протоколы DVMRP, PIM-DM, PIM-SM, MOSPF не поддерживаются. Используемые в данном практикуме коммутаторы Allied Telesys Rapier 24i помимо IGMP, поддерживают протоколы DVMRP, PIM-DM, PIM-SM, но требует активации специальной лицензии.

Подробное изучение протоколов DVMRP, PIM-DM, PIM-SM, MOSPF находится вне рассмотрения данной лабораторной работы.

Задание 4б. Настройка доставки multicast-трафика при древовидной архитектуре сети.

1. На маршрутизаторах включить поддержку IGMP в оснастке RaRA. (General->New Routing Protocol…)
2. Добавить интерфейсы, идущие от ядра сети как маршрутизатор (IGMP Router). Изучить настраиваемые параметры интерфейса в режиме маршрутизатора.
3. Добавить интерфейсы, идущие к ядру сети как прокси (IGMP Proxy).
4. К сетям подключить multicast-клиентов (использовать приложение UDPMulticast из лабораторной работы).
5. На интерфейсах маршрутизаторов при помощи Wireshark'а изучить трафик при пересылке данных между клиентами.
6. Отключить поддержку (Remove) IGMP протокола на маршрутизаторе.
7. Удалить созданные в задании 2 статические маршруты.

Задание 4с. Вернуть сеть лаборатории в исходное состояние.

1. Вернуть сеть лаборатории в исходное состояние в соответствии с рисунком 1.
2. Настроить все хосты на автоматическое получение сетевых настроек от DHCP-серверов либо назначить адреса статически в соответствии с исходной конфигурацией.

Список литературы:

1. Microsoft Developer Network Library
2. Anthony Jones, Jim Ohlund. Network Programming For Microsoft Windows. Microsoft Press 1999.
3. J.C. Mackin, Ian McLean. Self-Paced Training Kit. Implementing, Managing, and Maintaining a Microsoft Widows Server 2003 Network Infrastructure. Microsoft Press. 2006.

Примечание: Дополнить на следующий год заданиями по IPv6

Пример: http://www.v6.facebook.com. 3600 IN AAAA 2620:0:1cfe:face:b00c::3