Лабораторная работа № 5

Построение корпоративной сети с использованием стека протоколов TCP/IP

Цель работы: изучить принципы распределения адресного пространства в сетях TCP/IP, получить навыки формирования подсетей, определения маски подсети; изучить способы маршрутизации данных.

Краткие теоретические сведения

Стек протоколов TCP/IP является наиболее популярным на сегодняшний день, поскольку обеспечивает возможность взаимодействия как локальных, так и глобальных сетей. Наиболее широкое распространение получил стек протоколов TCP/IP версии 4. В состав данного стека входит множество протоколов, наиболее известными из которых являются *протокол управления передачей* (Transmission Control Protocol, TCP) и *протокол межсетевого взаимодействия* (Internet Protocol, IP). Учитывая тенденцию быстрого роста сети Internet, в основу которой положен TCP/IP, разработан стек протоколов TCP/IP версии 6, использующийся пока не в столь значительных масштабах. Основу новой версии стека протоколов составляет *протокол межсетевого взаимодействия нового поколения* (IP new generation, IPng или IPv6). В данной работе будет рассматриваться только TCP/IP версии 4.

Протокол межсетевого взаимодействия (IP) является протоколом сетевого уровня. Этот протокол определяет способ адресации сетей и отдельных узлов и обеспечивает поддержку маршрутизации потоков данных.

Каждому узлу сети TCP/IP (хосту) назначается 32-разрядный логический адрес, называемый IP-адресом и состоящий в общем случае из двух полей – адрес (номер) сети и адрес (номер) узла. Адрес сети уникально идентифицирует каждую сеть и назначается централизованно. Если сеть подключена к Internet, адрес сети может быть получен непосредственно в Сетевом информационном центре (Internet Network Information Center, InterNIC) или же назначен региональным провайдером услуг Internet (ISP), получившим блок адресов у InterNIC. Адрес хоста выбирается администратором сети, но должен быть уникальным в пределах сети. Для записи IP-адреса чаще всего используется десятичная форма записи с разделяющими точками (dotted decimal notation), в соответствии с которой каждый октет представляется десятичным числом в диапазоне от 0 до 255, например, 1.2.3.4. Общий формат IP-адреса представлен на рис. 1.

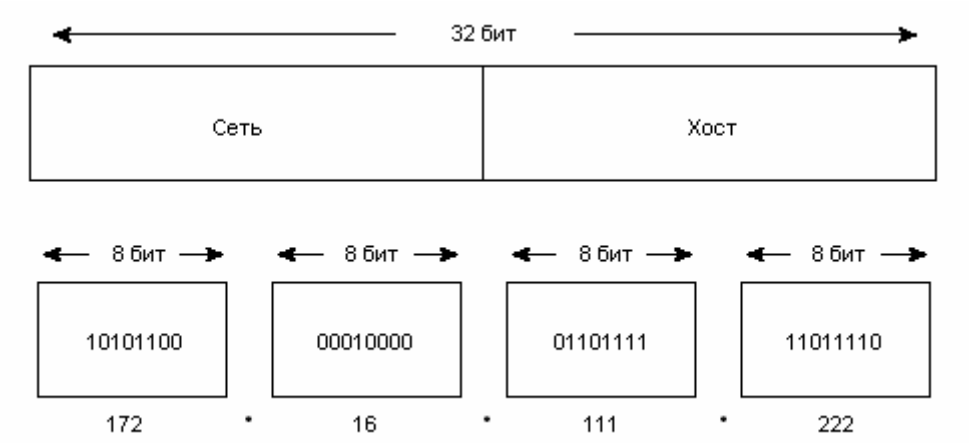


Рис. 1

Для выделения из IP-адреса номера сети и номера узла изначально были определены 5 классов адресов – классы A, B, C, D и E. Для общего пользования были доступны только классы A, B и C. Класс D предназначался для использования в сетях с групповой адресацией, класс E был зарезервирован для дальнейшего использования. Формат адреса для сетей классов A, B и C представлен на рис. 2.

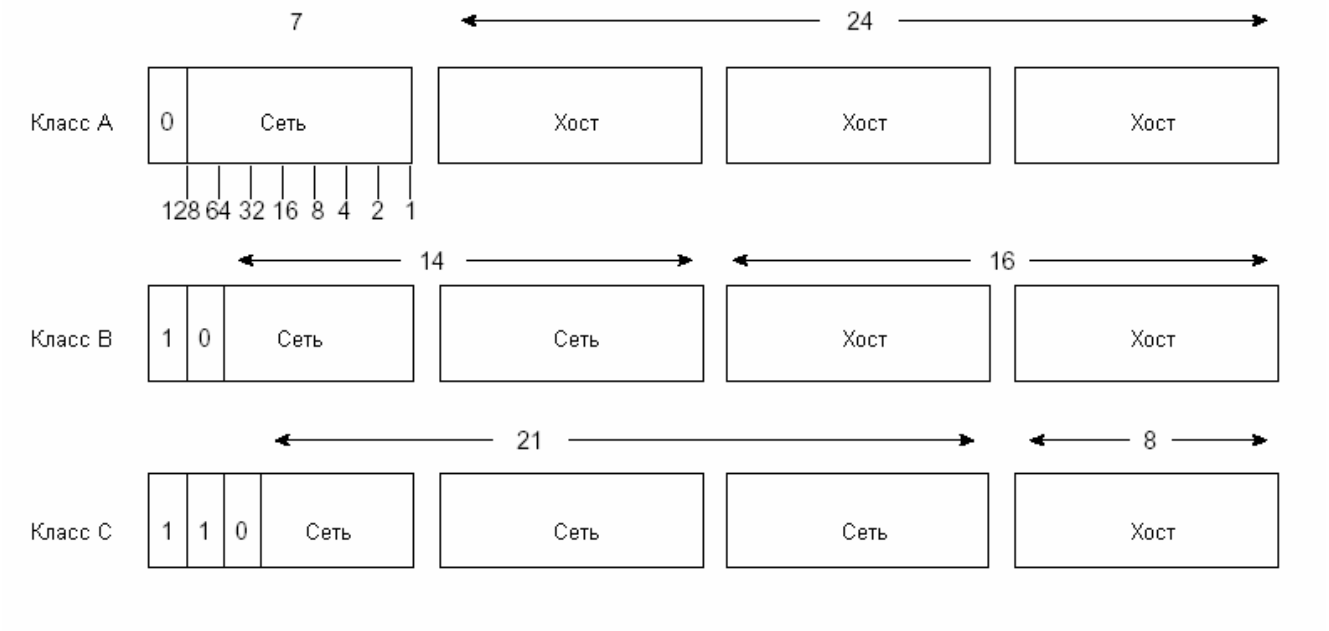


Рис. 2

Класс IP-адреса, а, следовательно, и соотношение между размерностями полей номера сети и номера узла, могут быть легко определены по старшим битам первого октета. Основные характеристики сетей различных классов приведены в табл. 1.

Табл. 1

Класс адреса	Формат адреса	Старшие биты	Диапазон адресов	Количество бит в адресе сети / хоста	Количество сетей	Количество хостов в сети
A	C.X.X.X	0	1.0.0.0 - 126.0.0.0	7/24	126	16 777 214
B	C.C.X.X	10	128.1.0.1 - 191.254.0.0	14/16	16384	65 543
C	C.C.C.X	110	192.0.0.0 - 223.255.254.0	22/8	4194304	254
D	Не определено	1110	224.0.0.0 - 239.255.255.255	Не определено	Не определено	Не определено
E	Не определено	1111	224.0.0.0 - 254.255.255.255	Не определено	Не определено	Не определено

Некоторые IP-адреса имеют специальное назначение и не могут быть использованы для узлов в реальных сетях (табл. 2).

Табл. 2

Тип адреса		**Назначение**
Все нули	0.0.0.0	Данный узел
[Номер сети].[Все нули]	192.168.1.0	Данная сеть
[Все нули].[Номер узла]	0.0.0.1	Узел данной сети
Все единицы	255.255.255.255	Все узлы данной сети
[Номер сети].[Все единицы]	192.168.1.255	Все узлы указанной сети
127.[Любое число]	127.0.0.1	"Петля" (loopback)

Для использования в частных (внутренних) сетях зарезервированы три блока IP-адресов:

10.0.0.0 – 10.255.255.255

172.16.0.0 – 172.31.255.255

192.168.1.0 – 192.168.255.255

Эти адреса предназначены для внутреннего использования и могут назначаться узлам в любой сети без предварительного согласования с какой-либо организацией. Очевидно, что эти адреса не являются уникальными в пределах глобальной сети, поэтому не могут быть использованы для адресации хостов из других сетей. Внешние сетевые шлюзы и маршрутизаторы отбрасывают все пакеты, отправляемые хостами, у которых имеются только внутренние адреса.

IP-сеть может быть разбита на более мелкие фрагменты, называемые подсетями. Каждая подсеть представляется как обычная IP-сеть, что придает большую гибкость схеме распределения адресов, позволяет более эффективно расходовать адреса и избежать их пустой траты. Преимуществом использования подсетей является также то, что для внешнего мира сеть организации представляется как единое целое, благодаря чему скрывается ее реальная внутренняя структура.

Для задания номера подсети выделяется некоторая часть разрядов из поля номера узла. Размерность поля номера подсети изменяется в зависимости от требуемого количества подсетей и количества узлов в каждой из них. Для выделения номера подсети используется *маска подсети* (subnet mask). Формат маски подсети аналогичен формату IP-адреса. Однако, маска подсети в двоичном представлении всегда содержит последовательность "1" в той части, которая соответствует номеру сети и подсети, и последовательность "0" в той части, которая соответствует номеру узла (рис. 3).

Рис. 3

Для сетей классов A, B и C в случае отсутствия подсетей используются маски 255.0.0.0, 255.255.0.0 и 255.255.255.0 соответственно. При необходимости использования подсетей требуется определить количество разрядов в полях номера подсети и номера узла. Заметим, что номера подсетей, состоящие из всех "0" или "1", использовать, как правило, не рекомендуется. Для расчета можно воспользоваться одним из следующих способов:

1. Исходя из необходимости, задается количество подсетей N_s. Длина поля номера подсети определяется по формуле $n_s = \lceil \log_2 (N_s + 2) \rceil$. Поле номера узла будет иметь размерность $n_h = 32 - (n_n + n_s)$ разрядов, где n_n – длина поля номера сети. Следовательно, максимальное количество хостов в каждой подсети равно $N_h = 2^{n_h} - 2$.
2. Исходя из необходимости, задается максимальное количество узлов в каждой подсети N_h. Длина поля номера узла определяется по формуле $n_h = \lceil \log_2 (N_h + 2) \rceil$. Поле номера подсети будет иметь размерность $n_s = 32 - (n_n + n_h)$ разрядов, где n_n – длина поля номера сети. Следовательно, максимальное количество подсетей равно $N_s = 2^{n_s} - 2$.

В данном случае предполагается, что во всех подсетях используются одинаковые (ровные) маски. Это может оказаться неприемлемым в том случае, когда количество узлов в разных подсетях существенно различается. Для устранения этого недостатка могут быть использованы неровные маски или маски переменной длины (variable length subnet mask). В этом случае процесс разбиения сети на подсети выполняется в несколько этапов, когда на первом этапе разбивается базовая сеть, а в дальнейшем полученные подсети в свою очередь разбиваются на подсети.

Например, предположим, что сети организации предоставлен IP-адрес 172.16.0.0. Необходимо определить 10 подсетей одинакового размера. В таком случае для задания номера подсети потребуется выделить 4 бита из поля номера узла, что позволит адресовать до 16 подсетей. Маска для каждой подсети будет задаваться в виде 255.255.240.0. Каждая подсеть позволяет адресовать 16382 узла. Адреса подсетей приведены в табл. 3.

Табл. 3

№	Адрес подсети	Адрес широковещательной передачи для подсети
1.	172.16.0.0	172.16.15.255
2.	172.16.16.0	172.16.31.255
3.	172.16.32.0	172.16.47.255
4.	172.16.48.0	172.16.63.255
5.	172.16.64.0	172.16.79.255
6.	172.16.80.0	172.16.95.255
7.	172.16.96.0	172.16.111.255
8.	172.16.112.0	172.16.127.255
9.	172.16.128.0	172.16.143.255
10.	172.16.144.0	172.16.159.255
11.	172.16.160.0	172.16.175.255
12.	172.16.176.0	172.16.191.255
13.	172.16.192.0	172.16.207.255
14.	172.16.208.0	172.16.223.255
15.	172.16.224.0	172.16.239.255
16.	172.16.240.0	172.16.255.255

В другом случае, предположим, организации предоставлен адрес 192.168.1.0. Адреса необходимо распределить между пятью взаимосвязанными сетями, в трех из которых имеется

50 узлов, в двух оставшихся по 30. Очевидно, что, используя ровные маски подсети, нельзя получить решение, удовлетворяющее условиям задачи. Поэтому необходимо воспользоваться масками переменной длины. На первом этапе при использовании маски подсети 255.255.255.192 сеть разбивается на четыре подсети, в каждой из которых может находиться до 64 хостов (табл. 4).

Табл. 4

№	Адрес подсети	Маска подсети	Адрес широковещательной передачи для подсети
1.	192.168.1.0	255.255.255.192	192.168.1.63
2.	192.168.1.64	255.255.255.192	192.168.1.127
3.	192.168.1.128	255.255.255.192	192.168.1.191
4.	192.168.1.192	255.255.255.192	192.168.1.255

После этого одна из полученных подсетей с помощью маски 255.255.255.224 разбивается еще на две подсети по 32 узла в каждой (табл. 5).

№	Адрес подсети	Маска подсети	Адрес широковещательной передачи для подсети
1.	192.168.1.0	255.255.255.192	192.168.1.63
2.	192.168.1.64	255.255.255.192	192.168.1.127
3.	192.168.1.128	255.255.255.192	192.168.1.191
4.	192.168.1.192	255.255.255.224	192.168.1.223
5.	192.168.1.224	255.255.255.224	192.168.1.255

В данном примере задействованы подсети, номера которых в двоичном представлении содержат все "0" и "1", что может оказаться неприемлемым при наличии аппаратных или программных маршрутизаторов, не поддерживающих бесклассовую маршрутизацию.

Важной функцией сетевого уровня является управление процессом маршрутизации пакетов, передаваемых через сеть. Для соединения источника и получателя должен быть установлен маршрут или набор маршрутов. Обычно маршруты фиксируются в каждом узле с помощью записей в соответствующей таблице маршрутов, показывающей по какому исходящему каналу должен направляться данный пакет. Алгоритмы выбора маршрутов, применяемые для установления соответствующих путей и формирования таблиц маршрутов, основываются на учете меры стоимости каждого маршрута. Стоимость может быть величиной фиксированной и определяться такими параметрами, как длина линии, скорость передачи или

ширина полосы (пропускная способность), безопасность, оцениваемая задержка при передаче сигнала и т.п.

Устройства, решающие данную задачу, называются маршрутизаторами (router). Маршрутизаторы объединяют отдельные сети в общую составную сеть. Внутренняя структура каждой сети в данном случае значения не имеет. К каждому маршрутизатору могут быть присоединены несколько сетей (по крайней мере две). В сложных составных сетях почти всегда существует несколько альтернативных маршрутов для передачи пакетов между двумя конечными узлами. Маршрут - это путь (последовательность маршрутизаторов), который должен пройти пакет от отправителя до пункта назначения. Маршрутизатор выбирает маршрут на основании своего представления о текущей конфигурации сети и соответствующего критерия выбора маршрута. Обычно в качестве критерия выступает время прохождения маршрута, которое в локальных сетях совпадает с длиной маршрута, измеряемой в количестве пройденных узлов маршрутизации (в глобальных сетях принимается в расчет и время передачи пакета по каждой линии связи).

В сети Internet, наиболее развитой на настоящий момент сети дейтаграммного типа, некоторые маршрутизаторы используются для перемещения информации через одну конкретную группу сетей, находящихся под одним и тем же административным началом и управлением (такой объект называется автономной системой – autonomous system). Маршрутизаторы, используемые для обмена информацией в пределах автономных систем, называются внутренними (interior routers); они используют различные протоколы внутренних шлюзов (Interior Gateway Protocol, IGP) для выполнения этой задачи. Маршрутизаторы, которые перемещают информацию между автономными системами, называются внешними (exterior routers); для этого они используют протоколы внешней маршрутизации (Exterior Gateway Protocol, EGP).

Для накопления информации о текущей конфигурации сети маршрутизаторы обмениваются маршрутной информацией между собой по специальному протоколу. Протоколы этого типа называются протоколами обмена маршрутной информацией (или протоколами маршрутизации). Протоколы обмена маршрутной информацией следует отличать от протоколов сетевого уровня, поскольку они служат для обмена служебной информацией, тогда как протоколы сетевого уровня предназначены для передачи пользовательских данных, аналогично протоколам канального уровня.

Для доставки пакетов протокола обмена маршрутной информацией используется протокол сетевого уровня, так как только он может передать информацию между маршрутизаторами, находящимися в разных сетях. Пакет протокола обмена маршрутной информацией помещается в поле данных пакета сетевого уровня. С помощью протоколов

обмена маршрутной информацией маршрутизаторы составляют карту межсетевых связей той или иной степени подробности и принимают решение о том, какому следующему узлу нужно передать пакет.

Протоколы обмена маршрутной информацией стека TCP/IP являются адаптивными, в свою очередь их можно разделить на две группы:

- протоколы вектора расстояний (distance vector);
- протоколы состояния связей (link state).

Протоколы, использующие алгоритм вектора расстояний, предписывают каждому маршрутизатору периодически и широковещательно рассылать по сети вектор расстояний от себя до всех известных ему сетей. Под расстоянием обычно понимается число промежуточных маршрутизаторов, через которые пакет должен пройти прежде, чем попадет в соответствующую сеть (hop). Может использоваться и другая метрика, учитывающая не только число промежуточных пунктов, но и время прохождения пакетов по связи между соседними маршрутизаторами. Получив вектор от соседнего маршрутизатора, каждый маршрутизатор добавляет к нему информацию об известных ему других сетях, о которых он узнал непосредственно (если они подключены к его портам) или из аналогичных объявлений других маршрутизаторов, а затем снова рассылает новое значение вектора по сети. В конце концов каждый маршрутизатор узнает информацию об имеющихся в интерсети сетях и о расстоянии до них через соседние маршрутизаторы.

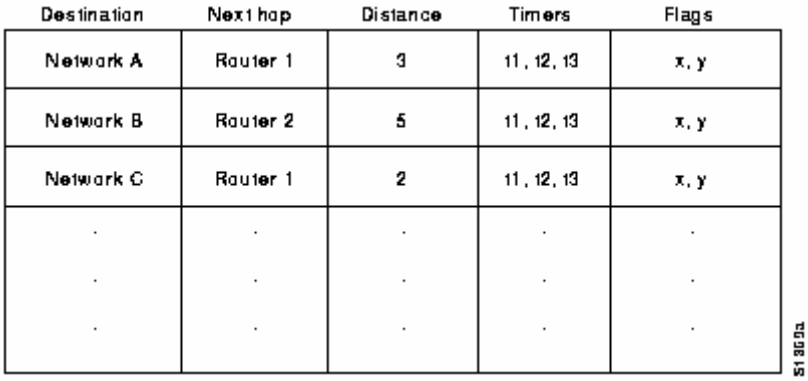
Алгоритмы вектора расстояний хорошо работают только в небольших сетях. В больших сетях они засоряют линии связи интенсивным широковещательным трафиком, к тому же изменения конфигурации могут отрабатываться по этому алгоритму не всегда корректно, так как маршрутизаторы не имеют точного представления о топологии связей в сети, а располагают только обобщенной информацией – вектором дистанций, к тому же полученной через посредников. Работа маршрутизатора в данном случае напоминает работу моста, так как точной топологической картины сети такой маршрутизатор не имеет.

Наиболее распространенным протоколом, основанным на алгоритме вектора расстояний, является протокол RIP (Routing Information Protocol – протокол информации маршрутизации), описанный в RFC 1058. Это один из старейших протоколов обмена маршрутной информацией, однако он до сих пор чрезвычайно распространен в вычислительных сетях. Существуют разновидности протокола RIP для различных сетей и различных стеков протоколов.

Согласно протоколу RIP все сети имеют номера (способ образования номера зависит от используемого в сети протокола сетевого уровня), а все маршрутизаторы – идентификаторы. Вектор расстояний представляет собой набор пар чисел, соответствующих номерам сетей и расстояниям до них. Вектора расстояний итерационно распространяются маршрутизаторами по

сети, и через несколько шагов каждый маршрутизатор имеет данные о достижимых для него сетях и о расстояниях до них. Если связь с какой-либо сетью обрывается, то маршрутизатор отмечает этот факт тем, что присваивает элементу вектора, соответствующему расстоянию до этой сети, максимально возможное значение, которое имеет специальный смысл – "связи нет". Таким значением в протоколе RIP является число 16.

Каждая запись данных в таблице маршрутизации RIP содержит необходимую информацию, включая конечный пункт назначения, следующую пересылку на пути к этому пункту назначения и метрику. В таблице маршрутизации может находиться также и другая информация, в том числе различные таймеры, связанные с данным маршрутом. Типичная таблица маршрутизации RIP показана на рис. 4.



Destination	Next hop	Distance	Timers	Flags
Network A	Router 1	3	11, 12, 13	x, y
Network B	Router 2	5	11, 12, 13	x, y
Network C	Router 1	2	11, 12, 13	x, y
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

S1 3693a

Рис. 4

RIP хранит только самые лучшие маршруты к пункту назначения. Если новая информация обеспечивает лучший маршрут, то эта информация заменяет старую маршрутную информацию. Изменения в топологии сети могут вызывать изменения в маршрутах, приводя к тому, например, что какой-нибудь новый маршрут становится лучшим маршрутом до конкретного пункта назначения. Когда имеют место изменения в топологии сети, то эти изменения отражаются в сообщениях о корректировке маршрутизации. Например, когда какой-нибудь маршрутизатор обнаруживает отказ одного из каналов или другого маршрутизатора, он повторно вычисляет свои маршруты и отправляет сообщения о корректировке маршрутизации. Каждый маршрутизатор, принимающий сообщение об обновлении маршрутизации, в котором содержится изменение, корректирует свои таблицы и распространяет это изменение.

RIP определяет ряд характеристик, предназначенных для более стабильной работы в условиях быстро изменяющейся топологии сети. В их число входит ограничение числа пересылок, временные удерживания изменений (hold-downs), расщепленные горизонты (split-horizons) и корректировки отмены (poison reverse updates).

Ограничение числа пересылок

RIP разрешает максимальное число пересылок, равное 15. Любому пункту назначения, который находится дальше, чем на расстоянии 15 пересылок, присваивается ярлык

"недосягаемого". Максимальное число пересылок RIP в значительной мере ограничивает его применение в крупных объединенных сетях, однако способствует предотвращению появления проблемы, называемой счетом до бесконечности (count to infinity), приводящей к зацикливанию маршрутов в сети. Проблема счета до бесконечности представлена на рис. 5.

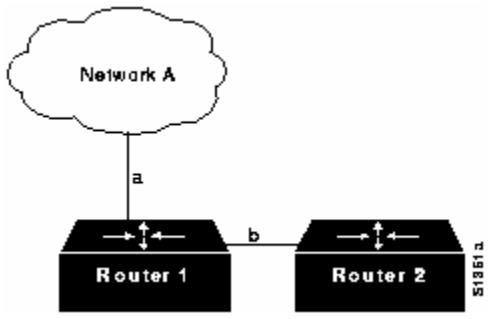


Рис. 5

Рассмотрим, что случится, если на рис. 3 канал *a* маршрутизатора 1 (R1), связывающий его с сетью A, откажет. R1 проверяет свою информацию и обнаруживает, что маршрутизатор 2 (R2) связан с сетью A каналом длиной в одну пересылку. Так как R1 знает, что он напрямую соединен с R2, то он объявляет о маршруте из двух пересылок до сети A и начинает направлять весь трафик в сеть A через R2. Это приводит к образованию маршрутной петли. Когда R2 обнаруживает, что R1 может теперь достичь сеть A за две пересылки, он изменяет запись своих собственных данных в таблице маршрутизации, чтобы показать, что он имеет тракт длиной в 3 пересылки до сети A. Эта проблема, а также данная маршрутная петля будут продолжаться бесконечно, или до тех пор, пока не будет навязано какое-нибудь внешнее граничное условие. Этим граничным условием является максимальное число пересылок RIP. Когда число пересылок превысит 15, данный маршрут маркируется как недосягаемый. Через некоторое время этот маршрут удаляется из таблицы.

Временные удерживания изменений

Временные удерживания изменений используются для того, чтобы помешать регулярным сообщениям о корректировке незаконно восстановить в правах маршрут, который оказался испорченным. Когда какой-нибудь маршрут отказывает, соседние маршрутизаторы обнаруживают это. Затем они вычисляют новые маршруты и отправляют сообщения об обновлении маршрутизации, чтобы информировать своих соседей об изменениях в маршруте. Эта деятельность приводит к появлению целой волны коррекций маршрутизации, которые фильтруются через сеть.

Приведенные в действие корректировки неодновременно прибывают во все устройства сети. Поэтому возможно, что какое-нибудь устройство, которое еще не получило информацию о каком-нибудь отказе в сети, может отправить регулярное сообщение о корректировке (в котором маршрут, который только что отказал, все еще числится исправным) в другое

устройство, которое только что получило уведомление об этом отказе в сети. В этом случае это другое устройство теперь будет иметь (и возможно, рекламировать) неправильную маршрутную информацию.

Команды о временном удерживании указывают маршрутизаторам, чтобы они на некоторое время придержали любые изменения, которые могут оказать влияние на только что удаленные маршруты. Этот период удерживания обычно рассчитывается таким образом, чтобы он был больше периода времени, необходимого для внесения какого-либо изменения о маршрутизации во всю сеть. Удерживание изменений предотвращает появление проблемы счета до бесконечности.

Расщепленные горизонты

Расщепленные горизонты используют преимущество того факта, что никогда не бывает полезным отправлять информацию о каком-нибудь маршруте обратно в том направлении, из которого пришла эта информация. Для иллюстрации этого положения рассмотрим рис. 6.

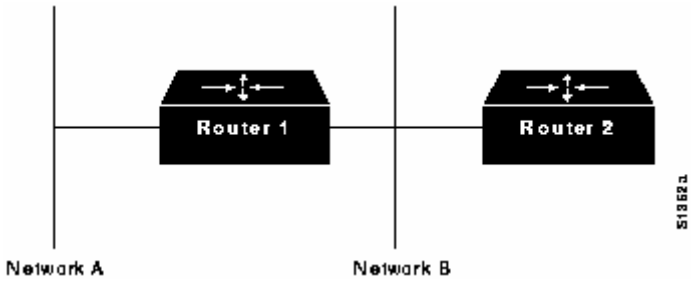


Рис. 6

Маршрутизатор 1 (R1) первоначально объявляет, что он располагает каким-то маршрутом до сети A. Маршрутизатору 2 (R2) нет оснований включать этот маршрут в свою корректировку, отсылаемую обратно маршрутизатору R1, так как R1 ближе к сети A. Правило расщепленного горизонта гласит, что R2 должен исключить (попасть на) этот маршрут при любых корректировках, которые он отправляет в R1.

Правило расщепленного горизонта помогает предотвратить маршрутные петли между двумя узлами. Например, рассмотрим случай, когда отказывает интерфейс R1 с сетью A. При отсутствии расщепленных горизонтов R2 продолжает информировать R1 о том, что он может попасть в сеть A через R1. Если R1 не располагает достаточным интеллектом, то он действительно может выбрать маршрут, предлагаемый R2, в качестве альтернативы для своей отказавшей прямой связи, что приводит к образованию петли маршрутизации. И хотя временное удерживание изменений должно предотвращать это, применение расщепленного горизонта обеспечивает дополнительную стабильность алгоритма.

Корректировки отмены маршрута

В то время как задачей расщепленных горизонтов является предотвращение образования маршрутных петель между соседними маршрутизаторами, корректировки отмены предназначены для устранения более крупных маршрутных петель. В основе их действия лежит положение о том, что увеличение значения показателей маршрутизации обычно указывает на наличие маршрутных петель. В этом случае отправляются корректировки отмены для удаления данного маршрута и помещения его в состояние временного удерживания.

На рис. 7 приведен пример сети, состоящей из шести маршрутизаторов, имеющих идентификаторы от 1 до 6, и из шести сетей от A до F, образованных прямыми связями типа "точка-точка".

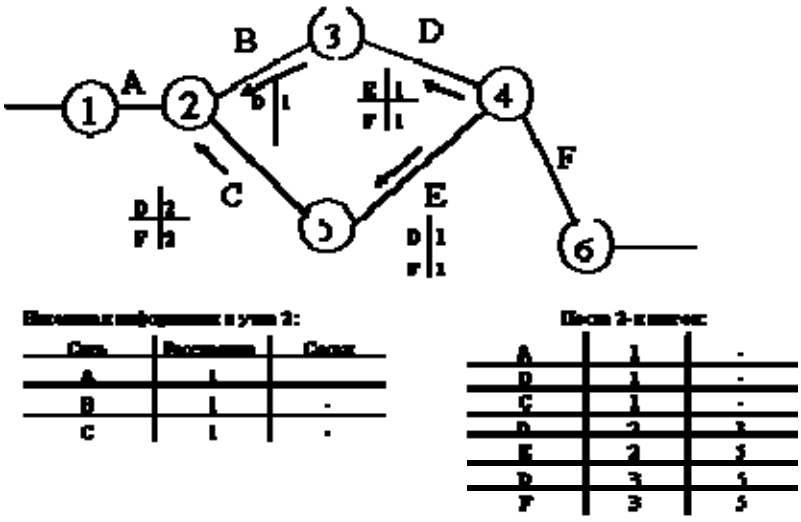


Рис. 7

На рисунке приведена начальная информация, содержащаяся в таблице маршрутизации маршрутизатора 2, а также состояние после двух итераций обмена маршрутными пакетами протокола RIP. После определенного числа итераций маршрутизатор 2 будет знать о расстояниях до всех сетей интерсети, причем у него может быть несколько альтернативных вариантов отправки пакета к сети назначения. Пусть в нашем примере сетью назначения является сеть D.

При необходимости отправить пакет в сеть D маршрутизатор просматривает свою базу данных маршрутов и выбирает порт, имеющий наименьшее расстояния до сети назначения (в данном случае порт, связывающий его с маршрутизатором 3).

Для адаптации к изменению состояния связей и оборудования с каждой записью таблицы маршрутизации связан таймер. Если за время тайм-аута не придет новое сообщение, подтверждающее этот маршрут, то он удаляется из маршрутной таблицы.

Протокол RIP использует эвристический алгоритм Беллмана-Форда, и решение, найденное с его помощью является не оптимальным, а близким к оптимальному.

Преимуществом протокола RIP является его вычислительная простота, а недостатками - увеличение трафика при периодической рассылке широковещательных пакетов и неоптимальность найденного маршрута.

На рис. 8 показан случай неустойчивой работы сети по протоколу RIP при изменении конфигурации - отказе линии связи маршрутизатора М1 с сетью 1. При работоспособном состоянии этой связи в таблице маршрутов каждого маршрутизатора есть запись о сети с номером 1 и соответствующим расстоянием до нее

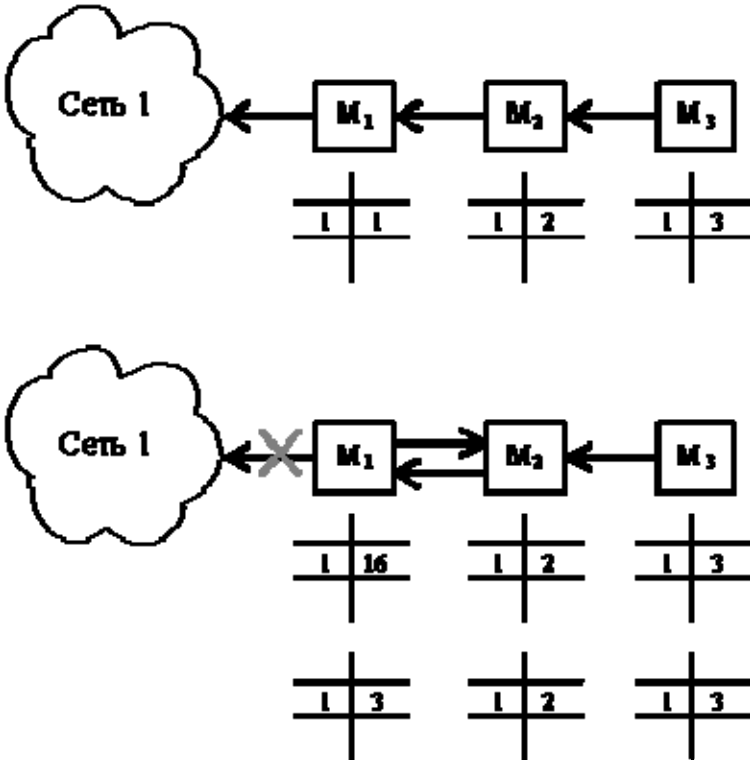


Рис. 8

При обрыве связи с сетью 1 маршрутизатор М1 отмечает, что расстояние до этой сети приняло значение 16. Однако получив через некоторое время от маршрутизатора М2 маршрутное сообщение о том, что от него до сети 1 расстояние составляет 2 пересылки, маршрутизатор М1 наращивает это расстояние на 1 и отмечает, что сеть 1 достижима через маршрутизатор 2. В результате пакет, предназначенный для сети 1, будет циркулировать между маршрутизаторами М1 и М2 до тех пор, пока не истечет время хранения записи о сети 1 в маршрутизаторе 2, и он не передаст эту информацию маршрутизатору М1.

Для исключения подобных ситуаций маршрутная информация об известной маршрутизатору сети не передается тому маршрутизатору, от которого она пришла.

Существуют и другие, более сложные случаи нестабильного поведения сетей, использующих протокол RIP, при изменениях в состоянии связей или маршрутизаторов сети.

Алгоритмы состояния связей обеспечивают каждый маршрутизатор информацией, достаточной для построения точного графа связей сети. Все маршрутизаторы работают на

основании одинаковых графов, что делает процесс маршрутизации более устойчивым к изменениям конфигурации. Широковещательная рассылка используется здесь только при изменениях состояния связей, что происходит в надежных сетях не так часто.

Для того, чтобы понять, в каком состоянии находятся линии связи, подключенные к его портам, маршрутизатор периодически обменивается короткими пакетами со своими ближайшими соседями. Этот трафик также широковещательный, но он циркулирует только между соседями и поэтому не так засоряет сеть.

Протоколом, основанным на алгоритме состояния связей, в стеке TCP/IP является протокол OSPF (Open Shortest Path First), обладающий многими особенностями, ориентированными на применение в больших гетерогенных сетях.

Протокол OSPF вычисляет маршруты в IP-сетях, сохраняя при этом другие протоколы обмена маршрутной информацией.

Непосредственно связанные (то есть достижимые без использования промежуточных маршрутизаторов) маршрутизаторы называются "соседями". Каждый маршрутизатор хранит информацию о том, в каком состоянии по его мнению находится сосед. Маршрутизатор полагается на соседние маршрутизаторы и передает им пакеты данных только в том случае, если он уверен, что они полностью работоспособны. Для выяснения состояния связей маршрутизаторы-соседи достаточно часто обмениваются короткими сообщениями HELLO.

Для распространения по сети данных о состоянии связей маршрутизаторы обмениваются сообщениями другого типа. Эти сообщения называются объявлениями о связях маршрутизатора (router links advertisement), или, точнее, о состоянии связей. OSPF-маршрутизаторы обмениваются не только своими, но и чужими объявлениями о связях, получая в конце концов информацию о состоянии всех связей сети. Эта информация и образует граф связей сети, который, естественно, один и тот же для всех маршрутизаторов сети.

Кроме информации о соседях, маршрутизатор в своем объявлении перечисляет IP-подсети, с которыми он связан непосредственно, поэтому после получения информации о графе связей сети, вычисление маршрута до каждой сети производится непосредственно по этому графу по алгоритму Дэйкстры. Более точно, маршрутизатор вычисляет путь не до конкретной сети, а до маршрутизатора, к которому эта сеть подключена. Каждый маршрутизатор имеет уникальный идентификатор, который передается в объявлении о состояниях связей. Такой подход дает возможность не тратить IP-адреса на связи типа "точка-точка" между маршрутизаторами, к которым не подключены рабочие станции.

Маршрутизатор вычисляет оптимальный маршрут до каждой адресуемой сети, но запоминает только первый промежуточный маршрутизатор из каждого маршрута. Таким образом, результатом вычислений оптимальных маршрутов является список строк, в которых

указывается номер сети и идентификатор маршрутизатора, которому нужно переслать пакет для этой сети. Указанный список маршрутов и является маршрутной таблицей, но вычислен он на основании полной информации о графе связей сети, а не частичной информации, как в протоколе RIP.

Описанный подход приводит к результату, который не может быть достигнут при использовании протокола RIP или других дистанционно-векторных алгоритмов. RIP предполагает, что все подсети определенной IP-сети имеют один и тот же размер, то есть, что все они могут потенциально иметь одинаковое число IP-узлов, адреса которых не перекрываются. Более того, классическая реализация RIP требует, чтобы выделенные линии "точка-точка" имели IP-адрес, что приводит к дополнительным затратам IP-адресов.

В OSPF такие требования отсутствуют: сети могут иметь различное число хостов и могут перекрываться. Под перекрытием понимается наличие нескольких маршрутов к одной и той же сети. В этом случае адрес сети в пришедшем пакете может совпасть с адресом сети, присвоенным нескольким портам.

Если адрес принадлежит нескольким подсетям в базе данных маршрутов, то продвигающий пакет маршрутизатор использует наиболее специфический маршрут, то есть адрес подсети, имеющей более длинную маску.

Например, если рабочая группа ответвляется от главной сети, то она имеет адрес главной сети наряду с более специфическим адресом, определяемым маской подсети. При выборе маршрута к хосту в подсети этой рабочей группы маршрутизатор найдет два пути, один для главной сети и один для рабочей группы. Так как последний более специфичен, то он и будет выбран. Этот механизм является обобщением понятия "маршрут по умолчанию", используемого во многих сетях.

Использование подсетей с различным количеством хостов является вполне естественным. Например, если в здании или кампусе на каждом этаже имеются локальные сети, и на некоторых этажах компьютеров больше, чем на других, то администратор может выбрать размеры подсетей, отражающие ожидаемые требования каждого этажа, а не соответствующие размеру наибольшей подсети.

В протоколе OSPF подсети делятся на три категории:

- "хост-сеть", представляющая собой подсеть из одного адреса,
- "тупиковая сеть", которая представляет собой подсеть, подключенную только к одному маршрутизатору,
- "транзитная сеть", которая представляет собой подсеть, подключенную к более чем одному маршрутизатору.

Транзитная сеть является для протокола OSPF особым случаем. В транзитной сети несколько маршрутизаторов являются взаимно и одновременно достижимыми. В широковещательных локальных сетях, таких как Ethernet или Token Ring, маршрутизатор может послать одно сообщение, которое получат все его соседи. Это уменьшает нагрузку на маршрутизатор, когда он посылает сообщения для определения существования связи или обновленные объявления о соседях. Однако, если каждый маршрутизатор будет перечислять всех своих соседей в своих объявлениях о соседях, то объявления займут много места в памяти маршрутизатора. При определении пути по адресам транзитной подсети может обнаружиться много избыточных маршрутов к различным маршрутизаторам. На вычисление, проверку и отбраковку этих маршрутов уйдет много времени.

Когда маршрутизатор начинает работать в первый раз (то есть инсталлируется), он пытается синхронизировать свою базу данных со всеми маршрутизаторами транзитной локальной сети, которые по определению имеют идентичные базы данных. Для упрощения и оптимизации этого процесса в протоколе OSPF используется понятие "выделенного" маршрутизатора, который выполняет две функции.

Во-первых, выделенный маршрутизатор и его резервный "напарник" являются единственными маршрутизаторами, с которыми новый маршрутизатор будет синхронизировать свою базу. Синхронизировав базу с выделенным маршрутизатором, новый маршрутизатор будет синхронизирован со всеми маршрутизаторами данной локальной сети.

Во-вторых, выделенный маршрутизатор делает объявление о сетевых связях, перечисляя своих соседей по подсети. Другие маршрутизаторы просто объявляют о своей связи с выделенным маршрутизатором. Это делает объявления о связях (которых много) более краткими, размером с объявление о связях отдельной сети.

Для начала работы маршрутизатора OSPF нужен минимум информации - IP-конфигурация (IP-адреса и маски подсетей), некоторая информация по умолчанию (default) и команда на включение. Для многих сетей информация по умолчанию весьма похожа. В то же время протокол OSPF предусматривает высокую степень программируемости.

Интерфейс OSPF (порт маршрутизатора, поддерживающего протокол OSPF) является обобщением подсети IP. Подобно подсети IP, интерфейс OSPF имеет IP-адрес и маску подсети. Если один порт OSPF поддерживает более, чем одну подсеть, протокол OSPF рассматривает эти подсети так, как если бы они были на разных физических интерфейсах, и вычисляет маршруты соответственно.

Интерфейсы, к которым подключены локальные сети, называются широковещательными (broadcast) интерфейсами, так как они могут использовать широковещательные возможности локальных сетей для обмена сигнальной информацией

между маршрутизаторами. Интерфейсы, к которым подключены глобальные сети, не поддерживающие широковещание, но обеспечивающие доступ ко многим узлам через одну точку входа, например сети X.25 или frame relay, называются нешироковещательными интерфейсами с множественным доступом или NBMA (non-broadcast multi-access). Они рассматриваются аналогично широковещательным интерфейсам за исключением того, что широковещательная рассылка эмулируется путем посылки сообщения каждому соседу. Так как обнаружение соседей не является автоматическим, как в широковещательных сетях, NBMA-соседи должны задаваться при конфигурировании вручную. Как на широковещательных, так и на NBMA-интерфейсах могут быть заданы приоритеты маршрутизаторов для того, чтобы они могли выбрать выделенный маршрутизатор.

Интерфейсы "точка-точка", подобные PPP, несколько отличаются от традиционной IP-модели. Хотя они и могут иметь IP-адреса и подмаски, но необходимости в этом нет.

В простых сетях достаточно определить, что пункт назначения достижим и найти маршрут, который будет удовлетворительным. В сложных сетях обычно имеется несколько возможных маршрутов. Иногда хотелось бы иметь возможности по установлению дополнительных критериев для выбора пути: например, наименьшая задержка, максимальная пропускная способность или наименьшая стоимость (в сетях с оплатой за пакет). По этим причинам протокол OSPF позволяет сетевому администратору назначать каждому интерфейсу определенное число, называемое метрикой, чтобы оказать нужное влияние на выбор маршрута.

Число, используемое в качестве метрики пути, может быть назначено произвольным образом по желанию администратора. Но по умолчанию в качестве метрики используется время передачи бита в 10-ти наносекундных единицах (10 Мб/с Ethernet назначается значение 10, а линии 56 Кб/с - число 1785). Вычисляемая протоколом OSPF метрика пути представляет собой сумму метрик всех проходимых в пути связей; это очень грубая оценка задержки пути. Если маршрутизатор обнаруживает более, чем один путь к удаленной подсети, то он использует путь с наименьшей стоимостью пути.

В протоколе OSPF используется несколько временных параметров, и среди них наиболее важными являются интервал сообщения HELLO и интервал отказа маршрутизатора (router dead interval).

HELLO - это сообщение, которым обмениваются соседние, то есть непосредственно связанные маршрутизаторы подсети, с целью установить состояние линии связи и состояние маршрутизатора-соседа. В сообщении HELLO маршрутизатор передает свои рабочие параметры и говорит о том, кого он рассматривает в качестве своих ближайших соседей. Маршрутизаторы с разными рабочими параметрами игнорируют сообщения HELLO друг друга, поэтому неверно сконфигурированные маршрутизаторы не будут влиять на работу сети.

Каждый маршрутизатор шлет сообщение HELLO каждому своему соседу по крайней мере один раз на протяжении интервала HELLO. Если интервал отказа маршрутизатора истекает без получения сообщения HELLO от соседа, то считается, что сосед неработоспособен, и распространяется новое объявление о сетевых связях, чтобы в сети произошел пересчет маршрутов.

Задание на работу

1. В соответствии с вариантом задания построить сеть предприятия с использованием средств маршрутизации потоков данных, обеспечив резервирование основных каналов связи.
2. Произвести распределение IP-адресов между узлами построенной сети.
3. Для полученной модели сети задать необходимые типы потоков данных между рабочими станциями и серверами и произвести имитационное моделирование работы сети.
4. Проанализировать среднюю загрузку сетевого коммуникационного оборудования и среды передачи данных, сравнить интенсивности потоков служебных данных при использовании различных протоколов обмена маршрутной информацией. Указать участки сети, уязвимые к перегрузкам, и определить средства повышения надежности функционирования сети.

Таблица 1. Варианты заданий.

№ варианта	Тип инфраструктуры	Тип трафика
1	1	2
2	2	3
3	3	4
4	4	1
5	5	3
6	6	4
7	7	1
8	8	2
9	1	4
10	2	1
11	3	2
12	8	3
13	5	1
14	6	2
15	7	3

Таблица 2. Тип инфраструктуры.

№ варианта	**Количество подсетей**	**Количество узлов в подсети**	**Предоставленные адреса**
1	4	60, 40, 30, 30	192.168.11.0
2	5	100, 100, 50, 50, 50	172.22.0.0
3	6	12, 14, 30, 60, 60, 60	192.168.13.0
4	4	60, 60, 50, 50	172.24.0.0
5	5	30, 30, 60, 60, 60	192.168.15.0
6	4	30, 30, 120, 160	172.26.0.0
7	4	30, 30, 60, 120	192.168.17.0
8	6	30, 30, 60, 60, 60, 120	172.28.0.0

Таблица 3. Тип моделируемого трафика.

№ варианта	Количество файловых серверов	Количество HTTP-серверов	Количество FTP-серверов	Количество серверов баз данных
1	3	1	2	2
2	3	2	1	2
3	2	1	2	3
4	2	2	1	3

Контрольные вопросы

1. Правила распределения адресов в сетях IP.
2. Классификация IP-адресов, понятие маски подсети.
3. Использование неровных масок подсетей.
4. Задача маршрутизации, способы маршрутизации.
5. Протоколы маршрутизации стека TCP/IP.

Дополнительная литература

1. Буров Є. Комп'ютерні мережі – Л.: БаК, 1999.
2. Кулаков Ю.А., Омелянский С. В. Компьютерные сети. Выбор, установка, использование и администрирование – К.: Юниор, 1999.
3. Кульгин М. Технологии корпоративных сетей – СПб.: Питер, 2000.
4. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы – СПб.: Питер, 1999.
5. CISCO Internetworking technology overview – Cisco, 1999.