

Информатика

Николай Борисович Линев

*Преподаватель подготовительных курсов
факультета вычислительной математики
и кибернетики МГУ им. М.В. Ломоносова.*



Путешествия пакетов по сети: готовим маршруты

Многие современные люди не представляют себе жизни без Интернета. Ведь так просто: сел за компьютер, вызвал программу-браузер, набрал адрес и... поехали: хочешь – ищи и просматривай информацию, хочешь – посылай и получай письма, хочешь – оформляй покупки на дом. За внешне простыми действиями скрывается сложный механизм, и эта статья поможет вам в нём разобраться.

Без чего в Интернете «гулять» невозможно

Давайте предположим, что вы хотите сесть дома за компьютер и найти какую-то информацию в Интернете. Для этого ваш компьютер должен быть подключён к Интернету, то есть должны выполняться следующие условия.

Ваш компьютер должен иметь сетевую карту. А поскольку вы хотите поискать что-то в Интернете, то за доставку данных будет отвечать Internet Protocol (IP), и вашей сетевой карте должен быть назначен IP-адрес. Если у вашего компьютера сетевая карта одна, то можно упрощённо говорить, что адрес назначен самому компьютеру.

IP-адрес представляет собой 32-битное двоичное число, однозначно определяющее сетевую карту в сети. По стандарту IP два компьютера могут напрямую обмениваться дан-

ными, если они находятся в одной IP-сети. Другими словами, у них должны совпадать идентификаторы (или адреса) сетей.

Идентификатор сети определяется логическим побитовым произведением IP-адреса на маску сети, сопровождающую IP-адрес. Маска сети также представляет собой 32-битное двоичное число, но в отличие от IP-адреса в маске запрещено чередование 0 и 1. То есть маска начинается с цепочки единиц, за которой следует цепочка нулей.

Для того чтобы получить идентификатор сети, требуется перевести десятичный вид IP-адреса в двоичный, выписать под ним двоичный вид сетевой маски и провести операцию побитового умножения. Выполнить эту операцию нужно дважды: для адреса, назначенного сете-

вой карте вашего компьютера, и для адреса сетевой карты, куда мы хотим отправить данные. Ведь для того чтобы получить, например, страничку какого-то сайта, нам сначала потребуется отправить на этот сайт запрос на получение нужных нам данных. Поскольку информации о маске сети, куда мы собираемся что-то отправлять, у нас нет, то используем маску с настроек своей сетевой карты (более подробно это обсуждалось в февральском номере журнала за 2010 год). И если идентификаторы сетей не совпали (а при отправке запроса к какому-либо ресурсу Интернета почти всегда так и будет), то дальнейшая отправка данных без специального устройства – маршрутизатора – становится невозможной. А значит, для связи с ресурсами Интернета или с другими сетями ваш компьютер (введём упрощение, что на компьютере установлен один сетевой адаптер) должен быть подключён к маршрутизатору. В просторечии маршрутизатор ещё называют *роутер*, от неправильного прочтения английского названия устройства – *Router*. Это может быть установленный у вас дома, например, ADSL-модем, работающий в режиме маршрутизатора, Интернет-центр или ещё какое-либо устройство класса *SOHO* (*Small Office/Home Office*). В домашних сетях обычно ставят устройства именно этого класса (рис. 1). А ваш домашний маршрутизатор подключён к маршрутизатору сети провайдера услуг Интернета. Возможен вариант, что вы сразу подключены через сеть коммутаторов к более мощному маршрутизатору провайдера. Это обычная ситуация при подключении пользователя по «витой паре», если у вас к Интернету подключён один компьютер. Но в любом случае маршрутизатор является устройством, стоящим на гра-

нице локальной сети, вашей домашней или провайдера, и обеспечивающим обмен данными с компьютерами, расположенными за её пределами. Если он установлен у вас дома, то вы являетесь тем человеком, который поддерживает его работоспособность, то есть администратором вашей домашней сети.



Рис. 1

Итак, маршрутизация включает в работу тогда, когда отправитель и получатель находятся в разных сетях. То есть результат произведения IP-адреса на маску компьютера отправителя и компьютера получателя различен. Основной задачей маршрутизатора является передача IP-пакетов в разные IP-сети. Плюс к этому маршрутизаторы умеют работать с различными типами сетей. Многие из вас наверняка встречались с такими терминами как ATM-сеть или Ethernet-сеть. Например, ваш домашний компьютер подключён к Интернету по технологии ADSL (пожалуй, этот тип подключения наиболее известен по торговой марке Стрим) и работает с ATM-сетью. А компьютер вашего одноклассника подключён по Ethernet. К нему в квартиру приходит кабель «витая пара» от коммутатора, установленного где-то в подъезде. Таких разновидностей сетей довольно много. Работают они по разным принципам и, как следствие, передать данные из одной такой

разновидности сети в другую без определённых преобразований невозможно. Маршрутизатор же такие преобразования умеет выполнять.

Каждый маршрутизатор хранит в оперативной памяти специальную таблицу – таблицу маршрутизации. Каждая строка этой таблицы несёт информацию о том, как передать IP-

пакет в какую-то определённую сеть; эти строки часто называют маршрутами. На основании этой таблицы маршрутизатор и осуществляет передачу. Возникает вопрос: а как в памяти маршрутизатора появляется эта таблица? Наш разговор сегодня – о формировании таблиц маршрутизации.

Статическая маршрутизация

Во-первых, часть строк таблицы формирует сам маршрутизатор. При включении маршрутизатор, как и компьютер, загружает в оперативную память операционную систему и файл настроек. Далее он опрашивает свои *интерфейсы*, к которым подключены сети. Интерфейсом называют комбинацию физического разъёма для подключения кабеля и тип и характеристики передаваемых сигналов. Тип сигнала – это то, что используется для передачи: электрические или световые импульсы, а к характеристикам можно отнести, например, количество используемых пар проводов, напряжение для медных кабелей, которые переносят электрические сигналы. Понятия «интерфейс» и «разъём» разделили из-за того, что на одном физическом разъёме может быть реализовано несколько разных интерфейсов. С беспроводными сетями ситуация похожая.

При настройке маршрутизатора администратор сети назначает IP-адрес и маску сети задёйствованным интерфейсам, причём IP-адрес интерфейса всегда должен принадлежать диапазону адресов сети, которая к этому интерфейсу подключена. Маршрутизатор считывает из настроек интерфейсов эти параметры, выполняет операцию логического умножения адреса на маску сети и получает адрес сети. Этот адрес маршрутизатор заносит к себе в таблицу маршрутизации и связыва-

ет адрес сети с интерфейсом. В итоге после обработки одного интерфейса появляется строка, говорящая, что сеть с данным адресом доступна через указанный интерфейс, и если на маршрутизатор приходит пакет, адресованный в эту сеть, то его надо переслать на этот интерфейс, чтобы он попал в сеть назначения.

Так происходит со всеми интерфейсами, и маршрутизатор автоматически в начале своей работы вносит в свою таблицу маршрутизации все сети, которые непосредственно к нему подключены. Никакой информации о других сетях он не имеет и, следовательно, передавать данные может только в подключённые сети. Но этого оказывается достаточно, чтобы ваш домашний маршрутизатор, имеющий всего два интерфейса, передал данные маршрутизатору провайдера.

Для того чтобы маршрутизатор смог передавать данные дальше в другие сети, он должен «узнать» об этих сетях. Как вариант, такую информацию может сообщить маршрутизатору администратор сети. В этом случае администратор специальными командами заполняет таблицу маршрутизации. Он вручную прописывает, на какой интерфейс должен быть передан пакет, адресованный в указанную сеть, и на каждом маршрутизаторе проделывает такую работу. Нетрудно догадаться, что работа очень трудоёмкая и кро-

потливая. Причём если маршрутизатор получает пакет и не находит в таблице маршрутизации сети назначения, то такой пакет он удаляет. Забыв где-то прописать маршрут в сеть, администратор столкнётся с необходимостью искать, в каком месте возникает потеря данных.

Давайте немного посчитаем. Предположим, что у нас есть сеть, состоящая из четырёх маршрутизаторов (рис. 2), образующих своим расположением четырёхугольник. Каждый маршрутизатор (он обозначается пиктограммой ) соединён кабелями с тремя оставшимися. К каждому маршрутизатору подключены по три локальных сети. Такая

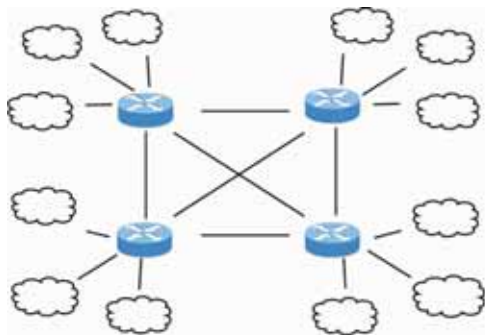


Рис. 2

топология называется *полносвязной*, *ячейковой*, или, в английском варианте, *Full-Mesh topology*. Итого двенадцать локальных сетей. Плюс к этому не надо забывать, что кабели, которыми соединены маршрутизаторы, тоже образуют отдельные IP-сети и их тоже нужно учитывать при формировании таблицы маршрутизации. Тогда на каждом из устройств администратору придётся прописать по девять маршрутов в локальные сети и ещё маршруты в сети, соединяющие удалённые маршрутизаторы. Если же количество локальных сетей увеличится, то пропорционально возрастёт и количество маршрутов в таблицах маршрутизации. Всё бы

ничего, можно потратить время и всё это прописать, но есть один неприятный момент.

Он заключается в том, что как в любом техническом построении, какие-либо компоненты сети могут отказать. Например, где-то произошёл обрыв кабеля, соединяющего два маршрутизатора. Тогда администратору необходимо выяснить, в какие сети по этому каналу передавались данные, и переписать все маршруты, в которых задействован повреждённый кабель. На эту работу уйдёт довольно много времени, а значит, всё это время между частями сети, которые соединял порванный кабель, данные передаваться не будут.

Описанный способ называется *статической маршрутизацией*. У него есть свои преимущества. Например, зная особенности работы пользователей сети и особенности линий связи, администратор может так прописать маршруты, чтобы равномерно загрузить все каналы в сети и не допустить перегрузок, то есть таких ситуаций, когда через интерфейс маршрутизатора в кабель должен пойти объём данных, превышающий пропускную способность интерфейса.

Представьте себе ситуацию с бассейном и двумя трубами. По первой в бассейн течёт воды больше, чем вытекает из бассейна через вторую трубу. Через некоторое время вода начнёт переливаться через край. Похожая ситуация с маршрутизатором. Если поток данных через интерфейс небольшой и маршрутизатор успевает отправлять пакеты в линию связи, то всё в порядке. Если скорости интерфейса оказывается недостаточно, то пакет помещается в специальный буфер памяти интерфейса и в нём дожидается отправки. При переполнении буфера вновь поступающие пакеты маршрутизатор удаляет.

Таким образом, распределение администратором сети движения данных с учётом характеристик каналов связи, или, как говорят, *балансировка трафика*, является большим достоинством применения статической маршрутизации. К ещё одному серьёзному достоинству вернёмся чуть позже.

Динамическая маршрутизация

И тогда единственным возможным решением становится переход на динамическую маршрутизацию, то есть включение и настройка специальных протоколов. Задачей этих протоколов является поиск лучших маршрутов во все сети для внесения их в таблицу маршрутизации. Плюс перестроение маршрутов в случае изменения топологии общей сети. Изменение топологии может быть вызвано отказом какого-либо канала или появлением нового. Это происходит, если к сети добавляют ещё один маршрутизатор с несколькими подключёнными к нему сетями. Естественно, что реакция любого динамического протокола на изменение топологии более быстрая, нежели реакция человека.

И опять возникает тот же вопрос: как динамические протоколы заполняют таблицу маршрутизации? Это зависит от алгоритма, реализованного в протоколе, и по таким алгоритмам все протоколы делят на два основных класса: дистанционно-векторные (Distance Vector) и протоколы на базе оценки качества линии связи (Link State).

В первом случае протоколы оценивают, какой маршрут лучше, на основании информации о количестве промежуточных маршрутизаторов между текущим маршрутизатором и сетью назначения. Например, маршрутизатор может передать пакет в сеть назначения двумя способами. При первом способе пакет должен

Из сказанного выше становится понятно, что основными проблемами применения статической маршрутизации являются размер и надёжность сети. Поэтому с некоторого момента роста сети или повышения вероятности отказов статическая маршрутизация может стать неприемлемой.

будет пройти два промежуточных маршрутизатора, при втором – промежуточный маршрутизатор один. Тогда второй вариант передачи считается лучшим. «Здесь короче путь, значит, здесь лучше!» – считают протоколы. Путь складывается из цепочки маршрутов. Количество промежуточных маршрутизаторов ещё называют *количеством прыжков пакета* (Hop Count). Отсюда и пошло название класса протоколов. Они как бы пытаются найти ответы на два вопроса: в каком направлении отправить пакет, то есть через какой интерфейс (вектор направления) и как далеко от маршрутизатора находится сеть назначения (дистанция)?

Во втором случае протоколы пытаются найти из всех возможных линий связи между маршрутизаторами, по которым можно в итоге передать пакет в сеть назначения, самые скоростные и по ним осуществляют пересылку пакетов. То есть из двух линий связи лучшей будет считаться та, которая обеспечивает более высокую скорость передачи. И следовательно, лучшим будет считаться маршрут через эту линию. Оптимальность маршрута определяется по параметру, который называют *стоимостью пути* (Cost). Этот параметр считается для каждой линии на основании скорости, а общий путь до сети назначения определяется суммой стоимостей всех задействованных линий связи. И тогда

может оказаться, что использовать одну низкоскоростную линию выгодней, чем передавать пакеты по цепочке более высокоскоростных.

Существует третий класс протоколов, который называют гибридным. Протоколы этого класса в алгоритмах поиска оптимального маршрута

используют и параметр количества прыжков, и оценивают скорость линий, получая некоторую усреднённую оценку. Общее название параметра, определяющего оптимальность маршрута – *метрика* (*Metric*).

Как работают дистанционно-векторные протоколы

Алгоритмы работы протоколов могут быть достаточно сложными. Самыми простыми из них считаются дистанционно-векторные протоколы. Они легко реализуются в операционных системах маршрутизаторов, не требовательны к аппаратным ре-

сурсам устройств, и это обеспечивает их достаточно активное использование и сегодня, несмотря на ряд существенных недостатков. Давайте сначала более подробно о них и поговорим.

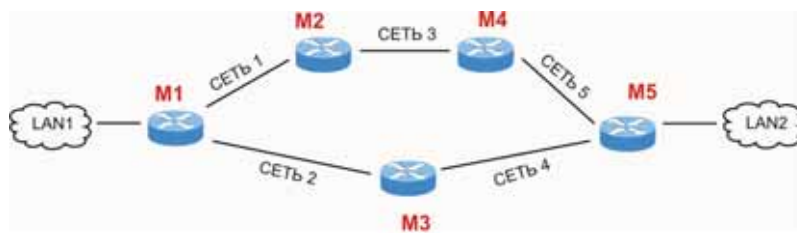


Рис. 3

Пусть у нас есть сеть маршрутизаторов, топология которой представлена на рис. 3, и на всех маршрутизаторах запущен дистанционно-векторный протокол маршрутизации. Как уже говорилось, каждый маршрутизатор заносит в свою таблицу маршрутизации маршруты в непосредственно подключённые сети. Метрика таких маршрутов равна нулю. Значит, в таблице маршрутизации маршрутизатора M1 будет три маршрута: в сеть LAN1, в сеть1 и в сеть2. Каждый маршрутизатор один раз в определённый промежуток времени организует отправку копии своей таблицы маршрутизации всем маршрутизаторам, которые подключены к нему линией связи. Такие маршрутизаторы называют также *соседними*, или *соседями*. У нас на схеме это M2 и M3. При первой такой отправке M1

пошлёт им информацию о своих трёх непосредственно подключённых сетях (а это и есть маршруты). Получив от соседа таблицу маршрутизации, каждый начинает анализировать полученные маршруты. Если маршрута в сеть в таблице маршрутизации у маршрутизатора нет, то он добавит его в свою таблицу, увеличив при этом присланное значение метрики на единицу. То есть маршрутизатор «понимает», что сеть, маршрут в которую он только что сохранил, находится ещё за одним промежуточным маршрутизатором, или ещё в одном прыжке. Если маршрут в сеть уже есть, маршрутизатор может перезаписать его, но только в том случае, если параметр метрика лучше у нового маршрута, чем у старого.

Описывать анализ присылаемой таблицы маршрутизации каждым

устройством слишком долго, да и необходимости в этом нет. Поэтому ограничимся выборочными примерами. Маршрутизатор M2 «знает» (имеет в своей таблице маршрутизации маршруты) сеть1 и сеть3, а маршрутов в сети LAN1 и сеть2 у него пока нет. Поэтому маршруты в сеть2 и в LAN1 после их получения от M1 он занесёт в свою таблицу, а маршрут в сеть1 отбросит со словами: «И без тебя знаю!». Оба маршрутизатора – M1 и M2 – автоматически определили маршрут в сеть1, и принимаемый от соседа маршрут в эту сеть будет всегда хуже.



Следовательно, после того как все маршрутизаторы сети первый раз отправили свои таблицы маршрутизации всем своим соседям, каждый из них внёс в таблицу маршрутизации маршруты в сети, расположенные за ближайшими соседями. Например, маршрутизатор M2 «узнаёт» маршруты в сети LAN1, сеть2 и сеть5. Что творится дальше, они пока не знают, маршрутов в более удалённые сети у них пока нет. Проходит ещё один такой же промежуток времени. Теперь каждый маршрутизатор шлёт соседям не только свои непосредственно подключённые сети, а ещё и сети, мар-

шруты в которые он получил при первой рассылке. Значит, после второй волны рассылок каждый маршрутизатор получает маршруты в сети, расположенные за двумя соседними устройствами.

Вернёмся к маршрутизатору M2. После второй рассылки он «узнаёт» о маршрутах во все сети. Ну и так далее, если количество маршрутизаторов в сети больше. В конце концов каждый маршрутизатор получит от своих соседей маршруты во все IP-сети общей сети. Тогда говорят, что для процесса маршрутизации *сеть сошлась*.

Кроме функции обмена маршрутами, в дистанционно-векторных протоколах рассылка таблицы маршрутизации играет ещё одну роль – это проверка работы канала связи. Если маршрутизатор принимает через свой интерфейс таблицу маршрутизации от соседа, значит, с линией связи всё в порядке. Получение таблицы маршрутизации с уже «известными» устройству маршрутами в этом случае работает как подтверждение работоспособности линий связи и ранее полученных маршрутов. Для каждой записи таблицы маршрутизатор ведёт счёт времени от момента получения маршрута. И когда в очередной раз приходит маршрут (обновление), он счётчик времени запускает заново. Если же определённое время обновление маршрута не приходит, значит, с линией что-то не так и маршрут в эту сеть надо удалять. После удаления маршрутов, которые использовали отказавшую линию, соседние маршрутизаторы, располагающие худшими маршрутами в отрезанный отказом сегмент, разошлют свои маршруты и заполнят этим образовавшиеся дырки в схеме маршрутов. Если же таких маршрутов не оказалось, сегмент сети будет отрезан от общей.

Ещё один нюанс. Давайте более внимательно посмотрим на маршрутизатор М3. Маршрут в сеть3 он может получить как от маршрутизатора М1, так и от маршрутизатора М5. Как же ему выбрать, какой из маршрутов занести в свою таблицу маршрутизации? Ведь значение метрики будет одинаковым. Ответ прост: в классической схеме маршрутизации он возьмёт маршрут в сеть3 от того соседа, который его раньше прислал. Второй, равнозначный маршрут от другого соседа, будет отбрасываться. Помним: маршрут заменяется только на лучший.

Когда наша сеть сойдётся, пакеты из сети LAN1 в сеть LAN2 и обратно будут передаваться через маршрутизатор М3. Действительно, например, маршрутизатор М5 будет получать маршруты в сеть LAN1 от соседей М3 и М4. Но М4 будет присылать маршрут в эту сеть с метрикой 2, а М3 – с метрикой 1. Значит, маршрут на маршрутизатор М3 лучше, он и будет внесён в таблицу.

Давайте теперь предположим, что линия связи между М3 и М5 по каким-то причинам перестала работать. Тогда М5 перестает получать обновления от М3 и по истечении времени удалит ранее полученные от него маршруты из своей таблицы маршрутизации. В частности, он удалит маршрут в сеть LAN1. Но М4 всё это время честно слал маршрут в сеть LAN1 маршрутизатору М5. До момента отказа этот маршрут маршрутизатором М5 отбрасывался. Но как только М5 удалит из своей таблицы маршрут в LAN1 через М3, у него не окажется маршрута в эту сеть и он примет его в очередной рассылке обновлений от М4 и занесёт в таблицу. Тогда пакеты между LAN1 и LAN2 начнут передаваться по верхней линии.

Таким образом, при работе дистанционно-векторных протоколов маршрутизаторы основываются в

построении маршрутов на том, что им «сказали» соседи. Именно поэтому такой принцип встречается под жаргонным названием «маршрутизация по слухам». Отправку маршрутизатором таблицы с маршрутами в его подключённые сети можно трактовать следующим образом:

– Соседи! – говорит маршрутизатор. – Это мои сети, я знаю, где они находятся! Поэтому, если к вам попадёт пакет, адресованный в одну из этих сетей, шлите мне, я передам!

Соседний маршрутизатор, получивший маршруты в такие сети, воспринимает их иначе:

– Соседи! – говорит он своими рассылками. – Вот он (и показывает пальцем, хоть это и нехорошо, на соседа, приславшего маршруты) сказал мне, что он знает, где эти сети находятся! И если к вам попадёт пакет, адресованный в одну из этих сетей, шлите мне, я передам ему, а он пусть разбирается!

На самом деле описанный выше принцип работы дистанционно-векторных протоколов представлен достаточно упрощённо. Реальные протоколы (наиболее известным представителем является Routing Information Protocol – RIP) содержат целый ряд дополнительных функций. Это и механизмы, ускоряющие сходимость сети, и механизм, блокирующий возникновение так называемых маршрутных петель, и ряд других вещей. Маршрутной петлей называют ситуацию, при которой отправленный маршрутизатором маршрут, пройдя некоторую цепочку маршрутизаторов, возвращается опять на этот маршрутизатор.

И, кстати, RIP входит в состав стека TCP/IP.

Как мы увидели, для работы протоколов маршрутизации необходима пересылка между маршрутизаторами маршрутной информации.

Это или пересылка маршрутов как в RIP, или другой информации, на основании которой можно построить маршруты в других протоколах маршрутизации. Объём этой информации у дистанционно-векторных протоколов пропорционален размеру сети. Причём речь идёт не столько о количестве маршрутизаторов в сети, сколько о числе подключённых к ним локальных сетей. И объём этого служебного трафика может оказаться достаточно велик. Но в любом случае передача маршрутной информации отбирает часть пропускной способности линий связи, снижая скорость передачи данных пользователями сети, чего не происходит в случае статической маршрутизации. Там понятия служебного трафика, обслуживающего маршрутизацию, просто нет, и это ещё одно большое

достоинство статической схемы.

Алгоритмы протоколов на базе оценки качества линий связи намного сложнее, и сейчас мы о них разговаривать подробно не будем. В очень упрощённом виде принцип их работы можно сравнить с известными задачами школьного курса информатики: определите по данной таблице минимальную стоимость проезда из пункта А в пункт В. То есть каждый маршрутизатор формирует у себя в памяти топологию сети, запоминает скорости линий связи в сети и на основании этой информации начинает искать способы достижимости каждой сети и выбирать наилучший. Примером протокола этого класса является *Open Shortest Path First (OSPF)*. Об этих протоколах подробнее поговорим в другой раз.

Калейдоскоп

Калейдоскоп

Калейдоскоп

Редкое явление

Летом 2004 г. произошло интересное астрономическое явление: в один из дней Венера оказалась между Солнцем и Землёй, и её диск размером $1/30$ солнечного перемещался по нему в течение примерно 6 часов. Случаются такие события довольно редко. Впервые его удалось наблюдать в 1631 г. (благодаря тому, что И. Кеплер сумел вычислить, когда Венера попадёт в пространство между Солнцем и Землёй), последний раз – почти 130 лет назад, в 1882 г. А в период между этими датами подобное явление происходило всего 2 раза. В мае 1761 г. его наблюдал М.В. Ломоносов. При этом наш великий соотечественник обратил внимание на то, что при прохождении Венерой краёв солнечного диска очертания светила размываются, и сделал открытие: объяснил это явление существованием у планеты атмосферы, которая рассеивает солнечный свет. Следующее покрытие Солнца Венерой состоится в 2012 г., но, к сожалению, на территории России увидеть его будет нельзя.