http://adminbook.ru/index.php?men1=5/2/1

**Сетевые протоколы**

* Стеки протоколов
* Протоколы канального уровня
* Протоколы межсетевого уровня
* Транспортные протоколы
* Прикладные протоколы

Как уже упоминалось ранее, в локальных сетях могут совместно работать компьютеры разных производителей, оснащенные различным набором устройств и обладающие несхожими техническими характеристиками. На практике это означает, что для обеспечения нормального взаимодействия этих компьютеров необходим некий единый унифицированный стандарт, строго определяющий алгоритм передачи данных в распределенной вычислительной системе. В современных локальных сетях, или, как их принято называть в англоязычных странах, LAN (Local Area Network), роль такого стандарта выполняют сетевые протоколы.  
Итак, сетевым протоколом, или протоколом передачи данных, называется согласованный и утвержденный стандарт, содержащий описание правил приема и передачи между несколькими компьютерами команд, файлов, иных данных, и служащий для синхронизации работы вычислительных машин в сети.  
Прежде всего следует понимать, что в локальных сетях передача информации осуществляется не только между компьютерами как физическими устройствами, но и между приложениями, обеспечивающими коммуникации на программном уровне. Причем под такими приложениями можно понимать как компоненты операционной системы, организующие взаимодействие с различными устройствами компьютера, так и клиентские приложения, обеспечивающие интерфейс с пользователем. Таким образом, мы постепенно приходим к пониманию многоуровневой структуры сетевых коммуникаций — как минимум, с одной стороны мы имеем дело с аппаратной конфигурацией сети, с другой стороны — с программной.  
Вместе с тем передача информации между несколькими сетевыми компьютерами — не такая уж простая задача, как это может показаться на первый взгляд. Для того чтобы понять это, достаточно представить себе тот круг проблем, который может возникнуть в процессе приема или трансляции каких-либо данных. В числе таких «неприятностей» можно перечислить аппаратный сбой либо выход из строя одного из обеспечивающих связь устройств, например, сетевой карты или концентратора, сбой прикладного или системного программного обеспечения, возникновение ошибки в самих передаваемых данных, потерю части транслируемой информации или ее искажение. Отсюда следует, что в локальной сети необходимо обеспечить жесткий контроль для отслеживания всех этих ошибок, и более того, организовать четкую работу как аппаратных, так и программных компонентов сети. Возложить все эти задачи на один-единственный протокол практически невозможно. Как быть?

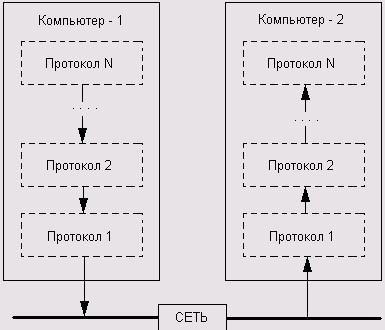
Выход нашелся в разделении протоколов на ряд концептуальных уровней, каждый из которых обеспечивает интерфейс между различными модулями программного обеспечения, установленного на работающих в сети компьютерах. Таким образом, механизм передачи какого-либо пакета информации через сеть от клиентской программы, работающей на о/щом компьютере, клиентской программе, работающей на другом компьютере, можно условно представить в виде последовательной пересылки этого пакета сверху вниз от некоего протокола верхнего уровня, обеспечивающего взаимодействие с пользовательским приложением, протоколу нижнего уровня, организующему интерфейс с сетью, его трансляции на компьютер-получатель и обратной передачи протоколу верхнего уровня уже на удаленной машине (рис. 2.1).

Рис. 2.1. Концептуальная модель многоуровневой системы протоколов

**Согласно такой схеме, каждый из уровней подобной системы обеспечивает собственный набор функций при передаче информации по локальной сети.**

Например, можно предположить, что протокол верхнего уровня, осуществляющий непосредственное взаимодействие с клиентскими программами, транслирует данные протоколу более низкого уровня, «отвечающему» за работу с аппаратными устройствами сети, преобразовывая их в «понятную» для него форму. Тот, в свою очередь, передает их протоколу, осуществляющему непосредственно пересылку информации на другой компьютер. На удаленном компьютере прием данных осуществляет аналогичный протокол «нижнего» уровня и контролирует корректность принятых данных, то есть определяет, следует ли транслировать их протоколу, расположенному выше в иерархической структуре, либо запросить повторную передачу. В этом случае взаимодействие осуществляется только между протоколами нижнего уровня, верхние уровни иерархии в данном процессе не задействованы. В случае если информация была передана без искажений, она транслируется вверх через соседние уровни протоколов до тех пор, пока не достигнет программы-получателя. При этом каждый из уровней не только контролирует правильность трансляции данных на основе анализа содержимого пакета информации, но и определяет дальнейшие действия исходя из сведений о его назначении. Например, один из уровней «отвечает» за выбор устройства, с которого осуществляется получение и через которое передаются данные в сеть, другой «решает», передавать ли информацию дальше по сети, или она предназначена именно этому компьютеру, третий «выбирает» программу, которой адресована принятая информация. Подобный иерархический подход позволяет не только разделить функции между различными модулями сетевого программного обеспечения, что значительно облегчает контроль работы всей системы в целом, но и дает возможность производить коррекцию ошибок на том уровне иерархии, на котором они возникли. Каждую из подобных иерархических систем, включающих определенный набор протоколов различного уровня, принято называть стеком протоколов.  
Вполне очевидно, что между теорией и практикой, то есть между концептуальной моделью стека протоколов и его практической реализацией существует значительная разница.

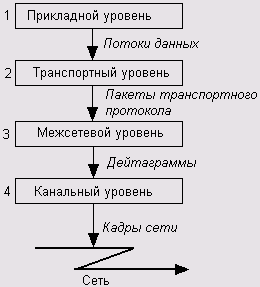
На практике принято несколько различных вариантов дробления стека протоколов на функциональные уровни, каждый из которых выполняет свой круг задач. Мы остановимся на одном из этих вариантов, который представляется наиболее универсальным. Данная схема включает четыре функциональных уровня, и так же, как и предыдущая диаграмма, описывает не конкретный механизм работы какого-либо стека протоколов, а общую модель, которая поможет лучше понять принцип действия подобных систем (рис. 2.2).  
Самый верхний в иерархической системе, прикладной уровень стека протоколов обеспечивает интерфейс с программным обеспечением, организующим работу пользователя в сети. При запуске любой программы, для функционирования которой требуется диалог с сетью, эта программа вызывает соответствующий протокол прикладного уровня. Данный протокол передает программе информацию из сети в доступном для обработки формате, то есть в виде системных сообщений либо в виде потока байтов. В точности таким же образом пользовательские приложения могут получать потоки данных и управляющие сообщения — как от самой операционной системы, так и от других запущенных на компьютере программ. То есть, обобщая, можно сказать, что протокол прикладного уровня выступает в роли своего рода посредника между сетью и программным обеспечением, преобразуя транслируемую через сеть информацию в «понятную» программе-получателю форму.

Рис. 2.2. Модель реализации стека протоколов

Основная задача протоколов транспортного уровня заключается в осуществлении контроля правильности передачи данных, а также в обеспечении взаимодействия между различными сетевыми приложениями. В частности, получая входящий поток данных, протокол транспортного уровня дробит его на отдельные фрагменты, называемые пакетами, записывает в каждый пакет некоторую дополнительную информацию, например идентификатор программы, для которой предназначены передаваемые данные, и контрольную сумму, необходимую для проверки целостности пакета, и направляет их на смежный уровень для дальнейшей обработки. Помимо этого протоколы транспортного уровня осуществляют управление передачей информации — например, могут запросить у получателя подтверждение доставки пакета и повторно выслать утерянные фрагменты транслируемой последовательности данных.

Некоторое недоумение может вызвать то обстоятельство, что протоколы транспортного уровня так же, как и протоколы прикладного уровня, взаимодействуют с сетевыми программами и координируют передачу данных между ними. Эту ситуацию можно прояснить на следующем примере: предположим, на подключенном к сети компьютере запущен почтовый клиент, эксплуатирующий два различных протокола прикладного уровня — РОРЗ ( Post Office Protocol) и SMTP (Simple Mail Transfer Protocol) — и программа загрузки файлов на удаленный сервер — FTP-клиент, работающий с протоколом прикладного уровня FTP (File Transfer Protocol). Все эти протоколы прикладного уровня опираются на один и тот же протокол транспортного уровня — TCP/IP (Transmission Control Protocol/Internet Protocol), который, получая поток данных от вышеуказанных программ, преобразует их в пакеты данных, где присутствует указание на конечное приложение, использующее эту информацию. Из рассмотренного нами примера следует, что данные, приходящие из сети, могут иметь различное назначение, и, соответственно, они обрабатываются различными программами, либо различными модулями одного и того же приложения. Во избежание путаницы при приеме и обработке информации каждая взаимодействующая с сетью программа имеет собственный идентификатор, который позволяет транспортному протоколу направлять данные именно тому приложению, для которого они предназначены.

Такие идентификаторы носят название программных портов. В частности, протокол прикладного уровня SMTP, предназначенный для отправки сообщений электронной почты, работает обычно с портом 25, протокол входящей почты РОРЗ — с портом 110, протокол Telnet — с портом 23. Задача перенаправления потоков данных между программными портами лежит па транспортных протоколах.  
На межсетевом уровне реализуется взаимодействие конкретных компьютеров распределенной вычислительной системы, другими словами, осуществляется процесс определения маршрута движения информации внутри локальной сети и выполняется отправка этой информации конкретному адресату. Данный процесс принято называть маршрутизацией. Получая пакет данных от протокола транспортного уровня вместе с запросом на его передачу и указанием получателя, протокол межсетевого уровня выясняет, на какой компьютер следует передать информацию, находится ли этот компьютер в пределах данного сегмента локальной сети или на пути к нему расположен шлюз, после чего трансформирует пакет в дейтаграмму — специальный фрагмент информации, передаваемый через сеть независимо от других аналогичных фрагментов, без образования виртуального канала (специально сконфигурированной среды для двустороннего обмена данными между несколькими устройствами) и подтверждения приема. В заголовок дейтаграммы записывается адрес компьютера-получателя пересылаемых данных и сведения о маршруте следования дейтаграммы. После чего она передается на канальный уровень.

**Шлюз — это программа, при помощи которой можно передавать информацию между двумя сетевыми системами, использующими различные протоколы обмена данными.**

Получая дейтаграмму, протокол межсетевого уровня определяет правильность ее приема, после чего выясняет, адресована ли она локальному компьютеру, или же ее следует направить по сети дальше. В случае, если дальнейшей пересылки не требуется, протокол межсетевого уровня удаляет заголовок дейтаграммы, вычисляет, какой из транспортных протоколов данного компьютера будет обрабатывать полученную информацию, трансформирует ее в соответствующий пакет и передает на транспортный уровень. Проиллюстрировать этот на первый взгляд сложный механизм можно простым примером. Предположим, на пеком компьютере одновременно используется два различных транспортных протокола: TCP/IP — для соединения с Интернетом и NetBEUI (NetBIOS Extended User Interface) для работы в локальной сети. В этом случае данные, обрабатываемые на транспортном уровне, будут для этих протоколов различны, однако на межсетевом уровне информация будет передаваться посредством дейтаграмм одного и того же формата.

Наконец, на канальном уровне осуществляется преобразование дейтаграмм в соответствующий сигнал, который через коммуникационное устройство транслируется по сети. В самом простом случае, когда компьютер напрямую подключен к локальной сети того или иного стандарта посредством сетевого адаптера, роль протокола канального уровня играет драйвер этого адаптера, непосредственно реализующий интерфейс с сетью. В более сложных ситуациях на канальном уровне могут работать сразу несколько специализированных протоколов, каждый из которых выполняет собственный набор функций.

# Протоколы канального уровня

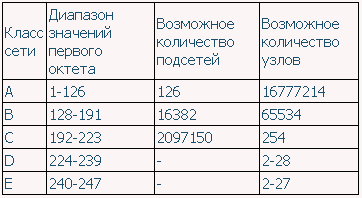
Протоколы, обеспечивающие взаимодействие компьютера с сетью на самом низком, аппаратном уровне, во многом определяют топологию локальной сети, а также ее внутреннюю архитектуру. В настоящее время на практике достаточно часто применяется несколько различных стандартов построения локальных сетей, наиболее распространенными среди которых являются технологии Ethernet, Token Ring, Fiber Distributed Data Interface (FDDI) и ArcNet.  
На сегодняшний день локальные сети, построенные на основе стандарта Ethernet, являются наиболее популярными как в нашей стране, так и во всем мире. На долю сетей Ethernet приходится почти девяносто процентов всех малых и домашних локальных сетей, что не удивительно, поскольку именно эта технология позволяет строить простые и удобные в эксплуатации и настройке локальные сети с минимумом затрат. Именно поэтому в качестве основного рассматриваемого нами стандарта будет принята именно технология Ethernet. Протоколы канального уровня поддержки Ethernet, как правило, встроены в оборудование, обеспечивающее подключение компьютера к локальной сети на физическом уровне. Стандарт Ethernet является широковещательным, то есть каждый подключенный к сети компьютер принимает всю следующую через его сетевой сегмент информацию — как предназначенную именно для этого компьютера, так и данные, направляемые на другую машину. Во всех сетях Ethernet применяется один и тот же алгоритм разделения среды передачи информации — множественный доступ с контролем несущей и обнаружением конфликтов (Carrier Sense Multiple Access with Collision Detection, CSMA/CD).  
В рамках технологии Ethernet сегодня различается несколько стандартов организации сетевых коммуникаций, определяющих пропускную способность канала связи и максимально допустимую длину одного сегмента сети, то есть расстояние между двумя подключенными к сети устройствами. Об этих стандартах мы побеседуем в следующей главе, посвященной изучению сетевого оборудования, пока же необходимо отметить, что в рамках стандарта Ethernet применяется, как правило, одна из двух различных топологий: конфигурация сети с общей шиной или звездообразная архитектура.

# Протоколы межсетевого уровня

Протоколы уровня межсетевого взаимодействия, как уже упоминалось ранее, предназначены для определения маршрутов следования информации в локальной сети, приема и передачи дейтаграмм, а также для трансляции принятых данных протоколам более высокого уровня, если эти данные предназначены для обработки на локальном компьютере. К протоколам межсетевого уровня принято относить протоколы маршрутизации, такие как RIP (Routing Internet Protocol) и OSPF (Open Shortest Path First), а также протокол контроля и управления передачей данных ICMP (Internet Control Message Protocol). Но вместе с тем одним из самых известных протоколов межсетевого уровня является протокол IP.

## Протокол IP

Протокол IP (Internet Protocol) используется как в глобальных распределенных системах, например в сети Интернет, так и в локальных сетях. Впервые протокол IP применялся еще в сети ArpaNet, являвшейся предтечей современного Интернета, и с тех пор он уверенно удерживает позиции в качестве одного из наиболее распространенных и популярных протоколов межсетевого уровня.  
Поскольку межсетевой протокол IP является универсальным стандартом, он нередко применяется в так называемых составных сетях, то есть сетях, использующих различные технологии передачи данных и соединяемых между собой посредством шлюзов. Этот же протокол «отвечает» за адресацию при передаче информации в сети. Как осуществляется эта адресация?  
Каждый человек, живущий на Земле, имеет адрес, по которому его в случае необходимости можно разыскать. Думаю, ни у кого не вызовет удивления то, что каждая работающая в Интернете или локальной сети машина также имеет свой уникальный адрес. Адреса в компьютерных сетях разительно отличаются от привычных нам почтовых. Боюсь, совершенно бесполезно писать на отправляемом вами в Сеть пакете информации нечто вроде «Компьютеру Intel Pentium III 1300 Mhz, эсквайру, Пэнии-Лэйн 114, Ливерпуль, Англия». Увидев такую надпись, ваша персоналка в лучшем случае фундаментально зависнет. Но если вы укажете компьютеру в качестве адреса нечто вроде 195.85.102.14, машина вас прекрасно поймет.  
Именно стандарт IP подразумевает подобную запись адресов подключенных к сети компьютеров. Такая запись носит название IP-адрес.  
Из приведенного примера видно, что IP-адрес состоит из четырех десятичных идентификаторов, или октетов, по одному байту каждый, разделенных точкой. Левый октет указывает тип локальной интрасети (под термином «интрасеть» (intranet) здесь понимается частная корпоративная или домашняя локальная сеть, имеющая подключение к Интернету), в которой находится искомый компьютер. В рамках данного стандарта различается несколько подвидов интрасетей, определяемых значением первого октета. Это значение характеризует максимально возможное количество подсетей и узлов, которые может включать такая сеть. В табл. 2.1 приведено соответствие классов сетей значению первого октета IP-адреса.

Таблица 2.1. Соответствие классов сетей значению первого октета IP-адреса

Адреса класса А используются в крупных сетях общего пользования, поскольку позволяют создавать системы с большим количеством узлов. Адреса класса В, как правило, применяют в корпоративных сетях средних размеров, адреса класса С — в локальных сетях небольших предприятий. Для обращения к группам машин предназначены широковещательные адреса класса D, адреса класса Е пока не используются: предполагается, что со временем они будут задействованы с целью расширения стандарта. Значение первого октета 127 зарезервировано для служебных целей, в основном для тестирования сетевого оборудования, поскольку IP-пакеты, направленные на такой адрес, не передаются в сеть, а ретранслируются обратно управляющей надстройке сетевого программного обеспечения как только что принятые. Кроме того, существует набор так называемых «выделенных» IP-адресов, имеющих особое значение. Эти адреса приведены в табл. 2.2.

Таблица 2.2. Значение выделенных IP-адресов

**ПРИМЕЧАНИЕ**Хостом принято называть любой подключенный к Интернету компьютер независимо от его назначения.

Как уже упоминалось ранее, небольшие локальные сети могут соединяться между собой, образуя более сложные и разветвленные структуры. Например, локальная сеть предприятия может состоять из сети административного корпуса и сети производственного отдела, сеть административного корпуса, в свою очередь, может включать в себя сеть бухгалтерии, планово-экономического отдела и отдела маркетинга. В приведенном выше примере сеть более низкого уровня является подсетью системы более высокого уровня, то есть локальная сеть бухгалтерии — подсеть для сети административного корпуса, а та, в свою очередь, — подсеть для сети всего предприятия в целом.  
Однако вернемся к изучению структуры IP-адреса. Последний (правый) идентификатор IP-адреса обозначает номер компьютера в данной локальной сети. Все, что расположено между правым и левым октетами в такой записи, — номера подсетей более низкого уровня. Непонятно? Давайте разберем на примере. Положим, мы имеем некий адрес в Интернете, на который хотим отправить пакет с набором свеженьких анекдотов. В качестве примера возьмем тот же IP-адрес— 195.85.102.14. Итак, мы отправляем пакет в 195-ю подсеть сети Интернет, которая, как видно из значения первого октета, относится к классу С. Допустим, 195-я сеть включает в себя еще 902 подсети, но наш пакет высылается в 85-ю. Она содержит 250 подсетей  
более низкого порядка, но нам нужна 102-я. Ну и, наконец, к 102-й сети подключено 40 компьютеров. Исходя из рассматриваемого нами адреса, подборку анекдотов получит машина, имеющая в этой сетевой системе номер 14. Из всего сказанного выше становится очевидно, что IP-адрес каждого компьютера, работающего как в локальной сети, так и в глобальных вычислительных системах, должен быть уникален.

Централизованным распределением IP-адресов в локальных сетях занимается государственная организация — Стенфордский международный научно-исследовательский институт (Stanford Research Institute, SRI International), расположенный в самом сердце Силиконовой долины — городе Мэнло-Парк, штат Калифорния, США. Услуга по присвоению новой локальной сети IP-адресов бесплатная, и занимает она приблизительно неделю. Связаться с данной организацией можно по адресу SRI International, Room EJ210, 333 Ravenswood Avenue, Menlo Park, California 94025, USA, no телефону в США 1-800-235-3155 или по адресу электронной почты, который можно найти на сайте http://www.sri.com. Однако большинство администраторов небольших локальных сетей, насчитывающих 5—10 компьютеров, назначают IP-адреса подключенным к сети машинам самостоятельно, исходя из описанных выше правил адресации в IP-сетях. Тацой подход вполне имеет право на жизнь, но вместе с тем произвольное назначение IP-адресов может стать проблемой, если в будущем такая сеть будет соединена с другими локальными сетями или в ней будет организовано прямое подключение к Интернету. В данном случае случайное совпадение нескольких IP-адресов может привести к весьма неприятным последствиям, например к ошибкам в маршрутизации передаваемых по сети данных или отказу в работе всей сети в целом.

Небольшие локальные сети, насчитывающие ограниченное количество компьютеров, должны запрашивать для регистрации адреса класса С. При этом каждой из таких сетей назначаются только два первых октета IP-адреса, например 197.112.Х.Х, на практике это означает, что администратор данной сети может создавать подсети и назначать номера узлов в рамках каждой из них произвольно, исходя из собственных потребностей.

Большие локальные сети, использующие в качестве базового межсетевой протокол IP, нередко применяют чрезвычайно удобный способ структуризации всей сетевой системы путем разделения общей IP-сети на подсети. Например, если вся сеть предприятия состоит из ряда объединенных вместе локальных сетей Ethernet, то в ней может быть выделено несколько структурных составляющих, то есть подсетей, отличающихся значением третьего октета IP-адреса. Как правило, в качестве каждой из подсетей используется физическая сеть какого-либо отдела фирмы, скажем, сеть Ethernet, объединяющая все компьютеры бухгалтерии. Такой подход, во-первых, позволяет  
излишне не расходовать IP-адреса, а во-вторых, предоставляет определенные удобства с точки зрения администрирования: например, администратор может открыть доступ к Интернету только для одной из вверенных ему подсетей или на время отключить одну из подсетей от локальной сети предприятия. Кроме того, в случае если сетевой администратор решит, что третий октет IP-адреса описывает номер подсети, а четвертый — номер узла в ней, то такая информация записывается в локальных таблицах маршрутизации сети вашего предприятия и не видна извне. Другими словами, данный подход обеспечивает большую безопасность.

Для того чтобы программное обеспечение могло автоматически выделять номера конкретных компьютеров из используемых в данной сетевой системе IP-адресов, применяются так называемые маски подсети. Принцип, по которому осуществляется распознавание номеров узлов в составе IP-адреса, достаточно прост: биты маски подсети, обозначающие номер самой IP-сети, должны быть равны единице, а биты, определяющие номер узла, — нулю. Именно поэтому в большинстве локальных IP-сетей класса С в качестве маски подсети принято значение 255.255.255.0: при такой конфигурации в состав общей сети может быть включено до 256 подсетей, в каждой из которых работает до 254 компьютеров. В ряде случаев это значение может изменяться, например, если возникла необходимость использовать в составе сети количество подсетей большее, чем 256, можно использовать маску подсети формата 255.255.255.195. В этой конфигурации сеть может включать до 1024 подсетей, максимальное число компьютеров в каждой из которых не должно превышать 60.

В локальных сетях, работающих под управлением межсетевого протокола IP, помимо обозначения IP-адресов входящих в сеть узлов принято также символьное обозначение компьютеров: например, компьютер с адресом 192.112.85.7 может иметь сетевое имя Localhost. Таблица соответствий IP-адресов символьным именам узлов содержится в специальном файле hosts, хранящемся в одной из системных папок; в частности, в операционной системе Microsoft Windows XP этот файл можно отыскать в папке flKCK:\Windows\system32\drivers\etc\. Синтаксис записи таблицы сопоставлений имен узлов локальной сети IP-адресам достаточно прост: каждый элемент таблицы должен быть расположен в новой строке, IP-адрес располагается в первом столбце, а за ним следует имя компьютера, при этом IP-адрес и имя должны быть разделены как минимум одним пробелом. Каждая из строк таблицы может включать произвольный комментарий, обозначаемый символом #. Пример файла hosts приведен ниже:

192.112.85.7 localhost # этот компьютер  
192.112.85.1 server # сервер сети  
192.112.85.2 director # компьютер приемной директора   
192.112.85.5 admin # компьютер системного администратора

Как правило, файл hosts создается для какой-либо конкретной локальной сети, и его копия хранится на каждом из подключенных к ней компьютеров. В случае, если один из узлов сети имеет несколько IP-адресов, то в таблице соответствий обычно указывается лишь один из них, вне зависимости от того, какой из адресов реально используется. При получении из сети IP-пакета, предназначенного для данного компьютера, протокол IP сверится с таблицей маршрутизации и на основе анализа заголовка IP-пакета автоматически опознает любой из IP-адресов, назначенных данному узлу.  
Помимо отдельных узлов сети собственные символьные имена могут иметь также входящие в локальную сеть подсети. Таблица соответствий IP-адресов именам подсетей содержится в файле networks, хранящемся в той же папке, что и файл hosts. Синтаксис записи данной таблицы сопоставлений несколько отличается от предыдущего, и в общем виде выглядит следующим образом: <сетевяе имя> <номер сети> [псевдонимы...] [#<конментарий>]  
где сетевое имя — имя, назначенное каждой подсети, номер сети — часть IP-адреса подсети (за исключением номеров более мелких подсетей, входящих в данную подсеть, и номеров узлов), псевдонимы — необязательный параметр, указывающий на возможные синонимы имен подсетей: они используется в случае, если какая-либо подсеть имеет несколько различных символьных имен; и, наконец, комментарий — произвольный комментарий, поясняющий смысл каждой записи. Пример файла networks приведен ниже:

loopback 127  
marketing 192.112.85 # отдел маркетинга  
buhgalteria 192.112.81 # бухгалтерия  
workshop 192.112.80 # сеть производственного цеха  
workgroup 192.112.10 localnetwork # основная рабочая группа

Обратите внимание на то обстоятельство, что адреса, начинающиеся на 127, являются зарезервированными для протокола IP, а подсеть с адресом 192.112.10 в нашем примере имеет два символьных имени, используемых совместно.  
Файлы hosts и networks не оказывают непосредственного влияния на принципиальный механизм работы протокола IP и используются в основном прикладными программами, однако они существенно облегчают настройку и администрирование локальной сети.

## Протокол IPX

Протокол IPX (Internet Packet Exchange) является межсетевым протоколом, используемым в локальных сетях, узлы которых работают под управлением операционных систем семейства Nowell Netware. Данный протокол обеспечивает передачу дейтаграмм в таких сетях без организации логического соединения — постоянного двустороннего обмена данными между двумя узлами сети, которое организуется протоколом транспортного уровня. Разработанный на основе технологий Nowell, этот некогда популярный протокол в силу несовместимости с чрезвычайно распространенным стеком протоколов TCP/IP в настоящее время медленно, но верно утрачивает свои позиции.  
Как и межсетевой протокол IP, IPX способен поддерживать широковещательную передачу данных посредством дейтаграмм длиной до 576 байт, 30 из которых занимает заголовок пакета. В сетях IPX используются составные адреса узлов, состоящих из номера сети, адреса узла и адреса прикладной программы, для которой предназначен передаваемый пакет информации, который также носит наименование гнезда или сокета. Для обеспечения обмена данными между несколькими сетевыми приложениями в многозадачной среде на узле, работающем под управлением протокола IPX, должно быть одновременно открыто несколько сокетов.  
Поскольку в процессе трансляции данных протокол IPX не запрашивает подтверждения получения дейтаграмм, доставка данных в таких сетях не гарантируется, и потому функции контроля над передачей информации возлагаются на сетевое программное обеспечение. Фактически IPX обеспечивает только инкапсуляцию транслируемых по сети потоков данных в дейтаграммы, их маршрутизацию и передачу пакетов протоколам более высокого уровня.  
Протоколам канального уровня IPX передает пакеты данных, имеющие следующую логическую структуру:

* контрольная сумма, предназначенная для определения целостности передаваемого пакета (2 байта);
* указание на длину пакета (2 байта);
* данные управления транспортом (1 байт);
* адрес сети назначения (4 байта);
* адрес узла назначения (6 байт);
* номер сокета назначения (2 байта);
* адрес сети-отправителя (4 байта);
* адрес узла-отправителя (6 байт);
* номер сокета-отправителя (2 байта);
* передаваемая информация (0-546 байт).

Протоколы канального уровня размещают этот пакет внутри кадра сети и передают его в распределенную вычислительную систему.

# Транспортные протоколы

Как уже упоминалось ранее, протоколы транспортного уровня обеспечивают контроль над передачей данных между межсетевыми протоколами и приложениями уровня операционной системы. В настоящее время в локальных сетях наиболее распространено несколько разновидностей транспортных протоколов.

## Протокол TCP

Протокол IP позволяет только транслировать данные. Для того чтобы управлять этим процессом, служит протокол TCP (Transmission Control Protocol), опирающийся на возможности протокола IP. Как же контролируется передача информации?  
Положим, вы хотите переслать по почте вашему другу толстый журнал, не потратив при этом денег на отправку бандероли. Как решить эту проблему, если почта отказывается принимать письма, содержащие больше нескольких бумажных листов? Выход простой: разделить журнал на страницы и отправлять их отдельными письмами. По номерам страниц ваш друг сможет собрать журнал целиком. Приблизительно таким же способом работает протокол TCP. Он дробит информацию на несколько частей, присваивает каждой части номер, по которому данные впоследствии можно будет соединить воедино, добавляет к ней «служебную» информацию и укладывает все это в отдельный «IP-конверт». Далее этот «конверт» отправляется по сети — ведь протокол межсетевого уровня умеет обрабатывать подобную информацию. Поскольку в такой схеме протоколы TCP и IP тесно связаны, их часто объединяют в одно понятие: TCP/IP. Размер передаваемых в Интернете TCP/IP-пакетов составляет, как правило, от 1 до 1500 байт, что связано с техническими характеристиками сети.  
Наверняка, пользуясь услугами обычной почтовой связи, вы сталкивались с тем, что обычные письма, посылки и иные почтовые отправления теряются и приходят совсем не туда, куда нужно. Те же проблемы характерны и для локальных сетей. На почте такие неприятные ситуации решают руководители почтовых отделений, а в сетевых системах этим занимается протокол TCP. Если какой-либо пакет данных не был доставлен получателю вовремя, TCP повторяет пересылку до тех пор, пока информация не будет принята корректно и в полном объеме.  
В действительности данные, передаваемые по электронным сетям, не только теряются, но зачастую искажаются из-за помех на линиях связи. Встроенные в TCP алгоритмы контроля корректности передачи данных решают и эту проблему. Одним из самых известных механизмов контроля правильности пересылки информации является метод, согласно которому в заголовок каждого передаваемого пакета записывается некая контрольная сумма, вычисленная компьютером-отправителем. Компьютер-получатель по аналогичной системе вычисляет контрольную сумму и сравнивает ее с числом, имеющимся в заголовке пакета. Если цифры не совпадают, TCP пытается повторить передачу.  
Следует отметить также, что при отправке информационных пакетов протокол TCP требует от компьютера-получателя подтверждения приема информации. Это организуется путем создания временных задержек при приеме-передаче — тайм-аутов, или ожиданий. Тем временем отправитель продолжает пересылать данные. Образуется некий объем уже переданных, но еще не подтвержденных данных. Иными словами, TCP организует двунаправленный обмен информацией, что обеспечивает более высокую скорость ее трансляции.  
При соединении двух компьютеров их модули TCP следят за состоянием связи. При этом само соединение, посредством которого осуществляется обмен данными, носит название виртуального или логического канала.  
Фактически протокол TCP является неотъемлемой частью стека протоколов TCP/IP, и именно с его помощью реализуются все функции контроля над передачей информации по сети, а также задачи ее распределения между клиентскими приложениями.

## Протокол SPX

В точности так же, как протокол TCP для IP-сетей, для сетей, построенных на базе межсетевого протокола IPX, транспортным протоколом служит специальный протокол SPX (Sequenced Pocket eXchange). В таких локальных сетях протокол SPX выполняет следующий набор функций:

* инициализация соединения;
* организация виртуального канала связи (логического соединения);
* проверка состояния канала;
* контроль передачи данных;
* разрыв соединения.

Поскольку транспортный протокол SPX и межсетевой протокол IPX тесно связаны между собой, их нередко объединяют в общее понятие — семейство протоколов IPX/SPX. Поддержка данного семейства протоколов реализована не только в операционных системах семейства Nowell Netware, но и в ОС Microsoft Windows 9x/Me/NT/2000/XP, Unix/Linux и OS/2.

Протоколы NetBIOS/NetBEUI

Разработанный компанией IBM транспортный протокол NetBIOS (Network Basic Input/Output System) является базовым протоколом для локальных  
сетей, работающих под управлением операционных систем семейств Nowell Netware и OS/2, однако его поддержка реализована также и в ОС Microsoft Windows, и в некоторых реализациях Unix-совместимых операционных систем. Фактически можно сказать, что данный протокол работает сразу на нескольких логических уровнях стека протоколов: на транспортном уровне он организует интерфейс между сетевыми приложениями в качестве надстройки над протоколами IPX/SPX, на межсетевом — управляет маршрутизацией дейтаграмм, на канальном уровне — организует обмен сообщениями между различными узлами сети.  
В отличие от других протоколов, NetBIOS осуществляет адресацию в локальных сетях на основе уникальных имен узлов и практически не требует настройки, благодаря чему остается весьма привлекательным для системных администраторов, управляющих сетями с небольшим числом компьютеров. В качестве имен хостов протоколом NetBIOS используются значащие последовательности длиной в 16 байт, то есть каждый узел сети имеет собственное уникальное имя (permanent name), которое образуется из сетевого адреса машины с добавлением десяти служебных байтов. Кроме этого, каждый компьютер в сетях NetBIOS имеет произвольное символьное имя, равно как произвольные имена могут иметь логические\*рабочие группы, объединяющие несколько работающих совместно узлов — такие имена могут назначаться и удаляться по желанию системного администратора. Имена узлов служат для идентификации компьютера в сети, имена рабочих групп могут служить, в частности, для отправки данных нескольким компьютерам группы или для обращения к целому ряду сетевых узлов одновременно.  
При каждом подключении к распределенной вычислительной системе протокол NetBIOS осуществляет опрос локальной сети для проверки уникальности имени узла; поскольку несколько узлов сети могут иметь идентичные групповые имена, определение уникальности группового имени не производится.  
Специально для локальных сетей, работающих на базе стандарта NetBIOS, корпорацией IBM был разработан расширенный интерфейс для этого протокола, который получил название NetBEUI (NetBIOS Extended User Interface). Этот протокол рассчитан на поддержку небольших локальных сетей, включающих не более 150-200 машин, и по причине того, что данный протокол может использоваться только в отдельных сегментах локальных сетей (пакеты NetBEUI не могут транслироваться через мосты — устройства, соединяющие несколько локальных сетей, нередко использующих различную среду передачи данных или различную топологию), этот стандарт считается устаревшим и более не поддерживается операционной системой Microsoft Windows XP, хотя его поддержка имеется в ОС семейства Windows 9х/МЕ/2000.

# Прикладные протоколы

Протоколы прикладного уровня служат для передачи информации конкретным клиентским приложениям, запущенным на сетевом компьютере. В IP-сетях протоколы прикладного уровня опираются на стандарт TCP и выполняют ряд специализированных функций, предоставляя пользовательским программам данные строго определенного назначения. Ниже мы кратко рассмотрим несколько прикладных протоколов стека TCP/IP.

## Протокол FTP

Как следует из названия, протокол FTP (File Transfer Protocol) предназначен для передачи файлов через Интернет. Именно на базе этого протокола реализованы процедуры загрузки и выгрузки файлов на удаленных узлах Всемирной Сети. FTP позволяет переносить с машины па машину не только файлы, но и целые папки, включающие поддиректории на любую глубину вложений. Осуществляется это путем обращения к системе команд FTP, описывающих ряд встроенных функций данного протокола.

## Протоколы РОРЗ и SMTP

Прикладные протоколы, используемые при работе с электронной почтой, называются SMTP (Simple Mail Transfer Protocol) и РОРЗ (Post Office Protocol), первый «отвечает» за отправку исходящей корреспонденции, второй — за доставку входящей.  
В функции этих протоколов входит организация доставки сообщений e-mail и передача их почтовому клиенту. Помимо этого, протокол SMTP позволяет отправлять несколько сообщений в адрес одного получателя, организовывать промежуточное хранение сообщений, копировать одно сообщение для отправки нескольким адресатам. И РОРЗ, и SMTP обладают встроенными механизмами распознавания адресов электронной почты, а также специальными модулями повышения надежности доставки сообщений.

## Протокол HTTP

Протокол HTTP (Hyper Text Transfer Protocol) обеспечивает передачу с удаленных серверов на локальный компьютер документов, содержащих код разметки гипертекста, написанный на языке HTML или XML, то есть веб-страниц. Данный прикладной протокол ориентирован прежде всего на предоставление информации программам просмотра веб-страниц, веб-браузерам, наиболее известными из которых являются такие приложения, как Microsoft Internet Explorer и Netscape Communicator.  
Именно с использованием протокола HTTP организуется отправка запросов удаленным http-серверам сети Интернет и обработка их откликов; помимо  
этого HTTP позволяет использовать для вызова ресурсов Всемирной сети адреса стандарта доменной системы имен (DNS, Domain Name System), то есть обозначения, называемые URL (Uniform Resource Locator) вида http:/ /www.domain.zone/page (l).

## Протокол TELNET

Протокол TELNET предназначен для организации терминального доступа к удаленному узлу посредством обмена командами в символьном формате ASCII. Как правило, для работы с сервером по протоколу TELNET на стороне клиента должна быть установлена специальная программа, называемая telnet-клиентом, которая, установив связь с удаленным узлом, открывает в своем окне системную консоль операционной оболочки сервера. После этого вы можете управлять серверным компьютером в режиме терминала, как своим собственным (естественно, в очерченных администратором рамках). Например, вы получите возможность изменять, удалять, создавать, редактировать файлы и папки, а также запускать на исполнение программы на диске серверной машины, сможете просматривать содержимое папок других пользователей. Какую бы операционную систему вы ни использовали, протокол Telnet позволит вам общаться с удаленной машиной «на равных». Например, вы без труда сможете открыть сеанс UNIX на компьютере, работающем под управлением MS Windows.

## Протокол UDP

Прикладной протокол передачи данных UDP (User Datagram Protocol) используется на медленных линиях для трансляции информации как дейтаграмм.  
Дейтаграмма содержит полный комплекс данных, необходимых для ее отсылки и получения. При передаче дейтаграмм компьютеры не занимаются обеспечением стабильности связи, поэтому следует принимать особые меры для обеспечения надежности.  
Схема обработки информации протоколом UDP, в принципе, такая же, как и в случае с TCP, но с одним отличием: UDP всегда дробит информацию по одному и тому же алгоритму, строго определенным образом. Для осуществления связи с использованием протокола UDP применяется система отклика: получив UDP-пакет, компьютер отсылает отправителю заранее обусловленный сигнал. Если отправитель ожидает сигнал слишком долго, он просто повторяет передачу.  
На первый взгляд может показаться, что протокол UDP состоит сплошь из одних недостатков, однако есть в нем и одно существенное достоинство: прикладные интернет-программы работают с UDP в два раза быстрее, чем с его более высокотехнологичным собратом TCP.

# Сквозные протоколы и шлюзы

Интернет — это единая глобальная структура, объединяющая на сегодня около 13 000 различных локальных сетей, не считая отдельных пользователей.

Раньше все сети, входившие в состав Интернета, использовали сетевой протокол IP. Однако настал момент, когда пользователи локальных систем, не использующих IP, тоже попросились в лоно Интернета. Так появились шлюзы.  
Поначалу через шлюзы транслировалась только электронная почта, но вскоре пользователям и этого стало мало.

Теперь посредством шлюзов можно передавать любую информацию — и графику, и гипертекст, и музыку, и даже видео.

Информация, пересылаемая через такие сети другим сетевым системам, транслируется с помощью сквозного протокола, обеспечивающего беспрепятственное прохождение IP-пакетов через не IP-сеть.