## **Интернет-адреса**

Поскольку Интернет - это глобальная сеть компьютеров, каждый компьютер, подключенный к Интернету, **должен** иметь уникальный адрес. Интернет-адреса представлены в виде **nnn.nnn.nnn.nnnn** где nnn должно быть числом от 0 до 255. Этот адрес известен как IP-адрес. (IP означает интернет-протокол; подробнее об этом позже.)

На рисунке ниже показаны два компьютера, подключенных к Интернету; ваш компьютер с IP-адресом 1.2.3.4 и другой компьютер с IP-адресом 5.6.7.8. Интернет представлен как абстрактный объект между ними. (По мере продвижения этой статьи интернет-часть диаграммы 1 будет объясняться и перерисовываться несколько раз по мере раскрытия деталей Интернета.)

|  |
| --- |
| Диаграмма 1 |
| Диаграмма 1 |

Если вы подключаетесь к Интернету через интернет-провайдера (ISP), вам обычно назначается временный IP-адрес на время сеанса удаленного доступа. Если вы подключаетесь к Интернету из локальной сети (LAN), ваш компьютер может иметь постоянный IP-адрес или он может получить временный от сервера DHCP (протокол динамической настройки хоста). В любом случае, если вы подключены к Интернету, ваш компьютер имеет уникальный IP-адрес.

|  |
| --- |
| **Проверьте это - программа Ping** |
| Если вы используете Microsoft Windows или разновидность Unix и у вас есть подключение к Интернету, существует удобная программа, позволяющая проверить, работает ли компьютер в Интернете. Это называется **ping**, вероятно, по звуку, издаваемому старыми системами гидролокатора подводных лодок.1 Если вы используете Windows, запустите окно командной строки. Если вы используете разновидность Unix, перейдите в командную строку. Тип пинговать www.yahoo.com. Программа ping отправит "ping" (фактически эхо-запрос ICMP (Internet Control Message Protocol)) на указанный компьютер. Пропингованный компьютер ответит ответом. Программа ping будет считать время, истекшее до тех пор, пока не придет ответ (если он придет). Кроме того, если вы введете доменное имя (т.е. www.yahoo.com) вместо IP-адреса, ping разрешит доменное имя и отобразит IP-адрес компьютера. Подробнее о доменных именах и разрешении адресов позже. |

## **Стеки протоколов и пакеты**

Итак, ваш компьютер подключен к Интернету и имеет уникальный адрес. Как он "общается" с другими компьютерами, подключенными к Интернету? Здесь следует привести пример: допустим, ваш IP-адрес 1.2.3.4, и вы хотите отправить сообщение на компьютер 5.6.7.8. Сообщение, которое вы хотите отправить, называется "Привет, компьютер 5.6.7.8!". Очевидно, что сообщение должно быть передано по любому типу провода, соединяющего ваш компьютер с Интернетом. Допустим, вы позвонили своему интернет-провайдеру из дома, и сообщение должно быть передано по телефонной линии. Следовательно, сообщение должно быть переведено из алфавитного текста в электронные сигналы, передано через Интернет, затем переведено обратно в алфавитный текст. Как это достигается? Благодаря использованию **стека протоколов**. Каждому компьютеру нужен один для связи в Интернете, и он обычно встроен в операционную систему компьютера (т. Е. Windows, Unix и т.д.). Используемый в Интернете стек протоколов называется стеком протоколов TCP / IP из-за двух основных используемых протоколов связи. Стек TCP / IP выглядит следующим образом:

|  |  |
| --- | --- |
| **Уровень протокола** | **Комментарии** |
| Уровень прикладных протоколов | Протоколы, специфичные для таких приложений, как WWW, электронная почта, FTP и т.д. |
| Уровень протокола управления передачей | TCP направляет пакеты к определенному приложению на компьютере, используя номер порта. |
| Уровень интернет-протокола | IP направляет пакеты на определенный компьютер, используя IP-адрес. |
| Аппаратный уровень | Преобразует двоичные пакетные данные в сетевые сигналы и обратно. (Например, сетевая карта ethernet, модем для телефонных линий и т.д.) |

Если бы мы проследили путь, по которому сообщение "Привет, компьютер 5.6.7.8!" перешло с нашего компьютера на компьютер с IP-адресом 5.6.7.8, произошло бы что-то вроде этого:

|  |
| --- |
| Диаграмма 2 |
| Диаграмма 2 |

1. Сообщение будет начинаться с вершины стека протоколов на вашем компьютере и продвигаться вниз.
2. Если отправляемое сообщение длинное, каждый уровень стека, через который проходит сообщение, может разбивать сообщение на более мелкие фрагменты данных. Это связано с тем, что данные, отправляемые через Интернет (и большинство компьютерных сетей), отправляются управляемыми порциями. В Интернете эти порции данных известны как **пакеты**.
3. Пакеты будут проходить через уровень приложений и переходить на уровень TCP. Каждому пакету присваивается **номер порта**. Порты будут объяснены позже, но достаточно сказать, что многие программы могут использовать стек TCP / IP и отправлять сообщения. Нам нужно знать, какая программа на конечном компьютере должна получить сообщение, потому что она будет прослушивать определенный порт.
4. После прохождения уровня TCP пакеты переходят на уровень IP. Здесь каждый пакет получает свой адрес назначения, 5.6.7.8.
5. Теперь, когда наши пакеты сообщений имеют номер порта и IP-адрес, они готовы к отправке через Интернет. Аппаратный уровень заботится о преобразовании наших пакетов, содержащих алфавитный текст нашего сообщения, в электронные сигналы и передаче их по телефонной линии.
6. На другом конце телефонной линии ваш интернет-провайдер имеет прямое подключение к Интернету. **Маршрутизатор** интернет-провайдера проверяет адрес назначения в каждом пакете и определяет, куда его отправить. Часто следующей остановкой пакета является другой маршрутизатор. Подробнее о маршрутизаторах и инфраструктуре Интернета позже.
7. В конечном итоге пакеты достигают компьютера 5.6.7.8. Здесь пакеты начинаются в нижней части стека TCP / IP целевого компьютера и продвигаются вверх.
8. По мере продвижения пакетов вверх по стеку все данные маршрутизации, добавленные стеком отправляющего компьютера (такие как IP-адрес и номер порта), удаляются из пакетов.
9. Когда данные достигают вершины стека, пакеты повторно собираются в их первоначальную форму: "Привет, компьютер 5.6.7.8!"

## **Сетевая инфраструктура**

Итак, теперь вы знаете, как пакеты передаются с одного компьютера на другой через Интернет. Но что находится между ними? Из чего на самом деле состоит Интернет? Давайте посмотрим на другую диаграмму:

|  |
| --- |
| Диаграмма 3 |
| Диаграмма 3 |

Здесь мы видим диаграмму 1, перерисованную более подробно. О физическом подключении через телефонную сеть к интернет-провайдеру, возможно, было легко догадаться, но помимо этого, возможно, потребуется некоторое объяснение.

Интернет-провайдер поддерживает пул модемов для своих абонентов с коммутируемым доступом. Этим управляет какой-либо компьютер (обычно выделенный), который управляет потоком данных из пула модемов к магистральному или выделенному линейному маршрутизатору. На эту установку можно ссылаться как на сервер портов, поскольку она "обслуживает" доступ к сети. Здесь также обычно собирается информация о выставлении счетов и использовании.

После того, как ваши пакеты проходят телефонную сеть и локальное оборудование вашего провайдера, они направляются на магистраль провайдера или магистраль, у которой провайдер покупает пропускную способность. Отсюда пакеты обычно проходят через несколько маршрутизаторов и по нескольким магистралям, выделенным линиям и другим сетям, пока не найдут свое назначение, компьютер с адресом 5.6.7.8. Но разве не было бы неплохо, если бы мы знали точный маршрут, по которому наши пакеты проходят через Интернет? Как оказалось, способ есть...

|  |
| --- |
| **Проверьте это - программа Traceroute** |
| Если вы используете Microsoft Windows или разновидность Unix и у вас есть подключение к Интернету, вот еще одна удобная интернет-программа. Этот называется **traceroute** и он показывает путь, по которому ваши пакеты отправляются к заданному интернет-пункту назначения. Как и ping, вы должны использовать traceroute из командной строки. В Windows используйте tracert www.yahoo.com. В командной строке Unix введите трассировка маршрута www.yahoo.com. Подобно ping, вы также можете вводить IP-адреса вместо доменных имен. Traceroute распечатает список всех маршрутизаторов, компьютеров и любых других интернет-объектов, через которые должны пройти ваши пакеты, чтобы добраться до места назначения. |

Если вы используете traceroute, вы заметите, что ваши пакеты должны проходить через множество объектов, чтобы добраться до места назначения. У большинства из них длинные имена, такие как sjc2-core1-h2-0-0.atlas.digex.net и fddi0-0.br4.SJC.globalcenter.net. Это интернет-маршрутизаторы, которые решают, куда отправлять ваши пакеты. На диаграмме 3 показано несколько маршрутизаторов, но только несколько. Диаграмма 3 предназначена для демонстрации простой структуры сети. Интернет намного сложнее.

## **Инфраструктура Интернета**

Магистраль Интернета состоит из множества крупных сетей, которые соединяются друг с другом. Эти крупные сети известны как **поставщики сетевых услуг** или **NSP**. К числу крупных NSP относятся UUNet, CerfNet, IBM, BBN Planet, SprintNet, PSINet, а также другие. Эти сети **равняются** друг с другом для обмена пакетным трафиком. Каждый NSP должен подключаться к трем **сетевым точкам доступа** или **NAP**-ам. Во время NAP трафик пакетов может перескакивать с магистрали одного NSP на магистраль другого NSP. NSP также соединяются на **городских биржах** или **MAE**s. MAE служат тем же целям, что и NAP, но находятся в частной собственности. NAP были первоначальными точками подключения к Интернету. Как NAP, так и MAE называются точками обмена данными в Интернете или **IX**-ми. NSP также продают пропускную способность небольшим сетям, таким как интернет-провайдеры и провайдеры меньшей пропускной способности. Ниже приведен рисунок, показывающий эту иерархическую инфраструктуру.

|  |
| --- |
| Диаграмма 4 |
| Диаграмма 4 |

Это неверное представление реальной части Интернета. Диаграмма 4 предназначена только для демонстрации того, как NSP могут взаимодействовать друг с другом и с более мелкими интернет-провайдерами. Ни один из физических сетевых компонентов не показан на диаграмме 4 так, как они показаны на диаграмме 3. Это связано с тем, что магистральная инфраструктура единого NSP сама по себе является сложным чертежом. Большинство NSP публикуют карты своей сетевой инфраструктуры на своих веб-сайтах, и их можно легко найти. Нарисовать реальную карту Интернета было бы практически невозможно из-за его размера, сложности и постоянно меняющейся структуры.

## **Иерархия интернет-маршрутизации**

Итак, как пакеты находят свой путь через Интернет? Каждый ли компьютер, подключенный к Интернету, знает, где находятся другие компьютеры? Пакеты просто "транслируются" на каждый компьютер в Интернете? Ответ на оба предыдущих вопроса - "нет". Ни один компьютер не знает, где находятся другие компьютеры, и пакеты отправляются не на каждый компьютер. Информация, используемая для доставки пакетов по назначению, содержится в таблицах маршрутизации, которые хранятся на каждом маршрутизаторе, подключенном к Интернету.

**Маршрутизаторы - это коммутаторы пакетов.** Маршрутизатор обычно подключается между сетями для маршрутизации пакетов между ними. Каждый маршрутизатор знает о своих подсетях и о том, какие IP-адреса они используют. Маршрутизатор обычно не знает, какие IP-адреса находятся "над" ним. Изучите диаграмму 5 ниже. Черные ящики, соединяющие магистрали, - это маршрутизаторы. Более крупные магистрали NSP вверху подключены в режиме NAP. Под ними расположено несколько подсетей, а под ними еще больше подсетей. Внизу расположены две локальные сети с подключенными компьютерами.

|  |
| --- |
| Диаграмма 5 |
| Диаграмма 5 |

Когда пакет поступает на маршрутизатор, маршрутизатор проверяет IP-адрес, введенный туда уровнем протокола IP на исходном компьютере. Маршрутизатор проверяет свою таблицу маршрутизации. Если сеть, содержащая IP-адрес, найдена, пакет отправляется в эту сеть. Если сеть, содержащая IP-адрес, не найдена, то маршрутизатор отправляет пакет по маршруту по умолчанию, обычно вверх по магистральной иерархии к следующему маршрутизатору. Будем надеяться, что следующий маршрутизатор будет знать, куда отправить пакет. Если этого не происходит, пакет снова направляется вверх, пока не достигнет магистрали NSP. Маршрутизаторы, подключенные к магистралям NSP, содержат самые большие таблицы маршрутизации, и здесь пакет будет перенаправлен в правильную магистраль, где он начнет свое путешествие "вниз" по все меньшим и меньшим сетям, пока не найдет место назначения.

## **Разрешение доменных имен и адресов**

Но что, если вы не знаете IP-адрес компьютера, к которому хотите подключиться? Что, если вам нужен доступ к веб-серверу, называемому *www.anothercomputer.com*? Как ваш веб-браузер узнает, в какой части Интернета находится этот компьютер? Ответом на все эти вопросы является **служба доменных имен** или **DNS**. DNS - это распределенная база данных, которая отслеживает имена компьютеров и соответствующие им IP-адреса в Интернете.

На многих компьютерах, подключенных к Интернету, размещена часть базы данных DNS и программное обеспечение, которое позволяет другим пользователям получать к ней доступ. Эти компьютеры известны как DNS-серверы. Ни один DNS-сервер не содержит всю базу данных; они содержат только ее подмножество. Если DNS-сервер не содержит доменного имени, запрошенного другим компьютером, DNS-сервер перенаправляет запрашивающий компьютер на другой DNS-сервер.

|  |
| --- |
| Диаграмма 6 |
| Диаграмма 6 |

Служба доменных имен структурирована как иерархия, аналогичная иерархии маршрутизации IP. Компьютер, запрашивающий разрешение имени, будет перенаправлен "вверх" по иерархии, пока не будет найден DNS-сервер, который может разрешить доменное имя в запросе. Рисунок 6 иллюстрирует часть иерархии. В верхней части дерева находятся корни домена. Некоторые из более старых и распространенных доменов видны ближе к вершине. Чего не показано, так это множества DNS-серверов по всему миру, которые формируют остальную часть иерархии.

При настройке подключения к Интернету (например, для локальной сети или коммутируемой сети в Windows) обычно указываются один основной и один или несколько дополнительных DNS-серверов как часть установки. Таким образом, любые интернет-приложения, которым требуется разрешение доменного имени, смогут корректно функционировать. Например, когда вы вводите веб-адрес в свой веб-браузер, браузер сначала подключается к вашему основному DNS-серверу. После получения IP-адреса для введенного вами доменного имени браузер затем подключается к целевому компьютеру и запрашивает нужную вам веб-страницу.

|  |
| --- |
| **Проверьте это - отключите DNS в Windows** |
| Если вы используете Windows 95 / NT и имеете доступ к Интернету, вы можете просматривать свои DNS-серверы и даже отключать их.  *Если вы используете коммутируемую сеть:* *Откройте окно коммутируемой сети (которое можно найти в проводнике Windows под вашим дисководом компакт-дисков и над сетевым окружением).* Щелкните правой кнопкой мыши на вашем интернет-соединении и выберите Свойства. В нижней части окна свойств подключения нажмите настройки TCP /IP... кнопка.  *Если у вас есть постоянное подключение к Интернету:* *Щелкните правой кнопкой мыши на сетевом окружении и выберите Свойства.* Выберите Свойства TCP / IP. Выберите вкладку Конфигурации DNS вверху.  Теперь вы должны посмотреть IP-адреса своих DNS-серверов. Здесь вы можете отключить DNS или установить для своих DNS-серверов значение 0.0.0.0. (Сначала запишите IP-адреса своих DNS-серверов. Вероятно, вам также придется перезапустить Windows.) Теперь введите адрес в свой веб-браузер. Браузер не сможет разрешить доменное имя, и вы, вероятно, получите неприятное диалоговое окно, объясняющее, что DNS-сервер не найден. Однако, если вы введете соответствующий IP-адрес вместо имени домена, браузер сможет получить нужную веб-страницу. (Используйте ping для получения IP-адреса перед отключением DNS.) Другие операционные системы Microsoft аналогичны. |

## **Пересмотр протоколов Интернета**

Как указывалось ранее в разделе о стеках протоколов, можно предположить, что в Интернете используется множество протоколов. Это правда; для функционирования Интернета требуется множество протоколов связи. К ним относятся протоколы TCP и IP, протоколы маршрутизации, протоколы управления доступом к среде, протоколы прикладного уровня и т.д. В следующих разделах описываются некоторые из наиболее важных и часто используемых протоколов в Интернете. Сначала обсуждаются протоколы более высокого уровня, за ними следуют протоколы более низкого уровня.

## **Прикладные протоколы: HTTP и Всемирная паутина**

Одной из наиболее часто используемых служб в Интернете является Всемирная паутина (WWW). Протокол приложения, который обеспечивает работу Интернета, - это **Hypertext Transfer Protocol** или **HTTP**. Не путайте это с языком разметки гипертекста (HTML). HTML - это язык, используемый для написания веб-страниц. HTTP - это протокол, который веб-браузеры и веб-серверы используют для связи друг с другом через Интернет. Это протокол прикладного уровня, потому что он находится поверх уровня TCP в стеке протоколов и используется конкретными приложениями для взаимодействия друг с другом. В данном случае приложениями являются веб-браузеры и веб-серверы.

HTTP - это текстовый протокол без установления соединения. Клиенты (веб-браузеры) отправляют запросы на веб-серверы для веб-элементов, таких как веб-страницы и изображения. После того, как запрос обслуживается сервером, соединение между клиентом и сервером через Интернет отключается. Для каждого запроса необходимо устанавливать новое соединение. Большинство протоколов ориентированы на подключение. Это означает, что два компьютера, обменивающиеся данными друг с другом, поддерживают соединение открытым через Интернет. Однако HTTP этого не делает. Прежде чем клиент сможет выполнить HTTP-запрос, необходимо установить новое соединение с сервером.

Вот что происходит, когда вы вводите URL-адрес в веб-браузере:

1. Если URL-адрес содержит доменное имя, браузер сначала подключается к серверу доменных имен и получает соответствующий IP-адрес для веб-сервера.
2. Веб-браузер подключается к веб-серверу и отправляет HTTP-запрос (через стек протоколов) на нужную веб-страницу.
3. Веб-сервер получает запрос и проверяет наличие нужной страницы. Если страница существует, веб-сервер отправляет ее. Если сервер не может найти запрошенную страницу, он отправит сообщение об ошибке HTTP 404. (404 означает "Страница не найдена", как, вероятно, знает каждый, кто пользовался Интернетом.)
4. Веб-браузер возвращает страницу обратно, и соединение закрывается.
5. Затем браузер анализирует страницу и ищет другие элементы страницы, необходимые для завершения веб-страницы. Обычно к ним относятся изображения, апплеты и т.д.
6. Для каждого необходимого элемента браузер выполняет дополнительные подключения и HTTP-запросы к серверу для каждого элемента.
7. Когда браузер завершит загрузку всех изображений, апплетов и т.д. страница будет полностью загружена в окне браузера.

|  |
| --- |
| **Проверьте это - используйте свой клиент Telnet для получения веб-страницы с использованием HTTP** |
| Telnet - это служба удаленного терминала, используемая в Интернете. В последнее время ее использование сократилось, но это очень полезный инструмент для изучения Интернета. В Windows найдите программу telnet по умолчанию. Он может быть расположен в каталоге Windows с именем telnet.exe. После открытия откройте меню терминала и выберите Настройки. В окне настроек проверьте Local Echo. (Это для того, чтобы вы могли видеть свой HTTP-запрос при его вводе.) Теперь откройте меню подключения и выберите Удаленная система. Введите www.google.com для имени хоста и 80 для порта. (Веб-серверы обычно прослушивают порт 80 по умолчанию.) Нажмите Подключиться. Теперь введите  ПОЛУЧИТЬ / HTTP/1.0  и дважды нажмите Enter. Это простой HTTP-запрос к веб-серверу для его корневой страницы. Вы должны увидеть, как промелькнет веб-страница, а затем должно появиться диалоговое окно, сообщающее вам, что соединение было потеряно. Если вы хотите сохранить извлеченную страницу, включите вход в систему в программе Telnet. Затем вы можете просмотреть веб-страницу и увидеть HTML, который использовался для ее написания. |

Большинство интернет-протоколов определяются интернет-документами, известными как **Запрос комментариев** или **RFC**. RFC можно найти в нескольких местах в Интернете. Соответствующие URL-адреса см. в разделе Ресурсы ниже. HTTP версии 1.0 указан в RFC 1945.

## **Прикладные протоколы: SMTP и электронная почта**

Другой широко используемой интернет-службой является электронная почта. Электронная почта использует протокол прикладного уровня, называемый **Simple Mail Transfer Protocol** или **SMTP**. SMTP также является текстовым протоколом, но в отличие от HTTP, SMTP ориентирован на подключение. SMTP также сложнее HTTP. В SMTP намного больше команд и соображений, чем в HTTP.

Когда вы открываете свой почтовый клиент, чтобы прочитать электронную почту, обычно происходит вот что:

1. Почтовый клиент (Netscape Mail, Lotus Notes, Microsoft Outlook и т.д.) открывает соединение со своим почтовым сервером по умолчанию. IP-адрес почтового сервера или доменное имя обычно настраиваются при установке почтового клиента.
2. Почтовый сервер всегда будет передавать первое сообщение для идентификации себя.
3. Клиент отправит команду SMTP HELO, на которую сервер ответит сообщением 250 OK.
4. В зависимости от того, проверяет ли клиент почту, отправляет ли почту и т.д. соответствующие SMTP-команды будут отправлены на сервер, который ответит соответствующим образом.
5. Эта транзакция запроса / ответа будет продолжаться до тех пор, пока клиент не отправит команду SMTP QUIT. Затем сервер попрощается, и соединение будет закрыто.

Простой "диалог" между SMTP-клиентом и SMTP-сервером показан ниже. **R:** обозначает сообщения, отправляемые сервером (получателем), а **S:** обозначает сообщения, отправляемые клиентом (отправителем). В этом примере SMTP показана почта, отправленная Смитом с хоста USC-ISIF на адрес   
 Джонса, Грина и Брауна с хоста BBN-UNIX. Здесь мы предполагаем, что   
 хост USC-ISIF напрямую связывается с хостом BBN-UNIX. Почта   
 принимается для Джонса и Брауна. У Грина нет почтового ящика по адресу   
 хост BBN-UNIX.  
  
 -------------------------------------------------------------  
  
 R: 220 BBN-UNIX.ARPA Simple Mail Transfer Service готова  
 S: HELO USC-ISIF.ARPA  
 R: 250 BBN-UNIX.ARPA  
  
 S: ПОЧТА ОТ:<Smith@USC-ISIF.ARPA>  
 R: 250 OK  
  
 S: RCPT КОМУ:<Джонс@BBN-UNIX.ARPA>  
 R: 250 ОК  
  
 S: RCPT ДЛЯ:<Green@BBN-UNIX.ARPA>  
 R: 550 Здесь нет такого пользователя  
  
 S: RCPT КОМУ:<Brown@BBN-UNIX.ARPA>  
 R: 250 OK  
  
 S: ДАННЫЕ  
 R: 354 Начинает ввод почты; заканчивается <CRLF>.<CRLF>  
 S: Бла-бла-бла...  
 S: ... и т.д. и т.п. и т.п.  
 S: .  
 R: 250 OK  
  
 S: ВЫЙТИ  
 R: 221 BBN-UNIX.Служба ARPA закрывает канал передачи  
Эта SMTP-транзакция взята из RFC 821, в котором указан SMTP.

## **Протокол управления передачей**

Под прикладным уровнем в стеке протоколов находится уровень TCP. Когда приложения открывают соединение с другим компьютером в Интернете, сообщения, которые они отправляют (используя определенный протокол прикладного уровня), передаются по стеку на уровень TCP. **TCP отвечает за маршрутизацию протоколов приложений к нужному приложению на целевом компьютере**. Для достижения этой цели используются номера портов. Порты можно рассматривать как отдельные каналы на каждом компьютере. Например, вы можете просматривать веб-страницы во время чтения электронной почты. Это связано с тем, что эти два приложения (веб-браузер и почтовый клиент) использовали разные номера портов. Когда пакет поступает на компьютер и поднимается по стеку протоколов, уровень TCP решает, какое приложение получит пакет, на основе номера порта.

TCP работает следующим образом:

* Когда уровень TCP получает данные протокола прикладного уровня сверху, он сегментирует их на управляемые "фрагменты", а затем добавляет заголовок TCP с определенной информацией TCP к каждому "фрагменту". Информация, содержащаяся в заголовке TCP, включает номер порта приложения, на которое необходимо отправить данные.
* Когда уровень TCP получает пакет от уровня IP ниже него, уровень TCP удаляет данные заголовка TCP из пакета, при необходимости выполняет некоторую реконструкцию данных, а затем отправляет данные в правильное приложение, используя номер порта, взятый из заголовка TCP.

Именно так TCP направляет данные, проходящие через стек протоколов, в нужное приложение.

TCP - это не текстовый протокол. **TCP - это ориентированный на соединение, надежный сервис передачи байтов**. Ориентированный на соединение означает, что два приложения, использующие TCP, должны сначала установить соединение перед обменом данными. Протокол TCP надежен, потому что для каждого полученного пакета отправителю отправляется подтверждение о доставке. Протокол TCP также включает контрольную сумму в свой заголовок для проверки полученных данных на ошибки. Заголовок TCP выглядит следующим образом:

|  |
| --- |
| Диаграмма 7 |
| Диаграмма 7 |

Обратите внимание, что в заголовке TCP нет места для IP-адреса. Это потому, что TCP ничего не знает об IP-адресах. Задача TCP заключается в надежной передаче данных прикладного уровня от приложения к приложению. Задача передачи данных с компьютера на компьютер - это работа IP.

|  |
| --- |
| **Проверьте это - Хорошо известные номера интернет-портов** |
| Ниже перечислены номера портов для некоторых наиболее часто используемых интернет-сервисов.   |  |  | | --- | --- | | FTP | 20/21 | | Telnet | 23 | | SMTP | 25 | | HTTP | 80 | | Арена Quake III | 27960 | |

## **Интернет-протокол**

В отличие от TCP, **IP - ненадежный протокол без установления соединения**. IP не заботится о том, попадет пакет в пункт назначения или нет. IP также не знает о соединениях и номерах портов. **Задача IP заключается в том, чтобы отправлять и маршрутизировать пакеты на другие компьютеры**. IP-пакеты являются независимыми объектами и могут поступать не по порядку или не поступать вообще. Работа TCP заключается в том, чтобы убедиться, что пакеты поступают в правильном порядке. Пожалуй, единственное, что IP имеет общего с TCP, - это способ, которым он получает данные и добавляет к данным TCP информацию собственного IP-заголовка. Заголовок IP выглядит следующим образом:

|  |
| --- |
| Диаграмма 8 |
| Диаграмма 8 |

Выше мы видим IP-адреса отправляющего и принимающего компьютеров в заголовке IP. Ниже показано, как выглядит пакет после прохождения через уровень приложений, уровень TCP и уровень IP. Данные прикладного уровня сегментируются на уровне TCP, добавляется заголовок TCP, пакет переходит на уровень IP, добавляется заголовок IP, а затем пакет передается через Интернет.

|  |
| --- |
| Диаграмма 9 |
| Диаграмма 9 |

## **Подведение итогов**

Теперь вы знаете, как работает Интернет. Но как долго он будет оставаться таким? Версия IP, используемая в настоящее время в Интернете (версия 4), допускает только 232 адреса. В конечном итоге свободных IP-адресов не останется. Удивлены? Не волнуйтесь. IP-версия 6 прямо сейчас тестируется на исследовательской магистрали консорциумом исследовательских институтов и корпораций. И что после этого? Кто знает. Интернет прошел долгий путь с момента своего создания в качестве исследовательского проекта Министерства обороны. Никто на самом деле не знает, во что превратится Интернет. Однако одно можно сказать наверняка. Интернет объединит мир так, как ни один другой механизм никогда не объединял. Информационная эпоха идет полным ходом, и я рад быть ее частью.