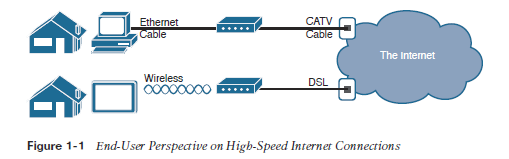
# Перспективы создания сетей

И так, ты новичок в сетях. Как и у многих людей, твой подход к сетям вероятно, как у пользователя сетей, в противоположность сетевому инженеру, который строит сети. Для некоторых, твой взгляд на сети вероятно строится на том, как ты используешь интернет, из дома, при помощи высокоскоростного интернет соединения, как цифровая абонентская линия (DSL - digital subscriber line) или кабельного TV, что показано на рисунке 1-1.

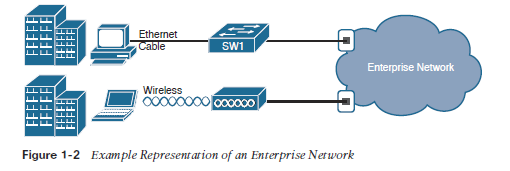


В верхней части рисунка показано типичное высокоскоростное подключение пользователя к интернету по кабелю. Компьютер подключён к кабельному модему через провод Ethernet. Кабельный модем затем подключается по коаксиальному кабелю к розетке CATV на стене - тот же самый тип (**the same kind of**) кабеля используется для подключения телевизора к розетке CATV. По причине того, что кабельный интернет сервис предоставляет (provide) сервис непрерывно, пользователь может сесть за компьютер и начать отправлять письма, просматривать web-сайты, совершать звонки через интернет, использовать другие инструменты и приложения.

В нижней части изображения используются две разные технологии. Первая, планшет использует беспроводную технологию, имя которой беспроводная локальная сеть (wireless LAN) или Wi-Fi, вместо (**instead**) использования Ethernet кабеля. В этом примере роутер использует другую технологию для подключения к Интернет - DSL (digital subscriber line).

И то и другое: и домашние сети, и сети, построенные для использования компаниями, используют аналогичные сетевые технологии. IT (The information technology) мир имеет отношение к сетям, созданным корпорацией или предприятием, с целью (**purpose**) доступа (**allowing**) своих сотрудников к средству связи, как корпоративная сеть. Небольшие сети дома, которые используют для бизнес-задач, часто называют SOHO (Small office /Home office) сетями.

Пользователи корпоративных сетей имеют некоторое представление о корпоративной сети в их компании или школе. Люди понимают (**realize**), что они используют сеть для большого числа задач. Пользователи ПК вероятно понимают, что их компьютер подключен через Ethernet кабель в соответствующую (**matching**) розетку, как показано на рисунке 1-2. Так же те же пользователи могут использовать беспроводное соединение Wi-Fi на своих ноутбуках, когда идут на встречу в конференц-зал. На рисунке 1-2 показаны эти два взгляда конечных пользователей на корпоративную сеть.



В схеме сети, облако является частью сети, чьи детали не важны для целей схемы. В данном случае рисунок 1-2 игнорирует детали того, как создается корпоративная сеть.

Некоторые пользователи вероятно даже не имеют представления (**not even have a concept**) о сети в целом. Зато, эти пользователи просто наслаждаются функциями сети, возможностью оставлять сообщения в социальных сетях, совершать телефонные звонки, искать информацию в интернете, слушать музыку и загружать бесчисленное количество приложений на их смартфоны, без заботы о том, как это работает или как их любимый девайс подключается к сети.

Вне зависимости от того (**Regardless of**), как много вы уже знаете о том, как сеть работает, эта книга и связанная сертификация помогут Вам изучить как сети выполняют их работу. Эта работа - это проста: перемещение информации от одного устройства к другому. Остальная часть этой главы и остальная часть этой книги раскрывают (проливают свет) (**reveals**) основы, как создавать SOHO и корпоративные сети, так что бы они могли передавать данные между двумя устройствами.

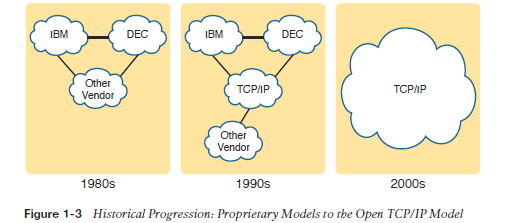
В строительном бизнесе большая работа происходит до того, как будет вбит первый гвоздь. Процесс начинается с некоторого планирования, с понимания того, как строить дом и некоторых архитектурных чертежей для строительства именно этого конкретного дома. Аналогично, дорога к построению какой-либо компьютерной сети не начинается с установки устройств и кабелей, а вместо этого (**instead**), со взгляда на архитектурные планы для этих современных сетей: модель TCP/IP.

# Сетевая модель TCP/IP

Сетевая модель, иногда также называют либо (**either**) **сетевой архитектурой** или **сетевой проект**, относится (refers) к исчерпывающему (**comprehensive**) набору документации. По отдельности каждый документ описывает (**describes**) одну небольшую функцию, необходимую (**required**) для сети; вместе эти документы определяют (**define**) все, что должно произойти для того, чтобы сеть работала. Некоторые документы определяют (**define**) протокол**,** который представляет собой набор логических правил, которым устройства должны следовать для связи. Другие документы определяют (**define**) некоторые физические требования (**requirements**) для сети. Например, документ может определять напряжение тока и уровень тока на конкретном (**particular**) кабеле при передаче данных. Вы можете думать о сетевой модели как об архитектурном чертеже для строительства дома. Конечно Вы можете построить дом без чертежа. Однако, чертеж может гарантировать что дом имеет хорошую основу и структуру, вследствие чего не упадет вниз, и он имеет правильные скрытые места для размещения (**accommodate**) сантехники, электричества, газа и т.д. Так же, множество разных людей, которые строят дом, используют чертеж – такие как плотники, электрики, каменщики, маляры и тд. знают, что, если они следуют чертежу, то их часть работы не будет причиной (**cause**) проблем для остальных рабочих. Аналогично, вы можете построить свою собственную сеть, написать свое собственное программное обеспечение, создать свои собственные сетевые карты и т.д. что бы создать сеть. Однако, гораздо проще просто купить и использовать продукты, которые уже соответствуют (**already conform**) какой-либо известной сетевой модели или чертежу. Так как производители сетевой продукции создают свои изделия согласно (**in mind**) определенной сетевой модели, их продукция должна хорошо работать вместе.

# История перехода к TCP/IP

Сегодня компьютерный мир использует сетевую модель TCP/IP. Однако, мир не всегда был таким простым. Давным-давно сетевые протоколы даже не существовали, в том числе TCP/IP. Производители создавали первые сетевые протоколы. Эти протоколы поддерживали только компьютеры этих производителей. Например, IBM опубликовала сетевую модель SNA в 1974 году. Другие производители так же создавали свои собственные (**own**) патентованные (**proprietary**) сетевые модели. Как результат, если твоя компания покупала компьютеры от трех разных вендоров, сетевые инженеры часто создавали три разные сети на основе сетевых моделей, созданных каждой (**each**) компанией, затем каким-нибудь образом (**somehow**) объединяли эти сети, комбинируя сеть в более сложный комплекс. Левая часть рисунка 1-3 показывает главную идею того, как могла выглядеть корпоративная сеть компании еще в 1980 году, перед тем как TCP/IP получила широкое распространение (**became common**) в корпоративных сетях.



Хотя (**Although**) собственные сетевые модели производителей-регуляторов часто работали хорошо, открытая (**open**), сетевая модель независимая от производителей будет способствовать конкуренции (**aid competition**) и уменьшению сложности (**reduce complexity**). Международная организация по стандартизации (ISO) взяла на себя (**took on**) задачу создать такую (**such**) модель, стартовав еще в конце 1970-х годов, начав работу над тем, что получило имя Open System Interconnection (OSI). ISO имела благородную цель (**noble goal**) в создании модели OSI: стандартизировать протоколы сети передачи данных для обеспечения (**allow**) связи между (**among**) всеми компьютерами на всей планете. ISO шла к этой грандиозной и благородной цели, с участниками (**participants**) из большинства технологически развитых стран на Земле, участвующих в этом процессе.

Вторая, менее формальная попытка создать открытую, независящую от производителей сетевую модель проросли вперед из контракта департамента обороны США (DOD). Исследователи из различных университетов вызвались помочь (**volunteered to help**) в дальнейшем (**further**) развитии протоколов, окружающих оригинальный работающий DOD. Эти старания привели к конкурирующей (**competing**) модели открытой сети под названием TCP/IP.

В течение 1990-х годов компании начали добавлять OSI, TCP/IP или и то и другое в свои корпоративные сети. Однако в конце 1990-х TCP/IP стал общим (**common**) выбором и OSI отпала. В центральной части рисунка 1-3 показана главная идея, на которой основывались корпоративные сети в этом десятилетии, до сих пор (**still**) сети основаны на нескольких сетевых моделях, но в том числе и TCP/IP. Здесь, в 20 веке TCP/IP доминирует. Собственные сетевые модели до сих пор существуют (**still** **exist**), но от них в основном (**mostly**) отказываются (**discarded**) в пользу (**in favor**) TCP/IP. Модель OSI, чья разработка пострадала отчасти (**suffered in part**) из-за процесса медленной формальной стандартизации в сравнении (**compared**) с TCP/IP, не была так успешна на рынке. И TCP/IP, сетевая модель, изначально созданная почти (**almost**) полностью (**entirely**) группой (**a** **bunch**) добровольцев, стала самой преуспевающей (**prolific**) сетевой моделью когда-либо, как показано на правой части рисунка 1-3. В этой главе, вы будете читать о некоторых из основ TCP/IP. Несмотря на то, что вы узнаете некоторые интересные факты о TCP/IP, истинная цель (**true goal**) этой главы заключается в том, чтобы помочь вам понять, какие сетевая архитектура и сетевая модель в реальности и как это работает. Также в этой главе Вы выучите некоторый жаргон, используемый в OSI. Будете ли Вы когда-нибудь работать на компьютере, использующем полную версию протоколов OSI, исключая использование TCP/IP. Наверное (**Probably**), нет. Однако Вы будете часто использовать термины, связанные (**relating**) с TCP/IP.

# Взгляд на сетевую модель tcp/IP

Модель TCP/IP определяет (**defines**) и ссылается (**references**) на большую коллекцию протоколов, которые позволяют (**allow**) компьютерам связываться. Для определения (**define**) протоколов TCP/IP использует документацию, которая носит название request for comments RFC (Вы можете найти эти RFC используя любой интернет поисковик). TCP/IP модель также избегает (**avoids**) повторений уже проделанной работы каким-либо другим органом стандартизации или консорциумом производителей, просто ссылаясь на стандарты или протоколы, созданные этими группами. Например, институт IEEE определяет Ethernet LANs. TCP/IP не определяет Ethernet в RFC, но ссылается к IEEE как к опции.

Легко можно сделать сравнение (**comparison**) между телефонами и компьютерами, которые использует TCP/IP. Ты идешь в магазин и покупаешь телефон от одного из массы (**dozen**) различных производителей. Когда ты приходишь домой и подключаешь телефон к тому же самому кабелю, к которому был подключён твой старый телефон, новый телефон работает. Производители телефонов знают стандарты для телефонов в твоей стране и создают телефоны, соответствующие (**to match**) этим стандартам.

Аналогично, когда ты покупаешь новый компьютер сегодня, он реализует (**implements**) модель TCP/IP к точке, и ты как обычно можешь взять компьютер из коробки, подключить правильный кабель, включить его и подключить к сети. Вы можете использовать браузер для подключения к вашим любимым сайтам. Как? Ну, ОС на компьютере реализует часть модели TCP/IP. Карта Ethernet или карта WI-FI, встроенные в компьютер реализует некоторые стандарты LAN на которое ссылается TCP/IP модель. Короче говоря, поставщики, которые создают аппаратное и программное обеспечение реализуют TCP/IP.

Для того, чтобы помочь понять людям сетевую модель, каждая модель разбивает функции на небольшое число категорий, называемые уровнями, каждый уровень включает протоколы и стандарты, которые связаны (**relate**) c этой категорией функции. TCP/IP на самом деле (**actually**) имеет две альтернативные модели, как показано на рисунке 1-4.



Модель в левой части показывает оригинальную модель TCP/IP, описанную в RFC 1122, с разбивкой TCP/IP на 4 уровня. Верхние два уровня (**Application**, **Transport**) фокусируются больше на приложениях, которым необходимо отправлять и получать данные. Нижний слой (**Link Layer**) фокусируется на передаче Bit-ов через каждое индивидуальное соединение, с Internet **Layer** фокусируются на доставке данных через весь путь (**entire path**) от начального передающего компьютера к финальному компьютеру получателю.

TCP/IP модель в правой части показывает более общие (**more common**) термины и уровни, используемые, когда люди говорят о TCP/IP сегодня.

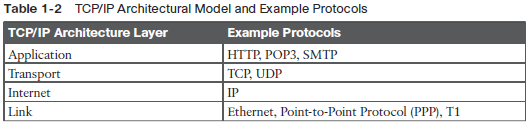
Он расширяет (**expands**) канальный уровень исходной модели на два отдельных (**separate**) уровня**: канал передачи данных** и **физический** (по аналогии двух нижних уровней модели OSI). Также, многие (**many**) люди обычно (**commonly**) используют слово "Network" вместо "Internet" для одного слоя.

Заметка

Канальный уровень изначальной модели TCP/IP также упоминается (**referred to**) как уровень сетевого доступа и сетевого интерфейса.

Многие из Вас уже слышали о нескольких протоколах TCP/IP, как примеры, перечисленные в таблице 1-2. Большинство протоколов и стандартов в этой таблице будет объяснено **(explained)** более подробно **(in more**

**detail)** если вы работаете по этой книге. После таблицы в этой секции будет тщательное рассмотрение (**a closer look**) уровней модели TCP/IP.



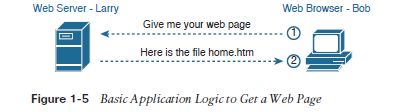
# TCP/IP прикладной уровень

Протоколы прикладного уровня TCP/IP предоставляют (**provide**) услуги программному обеспечению, работающему на компьютере. Прикладной уровень не определяет само приложение, но он определяет услуги, которые нужны приложению. Например, прикладной протокол HTTP определяет, как веб-браузеры могут вытащить содержимое веб-страницы с веб-сервера. Короче говоря, прикладной уровень обеспечивает интерфейс между программным обеспечением, запущенным на компьютере и самой сетью.

Возможно (**Arguably**), наиболее популярным приложением TCP/IP сегодня является веб-браузер. Многие крупные (**major**) поставщики программного обеспечения или (**either**) уже изменили или изменяют их прикладное программное обеспечение для поддержки доступа из веб-браузера. И к счастью, использовать Web-браузер легко: вы запускаете веб-браузер на компьютере и выбираете веб-сайт, введя имя веб-сайта, и появляется (**appears**) веб-страница.

# Взгляд на HTTP

Что в действительности должно было произойти, чтобы разрешить этой веб-странице появиться в вашем веб-браузер? Представьте себе (**Imagine**), что Боб открывает свой браузер. Его браузер был настроен автоматически запрашивать страницу по умолчанию веб-сервера Ларри или домашнюю страницу. Общая логика выглядит как на рисунке 1-5.



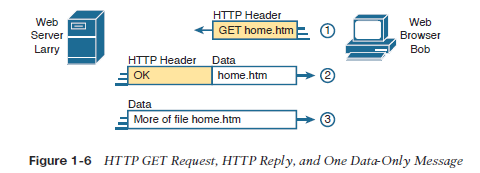
И так, что же на самом деле произошло? Первоначальный (**initial**) запрос (**request**) Боба на самом деле (**actually**) просит Ларри послать его домашнюю страницу обратно Бобу. Программное обеспечение веб-сервера Ларри настроено знать, что страница по умолчанию содержится (**contained**) в файле под названием home.htm. Боб получает файл от Ларри и отображает содержимое в окне браузера Боба.

# Механизм протокола HTTP

При ближайшем рассмотрении (**Taking a closer look**), этот пример показывает, как приложение на каждом конечном компьютере – особенно (**specifically**), приложение веб-браузера и приложение веб сервера – используют TCP/IP протокол прикладного уровня. Чтобы сделать запрос на веб-страницу и вернуть содержимое веб-страницы, приложения используют протокол передачи гипертекста (HTTP).

HTTP не существовало до тех пор, пока Тим Бернерс-Ли создал первый веб-браузер и веб-сервер в начале 90-х годов. Бернерс-Ли дал HTTP функциональность запросов для содержимого веб-страниц, в частности (**specifically**), путем предоставления веб-браузеру возможность запрашивать файлы с сервера и дал серверу способ вернуть содержимое этих файлов. Общая логика соответствует тому, что было показано на рисунке 1-5; Рисунок 1-6 показывает такую же идею, но с деталями, характерными (**specific**) для HTTP.

Полная версия большинства веб-адресов — также называется универсальный указатель ресурса (Uniform Resource Locators) или универсальные идентификаторы ресурсов (URI) — начинается с букв http, что означает, что HTTP используется для передачи веб-страницы.



Чтобы получить веб-страницу от Ларри, **на шаге 1**, Боб посылает сообщение с заголовком HTTP. Как правило (Generally), протоколы используют заголовки, как место для размещения информации, используемой этим протоколом. Этот заголовок HTTP включает запрос "**получить**" файл. Запрос обычно содержит (**contains**) имя файл (**home.htm, в данном случае**), или если имя файла не упоминается (**mentioned**), веб-сервер предполагает (**assumes**), что Боб хочет **веб-страницу по умолчанию.**

**Шаг 2** на Рисунке 1-6 показан ответ от веб-сервера Ларри. Сообщение начинается с HTTP заголовка с кодом возврата (200), что означает, что-то так же просто, как "OK" возвращается в заголовке. HTTP также определяет другие коды возврата, так что сервер может сообщить браузеру cработал ли запрос. (Вот еще один пример: если вы когда-нибудь видели “**Страница не была найдена**”, а затем получил HTTP 404 «не найдено» ошибка, вы получили код возврата HTTP 404). Второе сообщение также включает в себя первую часть запрашиваемого файла.

**Шаг 3** на рисунке 1-6 изображено другое сообщение с веб-сервера Ларри веб-браузеру Боба, но на этот раз без заголовка HTTP. HTTP передает данные путем отправки нескольких сообщений, каждое с частью файла. Вместо того (**Rather than**), чтобы тратить (**wasting**) пространство, посылая повторяющиеся заголовки HTTP, этот же список и та же информация, эти дополнительные сообщения просто опустят (**omit**) заголовок.

# Транспортный уровень TCP/IP

Хотя существует множество протоколов прикладного уровня TCP/IP, транспортный уровень включает в себя меньшее количество (**smaller number**) протоколов. Два наиболее часто (**most commonly**) используемые протокола транспортного уровня - Transmission Control Protocol (TCP) и User Datagram Protocol (UDP). Протоколы **транспортного уровня** предоставляют сервисы протоколам **прикладного уровня**, которые находятся (**reside**) на один уровень выше в модели TCP/IP. Как протокол транспортного уровня предоставляет услуги протоколу более высокого уровня? В данном разделе представлена (**introduces**) общая концепция, фокусирующаяся на единственной службе, предоставляемой TCP: исправление ошибок (error recovery). В последующих главах рассмотрим (**examine**) транспортный уровень более подробно и обсудим другие функции транспортного уровня.

**Основы TCP/IP Error Recovery**

Чтобы по достоинству оценить (**appreciate**) то, что делают протоколы транспортного уровня, вы должны думать об уровне выше транспортного уровня, о прикладном уровне. Почему? Ну, каждый уровень обеспечивает сервис для уровня над ним, как сервис восстановления ошибок (**Error Recovery**), предоставляемый протоколам прикладного уровня с помощью TCP.

Например, на рисунке 1-5, Боб и Ларри использовали HTTP для передачи домашней страницы с Веб-сервераЛарри в веб-браузер Боба. Но что случилось бы, если бызапрос Боба HTTP GET был потерян в транзите через TCP / IP сети? Или, что произошло бы,если ответ Ларри, который включал содержимое домашней страницы, был потерян? Хорошо, Вы вероятно думаете, в любом случае, страница не отобразилась (**shown up**) бы в браузере Боба.

TCP / IP необходим механизм, для гарантированной доставки данных по сети. Поскольку многие протоколы прикладного уровня вероятно хотят способ для гарантированной доставки данных по сети, создатели TCP включили функцию исправления ошибок (**error-recovery feature**). Чтобы оправиться от ошибок, TCP использует концепцию подтверждений приема (**acknowledgments**). Рисунок 1-7 описывает основную идею, как TCP замечает потерянные данные и просит отправителя повторить попытку.

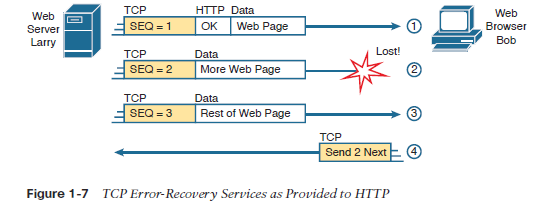


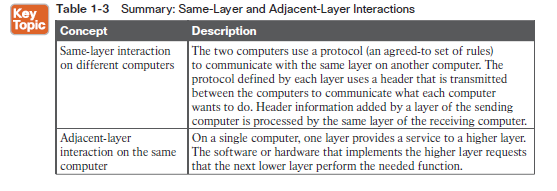
Рисунок 1-7 показывает, что веб-сервер Ларри отправляет веб-страницу в веб-браузер Боба, используя три отдельных (**separate**) сообщения. Обратите внимание (**Note**), что этот рисунок показывает те же HTTP-заголовки как рисунок 1-6, но он также показывает заголовок TCP. Заголовок TCP показывает **порядковый** номер (SEQ) с каждым сообщением. В этом примере сеть имеет проблемы, и сеть не доставит **сообщение TCP** (**так называемый сегмент**) с порядковым номером 2. Когда Боб получает сообщения с порядковыми номерами 1 и 3, но не получает сообщение с порядковым номером 2, Боб понимает, что это сообщение 2 было потеряно. Это осознание логики TCP Боба причина отправки TCP сегмента обратно Ларри, запрашивая Ларри отправить сообщение 2 снова.

**Взаимодействие аналогичных и смежных (соседних) уровней. Same-Layer and Adjacent-Layer Interactions**

Пример на рисунке 1-7 также демонстрирует функцию, называемую **взаимодействие соседних уровней (adjacent-layer interaction),** которая относится к понятиям о том, как соседние уровни в сетевой модели на одном компьютере работают вместе. В этом примере протокол более высокого уровня (**HTTP**) хочет восстановить ошибку и более высокий уровень использует протокол следующего нижнего уровня (**TCP**) для выполнения **(to perform)** **службы восстановления ошибок**; Нижний уровень предоставляет сервис уровню над (**above**) ним.

На рисунке 1-7 также показан пример подобной функции, называемой **взаимодействие аналогичных уровней (Same-Layer interaction)**. Когда конкретный уровень на одном компьютере хочет cвязаться с тем же уровнем на другом компьютере, два компьютера используют заголовки для хранения информации о том, как они хотят связаться. Например, на рисунке 1-7, Ларри установил (**set**) последовательность **(sequence)** чисел 1, 2, и 3 так, чтобы Боб мог заметить, когда некоторые из данных не поступили. Процесс **TCP** Ларри создал **TCP**-заголовок с порядковым (**sequence**) номером; процесс TCP Боба получил и отреагировал на сегмент (сообщение) **TCP**.

В таблице 1-3 приведены ключевые моменты, о том, как соседние уровни работают вместе на одном компьютере и как один уровень на одном компьютере работает с тем же самым сетевым уровнем на другом компьютере.

.

**Взаимодействие аналогичных уровней (Same-Layer interaction)**

Два компьютера используют протокол (согласно набору правил) для связи c тем же самым уровнем на другом компьютере. Протокол, определяемый для каждого уровня, использует заголовок, который передается между компьютерами, чтобы сообщить, что каждый компьютер хочет сделать. Информация заголовка, добавленная уровнем посылающего компьютера, **обрабатывается (is processed)** тем же уровнем принимающего компьютера

**Взаимодействие соседних уровней (Adjacent -Layer interaction)**

На одном компьютере, один уровень предоставляет услугу более высокому уровню. Программное обеспечение или оборудование, которое реализует более высокий уровень запрашивает следующий нижний уровень **выполнить (perform)** необходимую функцию.

# Сетевой уровень TCP/IP

**Прикладной уровень** включает множество протоколов. **Транспортный уровень** включает **меньше (fewer)** протоколов, **в первую очередь** **(most notably)**, TCP и UDP. **Сетевой уровень** TCP/IP включает небольшое число протоколов, но только один основной протокол: Интернет-протокол (**IP**). **Кстати** **(In fact)**, имя TCP/IP это просто имена двух **наиболее распространенных** **(most common)** протоколов (**TCP** и **IP**) разделенных **/**.

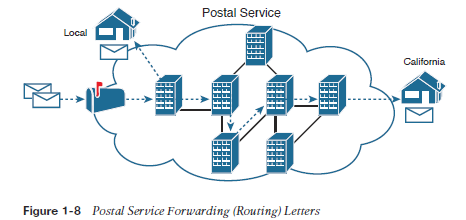
IP предоставляет **несколько** **(several)** функций, самые главные, адресация и маршрутизация. Этот раздел начинается с сравнения IP адресации и маршрутизации с другой широко известной системой, которая использует адресацию и маршрутизацию: **Почтовая служба**. После этого в данном разделе представлены IP адресация и маршрутизация. (Более подробно рассмотрено в Главе 4, «**Основы IPv4 адресация и маршрутизация**.»)

**Интернет протокол и почтовая служба.**

Представьте, что вы только что написали два письма: одно другу на другом конце страны и одно к другу на другом конце города. **Есть** **(Is there)** **много различий (much difference)** в том, как вы **относитесь** **(treat)** к каждому письму? Не совсем. Как правило вы бы просто положили их в один и тот же почтовый ящик и **ожидали (expect)** что почтовая служба доставит оба письма.

Почтовая служба, однако, должны думать о каждом письме по отдельности, а затем **принять решение (make a decision)** о том, куда отправить каждое письмо так, чтобы оно было доставлено. Для отправки письма через город людям в местном почтовом отделении, вероятно, просто нужно положить письмо в другую грузовую машину.

Для письма, которое должно идти через всю страну, почтовая служба отправляет письмо в другое почтовое отделение, потом в другое, и так далее, пока письмо не будет доставлено через всю страну. В каждом почтовом отделении, почтовая служба должна обработать письмо и выбрать, куда отправить его дальше.

Для того, чтобы все это работало, почтовая служба имеет регулярные маршруты для небольших грузовых автомобилей, больших грузовиков, самолетов, катеров, и так далее, для перемещения писем между зонами почтовой службы. Служба должна быть в **состоянии (able)** принимать и пересылать письма, и она должна принимать правильные **решения (decisions)** о том, куда отправить каждое письмо дальше, как показано на рисунке 1-8. Все еще размышляя о почтовой службе, рассмотрим разницу между лицом, отправившим письмо и работой, которую делает почтовая служба. Лицо, пославшее письма, **расчитывает (expects)**, что почтовая служба **в большинстве случаев (most of the time)** доставит письмо. Однако лицу, отправившему письмо не нужно знать детали того, какой именно путь прошло письмо. В тоже время, почтовая служба не создает письмо, но принимает письмо от клиента. Затем почтовая служба должны знать подробности об адресах и почтовых индексах для группировки адресов в более крупные группы, и она должна иметь возможность доставлять письма.

**Прикладной [Application]** и **транспортный [Transport]** **уровни** TCP/IP действуют как человек, посылающий письма через почтовую службу. Эти верхние уровни работают одинаково, **независимо от того (regardless of whether)** находятся **ли** конечные хосты компьютеров в одной локальной сети или разделены через весь интернет. Для отправки сообщения эти верхние уровни просят уровень ниже их, **сетевой уровень,** доставить сообщение.

**Канальный [Link] и Сетевой [Internet] -** нижние уровнимодели TCP/IP **действуют (act)** больше как почтовая служба для доставки этих сообщений к правильным адресатам (получателям). Для этого эти нижние уровни должны понимать **лежащую в основе (underlying)** физическую сеть, потому что они должны выбрать, как лучше передать данные с одного узла на другой.

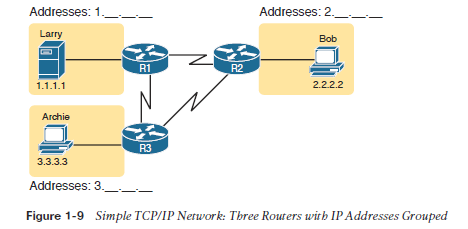
Итак, как это все касается сетей? Ну, **сетевой уровень** сетевой модели TCP / IP, **в первую очередь (primarily)** определяемый **интернет-протоколом (IP)**, работает **так же (much like)**, как почтовая служба. IP определяет, что каждый хост-компьютер **должен (should)** иметь разный IP-адрес, так же, как почтовая служба определяет адрес, который является уникальным для каждого дома, квартиры и бизнеса. Аналогично, IP-определяет процесс маршрутизации, **поэтому (so that)** устройства, называемые маршрутизаторы могут работать как почтовое отделение, **пересылая (forwarding)** пакеты данных **в следствие чего** **(so that)**, они будут доставлены в правильные пункты назначения. Подобно тому, как почтовая служба создала **необходимую (necessary)** инфраструктуру для доставки писем-почтовые отделения, сортировочные машины, грузовики, самолеты и персонал - **сетевой уровень** определяет детали того, как сетевая инфраструктура должна быть создана так, чтобы сеть могла передавать данные на все компьютеры в сети.

TCP / IP определяет две версии IP: IP версии 4 (IPv4) и IP версии 6 (IPv6). Мир **до сих пор (still)** **в основном (mostly)** использует IPv4, так что эта вводная часть книги использует IPv4 для всех **ссылок (references)** на IP. Позже в этой книге, часть VIII, "IP версии 6," обсуждает эту новую версию протокола IP.

**Основы адресации протокола Интернета**

IP определяет адреса по **нескольким (several)** важным **причинам (reasons).** Во-первых, каждое устройство, которое использует TCP/IP — каждый узел TCP/IP — нуждается в уникальном адресе, **чтобы (so that)** его можно было идентифицировать в сети. IP также определяет способ группировки адресов вместе, так же, как почтовая система группирует адреса, основанные на почтовых кодах (например, почтовые индексы в США).

Чтобы понять основы, **изучите (examine)** рисунок 1-9, который показывает **знакомые (familiar)** веб-сервер Ларри и веб-браузер Боба; но теперь, вместо того чтобы игнорировать сеть между этими двумя компьютерами, часть сетевой инфраструктуры включена.



Во-первых, заметьте, что рисунок 1-9 показывает некоторые примеры IP-адресов. Каждый IP-адрес имеет четыре числа, разделенных точками. В этом случае Ларри использует IP-адрес 1.1.1.1 и Боб использует 2.2.2.2. Этот стиль нумерации называется точечно десятичной нотацией (**DDN - dotted-decimal notation**).

Рисунок 1-9 также показывает три группы адресов. В этом примере все IP-адреса, которые начинаются с 1 должны быть в верхнем левом углу, как показано условным обозначением на рисунке в виде 1.\_ \_.\_ . Все адреса, которые начинаются с 2 должны быть справа, как показано условным обозначением в виде 2.\_.\_.\_. Наконец все IP-адреса, которые начинаются с 3 должно быть в нижней части рисунка.

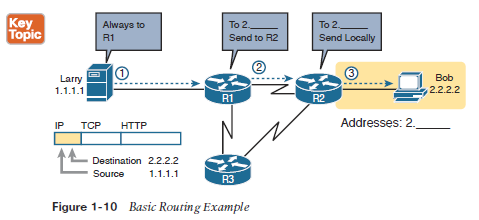
**Кроме того (In addition), на** рисунке 1-9 представлены значки, которые представляют IP-маршрутизаторы. Маршрутизаторы – это сетевые устройства, которые соединяют части сети TCP/IP вместе с целью маршрутизации (пересылки) IP-пакетов в правильное место назначения. Маршрутизаторы делают эквивалент работы, проделанной на каждом участке почтового отделения: они получают IP-пакеты на различные физические интерфейсы, принимают **решения (decisions)** на основе IP-адреса, включенного в пакет и затем физически переслают пакет на другой сетевой интерфейс.

**Основы IP адресации**

Сетевой уровень TCP/IP, используя протокол IP, **обеспечивает (provides)** услугу пересылки IP-пакетов от одного устройства к другому. **Любое устройство (Any device)** с IP-адресом может подключиться к сети TCP/IP и отправить пакеты. В этом разделе показан простой пример маршрутизации IP для перспективы.

Термин IP хост относится к любому устройству, независимо от размера или мощности, которое имеет IP-адрес и подключено к любой TCP / IP сети.

Рисунок 1-10 повторяет знакомый случай, в котором веб-сервер Ларри хочет отправить часть веб-страницы Бобу, но теперь с подробной информацией, относящейся к IP. В нижнем левом углу обратите внимание, что сервер Ларри готов к отправке знакомых данных приложения, **HTTP-заголовок** и **TCP заголовок**. **Кроме того (In addition),** сообщение теперь содержит заголовок IP. Заголовок IP включает в себя IP-адрес источника - IP-адрес Ларри (1.1.1.1) и IP-адрес назначения - IP-адрес Боба (2.2.2.2).



Шаг 1, c левой стороны рис. 1-10, начинается с Ларри, **находящегося (being)** в готовности к отправке IP-пакета. IP процесс Ларри выбирает отправку пакета на какой-то маршрутизатор - **некоторый (nearby)** маршрутизатор в одной и той же локальной сети - **в ожидании (expectation),** что маршрутизатор будет знать, как переслать пакет. (Эта логика очень похожа на вас или меня посылающего все наши письма, помещая их в **некоторый (nearby)** почтовый ящик.) Ларри не нужно знать ничего больше о топологии или других маршрутизаторах.

На шаге 2, маршрутизатор **R1** получает IP-пакет, и IP процесс **R1** принимает решение. R1 смотрит на адрес назначения (2.2.2.2), **сравнивает (compares)** этот адрес с известными ему IP - маршрутами, и выбирает переслать пакет на маршрутизатор **R2**. Этот **процесс пересылки IP пакетов** называется **маршрутизация IP** (или просто маршрутизация).

На шаге 3 маршрутизатор **R2** повторяет такую же логику, использующуюся в маршрутизаторе **R1**. **R2** в IP процесс будет сравнивать IP-адрес назначения пакета (2.2.2.2) **R2** с известным IP-маршрутов и сделать выбор для пересылки пакетов на право на Боба.

На шаге 3, маршрутизатор **R2** повторяет тот же тип логики, используемой маршрутизатором **R1**. IP процесс **R2** сравнит IP-адрес назначения пакетов (2.2.2.2) с известными IP-маршрутами **R2** и сделает выбор - переслать пакет направо, к Бобу.

Вы будете изучать IP **более углубленно (more depth),** чем любой другой протокол при подготовке к CCENT и CCNA. Практически в половине глав этой книги обсуждаются некоторые **особенности (feature)**, которые относятся к адресации, IP-маршрутизации и к тому, как маршрутизаторы **выполняют (perform)** маршрутизацию.

# Link уровень TCP/IP (Канальный уровень + Физический уровень)

Канальный уровень оригинальной модели TCP/IP определяет протоколы и **аппаратные средства (hardware required)**, необходимые для передачи данных через некоторую физическую сеть. Термин **Link (Связь, канал)** относится к физическим соединениям, или **каналам (связям)**, между двумя устройствами и протоколами, используемыми для управления этими **каналами (связями)**.

Так же, как каждый уровень в любой сетевой модели, **Канальный уровень TCP/IP** предоставляет услуги уровню над ним в сетевой модели. Когда IP процесс хоста или маршрутизатора выбирает отправить IP-пакет на другой маршрутизатор или узел, этот узел или маршрутизатор затем использует сведения **Канального уровня** для отправки этого пакета на следующий узел/маршрутизатор.

Поскольку каждый уровень предоставляет услугу уровню над ним, **найдите минутку (take a moment)**, чтобы подумать об алгоритмах IP, относящихся к рис. 1-10. В этом примере IP-алгоритм хоста Ларри выбирает отправку IP-пакета некоторому маршрутизатору (**R1**), при этом не указывается основной Ethernet. Ethernet-сеть, которая реализует протоколы **Канального уровня**, должна использоваться для доставки этого пакета от хоста Ларри к маршрутизатору R1. Рисунок 1-11 показывает четыре шага, которые должны быть сделаны **(occurs)** на **Канальном уровне** чтобы позволить Ларри отправить IP-пакет R1.

Рисунок 1-11 показывает Ethernet как ряд линий. Сетевые диаграммы часто используют это **условное обозначение (convention)** при изображении локальных сетей Ethernet, в тех случаях, когда фактические кабели локальной сети и сетевые устройства не важны для некоторого обсуждения, как в данном случае. Локальная сеть будет иметь кабели и устройства, такие как коммутаторы локальных сетей, которые не показаны на рисунке.

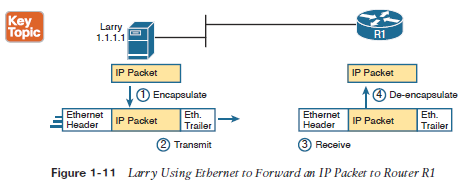


Рисунок 1-11 показывает четыре шага. Первые два происходят на Ларри, и два последних происходят на маршрутизаторе R1, следующим образом:

Шаг 1. Ларри инкапсулирует (упаковывает) пакет IP между заголовком Ethernet и Ethernet трейлером, создавая Ethernet кадр (frame).

Шаг 2. Ларри физически передает **биты** этого Ethernet кадра, используя электричество поступающее **(flowing)** через кабель Ethernet.

Шаг 3. Маршрутизатор R1 физически принимает электрический сигнал по кабелю, и повторно создает те же **биты** путем интерпретации значения электрических сигналов.

Шаг 4. Маршрутизатор R1 деинкапсулирует (распаковывает) IP-пакет из Ethernet кадра путем удаления и **отбрасывания (discarding)** Ethernet заголовка и трейлера.

К концу этого процесса, процессы **Канального уровня** Ларри и R1 работали вместе, чтобы доставить пакет от Ларри к роутеру R1.

Протоколы определяют и то и другое заголовки и трейлеры по одной общей причине, но заголовки существуют в начале сообщения, а трейлеры существуют в конце.

Канальный уровень включает большое количество протоколов и стандартов. Например, канальный уровень включает в себя все варианты протоколов Ethernet, **а также (along with)** несколько других стандартов локальной сети, которые были более популярны в **последние десятилетия (decades past)**. Канальный уровень включает стандарты глобальной сети (WAN) для разных физических носителей, которые **значительно (significantly)** **отличаются (differ)** в сравнении со стандартами LAN из-за больших расстояний, **связанных** (**involved**) с передачей данных. Этот уровень также включает популярные стандарты WAN, которые добавляют заголовки и трейлеры, как показано в целом на рис. 1-11 - протоколы, такие как протокол точка-точка (PPP) и Frame Relay. В главе 2 «Основы локальных сетей Ethernet» и в главе 3 «Основы глобальных сетей» далее развиваются эти темы для локальных сетей и глобальных сетей соответственно.

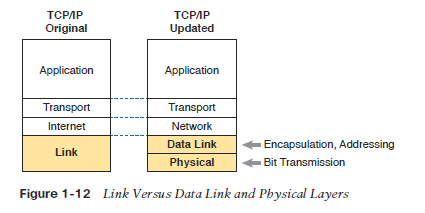
Короче говоря, канальный уровень TCP / IP включает в себя две **различные** **(distinct)** функции: функции, связанные с физической передачей данных, а также протоколы и правила, которые контролируют использование физических носителей. Пятислойная модель TCP / IP просто **разбивает (splits)** канальный уровень на два уровня (канал передачи данных и физический), чтобы соответствовать этой логике.

# TCP/IP модель и теминология

Прежде чем завершить это введение в модель TCP / IP, в этом разделе рассматриваются несколько **оставшихся (remaining)** деталей модели и некоторых **связанных (related)** терминов.

**Сравнение оригинальной и современной моделей TCP/IP.**

Исходная модель TCP / IP определяла один уровень – канальный уровень (**Link Layer**) – **ниже (below)** уровня Интернета (**Internet Layer**). Функции, определенные на исходном канальном уровне, можно разбить на две **основные (major)** категории: функции, связанные **непосредственно (directly)** с физической передачей данных, и только **косвенно (indirectly)** связанные с физической передачей данных. Например, в четырех шагах, показанных на рисунке 1-11, этапы 2 и 3 были специфическими для отправки данных, но этапы 1 и 4 - инкапсуляция и деинкапсуляция - были только косвенно связаны. Это разделение станет более ясным, когда вы прочтете дополнительные сведения о каждом протоколе и стандарте.

Сегодня большинство документов используют более современную версию модели TCP / IP, как показано на рисунке 1-12. Сравнивая эти два, верхние уровни идентичны, за исключением замены имени **Internet Layer** на **Network Layer**. Нижние уровни отличаются тем, что один канальный уровень в исходной модели разделен на два уровня, чтобы соответствовать разделению деталей физической передачи от других функций. На рисунке 1- 12 снова показаны две версии модели TCP / IP**, с акцентом (emphasis)** на эти **различия (distinctions).** 

**Терминология инкапсуляции данных**

Как вы можете видеть из **объяснений (explanations)** того, как выполняют свои задачи HTTP, TCP, IP и Ethernet, каждый уровень добавляет свой собственный заголовок (и для **Data-Link** протоколов также трейлер) к данным, предоставляемым более высоким уровнем. Термин инкапсуляция относится к процессу помещения заголовков (и иногда трейлеров) вокруг некоторых данных.

Многие примеры в этой главе показывают процесс инкапсуляции. Например, веб-сервер Larry инкапсулировал содержимое домашней страницы в HTTP-заголовок на рис. 1-6. Уровень TCP инкапсулировал заголовки и данные HTTP внутри заголовка TCP на рис. 1-7. IP инкапсулировал заголовки TCP и данные внутри заголовка IP на рисунке 1-10. Наконец, Ethernet канального уровня инкапсулировал IP-пакеты внутри заголовка и трейлера на рисунке 1-11.

Процесс, **посредством которого (by which)** хост TCP / IP отправляет данные, можно рассматривать как пятиэтапный процесс. Первые четыре шага относятся к инкапсуляции, **выполняемой (performed)** четырьмя уровнями TCP / IP, а последний шаг - **фактическая (actual)** физическая передача данных хостом. Фактически, если вы используете пятислойную модель TCP / IP, один шаг **соответствует (corresponds)** роли каждого уровня. Этапы суммируются в следующем списке:

**Шаг 1.** Создайте и инкапсулируйте данные приложения с любыми **требуемыми (required)** заголовками **Прикладного уровня**. Например, сообщение HTTP ОК может быть возвращено в HTTP-заголовке, за которым следует часть содержимого веб-страницы.

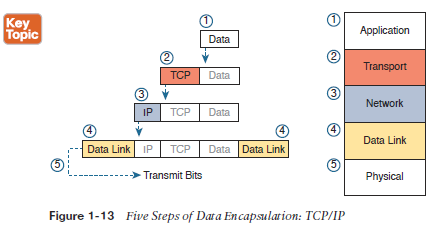
**Шаг 2**. Инкапсулируйте данные, **предоставляемые (supplied)** **Прикладным уровнем**, в заголовок **Транспортного уровня**. Для приложений конечного пользователя обычно используется заголовок TCP или UDP.

**Шаг 3.** Инкапсулируйте данные, предоставляемые **Транспортным уровнем**, в заголовок **Сетевого уровня** (IP). IP определяет IP-адреса, которые **однозначно (uniquely)** идентифицируют каждый компьютер.

**Шаг 4.** Инкапсулируйте данные, **предоставляемые (supplied)** **Сетевым уровнем**, в заголовок и трейлер **Канального уровня**. Этот уровень использует как заголовок, так и трейлер.

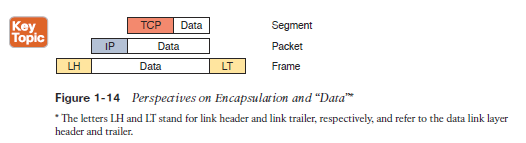
**Шаг 5.** Передайте биты. **Физический уровень** кодирует сигнал **на** **(onto)** среду для передачи кадра.

Цифры на рисунке 1-13 **соответствуют (correspond)** пяти шагам в этом списке, графически отображающим одни и те же понятия. Обратите внимание: поскольку на **Прикладном уровне** часто не требуется добавлять заголовок, на рисунке не отображается заголовок определенного Прикладного уровня.



**Имена TCP/IP сообщений**

Наконец, **обратите особое внимание (take particular care)** на запоминание терминов **сегмент**, **пакет** и **фрейм** и значение каждого из них. Каждый термин относится к заголовкам (и, возможно, к трейлерам), определенным конкретным уровнем, и к данным, инкапсулированным после этого заголовка. Однако каждый термин относится к другому уровню: **сегмент** для **Транспортного уровня**, **пакет** для **Сетевого уровня** и **фрейму** для **Канального уровня**. На Рисунке 1-14 показан каждый уровень вместе со связанным термином.



На Рисунке 1-14 также показаны инкапсулированные данные как просто «данные». При фокусировании на работе, выполняемой определенным уровнем, инкапсулированные данные обычно не имеют значения. Например, IP-пакет может действительно иметь заголовок TCP после заголовка IP, заголовок HTTP после заголовка TCP и данные для веб-страницы после HTTP-заголовка. Однако, обсуждая IP, вы, вероятно, просто заботитесь о IP-заголовке, поэтому все после IP-заголовка просто называют данными. Таким образом, при рисовании IP-пакетов все после заголовка IP обычно отображается просто как данные.

# Сетевая модель OSI

В какой-то момент в истории модели OSI многие люди думали, что OSI выиграет битву с обсуждавшимися ранее сетевыми моделями. Если бы это **произошло (occurred),** вместо того, чтобы запускать TCP / IP на каждом компьютере в мире, эти компьютеры работали бы с OSI.

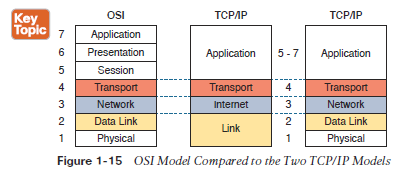
Однако OSI не выиграл эту битву. На самом деле, OSI **больше не существует (no longer exists)** в качестве сетевой модели, которая могла бы использоваться вместо TCP / IP, хотя некоторые из исходных протоколов, на которые ссылается модель OSI, все еще существуют.

Итак, почему OSI даже в этой книге? Терминология. В те годы, когда многие люди думали, что модель OSI станет **обычным явлением (commonplace)** в мире сетей (в основном в конце 1980-х и начале 1990-х годов), многие поставщики и протокольные документы начали использовать терминологию из модели OSI. Эта терминология **сохраняется (remains)** и сегодня. Итак, хотя вам никогда не понадобится работать с компьютером, который использует OSI, чтобы понять современную сетевую терминологию, вам нужно понять кое-что об OSI.

**Сравнение OSI и TCP/IP**

Модель OSI имеет **много общего** **(many similarities)** с моделью TCP/IP с базовой концептуальной точки зрения. Она имеет (семь) уровней, и каждый уровень определяет набор типичных сетевых функций. Как и в случае с TCP/IP, уровни OSI относятся к нескольким протоколам и стандартам, которые реализуют функции, определенные каждым уровнем. В других случаях, как и для TCP/IP, комитеты OSI не создавали новые протоколы или стандарты, а ссылались на другие протоколы, которые уже были определены. Например, IEEE определяет стандарты Ethernet, поэтому комитеты OSI **не тратили время (not waste time),** **указывая (specifying)** новый тип Ethernet; Она просто ссылался на стандарты IEEE Ethernet.

Сегодня модель OSI может использоваться как стандарт для сравнения с другими сетевыми моделями. На рис. 1-15 сравнивается модель с 7-ю уровнями OSI с четырьмя и пятью уровнями моделей TCP/IP.



Далее в этом разделе рассматриваются две причины, по которым мы по-прежнему используем сегодня терминологию OSI: описать другие протоколы и описать процесс инкапсуляции. **Попутно** **(Along the way)** текст кратко рассматривает каждый уровень модели OSI.

**Описание протоколов путем ссылки на уровни OSI**

**Даже сегодня (Even today)** сетевые документы часто описывают протоколы и стандарты TCP / IP, ссылаясь на уровни OSI, как по номеру уровня, так и по имени уровня. Например, общим описанием коммутатора локальной сети является «Коммутатор уровня 2», а «уровень 2» - это 2уровень OSI. Поскольку OSI имеет **четко определенный (well-defined)** набор функций, **связанный** **(associated)** с каждым из семи ее уровней, и, если вы знаете эти функции, вы можете понять, что люди имеют в виду, когда они ссылаются на продукт или функцию, относящуюся к уровню OSI.

В качестве другого примера, **Internet Layer** оригинальной TCP/IP, **реализованный** **(implemented)** в основном при помощи IP, **приравнивается (equates to)** **грубо говоря (most directly) к** **DataLink Layer** OSI. Поэтому большинство людей говорят, что IP - это протокол **сетевого уровня** или протокол **уровня 3**, используя терминологию и номерацию уровней OSI. Конечно, если вы пронумеровали **модель TCP/IP**, начиная снизу, IP будет **либо (either)** уровнем 2, либо 3, **в зависимости (depending)** от того, какую версию модели TCP/IP вы **желаете (care to)** использовать. Тем не менее, даже если IP является протоколом TCP/IP, все используют имена и номера уровней модели OSI, когда описывают IP или **любой другой (any other)** протокол **фактически (for that matter).**

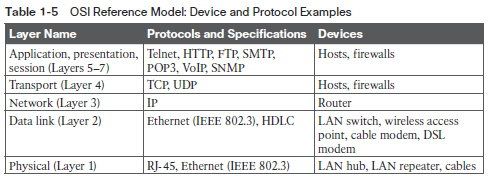
**Утверждение (The claim)**, что конкретный уровень TCP/IP аналогичен конкретному уровню OSI, является общим сравнением, но не детальным сравнением. Сравнение немного похоже на сравнение автомобиля с грузовиком: оба могут перевести вас из пункта А в пункт Б, но у них есть много особых отличий, например, грузовик с грузовым кузовом, в котором перевозится груз. Аналогично, оба сетевых уровня OSI и TCP/IP определяют логическую адресацию и маршрутизацию. Однако адреса имеют разные размеры, и логика маршрутизации работает **даже (even)** по-разному. Таким образом, сравнение уровней OSI с другими моделями протоколов - это общее сравнение основных целей, а не сравнение конкретных методов.

# Уровни OSI и их функции

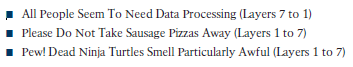
Сегодня, **поскольку (because)** большинство людей **оказывается** **(happen to be)** гораздо больше знакомы с функциями TCP/IP, чем с функциями OSI, один из лучших способов узнать о функциях разных уровней OSI - это подумать о функциях в модели TCP / IP и **найти связь (correlate)** с теми же у модели OSI. В целях изучения вы можете подумать о пяти уровнях OSI, точно также как вы это делали для 5 уровней TCP/IP. Например, прикладной уровень каждой модели определяет протоколы, которые будут использоваться **непосредственно (directly)** приложениями, а физический уровень каждого определяет электромеханические детали связи по физическим соединениям. В таблице 1-4 кратко описывается каждый уровень OSI.

|  |  |
| --- | --- |
| Уровень | Описание функций |
| 7 | **Application layer.** Предоставляет интерфейс от приложения к сети, поставляя протокол c действием, значимым для приложения, например, «получить объект веб-страницы». |
| 6 | **Presentation layer.** Этот уровень **согласовывает** **(negotiates)** форматы данных, такие как текст ASCII, или типы изображений, например, JPEG. |
| 5 | **Session layer.** Этот уровень предоставляет методы для группирования нескольких **двунаправленных** **(bidirectional)** сообщений в **рабочий поток (workflow)** для упрощения управления и облегчения **отката** **(backout)** работы, которая произошла, если весь рабочий процесс завершился неудачей. |
| 4 | **Transport layer.** По функциям, как и транспортный уровень TCP/IP. Этот уровень фокусируется на доставке данных между двумя конечными узлами (например, восстановление ошибок). |
| 3 | **Network layer.** Как и TCP/IP network (Internet) layer, этот уровень определяет логическую адресацию, маршрутизацию (forwarding) и протоколы маршрутизации, используемые для изучения маршрутов. |
| 2 | **Data link layer.** Как и TCP/IP data link layer, этот уровень определяет протоколы для доставки данных по определенному типу физической сети (например, the Ethernet data link protocols) |
| 1 | **Physical layer.** Этот уровень определяет физические характеристики среды передачи, включая разъемы, штырьки, использование контактов, электрические токи, кодирование, модуляцию света и т.д. |

В таблице 1-5 приведена выборка устройств и протоколов и их сопоставимых уровней OSI. Обратите внимание, что многие сетевые устройства должны понимать протоколы на нескольких уровнях OSI, поэтому уровень, указанный в таблице 1-5, фактически относится к самому высокому уровню, о котором обычно думает устройство при выполнении своей основной работы. Например, маршрутизаторы должны думать о концепциях Уровня 3, но они также должны поддерживать функции обоих уровней 1 и 2.



**Помимо (Besides)** запоминания основ функций каждого уровня OSI (как в таблице 1-4), а также примера протокола и устройства на каждом уровне (как в таблице 1-5), вы также должны запомнить имена уровней. Вы можете просто запомнить их, но некоторым людям нравится использовать мнемоническую фразу, чтобы облегчить запоминание. В следующих трех фразах первая буква каждого слова совпадает с первой буквой имени уровня OSI в порядке, указанном в круглых скобках:



# Концепции и преимущества уровней OSI

Хотя сетевые модели используют уровни, помогающие людям классифицировать и понимать множество функций в сети, сетевые модели используют уровни по многим причинам. Например, рассмотрим аналогию с другой почтовой службой. Лицу, пишущему письмо, не нужно думать о том, как почтовая служба будет доставлять письмо по всей стране. Почтовому работнику в середине страны не нужно беспокоиться о содержании письма. **Аналогично (Likewise)** сетевые модели, которые **делят** **(divide)** функции на разные уровни, **позволяют (enable)** одному программному пакету или аппаратно реализовывать функции с одного уровня и предполагать, что другое программное обеспечение будет выполнять функции, определенные другими уровнями.

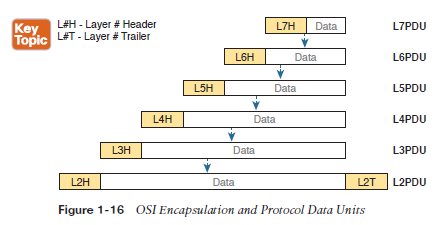
В следующем списке перечислены преимущества многоуровневых спецификаций протокола:

1. **Менее сложный:** По сравнению с отсутствием многоуровневой модели сетевые модели разбивают концепции на более мелкие части.
2. **Стандартные интерфейсы**: Стандартные **определения** **(definitions)** интерфейсов между каждым уровнем позволяют нескольким поставщикам создавать продукты, которые **выполняют** **(fill)** определенную роль, со всеми преимуществами открытой конкуренции.
3. **Легкость в изучении**: люди могут легче обсуждать и узнавать много деталей спецификации протокола.
4. **Более легкая разработка:** снижение сложности упрощает внесение изменений в программу и **ускоряет (faster)** разработку продукта.
5. **Многофункциональная (Multivendor) совместимость (interoperability):** Создание продуктов для **соответствия (to meet)** одним и тем же сетевым стандартам означает, что компьютеры и сетевое оборудование нескольких поставщиков могут работать в одной сети.
6. **Модульная инженерия:** один поставщик может написать программное обеспечение, которое реализует более высокие уровни - например, веб-браузер и другой поставщик может написать программное обеспечение, которое реализует более низкие уровни - например, встроенное программное обеспечение Microsoft TCP/IP в своих операционных системах.

# Терминология инкапсуляции OSI

Как и TCP / IP, каждый уровень OSI запрашивает сервисы со следующего нижнего уровня. Для предоставления услуг каждый уровень использует заголовок и, **возможно (possibly)**, трейлер. Нижний уровень инкапсулирует данные верхнего уровня за заголовком.

OSI использует более общий термин для обозначения сообщений, **а не (rather than)** кадра, пакета и сегмента. OSI использует термин **protocol data unit** (**PDU**). PDU представляет биты, которые включают заголовки и трейлеры для этого уровня, **а также (as well)** инкапсулированные данные. Например, IP-пакет, как показано на рисунке 1-14, с использованием терминологии OSI, является PDU, более конкретно PDU уровня 3 (сокращенно L3PDU), потому что IP является протоколом уровня 3. OSI просто ссылается на PDU Layer x (LxPDU), причем x ссылается на номер обсуждаемого уровня, как показано на рисунке 1-16.



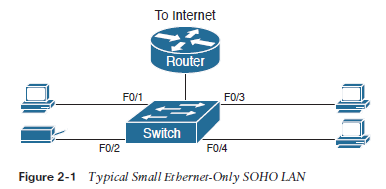
# **Обзор ЛВС**

Термин Ethernet относится к семейству стандартов LAN, которые вместе определяют **physical** и **data link** уровни самой популярной в мире технологии проводной ЛВС. Стандарты, определенные Институт инженеров по электротехнике и электронике (IEEE), определяет кабели, разъемы на концах кабелей, правила протокола и все остальное, что требуется для создания локальной сети Ethernet.

## **Typical SOHO LANs**

Для начала, во-первых, подумайте о **small office/home office (SOHO) LAN** сегодня**,** особенно о локальной сети, которая использует только технологию Ethernet LAN. Во-первых, LAN нуждается в устройстве, называемом коммутатором Ethernet LAN, который предоставляет множество физических портов, в которые могут быть подключены кабели. Ethernet использует Ethernet-кабели, что является общей **ссылкой (reference)** на любой кабель, который соответствует любому из нескольких стандартов Ethernet. ЛВС использует кабели Ethernet для подключения различных устройств или **узлов** (**nodes)** Ethernet к одному из портов Ethernet коммутатора.

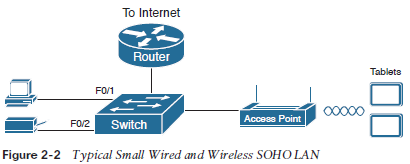
На рисунке 2-1 показано изображение локальной сети Ethernet SOHO. На рисунке показан один сетевой коммутатор, пять кабелей и пять других узлов Ethernet: три ПК, принтер и одно сетевое устройство, называемое маршрутизатором. (Маршрутизатор подключает ЛВС к глобальной сети, в данном случае к Интернету.)



Хотя на рисунке 2-1 показана простая Ethernet-сеть**, многие локальные сети SOHO Ethernet объединяют router и switch в одно устройство**. Продавцы продают интегрированные сетевые устройства потребительского класса, которые работают в качестве маршрутизатора и коммутатора Ethernet, а также выполняют другие функции**. Эти устройства обычно имеют надпись «роутер» на упаковке**, но во многих моделях также есть порты коммутатора Ethernet с четырьмя портами или восемью портами, встроенные в устройство.

Типичные SOHO LANs сегодня также поддерживают подключения к беспроводной локальной сети. Ethernet определяет только технологию проводной локальной сети; **Другими словами (in other words),** Ethernet-сети используют кабели. Однако вы можете создать одну локальную сеть, которая использует как технологию Ethernet LAN, так и технологию беспроводной локальной сети, которая также определяется IEEE. Беспроводные локальные сети, определенные стандартом IEEE, начиная с 802.11, используют радиоволны для отправки бит с одного узла на другой.

Большинство беспроводных локальных сетей **полагаются на (rely on)** **еще (yet)** одно сетевое устройство: **a wireless LAN access point (AP).** AP действует как Ethernet-коммутатор, поскольку все узлы беспроводной локальной сети обмениваются данными с коммутатором Ethernet путем отправки и приема данных с беспроводной точки доступа. Разумеется, в качестве беспроводного устройства AP не нуждается в портах Ethernet для кабелей, за исключением одного Ethernet-соединения для подключения AP к локальной сети Ethernet, как показано на рисунке 2-2.

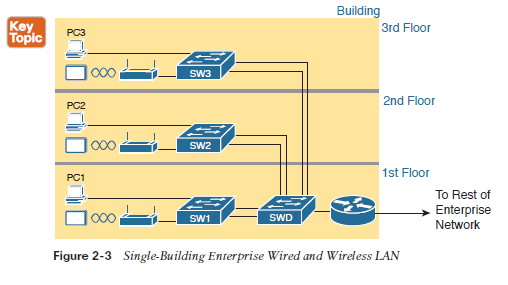


Обратите внимание, что на этом рисунке показан **router**, **Ethernet switch** и **wireless LAN access point (AP)** в виде трех отдельных устройств, чтобы вы могли лучше понимать разные роли. **Однако сегодня большинство сетей SOHO будут использовать одно устройство, которое часто обозначается как «беспроводной маршрутизатор», который выполняет все эти функции.**

# типичные локальные сети предприятия

Сети предприятий имеют **сходные потребности** **(similar needs)** по сравнению с сетью SOHO, но в гораздо большем **масштабе (scale)**. Например, корпоративные Ethernet-сети начинаются с LAN switches, установленных в шкафу для проводки за запертой дверью на каждом этаже здания. Электрики устанавливают кабели Ethernet из этого проводного шкафа в **небольшие комнаты (cubicles)** и конференц-залы, где устройствам, возможно, потребуется подключиться к локальной сети. В то же время большинство предприятий также поддерживают беспроводные локальные сети в одном и том же пространстве, что позволяет людям **передвигаться (roam)** и **вместе с тем (still)** работать, а также поддерживать **растущее (growing)** число устройств, не имеющих интерфейса Ethernet LAN.

На рисунке 2-3 показан концептуальный вид типичной локальной сети предприятия в **трехэтажном (three-story)** здании. На каждом этаже есть **Ethernet LAN switch** и **wireless LAN AP**. Чтобы обеспечить связь между этажами**, switch на этаже (per-floor switch)** соединяется с одним **централизованным распределительным коммутатором** **(centralized distribution switch).** Например, PC3 может отправлять данные на PC2, но сначала он будет проходить через switch SW3 на первый этаж к **распределительному коммутатору (SWD)** (**distribution switch**), а затем обратно через switch SW2 на втором этаже.



На рисунке также показан типичный способ подключения локальной сети к глобальной сети с использованием router. **LAN switch** и **wireless access points (AP)** работают для создания самой ЛВС. **Routers** подключаются как к локальной сети, так и к глобальной сети. Для подключения к локальной сети **Router** просто использует интерфейс Ethernet LAN и кабель Ethernet, как показано в правом нижнем углу на рисунке 2-3. В остальной части этой главы **основное внимание уделяется (focuses)** Ethernet в **частности (particular)**.

# Разнообразие стандартов физического уровня Ethernet

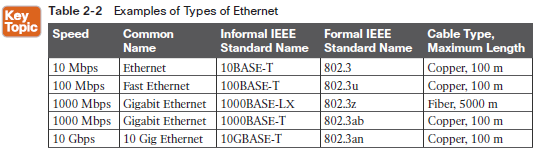
Термин «Ethernet» относится **ко всему (entire)** семейству стандартов. Некоторые стандарты определяют специфику отправки данных по конкретному типу кабелей и с определенной скоростью. Другие стандарты определяют протоколы или правила, которым узлы Ethernet должны следовать как часть локальной сети Ethernet. Все эти стандарты Ethernet исходят от IEEE и **включают число 802.3 в качестве начальной части стандартного имени.**

Ethernet поддерживает большое количество различных опций для физических Ethernet-соединений, учитывая его долгую историю за последние 40 **или около того лет** **(so years).** Сегодня Ethernet включает в себя множество стандартов для различных видов оптических и медных кабелей и для скоростей от 10 мегабит в секунду (Мбит / с) до 100 гигабит в секунду (Gbps). Стандарты также различаются по типу кабелей и разрешенной длине кабелей.

Самый фундаментальный выбор кабелей **имеет отношение к** (**has to do with**) материалам, используемым **внутри (inside)** кабеля для физической передачи бит: либо медных, либо стеклянных волокон. Использование кабелей неэкранированной витой пары (UTP) экономит деньги по сравнению с оптическими волокнами, а узлы Ethernet используют **провода** **(wires)** внутри кабеля для передачи данных по электрическим схемам. Волоконно-оптические кабели, более дорогая альтернатива, позволяют Ethernet-узлам посылать свет по стеклянным волокнам в центре кабеля. Хотя более дорогие оптические кабели обычно обеспечивают более длинные кабельные расстояния между узлами.

Чтобы быть готовым к выбору продуктов для приобретения для новой локальной сети Ethernet, сетевой инженер должен знать имена и функции различных стандартов Ethernet, поддерживаемых в продуктах Ethernet. IEEE определяет стандарты физического уровня Ethernet, используя пару соглашений об именах. Формальное имя начинается с 802.3, за которым следуют некоторые суффиксы. IEEE также использует более **значимые** **(meaningful)** сокращения имен, которые определяют скорость, а также подсказку о том, является ли кабельная система UTP (с суффиксом, который включает T) или волокном (с суффиксом, который включает в себя X).

В таблице 2-2 перечислены несколько стандартов Ethernet physical layer. Во-первых, в таблице перечислено достаточно имен, чтобы вы поняли **соглашения (conventions)** об именах IEEE. В нем также перечислены четыре наиболее распространенных стандарта, использующих кабели UTP, поскольку обсуждение этой книги Ethernet в основном посвящено опциям UTP.



**ПРИМЕЧАНИЕ.**

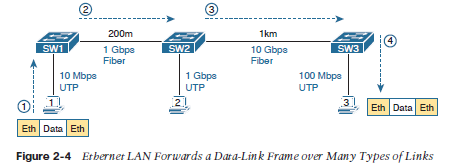
**Волоконно-оптические кабели содержат длинные тонкие нити из стекловолокна. Связанные Ethernet узлы посылают свет по стеклянному волокну в кабель, кодируя биты как изменения в свете.**

# Последовательный режим работы через все линки, использующие Ethernet Data Link уровня

Хотя Ethernet включает в себя множество стандартов **physical layer**, Ethernet действует как одна технология LAN, поскольку использует один и тот же **data link layer standard** по всем типам физических линий Ethernet. Этот стандарт определяет общий Ethernet-заголовок и трейлер. (в качестве напоминания заголовок и трейлер представляют собой байты **служебных (overhead)** данных, которые Ethernet использует для выполнения своей работы по отправке данных по локальной сети.) Независимо от того, **проходят (flows)** ли данные по кабелю UTP или любому оптоволоконному кабелю, и независимо от скорости, **data-link header and trailer** используют тот же формат.

В то время как стандарты физического уровня сосредоточены на отправке бит по кабелю, **the Ethernet data-link protocols** сосредоточены на отправке кадра Ethernet от источника к Ethernet-узлу назначения. С точки зрения передачи данных **nodes** строят и пересылают **frames**. Как впервые определено в главе 1 «Введение в TCP/IP Networking», термин frame **конкретно (specifically)** относится к **header** и **trailer** **data-link protocol**, а также к данным, инкапсулированным внутри этого заголовка и трейлера. Различные узлы Ethernet просто пересылают через все необходимые **Link**-и для доставки **frame** в нужное место назначения.

На рисунке 2-4 показан пример процесса. В этом случае PC1 отправляет Ethernet frame на PC3. Frame перемещается через **UTP Link** к коммутатору Ethernet SW1, затем по волоконно-оптическим линиям к Ethernet-коммутаторам SW2 и SW3, и, наконец, через другой **UTP Link** к PC3. Обратите внимание, что в этом примере биты фактически перемещаются с четырьмя различными скоростями: 10 Mbps, 1 Gbps, 10 Gbps, and 100 Mbps, **соответственно** **(respectively).**



Итак, что такое Ethernet LAN? Это комбинация пользовательских устройств, сетевых коммутаторов и различных типов кабелей. Каждый Link может использовать различные типы кабелей на разных скоростях. Тем не менее, все они работают вместе для доставки Ethernet-фреймов с одного устройства в локальной сети на другое устройство.

В остальной части главы копнем эту концепцию **немного глубже** **(little deeper)**, вначале рассматриваются детали построения физической сети Ethernet, после чего следует некоторое обсуждение правил пересылки **Frames** через Ethernet LAN.

# Building Physical Ethernet Networks with UTP

Для этого второго из трех основных разделов этой главы я сосредоточен на individual physical links between any two Ethernet nodes. Прежде чем сеть Ethernet **в целом** **(a whole)** сможет отправлять Ethernet-frames между пользовательскими устройствами, каждый node должен быть готов и способен отправлять данные по individual physical link. В этом разделе рассматриваются некоторые аспекты того, как Ethernet передает данные по этим links.

В этом разделе рассматриваются три наиболее часто используемых стандарта Ethernet: 10BASE-T (Ethernet), 100BASE-T (Fast Ethernet или FE) и 1000BASE-T (Gigabit Ethernet, или GE). В частности, в этом разделе рассматриваются детали отправки данных в обоих направлениях по кабелю UTP. Затем рассмотрим конкретную проводку кабелей UTP, используемых для сетей со скоростью 10 Мбит/с, 100 Мбит/с и 1000 Мб/с.

## Передача данных с использованием витых пар

Хотя верно, что Ethernet передает данные по кабелям UTP, физические средства для передачи данных используют электричество, которое протекает по проводам внутри кабеля UTP. Чтобы лучше понять, как Ethernet отправляет данные с использованием электричества, разобьем идею на две части: как создать электрическую цепь, а затем, как заставить этот электрический сигнал обмениваться данными 1 и 0.

Во-первых, для создания одной электрической цепи Ethernet определяет, как использовать два провода внутри одной витой пары проводов, как показано на рисунке 2-5. На рисунке не показан кабель UTP между двумя узлами, но вместо этого показаны два отдельных провода, которые находятся внутри кабеля UTP. Электрическая схема требует полного цикла, поэтому два узла, используя схему на своих Ethernet-портах, соединяют провода в одной паре, чтобы завершить цикл, позволяя электричеству течь.

Аннотация

**Сетевая модель** – это **общий (generic) термин (term),** относящийся к любому набору протоколов и стандартов, собранных в комплексную группу, и если устройства в сети следуют этому набору, то это позволяет им связываться. Примером являются TCP/IP и OSI.

Mainly – в основном

Mostly – глвным образом, в основном