**Введение**

Основой этого пути развития является **использование общепринятых моделей, описывающих сетевые правила и функции.**

**Что такое коммуникации?**

Для успешного обмена данными эти **устройства должны «знать»**, как обмениваться информацией.

Люди обмениваются идеями с использованием различных способов связи. При этом вне зависимости от выбранного способа связи все они содержат три общих элемента. **Первый — это источник сообщения, или отправитель.** Отправителем может быть человек или электронное устройство, которому нужно отправить сообщение другому человеку или устройству**. Второй элемент — это адресат, или получатель сообщения**. Адресат получает и интерпретирует сообщение. **Третий элемент, называемый каналом**, состоит из сетевой среды, по которой происходит передача данных. Это путь, по которому сообщение передается от источника к получателю.

Коммуникации начинаются с сообщения, или информации, которую нужно передать от источника к получателю. Отправка этого сообщения, посредством индивидуального общения или по сети, регулируется правилами, которые называются протоколами. Эти протоколы соответствуют типу способа коммуникации. При ежедневном личном общении правила для обмена данными по одному средству связи, например, по телефону, не обязательно совпадают с протоколом с помощью другого средства связи, например, при отправке письма.

Рассмотрим, например, индивидуальное общение двух людей, как показано на рисунке 1. До начала общения они должны договориться о способе общения. Если общение будет происходить с помощью голоса, сначала они должны договориться о том, на каком языке следует общаться. Затем, когда у них есть сообщение друг для друга, они должны уметь выразить его таким образом, чтобы оно стало понятным. Например, если используется английский язык, но употребляется неправильная структура предложения, сообщение может быть истолковано неверно. **Каждая из этих задач описывает протоколы, которые нужно применить для успешной коммуникации**. Это относится также к компьютерной коммуникации, как показано на рисунке 2.

**Установление правил**

До начала общения друг с другом люди должны использовать установленные правила или соглашения, регулирующие разговор. Выбор протоколов зависит от характеристик способа коммуникации, включая характеристики источника, адресата и канала. **Чтобы сообщение было успешно доставлено и понято, эти правила, или протоколы, необходимо соблюдать**. После согласования способа коммуникации (лично, по телефону, посредством писем, фотографий) применяемые протоколы должны учитывать следующие требования:

* установленные отправитель и получатель;
* общепринятые язык и грамматика;
* скорость и время доставки;
* требования к подтверждению или уведомлению о получении сообщения.

Хотя существует множество протоколов, которые должны взаимодействовать, к наиболее распространенным компьютерным протоколам относятся:

* кодирование сообщения;
* форматирование и инкапсуляция сообщения;
* размер сообщения;
* временные параметры сообщения;
* параметры доставки сообщения.

Каждый из этих пунктов будет обсуждаться более подробно далее.

**Кодирование сообщения**

Один из первых этапов отправки сообщения — кодирование. Кодирование — это процесс преобразования информации в форму, приемлемую для последующей передачи. Декодирование — обратный процесс, в результате которого информация преобразуется в исходный вид.

Представьте себе человека, который, планируя совместный отпуск с другом, звонит ему, чтобы обсудить пункт назначения предстоящей поездки, как показано на рисунке 1. Чтобы передать сообщение, отправителю прежде всего придётся преобразовать, или кодировать, свои мысли и представления о пункте назначения в слова. При разговоре по телефону слова преобразуются в звуки и модуляции разговорного языка, которые передают информацию. Абонент на другом конце телефонной линии слушает описание, воспринимает и декодирует звуки, мысленно создавая представление о том, что описывает отправитель.

Кодирование используется также при обмене данными с помощью компьютера, как показано на рисунке 2. Кодирование данных при обмене между узлами должно соответствовать среде связи. Прежде всего узел-отправитель преобразует передаваемое по сети сообщение в биты. Каждый бит кодируется набором звуков, световых волн или электрических импульсов, в зависимости от типа сети. Узел назначения принимает и декодирует сигналы и интерпретирует сообщение.

**Форматирование и инкапсуляция сообщения**

При отправке сообщения от источника к получателю необходимо использовать определённый формат или структуру. Формат зависит от типа сообщения и канала доставки.

Одна из наиболее распространенных форм письменного общения между людьми — письмо. Общепринятый формат личных писем не меняется столетиями. Во многих странах личное письмо состоит из следующих элементов:

* идентификатор получателя;
* обращение или приветствие;
* содержимое сообщения;
* заключительная фраза;
* идентификатор отправителя.

Кроме того, большинство личных писем нужно не только составить в правильном формате, но и запечатать, или инкапсулировать, в конверт для доставки, как показано на рисунке 1. На конверте, в специально отведённом месте, указывается адрес отправителя и получателя. Если адрес назначения или формат неверен, письмо не будет доставлено. Процесс размещения одного формата сообщения (письмо) внутри другого (конверт) называется инкапсуляцией. Деинкапсуляция производится в тот момент, когда получатель достаёт письмо из конверта.

Чтобы письмо дошло до получателя и было понято, автор должен использовать общепринятый формат. Аналогичным образом, для доставки и обработки письма в компьютерной сети необходимо придерживаться определённых правил форматирования. Подобно тому, как вкладывается в конверт письмо, инкапсулируются компьютерные сообщения. Для инкапсуляции каждого сообщения компьютера перед отправкой по сети используется особый формат, который называется кадром. Кадр действует примерно так же, как и конверт: в нём указаны адреса исходного узла и назначения, как показано на рисунке 2.

Формат и содержимое кадра зависят от типа сообщения и канала передачи. Узел назначения не может успешно получить и обработать неверно отформатированные сообщения.

**Размер сообщения**

Ещё одним правилом коммуникации является размер. В процессе разговора люди делят свои высказывания на фразы или предложения. Эти предложения ограничены в размерах так, что принимающее лицо может воспринять их за один раз, как показано на рисунке 1. В некоторых случаях разговор можно поделить на много более мелких предложений так, чтобы собеседник воспринял и понял каждую часть высказывания. Представьте, как бы воспринимался этот курс, если бы он выглядел как одно длинное предложение. В таком случае прочесть и понять его было бы довольно сложно.

Аналогичным образом, при передаче длинного сообщения от одного узла к другому по сети необходимо поделить его на части, как показано на рисунке 2. Размеры этих частей, или кадров, строго регулируются. Кроме всего прочего, они зависят от используемого канала. Слишком длинные или короткие кадры не доставляются.

Ограничения по размеру кадров заставляют исходный узел делить длинные сообщения на части, соответствующие требованиям к минимальному и максимальному размеру. Этот метод называется сегментацией. Каждая часть инкапсулируется с информацией об адресе в отдельный кадр и затем передаётся по сети. Узел-адресат распаковывает сообщения и собирает их вместе для обработки и интерпретации.

**Временные параметры сообщения**

Другим фактором, который влияет на качество приёма и понимания сообщения, является время. Расчёт времени позволяет людям определить, когда начать разговор, насколько быстро или медленно говорить и сколько времени ждать ответа. Это правила поддержания контакта.

**Метод доступа**

Метод доступа определяет, когда конкретный человек сможет отправить сообщение. Выбор времени зависит от среды. Например, в некоторых случаях говорить можно в любой момент, когда есть что сказать. В такой среде нужно дождаться, пока остальные прекратят разговор, а затем начать говорить. Если два человека начинают говорить одновременно, происходит информационная коллизия, и обоим приходится начинать сначала, как показано на рисунке 1. Компьютерам тоже необходимо выбирать метод доступа. Чтобы узнать, когда начать отправку сообщений и как реагировать на ошибки, узлам в сети нужно определить метод доступа.

**Управление потоком**

Временные параметры влияют также на количество отправляемой информации и скорость доставки. Если один человек говорит слишком быстро, другому сложно расслышать и понять сообщение, как показано на рисунке 2. Адресату придётся попросить его говорить медленнее. При отправке данных по сети узел-отправитель может передавать сообщения быстрее, чем узел назначения их принимает и обрабатывает. Управление потоком позволяет узлу-источнику и узлу назначения согласовать время для успешной связи.

**Тайм-аут ответа**

Если человек задает вопрос и не получает ответа за приемлемое время, он предполагает, что ответа не будет, и реагирует соответствующим образом, как показано на рисунке 3. Он может повторить вопрос или продолжить разговор. У сетевых узлов также существуют правила, определяющие время ожидания ответа и действие, выполняемое по истечении этого времени.

**Параметры доставки сообщения**

Оптимальный способ доставки сообщения может различаться, как показано на рисунке 1. В одних случаях информацию нужно передать только одному человеку. И наоборот, иногда информацию нужно одновременно передать группе людей или всем жителям определённого района. Разговор между двумя людьми — пример общения один на один. Если одно и то же сообщение нужно передать группе получателей, это доставка «один ко многим» или «один ко всем».

Кроме того, в некоторых случаях отправителю нужно убедиться, что сообщение доставлено успешно. Для этого получатель должен отправить подтверждение доставки. Если подтверждения не требуется, метод доставки сообщения называется неподтверждённым.

Узлы в сети используют аналогичные варианты доставки сообщений, как показано на рисунке 2.

Вариант доставки «один к одному» называется одноадресным (unicast). Это означает, что у сообщения есть только один адресат.

Если узел рассылает сообщения методом «один ко многим», это называется многоадресной или групповой рассылкой (multicast). Многоадресная рассылка предусматривает одновременную отправку одного и того же сообщения группе узлов.

Если всем сетевым узлам необходимо получить сообщение в одно и то же время, используется широковещательная рассылка (broadcast). Широковещательная рассылка представляет собой метод доставки сообщений «один ко всем». Кроме того, для узлов предусмотрены правила рассылки сообщений с подтверждением и без подтверждения.

**Протоколы: правила, регламентирующие способы обмена данными**

Как и в общении между людьми, различные сети и компьютерные протоколы должны иметь возможность взаимодействовать и совместно работать в сети, чтобы обеспечивать успешную связь. Группа взаимосвязанных протоколов, необходимых для выполнения функции связи, называется набором протоколов. Наборы протоколов реализуются узлами и сетевыми устройствами в программном обеспечении, аппаратном обеспечении, или в том и другом.

Один из лучших способов представить, как протоколы взаимодействуют в наборе, — рассмотреть взаимодействие в виде стека. Стек протоколов показывает, как в наборе реализованы отдельные протоколы. Протоколы рассматриваются с точки зрения уровней, причём каждый более высокий уровень обслуживания зависит от функций, определённых протоколами, которые показаны на более нижних уровнях. Нижние уровни стека относятся к перемещению данных по сети и предоставлению услуг верхним уровням, которые отвечают за наполнение пересылаемых сообщений. Судя по рисунку, мы можем использовать уровни для описания примера индивидуального общения. На нижнем, физическом уровне, есть два человека, каждый из которых может произнести слова вслух, используя голос. На втором уровне, уровне правил, достигнута договорённость общаться на разговорном языке. На верхнем уровне, уровне содержания, приведены фактически произносимые слова. В этом состоит содержание сообщения.

Если бы мы оказались свидетелями этого разговора, то фактически мы бы не увидели этих разбросанных в пространстве уровней. Использование уровней — это модель, которая даёт возможность эффективно разбить сложную задачу на части и описать принципы их работы.

**Сетевые протоколы**

На уровне общения между людьми некоторые коммуникационные правила формализованы, другие просто понятны на основе обычаев и практики. Для успешного обмена данными между устройствами сетевой набор протоколов должен описывать точные требования и взаимодействия. Сетевые протоколы определяют общий формат и набор правил для обмена сообщениями между устройствами. К некоторым распространённым сетевым протоколам относятся: IP, HTTP и DHCP.

На рисунках ниже показаны сетевые протоколы, описывающие следующие процессы:

* способ форматирования и структурирования сообщения, как показано на рисунке 1;
* процесс, в котором сетевые устройства обмениваются информацией о каналах передачи сообщения с другими сетями, как показано на рисунке 2;
* способ и время передачи сообщений об ошибках, а также системных сообщений между устройствами, как показано на рисунке 3;
* установка и завершение сеансов передачи данных, как показано на рисунке 4.

Например, IP определяет, каким образом пакет данных передаётся в пределах сети или в удалённую сеть. По протоколу IPv4 информация передаётся в определённом формате так, чтобы получатель мог правильно её интерпретировать. В этом не существует большого отличия от протокола, используемого при написании адреса на конверте при отправке письма. Информация должна соответствовать определённому формату, или письмо не сможет быть доставлено в почтовое отделение по месту назначения.

**Взаимодействие протоколов**

Примером использования набора протоколов в сетевой связи является взаимодействие между веб-сервером и веб-клиентом. Это взаимодействие использует ряд протоколов и стандартов в процессе обмена информацией между ними. Различные протоколы взаимодействуют друг с другом, чтобы гарантировать, что сообщения будут приняты и понятны обеими сторонами. Примерами этих протоколов являются следующие.

* **Протокол прикладного уровня** — протокол передачи гипертекста (HTTP): определяет, каким образом взаимодействуют веб-сервер и веб-клиент. HTTP определяет содержание и формат запросов и ответов, которыми обмениваются клиент и сервер. Программное обеспечение и веб-клиента, и веб-сервера реализует HTTP как часть приложения. Для управления процессом передачи сообщений между клиентом и сервером HTTP обращается к другим протоколам.
* **Транспортный протокол** — протокол управления передачей (TCP): управляет отдельными сеансами связи между серверами и клиентами в Интернете. TCP делит сообщения HTTP на более мелкие части, называемые сегментами. Эти сегменты передаются между веб-сервером и клиентскими процессами, запущенными на узле назначения. TCP также отвечает за управление размером и скоростью, с которой происходит обмен сообщениями между сервером и клиентом.
* **Интернет-протокол** — протокол IP: отвечает за приём форматированных сегментов от TCP, инкапсуляцию их в пакеты, присвоение им соответствующих адресов и их доставку по наилучшему пути к узлу назначения.
* **Протоколы сетевого доступа**: описывают две основные функции — связь по каналу передачи данных и физическую передачу данных по сетевой среде. Протоколы управления каналами передачи данных принимают пакеты от протокола IP и форматируют их для передачи в среде. Стандарты и протоколы физической передачи данных управляют тем, как сигналы посылаются и как они интерпретируются клиентами при получении. Одним из примеров протокола сетевого доступа является Ethernet.

**Наборы протоколов и отраслевые стандарты**

Как упоминалось ранее, набор протоколов представляет собой множество протоколов, которые используются вместе, чтобы обеспечить комплексный набор сетевых сервисов. Пакет протоколов может быть определён организацией по стандартизации или разработан поставщиком.

Протоколы IP, HTTP и DHCP являются частью набора протоколов Интернет, который называется протоколом управления передачей/протоколом IP (TCP/IP). Семейство протоколов TCP/IP является открытым стандартом, то есть эти протоколы находятся в свободном доступе для пользователей, и любой поставщик может внедрять эти протоколы на аппаратных средствах или в программном обеспечении.

Протокол, основанный на стандартах, одобряется в отрасли сетевых технологий и утверждается организацией по стандартизации. Использование стандартов в разработке и реализации протоколов гарантирует, что продукты от различных производителей могут взаимодействовать успешно. Если протокол не строго соблюдается конкретным изготовителем, такое оборудование или программное обеспечение может быть не в состоянии успешно поддерживать коммуникацию с продуктами других производителей.

В передаче данных, например, если одна сторона в сеансе связи использует протокол для управления односторонним режимом связи, а другая сторона использует протокол двусторонней связи, по всей вероятности, никакого обмена данными не произойдёт.

Некоторые протоколы являются проприетарными. «Частный» в данном контексте означает, что одна компания или поставщик контролирует определение протокола, а также то, как он функционирует. Некоторые частные протоколы могут использоваться различными организациями с разрешения владельца. Другие могут быть реализованы только на оборудовании, производимом поставщиком. Примерами частных протоколов являются AppleTalk и Novell Netware.

Некоторые компании могут работать вместе, чтобы создать собственный частный протокол. Нередко поставщик (или группа поставщиков) разрабатывает частный протокол для удовлетворения потребностей своих клиентов, а затем способствует принятию этого частного протокола в качестве открытого стандарта. Например, Ethernet был протоколом, первоначально разработанным Бобом Меткалфом (Bob Metcalfe) в исследовательском центре Palo Alto (PARC) компании Xerox в 70-х гг. прошлого столетия. Основав в 1979 году собственную компанию 3Com, Боб Меткалф сотрудничал с Digital Equipment Corporation (DEC), Intel и Xerox и продвигал стандарт «DIX» для Ethernet. В 1985 году Институт инженеров по электротехнике и электронике (IEEE) опубликовал стандарт IEEE 802.3, который был почти идентичен Ethernet. Сегодня 802.3 является общим стандартом, используемым в локальных сетях (LAN). Другой пример. Совсем недавно компания Cisco открыла протокол маршрутизации EIGRP, опубликовав RFC-документ, для удовлетворения потребностей клиентов, желающих использовать протокол в сети с компонентами разных производителей.

**Создание Интернета и разработка TCP/IP**

Семейство протоколов IP — это набор протоколов, необходимых для передачи и приёма информации с использованием Интернета. Этот протокол более известен как TCP/IP, потому что двумя первыми сетевыми протоколами, определёнными для этого стандарта, являлись TCP и IP. Основанный на открытых стандартах, TCP/IP заменил частные наборы протоколов отдельных поставщиков, такие как AppleTalk компании Apple и Internetwork Packet Exchange (IPX/SPX) компании Novell.

Первой сетью с коммутацией пакетов и предшественником современного Интернета была Advanced Research Projects Agency Network (ARPANET), которая появилась в 1969 году за счёт соединения базовых ЭВМ в четырёх местоположениях. ARPANET финансировался Министерством обороны США и использовался в университетах и научно-исследовательских лабораториях. Компания Bolt, Beranek and Newman (BBN) выступала подрядчиком, выполнив основную часть первоначальной разработки ARPANET, включая создание первого маршрутизатора, который назывался процессором интерфейсных сообщений (IMP).

В 1973 году Роберт Кан (Robert Kahn) и Винтон Серф (Vinton Cerf) начали работу над протоколом TCP для разработки следующего поколения ARPANET. TCP был разработан, чтобы заменить существовавшую в сети ARPANET программу сетевого управления (NCP). В 1978 году TCP был разделён на два протокола: TCP и IP. Позже к набору протоколов TCP/IP были добавлены другие протоколы, включая Telnet, FTP, DNS и многие другие.

Щёлкните по временной шкале на рисунке, чтобы увидеть подробную информацию о развитии других сетевых протоколов и приложений.

**Набор протоколов TCP/IP и процесс обмена данными**

В настоящее время этот набор включает в себя множество протоколов, как показано на рисунке 1. Нажмите на каждый протокол, чтобы просмотреть его описание. Протоколы сгруппированы в уровни с использованием протокольной модели TCP/IP. Протоколы TCP/IP входят в уровни, от межсетевого до прикладного, по модели TCP/IP. Протоколы более низких сетевых уровней (канального уровня или уровня доступа к сети) отвечают за доставку IP-пакета в физической среде передачи данных. Протоколы более низких сетевых уровней разработаны организациями по стандартизации, например IEEE.

Семейство протоколов TCP/IP реализовано в виде стека TCP/IP как на отправляющем, так и на принимающем узлах, и связано со сквозной доставкой данных приложений по сети. Протоколы 802.3 или Ethernet используются для передачи IP-пакета в физической среде передачи данных, используемой локальной сетью.

Рисунки 2 и 3 демонстрируют весь процесс обмена данными на примере веб-сервера, передающего данные клиенту.

Нажмите кнопку «Воспроизведение», чтобы просмотреть демонстрационные ролики.

1. Данные, которые требуется отправить — страница на гипертекстовом языке описания документов (HTML), имеющаяся на веб-сервере.

2. Заголовок протокола HTTP прикладного уровня добавляется в начало данных в формате HTML. Заголовок содержит различные данные, включая версию HTTP, которую использует сервер, а также код состояния, указывающий, что у него имеется информация для веб-клиента.

3. Протокол прикладного уровня HTTP передаёт форматированные данные веб-страницы на транспортный уровень. Протокол транспортного уровня TCP используется для управления отдельным сеансом связи между веб-сервером и веб-клиентом.

4. Затем информация IP добавляется перед сведениями TCP. IP назначает соответствующие IP-адреса источника и назначения. Такая информация называется IP-пакетом.

**5. Протокол Ethernet добавляет в начало и в конец IP-пакета информацию, называемую кадром канала передачи данных. Этот кадр доставляется на ближайший маршрутизатор на пути к клиенту. Маршрутизатор удаляет информацию Ethernet, анализирует IP-пакет, определяет наилучший путь для пакета, вставляет его в новый кадр и пересылает на следующий маршрутизатор по пути к месту назначения. Каждый маршрутизатор удаляет и добавляет новую информацию кадра канала передачи данных перед пересылкой пакета.**

6. Теперь эти данные передаются по объединённой сети, состоящей из среды и промежуточных устройств.

7. Клиент получает кадры канала передачи данных, в которых содержатся данные, и обрабатывает заголовки каждого протокола, а затем удаляет их в порядке, обратном тому, в котором они были добавлены. Информация Ethernet обрабатывается и удаляется, вслед за ней — информация протокола IP, затем TCP и, наконец, HTTP.

8. Затем данные веб-страницы передаются программному обеспечению браузера клиента.

**Открытые стандарты**

Открытые стандарты поощряют конкуренцию и инновации. Кроме того, они гарантируют, что продукт отдельной компании не сможет монополизировать рынок или получить несправедливое преимущество по сравнению с конкурентами. Хорошим примером этого является покупка беспроводного маршрутизатора для дома. Существует множество вариантов маршрутизаторов различных производителей, каждый из которых включает стандартные протоколы, такие как IPv4, DHCP, 802.3 (Ethernet) и 802.11 (беспроводная локальная сеть). Открытые стандарты также позволяют клиенту с операционной системой OS X от компании Apple загрузить веб-страницу с веб-сервера под управлением операционной системы Linux. Это связано с тем, что обе операционные системы используют протоколы открытых стандартов, например, из набора протоколов TCP/IP.

Организации по стандартизации играют важную роль в поддержании открытого Интернета со свободно доступной спецификацией и протоколами, которые могут быть реализованы любым поставщиком. Организация по стандартизации может разработать набор правил самостоятельно или в других случаях может выбрать частный протокол в качестве основы для стандарта. Если используется частный протокол, разработка стандарта, как правило, происходит с участием поставщика, который его создал.

Организации по стандартизации обычно являются нейтральными некоммерческими организациями, созданными для разработки и продвижения концепции открытых стандартов.

Организации по стандартизации:

* Общество Интернет (ISOC)
* Совет по архитектуре Интернета (IAB).
* Инженерная группа по развитию Интернета (IETF)
* Институт инженеров по электротехнике и электронике (IEEE)
* Международная организация по стандартизации (ISO)

Каждая из этих организаций будет обсуждаться более подробно на следующих страницах.

На рисунке нажмите каждый логотип, чтобы просмотреть информацию о стандартах.

**Преимущества использования многоуровневой модели**

Многоуровневая модель, например, модель TCP/IP, часто используется для описания взаимодействий между различными протоколами. Многоуровневая модель описывает взаимодействие протоколов внутри каждого уровня, а также взаимодействие с верхними и нижними уровнями.

При использовании многоуровневой модели для описания сетевых протоколов и эксплуатации существуют определённые преимущества. Использование многоуровневой модели:

* упрощает разработку протоколов, так как протоколы, работающие на определённом уровне, определяют формат обрабатываемых данных и предоставляют интерфейс к верхним и нижним уровням;
* способствует созданию поставщиками конкурирующих продуктов унифицированных решений;
* исключает возможности изменения технологий или функций одного уровня без учёта последствий для верхних и нижних уровней;
* предоставляет общий язык для описания функций сетевого взаимодействия.

Существует два основных типа моделей сети.

* **Протокольная модель** соответствует структуре определённого набора протоколов. Иерархический набор связанных протоколов в типичном случае соответствует всем функциональным возможностям, необходимым для взаимодействия сети, объединяющей людей, с сетью передачи данных. TCP/IP — протокольная модель, поскольку в ней описываются функции, которые выполняются на каждом уровне протоколов, входящих в семейство протоколов TCP/IP.
* **Эталонная модель** обеспечивает последовательность во всех сетевых протоколах и сервисах путем описания того, что необходимо сделать на определённом уровне, но не предписывает конкретные способы выполнения. Эталонная модель не используется как спецификация для непосредственной реализации и не обеспечивает достаточный уровень детализации, чтобы точно определить сервисы сетевой архитектуры. Основная цель эталонной модели — способствовать более ясному пониманию соответствующих функций и процессов.

Модель взаимодействия открытых систем (OSI) является наиболее широко известной межсетевой эталонной моделью. Она используется для проектирования сетей передачи данных, технических требований к операциям, а также для поиска и устранения неполадок.

Как показано на рисунке, TCP/IP и OSI — основные используемые модели функциональности сети. Разработчики сетевых протоколов, сервисов или устройств могут создавать собственные модели для представления продуктов. В конечном итоге разработчикам необходимо взаимодействовать с отраслью путем соотнесения продуктов и сервисов либо с моделью OSI, моделью TCP/IP, либо с обеими.

**Эталонная модель OSI**

Исходная модель взаимодействия открытых систем (OSI) была разработана организацией ISO для предоставления платформы и создания набора протоколов открытых систем. Идея заключалась в том, чтобы этот набор протоколов можно было использовать для разработки международной сети, которая бы не зависела от запатентованных систем.

В конечном итоге скорость принятия и расширения Интернета на основе TCP/IP вызвала задержку разработки и внедрения набора протоколов OSI. Хотя некоторые из разработанных протоколов, использующих OSI, широко применяются сегодня, 7-уровневая модель OSI внесла большой вклад в разработку других протоколов и продуктов для всех типов новых сетей.

Модель OSI представляет расширенный список возможностей и сервисов, которые могут происходить на каждом уровне. Кроме того, она описывает взаимодействие каждого уровня с уровнями, расположенными рядом. Несмотря на то, что содержимое данного курса структурировано по эталонной модели OSI, основное внимание в обсуждении отводится протоколам, определяемым в модели протоколов TCP/IP. Нажмите на название уровня, чтобы просмотреть подробные сведения о каждом из них.

**Модель протокола TCP/IP**

Протокольная модель сетевого взаимодействия TCP/IP была создана в начале 70-х годов и нередко называется моделью сети Интернет. Как показано на рисунке, такая модель определяет четыре категории функций, необходимых для успешного взаимодействия. Архитектура протоколов TCP/IP построена на основе этой модели. Поэтому модель сети Интернет обычно называют моделью TCP/IP.

Большинство протокольных моделей описывают стек протоколов определённого производителя. Тем не менее, поскольку модель TCP/IP представляет собой открытый стандарт, ни одна компания не вправе контролировать её определение. Определения и протоколы TCP/IP рассматриваются на общедоступном форуме и определяются в общедоступных стандартах RFC. RFC содержат как официальные технические характеристики протоколов обмена данными, так и ресурсы, описывающие применение протоколов.

RFC также содержат технические и организационные документы, касающиеся сети Интернет, в том числе техническую информацию и документы о политиках, составляемых организацией IETF.

**Сравнение моделей OSI и TCP/IP**

Семейство протоколов TCP/IP может быть описано с точки зрения эталонной модели OSI. В модели взаимодействия открытых систем (OSI) уровень доступа к сети и уровень приложений в модели TCP/IP дополнительно подразделяются для описания отдельных функций, которые реализуются на этих уровнях.

На уровне доступа к сети семейство протоколов TCP/IP не определяет список протоколов, используемых для передачи по физической среде; оно описывает только передачу с межсетевого уровня физическим сетевым протоколам. Уровни 1 и 2 модели OSI описывают необходимые процедуры для доступа к среде передачи и физическим средствам отправки данных по сети.

Как показано на рисунке, полное совпадение двух сетевых моделей происходит на уровнях 3 и 4 модели OSI. Уровень 3 (сетевой уровень) обычно используется для описания ряда процессов, которые возникают во всех сетях передачи данных при адресации и маршрутизации сообщений в объединённой сети. IP — это протокол из набора протоколов TCP/IP, реализующий функциональность, описанную на уровне 3 OSI. IP — протокол из набора протоколов TCP/IP, реализующий функциональность, описанную на уровне 3 OSI.

Уровень 4 (транспортный уровень модели OSI) описывает общие сервисы и функции, которые предоставляют упорядоченную и надёжную доставку данных между узлами источника и назначения. Эти функции включают подтверждение, устранение ошибок и восстановление последовательности данных. На этом уровне основные функции обеспечивают протоколы TCP и UDP (Протокол пользовательских датаграмм) семейства TCP/IP.

Прикладной уровень TCP/IP включает в себя ряд протоколов, которые предоставляют определённую функциональность различным приложениям для конечных пользователей. Уровни 5, 6 и 7 модели OSI используются в качестве ссылки для разработчиков и поставщиков прикладного программного обеспечения в производстве сетевой продукции.

**Передача сообщений**

Теоретически одно сообщение, например, музыкальное видео или сообщение электронной почты, может быть отправлено по сети от источника к месту назначения как один массивный и бесперебойный поток битов. Если бы сообщения действительно передавались, это означало бы, что другие устройства не смогли бы отправлять и получать сообщения в той же сети в течение всего процесса передачи данных. Большие потоки данных приводили бы к существенным задержкам. Кроме того, если какое-либо из звеньев инфраструктуры сети отказало бы во время передачи данных, целое сообщение было бы утрачено и его необходимо было бы передать повторно в полном объёме.

В этом случае следует разделить данные на более мелкие и управляемые части для передачи по сети. Такое разделение потока данных на более мелкие части называется сегментацией. Сегментация сообщения предоставляет два основных преимущества:

* путём отправки небольших отдельных частей от источника к получателю в сети можно поддерживать множество различных чередующихся обменов сообщениями; процесс, используемый для доставки с чередованием частей отдельных обменов сообщениями в сети, называется мультиплексированием. Нажмите на каждую из кнопок на рис. 1, а затем нажмите кнопку «Воспроизведение» для просмотра анимации о сегментации и мультиплексировании.
* Сегментация может повысить надёжность сетевого взаимодействия. Отдельные части каждого сообщения не обязательно следуют по одному и тому же пути по сети от источника к получателю. Если определённый путь будет переполнен трафиком или сетевое оборудование выйдет из строя, отдельные части сообщения могут быть отправлены к месту назначения по другому пути. Если какую-либо часть сообщения не удаётся доставить к месту назначения, необходимо будет повторно передать только отсутствующие компоненты.

Недостаток использования сегментации и мультиплексирования для передачи сообщений через сеть — уровень сложности, которая добавляется ко всему процессу. Представьте себе, что необходимо отправить письмо из 100 страниц, но каждый конверт вмещает только одну страницу. Процесс написания адресов, маркировка, получение и открытие всех 100 конвертов отнимает много времени у отправителя и получателя.

В области коммуникаций все сегменты сообщения должны пройти подобный процесс, чтобы сообщение было доставлено до нужного места назначения и могло быть собрано в правильное содержимое исходного сообщения, как показано на рисунке 2.

Различные типы устройств в сети участвуют в обеспечении надёжной доставки всех частей сообщения в место назначения.

**Протокольные блоки данных (PDU)**

По мере того, как данные приложений передаются по стеку протоколов на пути к передаче по сетевой среде, различные протоколы добавляют в них информацию на каждом из уровней. Обычно это называется процессом инкапсуляции.

Форма, которую принимает массив данных на каждом из уровней, называется протокольными блоками данных (PDU). В ходе инкапсуляции каждый последующий уровень инкапсулирует PDU, полученный от вышестоящего уровня в соответствии с используемым протоколом. На каждом этапе процесса PDU получает другое имя для отражения новых функций. Несмотря на то, что не существует универсальной схемы именования для PDU, в этом курсе PDU называются в соответствии с терминологией набора протоколов TCP/IP, как показано на рисунке.

* **Данные** — общий термин для обозначения PDU прикладного уровня
* **Сегмент** — PDU транспортного уровня
* **Пакет** — PDU сетевого уровня
* **Кадр** — PDU уровня канала данных
* **Биты** — PDU физического уровня, используемый при физической передаче данных по среде передачи

**Инкапсуляция**

Инкапсуляция данных — процесс, который добавляет к данным содержимое заголовка дополнительного протокола перед передачей. В большинстве форм передачи данных первоначальные данные подвергаются инкапсуляции нескольких протоколов до начала передачи.

При отправке сообщения по сети стек протоколов на узле работает от верхнего уровня к нижнему. В примере с веб-сервером мы можем использовать модель TCP/IP, чтобы иллюстрировать процесс отправки клиенту веб-страницы в формате HTML.

Протокол прикладного уровня (HTTP) запускает процесс, предоставляя отформатированные данные HTML веб-страницы транспортному уровню. В нём данные приложения разбиваются на сегменты TCP. Каждому сегменту TCP присваивается метка, называемая заголовком и содержащая информацию о том, какой процесс, запущенный на компьютере назначения, должен получить сообщение. Кроме того, он содержит информацию, которая помогает процессу назначения собрать данные обратно в исходный формат.

Транспортный уровень инкапсулирует данные веб-страницы в формате HTML в сегменте и передаёт его на межсетевой уровень, где реализован протокол IP. В нём весь сегмент TCP инкапсулируется в IP-пакет, и к нему добавляется еще одна метка, называемая заголовком IP. В заголовке IP указываются IP-адреса узлов отправителя и получателя, а также данные, необходимые для доставки пакета соответствующему процессу назначения.

Далее этот пакет IP передаётся на уровень сетевого доступа, где он инкапсулируется — к нему добавляются заголовок кадра и Концевик. В каждом заголовке кадра указываются физические адреса источника и назначения. Физический адрес служит для идентификации устройства в локальной сети. В концевике указывается информация для проверки ошибок. Наконец, биты кодируются в среду передачи сетевой интерфейсной платой сервера. На рисунке нажмите кнопку «Воспроизведение», чтобы увидеть процесс инкапсуляции.

**Деинкапсуляция**

Обратный процесс на принимающем узле называется деинкапсуляцией. Деинкапсуляция — процесс, который выполняется приёмным устройством, чтобы удалить один или несколько заголовков протоколов. Данные деинкапсулируются по мере продвижения по стеку к приложениям для конечных пользователей. На рисунке нажмите кнопку «Воспроизведение», чтобы увидеть процесс деинкапсуляции.

**Сетевые адреса и адреса канала передачи данных**

Модель взаимодействия открытых систем (OSI) описывает процессы кодирования, форматирования, сегментации и инкапсуляции данных для передачи по сети. Сетевой и канальный уровни отвечают за предоставление данных от отправляющего устройства, или источника, к принимающему устройству, или устройству назначения. Протоколы на обоих уровнях содержат адреса источника и назначения, которые имеют различные цели.

**Сетевой адрес**

Логический адрес сетевого уровня (уровень 3) содержит информацию, необходимую для доставки IP-пакета между устройствами. IP-адрес уровня 3 имеет две части: префикс сети и узловую часть. Префикс сети используется маршрутизаторами, чтобы передать пакет в соответствующую сеть. Узловая часть используется последним маршрутизатором для доставки пакета к устройству назначения.

IP-пакет содержит два IP-адреса:

* **IP-адрес источника** — IP-адрес отправляющего устройства.
* **IP-адрес назначения** — IP-адрес принимающего устройства. IP-адрес назначения используется маршрутизаторами для передачи пакета к месту назначения.

**Адрес канала передачи данных**

Физический адрес канального уровня (уровень 2) играет другую роль. Назначение адреса канала передачи данных — доставлять кадр канала передачи данных с одного сетевого интерфейса на другой в одной и той же сети. Прежде чем IP-пакет можно будет отправить по проводной или беспроводной сети, его необходимо инкапсулировать в кадр канала передачи данных для последующей передачи по физической среде реальной сети. Локальные сети Ethernet и беспроводные локальные сети — два примера сетей с различными физическими носителями, каждая из которых имеет собственный тип протокола канального уровня.

IP-пакет инкапсулируется в кадр канала передачи данных для доставки в сеть назначения. Добавляются адреса канального уровня источника и назначения, как показано на рисунке.

* **Адрес канального уровня источника** — физический адрес устройства, отправляющего пакет. Первоначально им является NIC источника IP-пакета.
* **Адрес канального уровня назначения** — физический адрес сетевого интерфейса либо следующего маршрутизатора, либо интерфейса устройства назначения.

**Обмен данными с помощью устройства в одной сети**

Чтобы узнать, насколько успешно передано сообщение в сети, важно понять роль как адресов сетевого уровня, так и адресов канального уровня при взаимодействии устройства с другим устройством в одной и той же сети. Для иллюстрации этого примера есть клиентский компьютер PK1, который взаимодействует с файловым сервером (FTP-сервер), находящимся в той же IP-сети.

**Сетевые адреса**

Адреса сетевого уровня, или IP-адреса, представляют собой сетевые адреса источника и назначения. Сетевая часть адреса будет единой; отличаться будет только часть адреса, описывающая отдельный узел или устройство.

* **IP-адрес источника** — IP-адрес устройства-отправителя, клиентский компьютер ПК1: 192.168.1.110.
* **IP-адрес назначения** — IP-адрес принимающего устройства, FTP-сервер: 192.168.1.9.

**Адреса канала передачи данных**

Если отправитель и получатель IP-пакета находятся в одной и той же сети, кадр канала передачи данных отправляется напрямую принимающему устройству. В сети Ethernet адреса канала передачи данных называются MAC-адресом Ethernet. MAC-адреса представляют собой 48-битные адреса, которые физически встроены на сетевой интерфейсной плате Ethernet. MAC-адрес также называется физическим или аппаратным адресом (BIA).

* **MAC-адрес источника** — это адрес канального уровня, или MAC-адрес Ethernet устройства, отправляющего IP-пакет, PK1. MAC-адрес сетевой интерфейсной платы Ethernet (NIC) ПК1: 0A-AA-AA-AA-AA-AA.
* **MAC-адрес назначения** — адрес канального уровня принимающего устройства, если получающее устройство находится в той же сети, что и устройство-отправитель. В этом примере MAC-адрес получателя — MAC-адрес FTP-сервера: 0C-CC-CC-CC-CC-CC.

Адреса источника и назначения добавляются в кадр Ethernet. Теперь кадр с инкапсулированным IP-пакетом можно передавать с PK1 сразу на FTP-сервер.

**МАС- и IP-адреса**

Теперь вполне очевидно: чтобы отправить данные другому узлу в той же локальной сети, исходный узел должен знать и физический, и логический адрес узла назначения. После этого он может создать кадр и отправить его по сетевой среде. Исходный узел может узнать IP-адрес назначения несколькими способами. Например, с помощью Службы доменных имен (DNS) или по тому, что адрес введён в приложении вручную (например, если пользователь указывает IP-адрес требуемого FTP-сервера). Но как узел определяет MAC-адрес Ethernet другого устройства?

Большинство сетевых приложений находят местоположение взаимодействующих узлов только по логическому IP-адресу. MAC-адрес канала передачи данных необходим для доставки инкапсулированного в кадре Ethernet IP-пакета по сети к месту назначения.

С помощью IP-протокола, который называется протоколом разрешения адресов (ARP), узел определяет MAC-адрес любого узла в той же локальной сети. Передающий узел отправляет сообщение с запросом ARP по всей локальной сети. ARP-запрос представляет собой широковещательное сообщение. ARP-запрос содержит IP-адрес устройства назначения. Каждое устройство в локальной сети анализирует ARP-запрос на предмет содержания своего IP-адреса. Только устройство с IP-адресом, представленным в ARP-запросе, возвращает ответ ARP. Ответ ARP включает в себя МАС-адрес, связанный с IP-адресом в ARP-запросе.

**Шлюз по умолчанию**

Метод, с помощью которого узел отправляет сообщения адресату в удалённой сети, отличается от метода отправки адресату в той же локальной сети. При отправке узлу, подключённому к той же сети, сообщение направляется напрямую. Узел с помощью протокола ARP определяет MAC-адрес узла назначения. Он включает IP-адрес в заголовок пакета и инкапсулирует пакет в кадр, где находится MAC-адрес назначения, а затем передает его далее.

Если узлу нужно отправить сообщение в удалённую сеть, необходимо использовать маршрутизатор, который также называется шлюзом по умолчанию. Шлюз по умолчанию — это IP-адрес интерфейса маршрутизатора в той же сети, в которой находится отправляющий узел.

Для каждого узла в локальной сети важно правильно задать адрес шлюза по умолчанию. Если в настройках узла TCP/IP адрес шлюза по умолчанию не указан или указан неверно, сообщения, адресованные узлам в удалённых сетях, доставлены не будут.

На рисунке узлы в локальной сети используют маршрутизатор R1 как шлюз по умолчанию, поскольку его адрес 192.168.1.1 указан в настройках TCP/IP. Если назначение блоков PDU относится к другой сети IP, узлы отправляют их к шлюзу по умолчанию на маршрутизаторе для последующей передачи.