

ТЕХНОЛОГИИ ФИЗИЧЕСКОГО УРОВНЯ ПЕРЕДАЧИ ДАННЫХ

Занятие №25

Модель OSI

1. Архитектура и стандартизация сетей
2. Декомпозиция задачи сетевого взаимодействия
3. Многоуровневый подход
4. Протокол и стек протоколов
5. Общая характеристика модели OSI
6. Физический уровень
7. Канальный уровень
8. Сетевой уровень
9. Транспортный уровень
10. Сеансовый уровень
11. Уровень представления
12. Прикладной уровень
13. Вопросы

Архитектура и стандартизация сетей

Архитектура подразумевает представление сети в виде системы элементов, каждый из которых выполняет определенную частную функцию, при этом все элементы вместе согласованно решают общую задачу взаимодействия компьютеров. Другими словами, архитектура сети отражает декомпозицию общей задачи взаимодействия компьютеров на отдельные подзадачи, которые должны решаться отдельными элементами сети.

Одним из важных элементов архитектуры сети является коммуникационный протокол — формализованный набор правил взаимодействия узлов сети.

Прорывом в стандартизации архитектуры компьютерной сети стала разработка модели взаимодействия открытых систем (Open System Interconnection, OSI). Модель OSI является международным стандартом и определяет способ декомпозиции задачи взаимодействия «по вертикали», поручая эту задачу коммуникационным протоколам семи уровней. Уровни образуют иерархию, известную как стек протоколов, где каждый вышестоящий уровень использует нижестоящий в качестве удобного инструмента для решения своих задач.

Стандартная архитектура компьютерной сети определяет также распределение протоколов между элементами сети — конечными узлами (компьютерами) и промежуточными узлами (повторителями, концентраторами, мостами, коммутаторами, маршрутизаторами, шлюзами).

Промежуточные узлы выполняют только транспортные функции стека протоколов, передавая трафик между конечными узлами.

Конечные узлы поддерживают весь стек протоколов, предоставляя информационные услуги, например веб-сервис. Такое распределение функций означает смещение «интеллекта» сети на ее периферию.

Декомпозиция задачи сетевого взаимодействия

Организация взаимодействия между устройствами сети является сложной задачей. Для решения сложных задач используется известный универсальный прием — *декомпозиция*, то есть разбиение одной сложной задачи на несколько более простых задач-модулей.

Декомпозиция состоит в четком определении:

- функций каждого модуля,
- порядке их взаимодействия (то есть межмодульных интерфейсов).

В результате такого логического упрощения задачи появляется возможность независимого тестирования, разработки и модификации модулей. Так, любой из показанных на рис. 1 модулей может быть переписан заново.

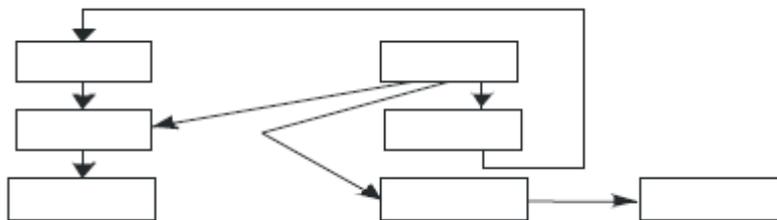


Рис. 1. Пример декомпозиции задачи

Многоуровневый подход

Еще более эффективной концепцией, развивающей идею декомпозиции, является *многоуровневый подход*. После представления исходной задачи в виде множества модулей эти модули группируют и упорядочивают по уровням, образующим иерархию. В соответствии с принципом иерархии для каждого промежуточного уровня можно указать непосредственно примыкающие к нему соседние вышележащий и нижележащий уровни (рис. 2).

Группа модулей, составляющих каждый уровень, для решения своих задач должна обращаться с запросами только к модулям соседнего нижележащего уровня. С другой стороны, результаты работы каждого из модулей, отнесенных к некоторому уровню, могут быть переданы только модулям соседнего вышележащего уровня. Такая иерархическая декомпозиция задачи предполагает четкое определение функций и интерфейсов не только отдельных модулей, но и каждого уровня.

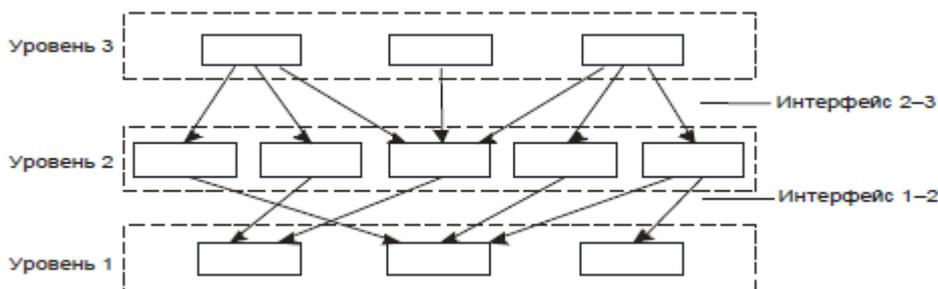


Рис. 2. Многоуровневый подход — создание иерархии задач

Межуровневый интерфейс, называемый также интерфейсом услуг, определяет набор функций, которые нижележащий уровень предоставляет вышележащему (рис. 2).

Такой подход дает возможность проводить разработку, тестирование и модификацию отдельного уровня независимо от других уровней. Иерархическая декомпозиция позволяет, двигаясь от более низкого уровня к более высокому, переходить ко все более и более абстрактному, а значит, и более простому представлению исходной задачи.

Протокол и стек протоколов

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют, по меньшей мере, *две стороны*, то есть в данном случае необходимо организовать согласованную работу двух иерархий аппаратных и программных средств, работающих на разных

компьютерах. Оба участника сетевого обмена должны принять множество соглашений. Например, они должны согласовать уровни и форму электрических сигналов, способ определения размера сообщений, договориться о методах контроля достоверности и т. п. Другими словами, соглашения должны быть приняты на всех уровнях, начиная от самого низкого — уровня передачи битов и заканчивая самым высоким, реализующим обслуживание пользователей сети. На рис. 3. показана модель взаимодействия двух узлов.

С каждой стороны средства взаимодействия представлены четырьмя уровнями. Каждый уровень поддерживает интерфейсы двух типов:

1. Интерфейсы услуг с выше- и нижележащим уровнями «своей» иерархии средств.
2. Интерфейс со средствами взаимодействия другой стороны, расположенными на том же уровне иерархии. Этот тип интерфейса называют протоколом.

Таким образом, протокол всегда является одноранговым интерфейсом.

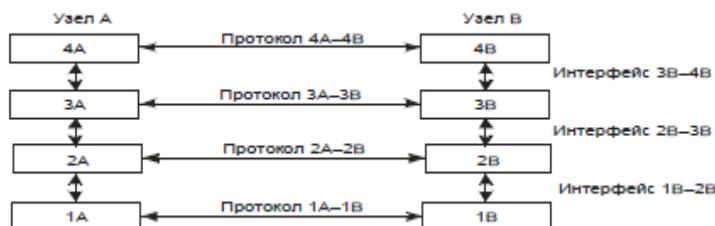


Рис. 3. Взаимодействие двух узлов

В сущности, термины «протокол» и «интерфейс» выражают одно и то же понятие — формализованное описание процедуры взаимодействия двух объектов, но традиционно в сетях за ними закрепили разные области действия: протоколы определяют правила взаимодействия модулей одного уровня в разных узлах, а интерфейсы — правила взаимодействия модулей соседних уровней в одном узле.

Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называется стеком протоколов.

Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней, как правило, программными средствами.

Программный модуль, реализующий некоторый протокол, называют протокольной сущностью, или, для краткости, тоже протоколом.

Протокольные сущности одного уровня двух взаимодействующих сторон обмениваются сообщениями в соответствии с определенным для них протоколом.

Сообщения состоят из заголовка и поля данных (иногда оно может отсутствовать).

Обмен сообщениями является своеобразным языком общения, с помощью которого каждая из сторон «объясняет» другой стороне, что необходимо сделать на каждом этапе взаимодействия. Работа каждого протокольного модуля состоит в чтении, анализе заголовков поступающих к нему сообщений и выполнении связанных с этим действий.

Заголовки сообщений разных протоколов имеют разную структуру, что соответствует различиям в их функциональности. Понятно, что чем сложнее структура заголовка сообщения, тем более сложные функции возложены на соответствующий протокол.

Общая характеристика модели OSI

Из того, что протокол является соглашением, принятым двумя взаимодействующими узлами сети, совсем не следует, что он обязательно является стандартным. Но на практике при реализации сетей стремятся использовать стандартные протоколы. Это могут быть фирменные, национальные или международные стандарты.

В начале 80-х годов ряд международных организаций по стандартизации, в частности International Organization for Standardization (ISO), часто называемая также

International Standards Organization, а также International Telecommunications Union (ITU) – разработали стандартную модель взаимодействия открытых систем (Open System Interconnection, OSI).

Эта модель сыграла значительную роль в развитии компьютерных сетей.

Назначение модели OSI состоит в обобщенном представлении средств сетевого взаимодействия. Она разрабатывалась в качестве своего рода универсального языка сетевых специалистов, именно поэтому ее называют справочной моделью.

Модель OSI определяет:

1. уровни взаимодействия систем в сетях с коммутацией пакетов,
2. стандартные названия уровней,
3. функции, которые должен выполнять каждый уровень.

Модель OSI не содержит описаний реализаций конкретного набора протоколов. В модели OSI (рис. 4.) средства взаимодействия делятся на *семь* уровней:

1. прикладной,
2. представления,
3. сеансовый,
4. транспортный,
5. сетевой,
6. канальный,
7. физический.

Каждый уровень имеет дело с совершенно определенным аспектом взаимодействия сетевых устройств.

Модель OSI описывает только системные средства взаимодействия, реализуемые операционной системой, системными программными и аппаратными средствами. Модель не включает средства взаимодействия приложений конечных пользователей. Важно различать уровень взаимодействия приложений и прикладной уровень семиуровневой модели.

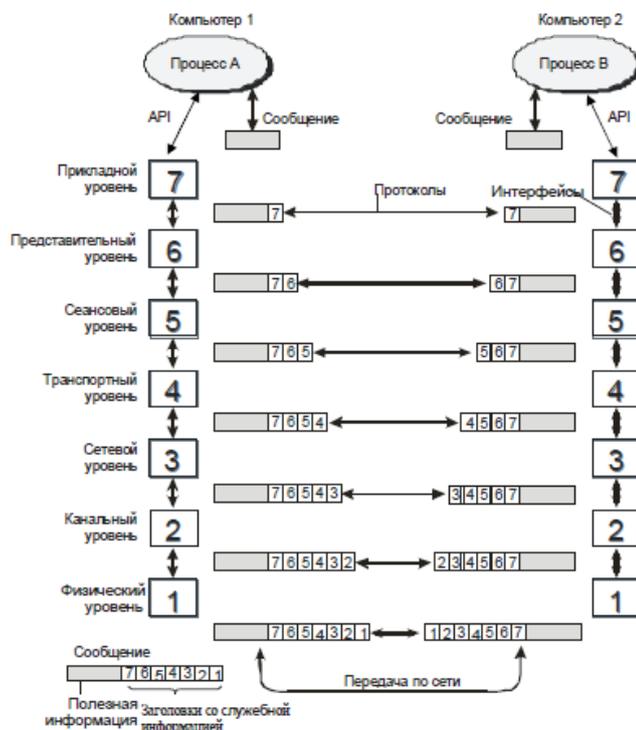


Рис. 4. Модель взаимодействия открытых систем ISO/OSI

Приложения могут реализовывать собственные протоколы взаимодействия, используя для этих целей многоуровневую совокупность системных средств. Именно для

этого в распоряжение программистов предоставляется прикладной программный интерфейс (Application Program Interface, API).

Итак, пусть приложение узла А хочет взаимодействовать с приложением узла В. Для этого приложение А обращается с запросом к прикладному уровню, например к файловой службе. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата.

После формирования сообщения прикладной уровень направляет его вниз по стеку уровню представления. Протокол уровня представления на основании информации, полученной из прикладного уровня, выполняет требуемые действия и добавляет к сообщению собственную служебную информацию — заголовок уровня представления, в котором содержатся указания для протокола уровня представления машины-адресата. Полученное в результате сообщение передается вниз сеансовому уровню, который в свою очередь добавляет свой заголовок и т. д. (Некоторые реализации протоколов помещают служебную информацию не только в начале сообщения в виде заголовка, но и в конце в виде так называемого концевика.) Наконец, сообщение достигает нижнего, физического уровня, который, собственно, и передает его по линиям связи машине-адресату. К этому моменту сообщение «обрастает» заголовками всех уровней (рис. 5.).

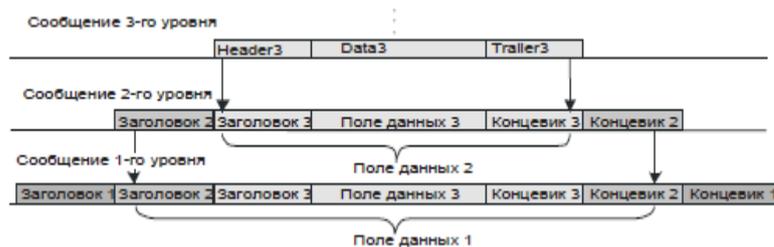


Рис. 5. Вложенность сообщений различных уровней

Физический уровень помещает сообщение на физический выходной интерфейс компьютера 1, и оно начинает свое «путешествие» по сети (до этого момента сообщение передавалось от одного уровня другому в пределах компьютера 1).

Когда сообщение по сети поступает на входной интерфейс компьютера 2, оно принимается его физическим уровнем и последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует и обрабатывает заголовок своего уровня, выполняя соответствующие функции, а затем удаляет этот заголовок и передает сообщение вышележащему уровню.

Как видно из описания, протокольные сущности одного уровня не общаются между собой непосредственно, в этом общении всегда участвуют посредники — средства протоколов нижележащих уровней. И только физические уровни различных узлов взаимодействуют непосредственно.

Физический уровень

Физический уровень (physical layer) имеет дело с передачей потока битов по физическим каналам связи, таким как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал.

Функции физического уровня реализуются на всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом.

Примером протокола физического уровня может служить спецификация 10BaseT технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического сегмента 100 метров, манчестерский код для

представления данных в кабеле, а также некоторые другие характеристики среды и электрических сигналов.

В стандартах ISO для обозначения единиц обмена данными, с которыми имеют дело протоколы разных уровней, используется общее название протокольная единица данных (Protocol Data Unit, PDU). Для обозначения единиц обмена данными конкретных уровней часто используются *специальные* названия, в частности: сообщение, кадр, пакет, дейтаграмма.

Физический уровень не вникает в смысл информации, которую он перелает. Для него эта информация представляет однородный поток битов, которые нужно доставить без искажений и в соответствии с заданной тактовой частотой (интервалом между соседними битами).

Канальный уровень

Канальный уровень (data link layer) является первым уровнем (если идти снизу вверх), который работает в режиме коммутации пакетов. На этом уровне PDU (протокольный блок данных) обычно носит название кадр (frame).

Функции средств канального уровня определяются по-разному для локальных и глобальных сетей:

- В локальных сетях канальный уровень должен обеспечивать доставку кадра между любыми узлами сети. При этом предполагается, что сеть имеет типовую топологию, например общую шину, кольцо, звезду или дерево (иерархическую звезду). Примерами технологий локальных сетей, применение которых ограничено типовыми топологиями, являются Ethernet, FDDI, Token Ring.
- В глобальных сетях канальный уровень должен обеспечивать доставку кадра только между двумя соседними узлами, соединенными индивидуальной линией связи. Примерами двухточечных протоколов (как часто называют такие протоколы) могут служить широко распространенные протоколы PPP (Point to Point Protocol) и HDLC. На основе двухточечных связей могут быть построены сети произвольной топологии.

Для связи локальных сетей между собой или для доставки сообщений между любыми конечными узлами глобальной сети используются средства более высокого сетевого уровня.

Одной из функций канального уровня является поддержание интерфейсов с нижележащим физическим уровнем и вышележащим сетевым уровнем. Сетевой уровень направляет канальному уровню пакет для передачи в сеть или принимает от него пакет, полученный из сети. Физический уровень используется канальным как инструмент, который принимает и передает в сеть последовательности битов.

Начнем рассмотрение работы канального уровня, начиная с момента, когда сетевой уровень отправителя передает канальному уровню пакет, а также указание, какому узлу его передать. Для решения этой задачи канальный уровень создает кадр, который имеет поле данных и заголовок. Канальный уровень помещает (*инкапсулирует*) пакет в поле данных кадра и заполняет соответствующей служебной информацией заголовок кадра. Важнейшей информацией заголовка кадра является адрес назначения, на основании которого коммутаторы сети будут продвигать пакет.

Одной из задач канального уровня является *обнаружение и коррекция ошибок*. Для этого канальный уровень фиксирует границы кадра, помещая специальную последовательность битов в его начало и конец (преамбула (7 байт 10101010, 8-ой байт 10101011), флаг (01111110), необходимые для синхронизации кадра), а затем вставляет в кадр контрольную сумму, которая называется также контрольной последовательностью кадра (Frame Check Sequence, FCS). Контрольная сумма вычисляется по некоторому алгоритму как функция от всех байтов кадра. По значению FCS узел назначения сможет определить, были или нет искажены данные кадра в процессе передачи по сети.

Однако прежде, чем переправить кадр физическому уровню для непосредственной передачи данных в сеть, канальному уровню может потребоваться решить еще одну важную задачу. Если в сети используется разделяемая среда, то прежде чем физический уровень начнет передавать данные, канальный уровень должен *проверить доступность среды*.

Если разделяемая среда освободилась (когда она не используется, то такая проверка, конечно, пропускается), кадр передается средствами физического уровня в сеть, проходит по каналу связи и поступает в виде последовательности битов в распоряжение физического уровня узла назначения. Этот уровень в свою очередь передает полученные биты «наверх» канальному уровню своего узла. Последний группирует биты в кадры, снова вычисляет контрольную последовательность кадра полученных данных по тому же алгоритму и сравнивает результат с контрольной последовательностью, переданной в кадре. Если они совпадают, кадр считается правильным. Если же контрольные последовательности не совпадают, фиксируется ошибка. В функции канального уровня входит не только обнаружение ошибок, но и исправление их за счет повторной передачи поврежденных кадров. Однако эта функция не является обязательной и в некоторых реализациях канального уровня она отсутствует, например в Ethernet, Token Ring, FDDI и Frame Relay.

Все вышеназванные функции канального уровня иногда выделяют в отдельный подуровень управления доступом к среде (Media Access Control, MAC), а функции интерфейса с вышележащим сетевым уровнем выделяют в подуровень управления логической связью (Logical Link Control).

Протокол канального уровня обычно работает в пределах сети, являющейся одной из составляющих более крупной составной сети, объединенной протоколами сетевого уровня.

Адреса, с которыми работает протокол канального уровня, используются для доставки кадров только в пределах этой сети, а для перемещения пакетов между сетями применяются уже адреса следующего, сетевого, уровня.

В локальных сетях канальный уровень поддерживает весьма мощный и законченный набор функций по пересылке сообщений между узлами сети. В некоторых случаях протоколы канального уровня локальных сетей оказываются самодостаточными транспортными средствами и могут допускать работу непосредственно поверх себя протоколов прикладного уровня или приложений без привлечения средств сетевого и транспортного уровней. Тем не менее, для качественной передачи сообщений в сетях с произвольной топологией функций канального уровня оказывается недостаточно.

Это утверждение в еще большей степени справедливо для глобальных сетей, в которых протокол канального уровня реализует достаточно простую функцию передачи данных между соседними узлами.

В качестве примера приведены на рис. 6 протоколы сетевых технологий семейств Ethernet и WLAN, а также технологии Token Ring, разработанных 802 комитетом Института инженеров по электронике и радиоэлектронике (IEEE).

Сетевой уровень

Сетевой уровень (network layer) служит для образования единой транспортной системы, объединяющей несколько сетей, называемой составной сетью, или интернетом.

Не следует путать интернет (со строчной буквы) с Интернетом (с прописной буквы).

Интернет - это самая известная и охватывающая весь мир реализация составной сети, построенная на основе технологии TCP/IP.

На рис. 7. показаны несколько сетей, каждая из которых использует собственную технологию канального уровня: Ethernet, FDDI, Token Ring, ATM, Frame Relay. На базе этих технологий каждая из указанных сетей может связывать между собой любых

пользователей, но только *своей* сети, и не способна обеспечить передачу данных в другую сеть.

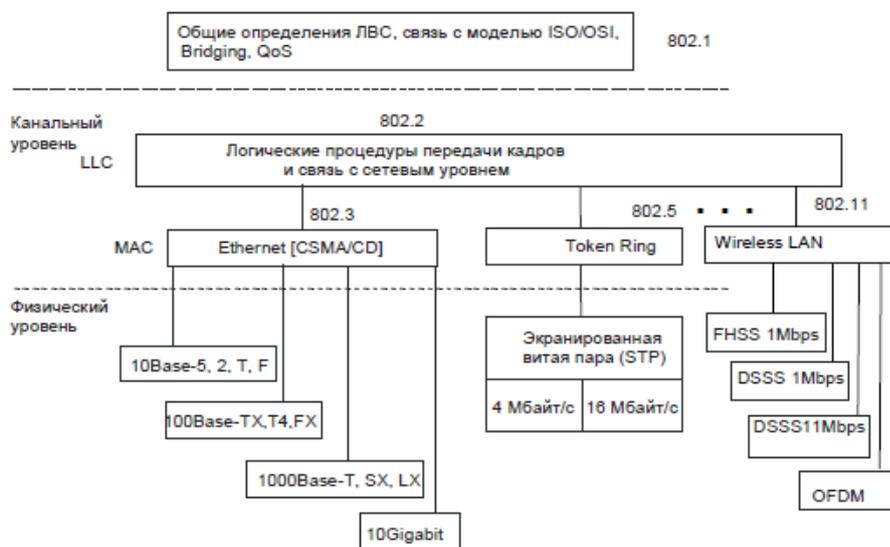


Рис. 6. Структура стандартов IEEE 802.x

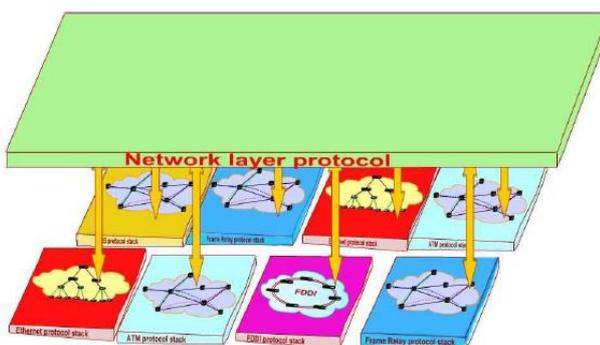


Рис. 7. Необходимость сетевого уровня

Причина такого положения вещей очевидна и кроется в существенных отличиях одной технологии от другой. Даже наиболее близкие технологии LAN — Ethernet, FDDI, Token Ring, — имеющие одну и ту же систему адресации (адреса подуровня MAC, называемые MAC-адресами), отличаются друг от друга форматом используемых кадров и логикой работы протоколов. Еще больше отличий между технологиями LAN и WAN. Во многих технологиях WAN задействована техника предварительно устанавливаемых виртуальных каналов, идентификаторы которых применяются в качестве адресов. Все технологии имеют собственные форматы кадров (в технологии ATM кадр даже называется иначе — ячейкой) и, конечно, собственные стеки протоколов.

Чтобы связать между собой сети, построенные на основе столь отличающихся технологий, нужны *дополнительные средства*, и такие средства предоставляет сетевой уровень.

Функции сетевого уровня реализуются:

- группой протоколов;
- специальными устройствами - маршрутизаторами.

Одной из функций маршрутизатора является *физическое соединение сетей*. Маршрутизатор имеет несколько сетевых интерфейсов, подобных интерфейсам компьютера, к каждому из которых может быть подключена одна сеть. Таким образом, все интерфейсы маршрутизатора можно считать узлами разных сетей. Маршрутизатор

может быть реализован программно, на базе универсального компьютера (например, типовая конфигурация Unix или Windows включает программный модуль маршрутизатора). Однако чаще маршрутизаторы реализуются на базе специализированных аппаратных платформ. В состав программного обеспечения маршрутизатора входят протокольные модули сетевого уровня.

Чтобы связать сети, показанные на рис. 7, необходимо соединить все эти сети маршрутизаторами и установить протокольные модули сетевого уровня на все конечные узлы пользователей, которые хотели бы связываться через составную сеть (рис. 8).

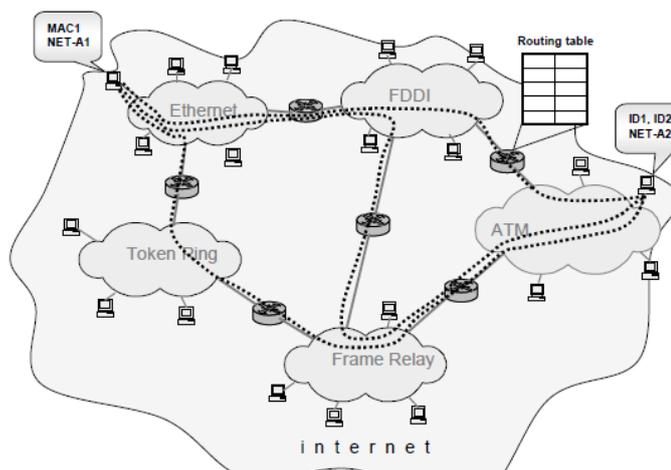


Рис. 8. Пример internet

Данные, которые необходимо передать через составную сеть, поступают на сетевой уровень от вышележащего транспортного уровня. Эти данные снабжаются заголовком сетевого уровня. Данные вместе с заголовком образуют пакет — так называется PDU сетевого уровня.

Заголовок пакета сетевого уровня имеет унифицированный формат, не зависящий от форматов кадров канального уровня тех сетей, которые могут входить в составную сеть, и несет наряду с другой служебной информацией данные об адресе назначения этого пакета.

Для того чтобы протоколы сетевого уровня могли доставлять пакеты любому узлу составной сети, эти узлы должны иметь адреса, уникальные в пределах данной составной сети. Такие адреса называются сетевыми, или глобальными. Каждый узел составной сети, который намерен обмениваться данными с другими узлами составной сети, должен иметь сетевой адрес наряду с адресом, назначенным ему на канальном уровне. Например, на рис. 3.9 компьютер в сети Ethernet, входящей в составную сеть, имеет адрес канального уровня MAC1 и адрес сетевого уровня NET-A1; аналогично в сети ATM узел, адресуемый идентификаторами виртуальных каналов ID1 и ID2, имеет сетевой адрес NET-A2. В пакете в качестве адреса назначения должен быть указан адрес сетевого уровня, на основании которого определяется маршрут пакета. *Определение маршрута* является важной задачей сетевого уровня. Маршрут описывается последовательностью сетей (или маршрутизаторов), через которые должен пройти пакет, чтобы попасть к адресату. Например, на рис. 8 штриховой линией показано 3 маршрута, по которым могут быть переданы данные от компьютера А к компьютеру Б. Маршрутизатор собирает информацию о топологии связей между сетями и на ее основании строит таблицы коммутации, которые в данном случае носят специальное название таблиц маршрутизации.

В соответствии с многоуровневым подходом сетевой уровень для решения своей задачи обращается к нижележащему канальному уровню. Весь путь через составную сеть

разбивается на участки от одного маршрутизатора до другого, причем каждый участок соответствует пути через отдельную сеть.

Для того чтобы передать пакет через очередную сеть, сетевой уровень помещает его в поле данных кадра соответствующей канальной технологии, указывая в заголовке кадра канальный адрес интерфейса следующего маршрутизатора. Сеть, используя свою канальную технологию, доставляет кадр с инкапсулированным в него пакетом по заданному адресу.

Маршрутизатор извлекает пакет из прибывшего кадра и после необходимой обработки передаст пакет для дальнейшей транспортировки в следующую сеть, предварительно упаковав его в новый кадр канального уровня в общем случае другой технологии. Таким образом, сетевой уровень играет роль координатора, организующего совместную работу сетей, построенных на основе разных технологий.

На сетевом уровне определяются два вида протоколов. Первый вид — маршрутизируемые протоколы — реализуют продвижение пакетов через сеть. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. Пример: межсетевой протокол (Internet Protocol, IP). В его задачу входит продвижение пакета между сетями — от одного маршрутизатора до другого до тех пор, пока пакет не попадет и сеть назначения.

Однако часто к сетевому уровню относят и другой вид протоколов, называемых маршрутизирующими протоколами, или протоколами маршрутизации. С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений, на основании которой осуществляется выбор маршрута продвижения пакетов. Пример: протоколы *маршрутизации* RIP, OSPF, BGP.

Транспортный уровень

На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением.

Транспортный уровень (transport layer) обеспечивает приложениям или верхним уровням стека — прикладному, представления и сеансовому — передачу данных с той степенью надежности, которая им требуется.

Модель OSI определяет пять классов транспортного сервиса от низшего класса 0 до высшего класса 4. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное — способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

Выбор класса сервиса транспортного уровня определяется:

- С одной стороны, тем, в какой степени задача обеспечения надежности решается самими приложениями и протоколами более высоких, чем транспортный, уровней.
- С другой стороны, этот выбор зависит от того, насколько надежной является система транспортировки данных в сети, обеспечиваемая уровнями, расположенными ниже транспортного, — сетевым, канальным и физическим.

Так, если качество каналов передачи связи очень высокое и вероятность возникновения ошибок, не обнаруженных протоколами более низких уровней, невелика, то разумно воспользоваться одним из облегченных сервисов транспортного уровня, не обремененных многочисленными проверками, квитированием и другими приемами повышения надежности.

Если же транспортные средства нижних уровней очень ненадежны, то целесообразно обратиться к наиболее развитому сервису транспортного уровня, который

работает, используя максимум средств для обнаружения и устранения ошибок, включая предварительное установление логического соединения, контроль доставки сообщений по контрольным суммам и циклической нумерации пакетов, установление тайм-аутов доставки и т. п.

Все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами конечных узлов сети — компонентами их сетевых операционных систем. В качестве примера транспортных протоколов можно привести протоколы TCP и UDP стека TCP/IP и протокол SPX стека NetWare разработки фирмы Novell.

Протоколы нижних четырех уровней обобщенно называют сетевым транспортом, или транспортной подсистемой, так как они полностью решают задачу транспортировки сообщений с заданным уровнем качества в составных сетях с произвольной топологией и различными технологиями. Оставшиеся три верхних уровня решают задачи предоставления прикладных сервисов, используя нижележащую транспортную подсистему.

Сеансовый уровень

Сеансовый уровень (session layer) обеспечивает управление взаимодействием сторон: фиксирует, какая из сторон является активной в настоящий момент, и предоставляет средства синхронизации сеанса. Эти средства позволяют в ходе длинных передач сохранять информацию о состоянии этих передач в виде контрольных точек, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, а не начинать все с начала.

На практике немногие приложения используют сеансовый уровень, и он редко реализуется в виде отдельных протоколов. Функции этого уровня часто объединяют с функциями прикладного и представительного уровней и реализуют в одном протоколе.

Уровень представления

Уровень представления (presentation layer), как явствует из его названия, обеспечивает представление передаваемой по сети информации, не меняя при этом ее содержания. За счет уровня представления информация, передаваемая прикладным уровнем одной системы, всегда понятна прикладному уровню другой системы. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов, например кодов ASCII и EBCDIC.

На этом уровне могут выполняться шифрование и дешифрирование данных, благодаря которым секретность обмена данными обеспечивается сразу для всех прикладных служб. Примером такого протокола является протокол SSL (Secure Socket Layer — слой защищенных сокетов), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP.

Прикладной уровень

Прикладной уровень (application layer) - это в действительности просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые веб-страницы, а также организуют свою совместную работу, например, по протоколу электронной почты.

Единица данных, которой оперирует прикладной уровень, обычно называется сообщением.

Существует очень большое разнообразие протоколов и соответствующих служб прикладного уровня. Приведем в качестве примера несколько наиболее распространенных реализаций сетевых файловых служб: NFS и FTP в стеке TCP/IP. SMB в Microsoft Windows. NCP в операционной системе Novell NetWare.

Вопросы.

1. Что стандартизирует модель OSI?
2. Можно ли представить еще один вариант модели взаимодействия открытых систем с другим количеством уровней, например 8 или 5?
3. Ниже перечислены оригинальные (англоязычные) названия семи уровней модели OSI. Отметьте, какие из названий уровней не соответствуют стандарту?

- physical layer
- data link layer
- network layer
- transport layer
- seances layer
- presentation layer
- application layer

4. Какие из приведенных утверждений вы считаете ошибочными:

- протокол — это программный модуль, решающий задачу взаимодействия систем;
- протокол — это формализованное описание правил взаимодействия, включающих последовательность обмена сообщениями и их форматы;
- термины «интерфейс» и «протокол», в сущности, являются синонимами.

5. На каком уровне модели OSI работает прикладная программа?

6. Как вы считаете, протоколы транспортного уровня устанавливаются только на конечных узлах, только на промежуточном коммуникационном оборудовании (маршрутизаторах) или и там, и там?

7. На каком уровне модели OSI работают сетевые службы?

8. Ниже перечислены некоторые сетевые устройства:

- маршрутизатор;
- коммутатор;
- мост;
- повторитель;
- сетевой адаптер;
- концентратор.

В каком из этих устройств реализуются функции физического уровня модели OSI? Канального уровня? Сетевого уровня?

9. Какое название традиционно используется для единицы передаваемых данных на каждом из уровней? Заполните таблицу.

	Пакет	Сообщение	Кадр	Поток	Сегмент
Канальный уровень					
Сетевой уровень					
Транспортный уровень					
Сеансовый уровень					
Уровень представления					
Прикладной уровень					