

1. Этапы развития технологий построения ТКС

В современных компьютеризированных средствах связи практически всю «функциональную» начинку определяет записанная в память программа, управляющая работой микропроцессора. При этом практически невозможно сопоставить отдельные функции отдельным электронным элементам.

Последовательность команд, выполняемых компьютером по заданной программе, обычно представляют в виде алгоритма, изображение которого имеет форму упорядоченных по вертикали функциональных модулей. Подобное изображение действительно напоминает здание и позволяет для потерявших наглядность процессов преобразования информации в современных средствах связи воспользоваться аналогией строительного понятия «архитектура».

Другим примером использования новых аналогий из области строительства является выражение «*технологии построения ТКС*», в которое вкладывается смысл не строительства стационарных объектов или развертывания полевых средств связи, а выбор определенного взаимосвязанного функционального наполнения элементов сети в виде той или иной «архитектуры».

Конкретные сетевые технологии (технологии построения сети) фактически определяют правила, по которым работает сеть в течение заданного (внешней системой управления) времени, расходуя соответствующие данным технологиям ресурсы и выполняя текущие требования пользователей (абонентов) по связи в определенных (как правило, мешающих) условиях воздействия внешней среды.

Следует отметить, что пользователей, в принципе, не интересует, какие именно технологии реализованы в сети. Пользователям нужен результат деятельности сети в виде услуг по связи, предоставляемых сетью в определенных (мешающих) условиях и за определенную оплату ресурсов (или в обмен на выделяемые ресурсы). Но взаимосвязь качества услуг, допустимых внешних условий и потребляемых ресурсов (описываемых внешними функциональными характеристиками), как раз, и зависит от используемых в сети технологий (описываемых внутренними функциональными характеристиками).

Результатом слияния отраслей обработки и обмена информацией явилось появление информационных сетей, реализующих все множество информационных процессов обработки и передачи информации.

Информационная сеть (ИС) – это сложная распределенная в пространстве техническая система, представляющая собой функционально связанную совокупность программно-технических средств обработки и обмена информацией и состоящая из территориально распределенных информационных узлов (подсистем обработки информации) и каналов передачи информации, соединяющих данные узлы.

Обобщенно функциональную архитектуру ИС можно представить в виде трехуровневой концептуальной модели.

Первый уровень (внутренний) описывает функции и правила взаимосвязи при передаче различных видов информации между территориально удаленными абонентскими системами через физические каналы связи (передачи) и реализуется **транспортной сетью** (ранее подобные функции выполняла первичная сеть связи).

Второй уровень (промежуточный) описывает функции и правила обмена информацией в интересах взаимосвязи прикладных процессов различных абонентских систем и реализуется **телекоммуникационной сетью**, представляющей собой единую

инфраструктуру для обмена различными видами информации в интересах пользователей информационной сети (ранее подобные функции выполняли различные вторичные сети связи).

Третий уровень (внешний) образуется совокупностью прикладных процессов, размещенных в территориально удаленных абонентских системах, являющихся потребителями информации и выполняющих ее содержательную обработку. Третий уровень, дополняя первый и второй указанными функциями обработки информации, образует внешний облик *информационной сети*.

2. Эталонная модель взаимосвязи открытых систем

Информационный процесс взаимодействия пользователей в ИС начинается и заканчивается вне самой сети и включает ряд вложенных этапов, одним из которых является реализация телекоммуникационной сетью процесса *взаимосвязи* в интересах *взаимодействия* информационных процессов, реализующих содержательную обработку формализованных сообщений при решении той или иной прикладной задачи.

Указанный процесс взаимосвязи также может быть представлен в виде последовательности разнообразных и, как правило, многократных функциональных преобразований информационных сообщений в различных сетевых элементах из одной цифровой формы в другую и из одного вида физических (электрических) сигналов в другие.

Для различных телекоммуникационных сетей, создававшихся в различное время различными производителями, группирование указанных функциональных преобразований различно. Отличается также количество выделяемых этапов и функций процесса взаимосвязи, зачастую объединяемых в рамках той или иной функциональной архитектуры ИС (ТКС) в отдельные уровни или слои. В настоящее время существует ряд различных архитектур ставших де-факто или де-юре международными открытыми (общепринятыми) стандартами.

Примером наиболее известной и детально проработанной архитектуры является семиуровневая *эталонная модель взаимосвязи открытых систем* (ЭМВОС), предложенная Международной организацией стандартов. Данная архитектура ориентированна на описание реализации только функций взаимосвязи при взаимодействии ИП, выполняющих функции содержательной обработки информации в территориально распределенных узлах ИС (поэтому мы будем называть данную архитектуру архитектурой ТКС, а не архитектурой ИС).

В англоязычном обозначении ЭМВОС иногда подчеркивается принадлежность данной модели *взаимосвязи открытых систем* (ВОС) (*OSI – Open System Interconnection*) к разработкам МОС (*ISO – International Standards Organization*) в виде OSI/ISO.

Следует отметить, что в русскоязычной литературе часто аббревиатура ЭМВОС расшифровывается как эталонная модель «взаимодействия», не «взаимосвязи» открытых систем, что является следствием неточного перевода слова «*Interconnection*».

Основным российским стандартом, который определяет принципы архитектуры взаимосвязи открытых систем является ГОСТ 28906–91 «Системы обработки информации. Взаимосвязь открытых систем. Базовая эталонная модель». Этот стандарт подготовлен методом прямого применения стандартов МОС 7498–84, МОС 7498–84 Доп. 1 и полностью им соответствует. Аналогичные рекомендации

содержатся в восьмом томе Синей книги *CCITT* (*Consultative Committee for International Telegraphy and Telephony*) (МККТТ – Международный консультативный комитет по телефонии и телеграфии), ныне секции стандартизации Международного союза электросвязи (МСЭ) (*ITU – International Telecommunication Union*) – рекомендация X.200.

Понятие «открытости» систем означает взаимное признание и поддержку соответствующих стандартов взаимосвязи и не связано с их конкретной реализацией и с используемыми техническими (программными) средствами.

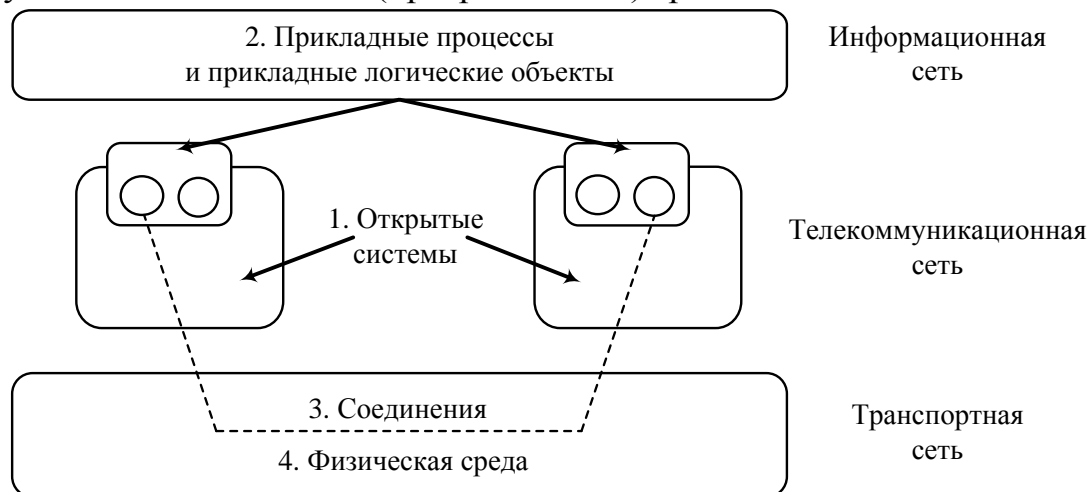


Рис. 1. Основные элементы ЭМВОС и их соотношение с архитектурой ИС

Основу ЭМВОС составляют четыре элемента представленные на рис. 1, согласно которому компоненты прикладных процессов, называемые прикладными логическими объектами (далее для краткости – логические объекты), реализуют процессы взаимосвязи открытых систем по устанавливаемым соединениям через среду ВОС, под которой понимается совокупность взаимодействующих реальных открытых систем вместе с физической средой для ВОС, предназначенной для передачи информации между ними. В качестве физической среды для ВОС обычно выступают цифровые каналы передачи различной физической природы.

Эталонной моделью в рамках среды ВОС наряду с вариантом взаимосвязи с установлением соединения (с использованием постоянных или коммутируемых виртуальных каналов) предусмотрен также вариант взаимосвязи без установления соединения, что соответствует дейтаграммному режиму работы сети с коммутацией пакетов (без использования виртуальных каналов). В целом же большое многообразие и сложность функций взаимосвязи привели к необходимости их иерархического разделения на группы (слои, уровни) в рамках открытой системы и создания многоуровневой архитектуры телекоммуникационных сетей.

Уровневая организация ЭМВОС

Любая многоуровневая организация специализированных систем является заведомо избыточной и не эффективной для конкретных условий применения, но она значительно упрощает построение открытых систем (общего пользования), предназначенных для работы в многообразных условиях и состоящих из множества элементов, согласованно выполняющих функции отдельных уровней, будучи разработанными различными независимыми производителями.

При решении вопроса о том, где должны быть проведены границы между уровнями

и сколько должно быть уровней, разработчики ЭМВОС опирались на определенные **принципы разбиения на уровни**, основными из которых являются следующие:

- число уровней не должно быть слишком большим;
- проводить границу между уровнями следует в том месте, где описание услуг является наиболее простым, количество операций через границу минимально и уже существует подходящий стандартный интерфейс;
- создавать отдельные уровни следует для выполнения специфических функций, отличающихся по реализующим их процессам или техническим решениям;
- следует формировать уровни из легко локализуемых функций с обеспечением возможности их обновления независимо от функций соседних уровней;
- для каждого уровня следует создавать интерфейсы только с вышележащим и нижележащим уровнями;
- возможно образование подуровней в рамках одного уровня в том случае, когда этого требуют специфические виды услуг (должна быть предусмотрена возможность обхода подуровней).

Руководствуясь указанными принципами, в ЭМВОС были выделены семь уровней, перечисляемых обычно сверху вниз:

- 7 уровень – прикладной (*application layer*);
- 6 уровень – представления данных или представительный (*presentation layer*);
- 5 уровень – сеансовый (*session layer*);
- 4 уровень – транспортный (*transport layer*);
- 3 уровень – сетевой (*network layer*);
- 2 уровень – звена данных или канальный (*data link layer*);
- 1 уровень – физический (*physical layer*).

Описание уровней ЭМВОС опирается на ряд формализованных понятий, перечисленных ниже с краткими пояснениями:

Протокол – совокупность правил взаимодействия равноправных логических объектов (различных открытых систем).

Межуровневый интерфейс – совокупность правил взаимодействия логических объектов соседних уровней при предоставлении N -услуг объектам $(N+1)$ -уровня.

Основными функциями всех уровней являются:

- выбор протокола;
- установление и разрыв соединения;
- мультиплексирование и расщепление соединений;
- передача нормальных (обычных) данных;
- передача срочных (внеочередных) данных (с приоритетом);
- управление потоком данных (задержками, скоростью и размером ПБД);
- сегментирование (сборка) или блокирование (деблокирование) данных;
- организация последовательности данных (нумерация);
- защита от ошибок (исправление, обнаружение и сброс и/или повторение);
- маршрутизация (адресование и распределение потоков данных).

Отличия состава данных функций и их количественных параметров для отдельных уровней являются отличительными признаками реальных сетевых технологий, несовпадающих с ЭМВОС в полном объеме.

Часто иерархически организованный набор протоколов различных уровней конкретных сетевых технологий, называется **стеком** протоколов.

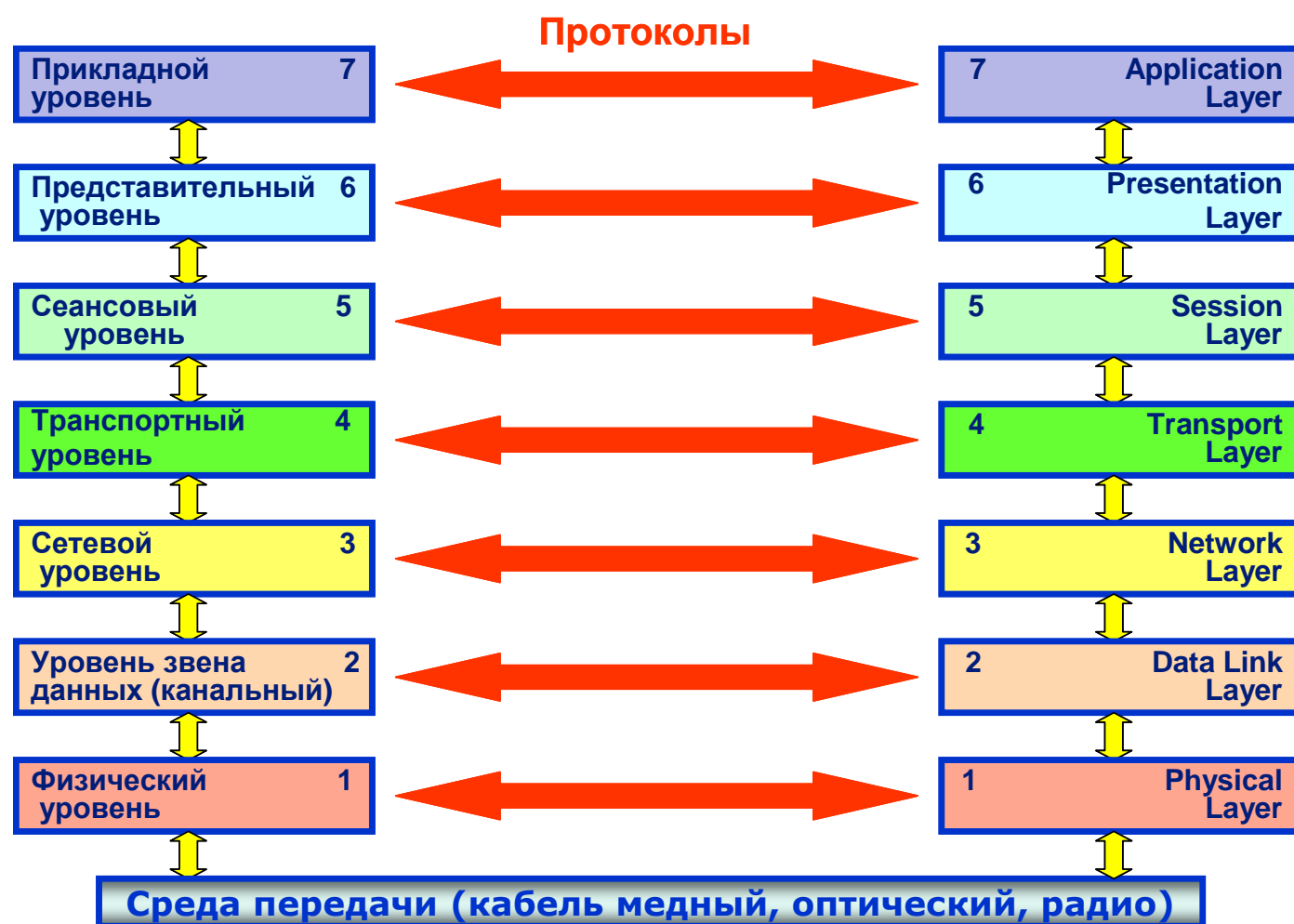
Важным для количественной оценки итогов предоставления N -услуг является задание **параметров качества услуг**, основными из которых являются:

- параметры задержки передачи информации;
- параметры искажений информации;
- параметры потерь информации;
- параметры неправильной адресации;
- параметры защищенности от несанкционированного доступа.

В основном данные параметры являются вероятностными (усредненными или граничными). Определение взаимосвязи данных параметров с параметрами протоколов, доступными ресурсами и мешающими условиями является основной задачей при оценке качества сетевых технологий.

Значения параметров качества услуг нижних уровней влияют на значения параметров качества услуг верхних уровней. В конечном счете, значения параметров качества услуг верхнего уровня определяют качество сервиса (**QoS** – *Quality of Service*), предоставляемого сетью связи в лице конкретных **сетевых служб**.

Эталонная модель взаимосвязи открытых систем (OSI - Open System Interconnection)



Физический уровень

Физический уровень (Physical layer) имеет дело с передачей битов по физическим каналам связи, таким, например, как коаксиальный кабель, витая пара,

оптоволоконный кабель или цифровой территориальный канал. К этому уровню имеют отношение характеристики физических сред передачи данных, такие как полоса пропускания, помехозащищенность, волновое сопротивление и другие. На этом же уровне определяются характеристики электрических сигналов, передающих дискретную информацию, например, крутизна фронтов импульсов, уровни напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи сигналов. Кроме этого, здесь стандартизируются типы разъемов и назначение каждого контакта.

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

Примером протокола физического уровня может служить спецификация 10-Base-T 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.

Канальный уровень

На физическом уровне просто пересылаются биты. При этом не учитывается, что в некоторых сетях, в которых линии связи используются (разделяются) попеременно несколькими парами взаимодействующих компьютеров, физическая среда передачи может быть занята. Поэтому одной из задач канального уровня (Data Link layer) является проверка доступности среды передачи. Другой задачей канального уровня является реализация механизмов обнаружения и коррекции ошибок. Для этого на канальном уровне биты группируются в наборы, называемые кадрами (frames). Канальный уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность бит в начало и конец каждого кадра, для его выделения, а также вычисляет контрольную сумму, обрабатывая все байты кадра определенным способом и добавляя контрольную сумму к кадру. Когда кадр приходит по сети, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы не совпадают, то фиксируется ошибка. Канальный уровень может не только обнаруживать ошибки, но и исправлять их за счет повторной передачи поврежденных кадров. Необходимо отметить, что функция исправления ошибок не является обязательной для канального уровня, поэтому в некоторых протоколах этого уровня она отсутствует, например, в Ethernet и frame relay.

В протоколах канального уровня, используемых в локальных сетях, заложена определенная структура связей между компьютерами и способы их адресации. Хотя канальный уровень и обеспечивает доставку кадра между любыми двумя узлами локальной сети, он это делает только в сети с совершенно определенной топологией связей, именно той топологией, для которой он был разработан. К таким типовым топологиям, поддерживаемым протоколами канального уровня локальных сетей, относятся общая шина, кольцо и звезда, а также структуры, полученные из них с помощью мостов и коммутаторов. Примерами протоколов канального уровня являются протоколы Ethernet, Token Ring, FDDI, 100VG-AnyLAN.

В локальных сетях протоколы канального уровня используются компьютерами,

мостами, коммутаторами и маршрутизаторами. В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

В глобальных сетях, которые редко обладают регулярной топологией, канальный уровень часто обеспечивает обмен сообщениями только между двумя соседними компьютерами, соединенными индивидуальной линией связи. Примерами протоколов «точка-точка» (как часто называют такие протоколы) могут служить широко распространенные протоколы PPP и LAP-B. В таких случаях для доставки сообщений между конечными узлами через всю сеть используются средства сетевого уровня. Именно так организованы сети X.25. Иногда в глобальных сетях функции канального уровня в чистом виде выделить трудно, так как в одном и том же протоколе они объединяются с функциями сетевого уровня. Примерами такого подхода могут служить протоколы технологий ATM и frame relay.

В целом канальный уровень представляет собой весьма мощный и законченный набор функций по пересылке сообщений между узлами сети. В некоторых случаях протоколы канального уровня оказываются самодостаточными транспортными средствами и могут допускать работу поверх них непосредственно протоколов прикладного уровня или приложений, без привлечения средств сетевого и транспортного уровней. Например, существует реализация протокола управления сетью SNMP непосредственно поверх Ethernet, хотя стандартно этот протокол работает поверх сетевого протокола IP и транспортного протокола UDP. Естественно, что применение такой реализации будет ограниченным - она не подходит для составных сетей разных технологий, например Ethernet и X.25, и даже для такой сети, в которой во всех сегментах применяется Ethernet, но между сегментами существуют петлевидные связи. А вот в двухсегментной сети Ethernet, объединенной мостом, реализация SNMP над канальным уровнем будет вполне работоспособна.

Тем не менее для обеспечения качественной транспортировки сообщений в сетях любых топологий и технологий функций канального уровня оказывается недостаточно, поэтому в модели OSI решение этой задачи возлагается на два следующих уровня - сетевой и транспортный.

Сетевой уровень

Сетевой уровень (Network layer) служит для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут использовать совершенно различные принципы передачи сообщений между конечными узлами и обладать произвольной структурой связей. Функции сетевого уровня достаточно разнообразны. Начнем их рассмотрение на примере объединения локальных сетей.

Протоколы канального уровня локальных сетей обеспечивают доставку данных между любыми узлами только в сети с соответствующей типовой топологией, например топологией иерархической звезды. Это очень жесткое ограничение, которое не позволяет строить сети с развитой структурой, например, сети, объединяющие несколько сетей предприятия в единую сеть, или высоконадежные сети, в которых существуют избыточные связи между узлами. Можно было бы усложнять протоколы канального уровня для поддержания петлевидных избыточных связей, но принцип разделения обязанностей между уровнями приводит к другому решению. Чтобы с одной стороны сохранить простоту процедур передачи данных для типовых топологий, а с другой допустить использование произвольных топологий, вводится дополнительный сетевой уровень.

На сетевом уровне сам термин сеть наделяют специфическим значением. В данном

случае под сетью понимается совокупность компьютеров, соединенных между собой в соответствии с одной из стандартных типовых топологий и использующих для передачи данных один из протоколов канального уровня, определенный для этой топологии.

Внутри сети доставка данных обеспечивается соответствующим канальным уровнем, а вот доставкой данных между сетями занимается сетевой уровень, который и поддерживает возможность правильного выбора маршрута передачи сообщения даже в том случае, когда структура связей между составляющими сетями имеет характер, отличный от принятого в протоколах канального уровня. Сети соединяются между собой специальными устройствами, называемыми маршрутизаторами. Маршрутизатор - это устройство, которое собирает информацию о топологии межсетевых соединений и на ее основании пересылает пакеты сетевого уровня в сеть назначения. Чтобы передать сообщение от отправителя, находящегося в одной сети, получателю, находящемуся в другой сети, нужно совершить некоторое количество транзитных передач между сетями, илихопов (от hop - прыжок), каждый раз выбирая подходящий маршрут. Таким образом, маршрут представляет собой последовательность маршрутизаторов, через которые проходит пакет.

Проблема выбора наилучшего пути называется маршрутизацией, и ее решение является одной из главных задач сетевого уровня. Эта проблема осложняется тем, что самый короткий путь не всегда самый лучший. Часто критерием при выборе маршрута является время передачи данных по этому маршруту; оно зависит от пропускной способности каналов связи и интенсивности трафика, которая может изменяться с течением времени. Некоторые алгоритмы маршрутизации пытаются приспособиться к изменению нагрузки, в то время как другие принимают решения на основе средних показателей за длительное время. Выбор маршрута может осуществляться и по другим критериям, например надежности передачи.

В общем случае функции сетевого уровня шире, чем функции передачи сообщений по связям с нестандартной структурой, которые мы сейчас рассмотрели на примере объединения нескольких локальных сетей. Сетевой уровень решает также задачи согласования разных технологий, упрощения адресации в крупных сетях и создания надежных и гибких барьеров на пути нежелательного трафика между сетями.

Сообщения сетевого уровня принято называть пакетами (packets). При организации доставки пакетов на сетевом уровне используется понятие «номер сети». В этом случае адрес получателя состоит из старшей части - номера сети и младшей - номера узла в этой сети. Все узлы одной сети должны иметь одну и ту же старшую часть адреса, поэтому термину «сеть» на сетевом уровне можно дать и другое, более формальное определение: сеть - это совокупность узлов, сетевой адрес которых содержит один и тот же номер сети.

На сетевом уровне определяются два вида протоколов. Первый вид - сетевые протоколы (routed protocols) - реализуют продвижение пакетов через сеть. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. Однако часто к сетевому уровню относят и другой вид протоколов, называемых протоколами обмена маршрутной информацией или просто протоколами маршрутизации (routing protocols). С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений. Протоколы сетевого уровня реализуются программными модулями операционной системы, а также программными и аппаратными средствами маршрутизаторов.

На сетевом уровне работают протоколы еще одного типа, которые отвечают за отображение адреса узла, используемого на сетевом уровне, в локальный адрес сети. Такие протоколы часто называют протоколами разрешения адресов - Address Resolution Protocol, ARP. Иногда их относят не к сетевому уровню, а к канальному, хотя тонкости классификации не изменяют их сути.

Примерами протоколов сетевого уровня являются протокол межсетевого взаимодействия IP стека TCP/IP и протокол межсетевого обмена пакетами IPX стека Novell.

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 и сеансовому - передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем. 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, а 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, например, 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 изначально очень ненадежны, то целесообразно обратиться к наиболее развитому сервису транспортного уровня, который работает, используя максимум средств для обнаружения и устранения ошибок, - с помощью предварительного установления логического соединения, контроля доставки сообщений по контрольным суммам и циклической нумерации пакетов, установления тайм-аутов доставки и т. п.

Как правило, 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 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. Остальные три верхних уровня решают задачи предоставления прикладных сервисов на основании имеющейся транспортной подсистемы.

Сеансовый уровень

Сеансовый уровень (Session layer) обеспечивает управление диалогом: фиксирует, какая из сторон является активной в настоящий момент, предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, 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, хотя функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

Представительный уровень

Представительный уровень (Presentation layer) имеет дело с формой представления 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 может выполняться шифрование и дешифрование данных, благодаря которому секретность обмена данными обеспечивается сразу для всех прикладных служб. Примером такого протокола является протокол Secure Socket Layer (SSL), 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 Web-страницы, а также организуют свою совместную работу, например, с помощью протокола электронной почты. Единица данных, которой оперирует прикладной уровень, обычно называется сообщением (message).

Существует очень большое разнообразие служб прикладного уровня. Приведем в качестве примера хотя бы несколько наиболее распространенных реализации файловых служб: NCP в операционной системе Novell NetWare, SMB в Microsoft Windows NT, NFS, FTP и TFTP, входящие в стек TCP/IP.