

2. Описание уровней модели ISO/OSI

1. Физический уровень (Physical layer)

- Преобразует данные в оптические или электрические сигналы, соответствующие 0 и 1 бинарного потока, и передает эти биты по физическим каналам связи (коаксиальный кабель, витая пара, оптоволоконный кабель, беспроводные сети) на приемный узел.

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

2. Канальный уровень (Data Link layer)

- Проверка доступности среды передачи.

- Обнаружение и коррекции ошибок. Биты группируются в наборы, называемые кадрами (frames). Проверка контрольной суммы.

- В целом канальный уровень представляет собой весьма мощный и законченный набор функции по пересылке сообщений между узлами сети.

- Примерами протоколов канального уровня являются протоколы Ethernet, Token Ring, FDDI, 100VG-AnyLAN, ARCNet, ATM.

3. Сетевой уровень (Network layer)

- Согласование разных технологий

- упрощение адресации в крупных сетях

- создает надежных и гибких барьеров на пути нежелательного трафика между сетями

- сообщения сетевого уровня принято называть пакетами

- примеры протоколов: IP/IPv4/IPv6, IPx

4. Транспортный уровень (Transport layer)

- Делит потоки информации на достаточно малые фрагменты (пакеты) для передачи их на сетевой уровень

- Гарантированная доставка данных (подтверждение получения пакетов)

- Управление потоком данных (регулируем скорость передачи во избежание потери пакетов)

- Обнаружение ошибок на уровне отправителя и получателя (позволяет принимающей системе обнаружить поврежденные пакеты)

- Примеры: TPC, UDP, SPX.

5. Сеансовый уровень (Session layer)

- обеспечивает управление диалогом: фиксирует, какая из сторон является активной в настоящий момент, представляет средства синхронизации.

- редко реализуется в виде отдельных протоколов.

6. Представительный уровень (Presentation layer)

- имеет дело с формой представления передаваемой по сети информации, не меняя при этом ее содержание

- обеспечивает преобразование данных (кодирование, компрессия и т.п.)

7. Прикладной уровень (Application layer)

- набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделенным ресурсам, таким как файлы, принтеры или гипертекстовые web-страницы, эл. почта и т.д.

- единица данных, которая оперирует прикладной уровень, обычно называется сообщением

- примеры протоколов: HTTP, POP3, SMTP, FTP.

1. Описание протоколов на каждом уровне модели ISO/OSI

Любой протокол модели OSI должен взаимодействовать либо с протоколами своего уровня, либо с протоколами на единицу выше и/или ниже своего уровня. Взаимодействия с протоколами своего уровня называются горизонтальными, а с уровнями на единицу выше или ниже — вертикальными. Любой протокол модели OSI может выполнять только функции своего уровня и не может выполнять функций другого уровня, что не выполняется в протоколах альтернативных моделей.

1. Протоколы прикладного уровня: RDP, HTTP, SMTP, SNMP, POP3, FTP, XMPP, OSCAR, Modbus, SIP, TELNET и другие.

Определения протокола прикладного уровня и уровня представления очень размыты, и принадлежность протокола к тому или иному уровню, например протокола HTTPS зависит от конечного сервиса который предоставляет приложение.

В том случае если протокол, например HTTPS, используется для просмотра некоей простой интернет страницы через браузер - его можно рассматривать как протокол прикладного уровня. В том же случае если протокол HTTPS используется как низкоуровневый протокол для передачи финансовой информации например по протоколу ISO 8583 , то протокол HTTPS будет являться протоколом уровня представления , а протокол ISO 8583 - будет протоколом уровня приложения. То же касается иных протоколов прикладного уровня указанных в данной статье.

1. Протоколы уровня представления: AFP — Apple Filing Protocol, ICA — Independent Computing Architecture, LPP — Lightweight Presentation Protocol, NCP — NetWare Core Protocol, NDR — Network Data Representation, XDR — eXternal Data Representation, X.25 PAD — Packet Assembler/Disassembler Protocol.
2. Протоколы транспортного уровня: ATP (AppleTalk Transaction Protocol), CUDP (Cyclic UDP), DCCP (Datagram Congestion Control Protocol), FCP (Fibre Channel Protocol), IL (IL Protocol), NBF (NetBIOS Frames protocol), NCP (NetWare Core Protocol), SCTP (Stream Control Transmission Protocol), SPX (Sequenced Packet Exchange), SST (Structured Stream Transport), TCP (Transmission Control Protocol), UDP (User Datagram Protocol).
3. Протоколы сетевого уровня маршрутизируют данные от источника к получателю. Работающие на этом уровне устройства (маршрутизаторы) условно называют устройствами третьего уровня (по номеру уровня в модели OSI).

Протоколы сетевого уровня: IP/IPv4/IPv6 (Internet Protocol), IPX (Internetwork Packet Exchange, протокол межсетевого обмена), X.25 (частично этот протокол реализован на уровне 2), CLNP (сетевой протокол без организации соединений), IPsec (Internet Protocol Security).

1. Протоколы канального уровня: ARCnet, ATM, Controller Area Network (CAN), Econet, IEEE 802.3 (Ethernet), Ethernet Automatic Protection Switching (EAPS), Fiber Distributed Data Interface (FDDI), Frame Relay, High-Level Data Link Control (HDLC), IEEE 802.2 (предоставляет функции LLC для подуровня IEEE 802 MAC), Link Access Procedures, D channel (LAPD), IEEE 802.11 wireless LAN, LocalTalk, Multiprotocol Label Switching (MPLS), Point-to-Point Protocol (PPP), Point-to-Point Protocol over Ethernet (PPPoE), Serial Line Internet Protocol (SLIP, устарел), StarLan, Token ring, Unidirectional Link Detection[en] (UDLD), x.25, ARP.

При разработке стеков протоколов на этом уровне решаются задачи помехоустойчивого кодирования. К таким способам кодирования относится код Хемминга, блочное кодирование, код Рида — Соломона.

1. При разработке стеков протоколов на этом уровне решаются задачи синхронизации и линейного кодирования. К таким способам кодирования относится код NRZ, код RZ, MLT-3, PAM5, Манчестер II.

Протоколы физического уровня: IEEE 802.15 (Bluetooth), IRDA, EIA RS-232, EIA-422, EIA-423, RS-449, RS-485, DSL, ISDN, SONET/SDH, 802.11 Wi-Fi, Etherloop, GSM Um radio interface, ITU и ITU-T, TransferJet[en], ARINC 818, G.hn/G.9960.

1. Описание процесса инкапсуляции

Инкапсуляция в компьютерных сетях — это метод построения модульных сетевых протоколов, при котором логически независимые функции сети абстрагируются от нижележащих механизмов путём включения(инкапсулирования) этих механизмов в более высокоуровневые объекты. Например, когда приложению требуется послать сообщение с помощью UDP, то производится последовательность действий:

в первую очередь приложение заполняет специальную структуру данных, в которой указывает информацию о получателе (сетевой протокол, IP-адрес, порт UDP);

передаёт сообщение, его длину и структуру с информацией о получателе обработчику протокола UDP (транспортный уровень);

обработчик UDP формирует датаграмму, в которой в качестве данных выступает сообщение, а в заголовках находится UDP-порт получателя (а также другие данные);

обработчик UDP передаёт сформированную датаграмму обработчику IP (сетевой уровень);

обработчик IP рассматривает переданную UDP датаграмму как данные и предваряет их своим заголовком (в котором, в частности, находится IP-адрес получателя, взятый из той же структуры данных приложения, и номер верхнего протокола);

полученный пакет обработчик IP передаёт на канальный уровень, который опять-таки рассматривает данный пакет как «сырые» данные;

обработчик канального уровня, аналогично предыдущим обработчикам, добавляет в начало свой заголовок (в котором так же указывается номер протокола верхнего уровня, в нашем случае это 0x0800(IP)) и, в большинстве случаев, добавляет конечную контрольную сумму, тем самым формируя кадр;

далее полученный кадр передаётся на физический уровень, который осуществляет преобразование битов в электрические или оптические сигналы и посылает их в среду передачи.

Итак, говоря более простым языком, инкапсуляция — это включение всего пакета одного протокола (то есть его заголовки и данные) внутрь пакета другого протокола в качестве передаваемой информации.

**Контрольные вопросы**

1. Чем обусловлена необходимость разработки эталонной модели OSI/ISO?
2. Дать характеристику каждого из семи уровней модели OSI/ISO?
3. Как соотносятся аппаратные и программные средства на различных уровнях модели?
4. Чем отличается модель, используемая в локально-вычислительной сети от модели OSI/ISO?
5. Применение концепции прозрачности к различным аспектам сети.
6. Темпы технологических изменений обусловили необходимость определения стандартов, к которым новые системы могли бы сходиться, а не стандартизировать процедуры постфактум; это обратная сторона традиционного подхода к разработке стандартов.
7. **физический уровень** имеет дело с передачей битов по физическим каналам связи, таким, 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.

Одной из задач **канального уровня** является проверка доступности среды передачи. Другая задача канального уровня – 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.

**Сетевой уровень** служит для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут использовать различные принципы передачи сообщений между конечными узлами и обладать произвольной структурой связей. Внутри одной сети доставка данных обеспечивается канальным уровнем, а вот доставкой данных между различными 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 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.

Транспортный уровень 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.

Сеансовый уровень 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 сначала. 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 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, в кодах 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 – это набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые Web-страницы, а также организуют совместную работу, например, с помощью протокола электронной почты. Единица данных, которой оперирует прикладной уровень, обычно называется сообщением.

1. Аппаратные и программные средства взаимодействуют друг с другом благодаря уровням и инкапсуляции.
2. В модели OSI мы имеем разные уровни взаимодействия систем, а ЛВС на одном уровне.
3. Концепция прозрачности применима к различным аспектам сети. Например, прозрачность расположения означает, что от пользователя не требуется знать местонахождение программных и аппаратных ресурсов, таких как процессоры, принтеры, файлы и базы данных. Имя ресурса не должно включать информацию о месте его расположения, поэтому имена типа mashine1:prog.c или \\ftp\_serv\pub прозрачными не являются. Аналогично, прозрачность перемещения означает, что ресурсы могут свободно перемещаться из одного компьютера в другой без изменения имен. Еще одним из возможных аспектов прозрачности является прозрачность параллелизма, которая заключается в том, что процесс распараллеливания вычислений происходит автоматически, без участия программиста, при этом система сама распределяет параллельные ветви приложения по процессорам и компьютерам сети. В настоящее время нельзя сказать, что свойство прозрачности в полной мере присуще многим вычислительным сетям, это скорее цель, к которой стремятся разработчики современных сетей.