



# Модель OSI

общая характеристика модели

# Модель OSI

- Модель взаимодействия открытых систем (Open System Interconnection, OSI) создана в 80-е годы 20 века международной организацией по стандартизации **ISO** при поддержке других организаций по стандартизации.

# Модель OSI

- Описывает системные средства взаимодействия, реализуемые операционной системой, системными утилитами, системными аппаратными средствами.
- Очень быстро стала одной из основных моделей, описывающих процесс передачи данных между компьютерами.

# Уровни модели OSI



# 1) Физический уровень (Physical Layer)

Протоколы физического уровня обеспечивают непосредственный доступ к среде передачи данных для протоколов канального и последующих уровней.

Данные передаются в виде последовательностей битов (для последовательных протоколов) или групп битов (для параллельных протоколов).

На этом уровне определяются набор сигналов, которыми обмениваются системы, параметры этих сигналов (временные и электрические) и последовательность формирования сигналов при выполнении процедуры передачи данных.

# Функции физического уровня

- устанавливает и разъединяет физические соединения;
- передает последовательность сигналов;
- "прослушивает" в нужных случаях каналы;
- выполняет идентификацию каналов;
- оповещает о появлении неисправностей и отказов.

Кроме того, на данном уровне формулируются требования к электрическим, физическим и механическим характеристикам среды передачи, передающих и соединительных устройств.

## 2) Канальный уровень или уровень звена данных (Data Link Layer)

Обеспечивает передачу данных в среде передачи по физическому носителю. В канале формируется стартовый сигнал передачи данных, организуется начало передачи, производится сама передача, проводится проверка правильности процесса, производится отключение канала при сбоях и восстановление после ликвидации неисправности, формирование сигнала на окончание передачи и перевода канала в ждущий режим.

# Функции канального уровня

- организация (установление, управление, расторжение) канальных соединений и идентификация их портов;
- передача блоков данных;
- обнаружение и исправление ошибок;
- управление потоками данных;
- обеспечение прозрачности логических каналов (передачи по ним данных, закодированных любым способом).



# Функции канального уровня

На канальном уровне данные передаются в виде блоков, которые называются кадрами.

Тип используемой среды передачи и её топология во многом определяют вид кадра протокола транспортного уровня, который должен быть использован.

При использовании топологии "общая шина" (Common Bus) и "один-ко-многим" (Point-to-Multipoint) средства протокола канального уровня задают физические адреса, с помощью которых будет производиться обмен данными в среде передачи и процедура доступа к этой среде. Примерами таких протоколов являются протоколы Ethernet (в соответствующей части) и HDLC.

### 3) Сетевой уровень (Network Layer)

Основной задачей является определение пути, который будет использован для доставки пакетов данных при работе протоколов верхних уровней (маршрутизация).

Для того чтобы пакет был доставлен до какого-либо заданного хоста, этому хосту должен быть поставлен в соответствие известный передатчику сетевой адрес. Группы хостов, объединенные по территориальному принципу, образуют сети.

Для упрощения задачи маршрутизации сетевой адрес хоста составляется из двух частей: адреса сети и адреса хоста. Таким образом, задача маршрутизации распадается на две - поиск сети и поиск хоста в этой сети.

# Функции сетевого уровня

- создание сетевых соединений и идентификация их портов;
- обнаружение и исправлений ошибок, возникающих при передаче через коммуникационную сеть;
- управление потоками пакетов;
- организация (упорядочение) последовательностей пакетов;
- маршрутизация и коммутация;
- сегментация и объединение пакетов;
- возврат в исходное состояние;
- выбор видов сервиса.

## 4) Транспортный уровень (Transport Layer)

Предназначен для управления потоками сообщений и сигналов. Этот механизм позволяет надёжно обеспечивать передачу данных по сетям с разнородной структурой.

Управление потоком заключается в обязательном ожидании передатчиком подтверждения приема обусловленного числа сегментов приемником. Количество сегментов, которое передатчик может отправить без подтверждения их получения от приемника, называется окном.

## 4) Транспортный уровень (Transport Layer)

Существует два типа протоколов транспортного уровня - сегментирующие протоколы и дейтаграммные протоколы.

Сегментирующие протоколы транспортного уровня разбивают исходное сообщение на блоки данных транспортного уровня - сегменты. Основной функцией таких протоколов является обеспечение доставки этих сегментов до объекта назначения и восстановление сообщения.

Дейтаграммные протоколы не сегментируют сообщение, они отправляют его одним пакетом вместе с адресной информацией. Пакет данных, который называется "дейтаграмма" (Datagram), маршрутизируется в сетях с переключением адресов или передается по локальной сети прикладной программе или пользователю.

## 4) Транспортный уровень (Transport Layer)

Может выполняться также согласование сетевых уровней различных несовместимых сетей через специальные шлюзы.

Рассматриваемый уровень определяет адресацию абонентских систем и административных систем. Главной задачей транспортного уровня является использование виртуальных каналов, проложенных между взаимодействующими абонентскими системами и административными системами, для передачи в пакетах блоков данных.

# Функции транспортного уровня

- управление передачей блоков данных и обеспечение их целостности;
- обнаружение ошибок, их частичная ликвидация, сообщение о неисправленных ошибках;
- восстановление передачи после отказов и неисправностей;
- укрупнение либо разукрупнение блоков данных;
- предоставление приоритетов при передаче блоков;
- передача подтверждений о переданных блоках данных;
- ликвидация блоков при тупиковых ситуациях в сети.

Кроме этого, транспортный уровень может восстанавливать блоки данных, потерянные на

## **5) Сеансовый уровень или уровень сессии (Session Layer)**

На этом уровне устанавливаются, обслуживаются и прекращаются сессии между представительными объектами приложений (прикладными процессами).

Сеансовый уровень обеспечивает взаимодействие с транспортным уровнем, координирует прием и передачу данных одного сеанса связи, содержит функции управления паролями, подсчета платы за использование ресурсов сети и т.д.



## **5) Сеансовый уровень или уровень сессии (Session Layer)**

В качестве примера можно рассмотреть протокол RPC (Remote Procedure Call). Он предназначен для отображения результатов выполнения процедуры на удаленном хосте. В процессе выполнения этой процедуры между приложениями устанавливается сеансовое соединение. Назначением данного соединения является обслуживание запросов, которые возникают, например, при взаимодействии приложения-сервера с приложением-клиентом.

# Функции сеансового

## уровня

- установление и завершение на сеансовом уровне соединения между партнерами;
- выполнение нормального и срочного обмена данными между прикладными процессами;
- синхронизация работы сеансовых соединений;
- извещение прикладных процессов об исключительных ситуациях;
- установление в прикладном процессе меток, позволяющих после отказа либо ошибки восстановить его выполнение от ближайшей метки;
- прерывание в нужных случаях прикладного процесса и его корректное возобновление;
- прекращение сеанса без потери данных;
- передачу особых сообщений о ходе проведения сеанса.

## 6) Уровень представления (Presentation Layer)

На этом уровне информация преобразуется к виду, в каком это требуется для выполнения прикладных процессов.

Уровень представления обеспечивает кодирование данных, выдаваемых прикладными процессами, и интерпретацию передаваемых данных.

Например, выполняются алгоритмы преобразования формата представления данных для печати - ASCII или КОИ-8. Если для визуализации используется дисплей, то данные по заданному алгоритму формируются в виде страницы, которая выводится на экран.

# Функции уровня представления

- выбор образа представлений из возможных вариантов;
- изменение образа представления в заданный виртуальный образ;
- преобразование синтаксиса данных (кодов, символов) в стандартный;
- определение формата данных.

## **7) Уровень приложения, или прикладной уровень (Application Layer)**

Протоколы уровня предназначены для обеспечения доступа к ресурсам сети и программам-приложениям пользователя. На данном уровне определяется интерфейс с коммуникационной частью приложений. В качестве примера протоколов прикладного уровня можно привести протокол Telnet, который обеспечивает доступ пользователя к «хосту» (главному вычислительному устройству, одному из основных элементов в многомашинной системе или любому устройству, подключенному к сети и использующему протоколы TCP/IP) в режиме удаленного терминала.

# Функции прикладного уровня

- описание форм и методов взаимодействия прикладных процессов;
- выполнение различных видов работ (управление заданиями, передача файлов, управление системой и т.д.);
- идентификацию пользователей (партнеров взаимодействия) по их паролям, адресам, электронным подписям;
- определение функционирующих абонентов;
- объявление о возможности доступа к новым прикладным процессам;
- определение достаточности имеющихся ресурсов;
- посылку запросов на соединение с другими прикладными процессами;

# Функции прикладного уровня

- подачу заявок представительному уровню на необходимые методы описания информации;
- выбор процедур планируемого диалога процессов;
- управление данными, которыми обмениваются прикладные процессы;
- синхронизацию взаимодействия прикладных процессов;
- определение качества обслуживания (время доставки блоков данных, допустимой частоты ошибок и т.д.);
- соглашение об исправлении ошибок и определении достоверности данных;
- согласование ограничений, накладываемых на синтаксис (наборы символов, структура данных).

## **7) Уровень приложения, или прикладной уровень (Application Layer)**

Прикладной уровень часто делится на два подуровня.

Верхний подуровень включает сетевые службы.

Нижний – содержит стандартные сервисные элементы, поддерживающие работу сетевых служб.



# ***Сетезависимые и сетезависимые уровни***

Перечисленные функции всех уровней можно отнести к одной из двух групп: либо к функциям, ориентированным на работу с приложениями вне зависимости от устройства сети, либо к функциям, зависящим от конкретной технической реализации сети.

# ***Сетезависимые и сетезависимые уровни***

Три верхних уровня – прикладной, представительный и сеансовый – ориентированы на приложения и практически *не зависят* от технических особенностей построения сети. На протоколы этих уровней не влияют какие-либо изменения в топологии сети, замена оборудования или переход на другую сетевую технологию.

Три нижних уровня – физический, канальный и сетевой – являются *сетезависимыми*, протоколы этих уровней тесно связаны с технической реализацией сети и используемым коммуникационным оборудованием.

# ***Сетезависимые и сетезависимые уровни***

Транспортный уровень является промежуточным, он скрывает все детали функционирования нижних уровней от верхних.

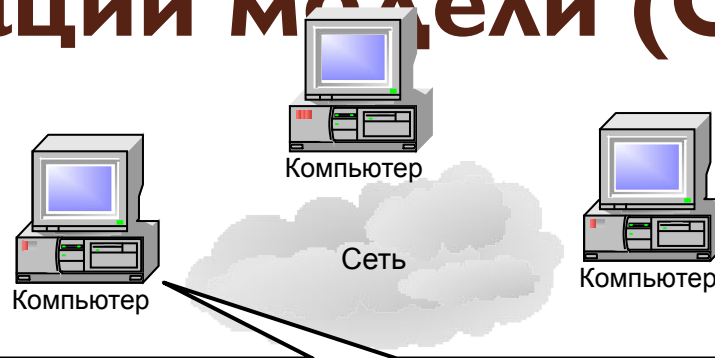
Это позволяет разрабатывать приложения, не зависящие от технических средств непосредственной транспортировки сообщений.



В некоторых случаях семь уровней модели OSI объединяют в три группы:

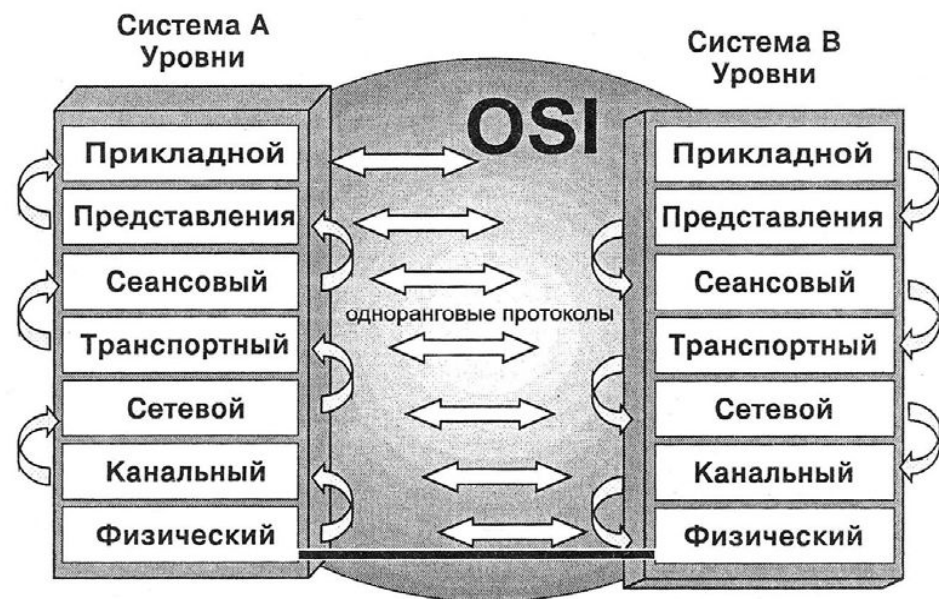
- группа «утилит и приложений» – 1-й уровень;
- группа «преобразования информации» – с 2-го по 5-й уровни;
- группа «физическая» – 6-й и 7-й уровни.

# Схема практической реализации модели (OSI)



# Взаимодействие компьютеров

Компьютер с установленной на нем сетевой операционной системой взаимодействует с другим компьютером с помощью протоколов всех семи уровней



# **Взаимодействие компьютеров**


Осуществляется через различные коммуникационные устройства: концентраторы, модемы, мосты, коммутаторы, маршрутизаторы, мультиплексоры.

# Взаимодействие компьютеров


Коммуникационное устройство в зависимости от типа может работать:

- только на физическом уровне (повторитель);
- на физическом и канальном уровнях (мост);
- на физическом, канальном и сетевом уровнях, иногда захватывая и транспортный уровень (маршрутизатор).






**Эталонная модель взаимосвязи  
(взаимодействия) открытых систем  
(OSI) описывает и реализует  
стандартизованную систему  
взаимодействия в процессах обмена  
информацией и данными между  
прикладными программами и  
системами в вычислительных сетях.**



Для реализации конкретных вычислительных, информационных и коммуникационных процессов и процедур на базе моделей среды открытых систем (OSE) и взаимодействия открытых систем (OSI) создаются разнообразные прикладные службы.

# Прикладные службы, реализуемые на базе принципа открытых систем

Бизнес-моделирование Моделирование проектов	Службы HTI	GUI OSF Motif	Браузеры MSIE Netscape	Ввод/вывод AVI,MIDI,GIF,JPEG MPEG,VOC,WAV,PDF	Kerberos
	Операционная среда приложения				
Разработка		ODA, ODIF, SGML, CGM, TIFF, OLE ...	Бизнес-приложения		Управление
	Службы расширения приложений	Информационные службы	FTP,FTAM,SQL, CLI,DFS,DTP...		
		Службы транзакций	DTP, XA...		
		Службы коммерции	MIME,SMTP,POP3, IMAP4,X.400,EDI,HTTP, CGI,SSL,TAPI,TSAPI,...		
	Распределенные службы	Объекты ORB, COM/DECOM...	Распределение X.500,LDAP DNS,NTP	Средства связи OSF RPC	
Сетевые службы	IPS Ethernet, TR, ATM,X.25,FR,rS232	ISO OSI	SNA		
Базовые службы	Ввод/вывод MCA,EISA,I2O	Хранение SCSI,IDE, RAID,...	OS UNIX,Win95,NT BackOffice	POSIX 1003.7 MMC,DME,DMI SNMP,CMIP GDMO	
С C++,SGML HTML,VRML PERL,VBScript Java,ActiveX OLE Реализация					



Стандартизация интерфейсов обеспечивает полную прозрачность взаимодействия вне зависимости от того, каким образом устроены уровни в конкретных реализациях (службах) модели.



**Благодарю  
за внимание**