

ИЕРАРХИЧЕСКИЕ МОДЕЛИ УПРАВЛЕНИЯ КОМПЬЮТЕРНЫМИ СЕТЯМИ

1.0.1.1

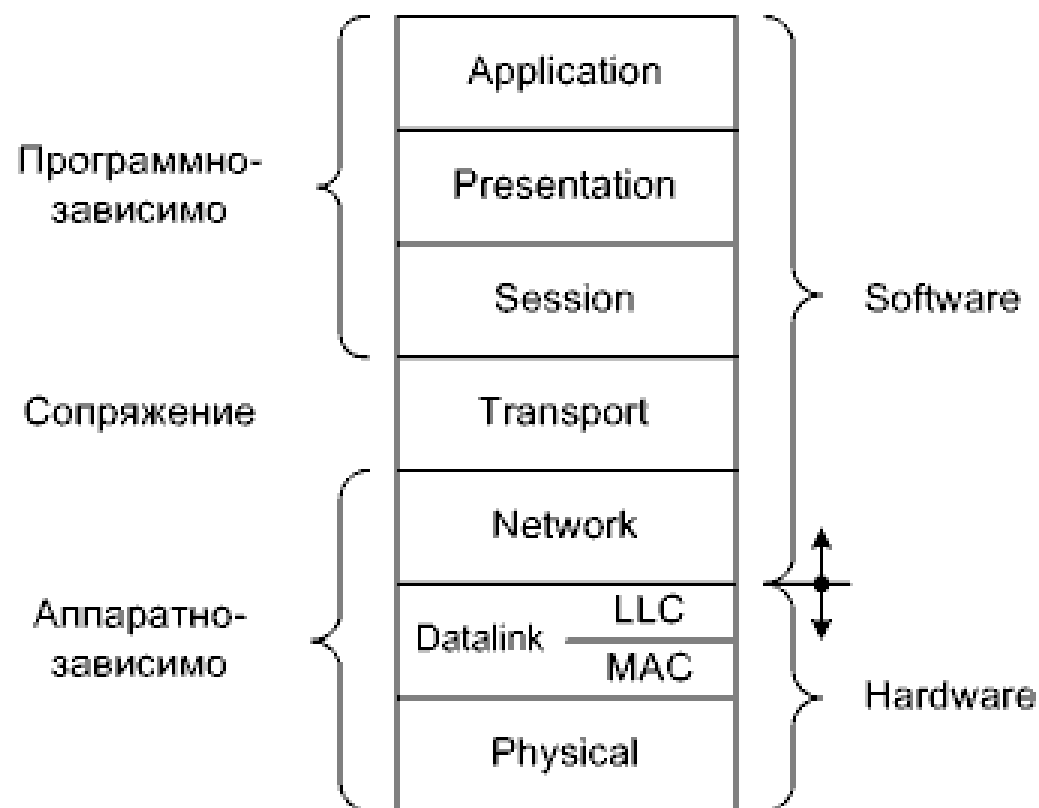
Из всех моделей КС наиболее фундаментальной является *открытая модель взаимодействия систем* -- Open System Interconnection (OSI), разработанная ISO.

1.0.1.2

Что понимают под открытостью модели?

1.0.1.3

Модель включает семь уровней.



На вершине иерархии находится человек, но абонентами КС являются взаимодействующие программы.

1.0.2.1

Как вы думаете, зачем нужен физический уровень?
И какую область очерчивает физический уровень?

1.0.2.2

На *физическом* (physical) уровне формализуют подключение того либо иного сетевого устройства к СрПД.

Соответственно в пространстве физический уровень охватывает «точку» подключения.

1.0.2.3

Какие задачи возложены на физический уровень?

Попробуйте перечислить специфические понятия физического уровня?

1.0.2.4

Специфическими понятиями физического уровня являются:

- среда;
- разъем (физический порт);
- несущая (частота);
- модуляция;
- сигнал.

Фундаментальная задача физического уровня заключается в передаче сигнала.

1.0.2.5

Что такое несущая (carrier)?

Что такое модуляция (modulation)?

Назовите базовые способы модуляции.

1.0.2.6

Чем отличается симметричная электрическая цепь от несимметричной?

1.0.2.7

Подумайте, зачем нужен канальный уровень?

Какую область очерчивает канальный уровень?

Какие новые задачи возникают при переходе от физического уровня к канальному?

1.0.2.8

На *канальном* (datalink) уровне формализуют взаимодействие станций в пределах сегмента.

Любое устройство, способное передавать или принимать сетевой трафик принято называть *станцией* или, по-другому, *узлом* (node). Примерами станций могут быть: ПК, сервер, маршрутизатор и так далее.

Физически любая КС состоит из некоторого, большего или меньшего, количества сегментов. *Сегментом* (segment) называют множество станций, объединенных посредством одной СрПД, то есть «видящих» друг друга непосредственно. Технологически сегменты могут быть самыми разными.

В традиционном понимании СрПД соответствует *физическому соединению* (link). Но многие современные технологии предполагают опциональное или обязательное наличие в СрПД «прозрачных» устройств-посредников, таких как преобразователи сред или коммутаторы.

1.0.2.9

Специфическими понятиями канального уровня являются:

- сегмент сети;
- физическая и логическая топология сегмента;
- пакет (кадр);
- бит- и байт-стаффинг;
- адресация в пределах сегмента;
- канальный код;
- код проверки целостности пакета (кадра);
- алгоритм доступа к моноканалу.

Эти понятия более подробно будут рассмотрены далее в соответствующих разделах.

1.0.2.10

Каждый из уровней модели OSI может быть реализован достаточно сложно, но канальный уровень особенно сложен. Поэтому его разделяют на два подуровня:

1. MAC (Media Access Control) -- контроль доступа к СрПД.
2. LLC (Logical Link Control) -- контроль логического соединения.

На подуровне MAC, более низком, выполняется взаимодействие с физическим уровнем, то есть средозависимые операции, такие как формирование и распознавание пакетов, адресация, канальное кодирование и другие.

На подуровне LLC, более высоком, выполняется взаимодействие с сетевым уровнем, то есть средонезависимые операции, такие как разбиение данных на пакеты, сборка данных из пакетов, определение соответствующей подсистемы сетевого уровня и другие.

1.0.2.11

Подумайте, зачем нужен сетевой уровень?
Какую область очерчивает сетевой уровень?

1.0.2.12

Сетевой уровень позволяет «выйти» за пределы сегмента.

На *сетевом* (network) уровне формализуют построение полноценной КС произвольного масштаба, охватывающей произвольное количество сегментов.

1.0.2.13

Специфическими понятиями сетевого уровня являются:

- пакет (собственно пакет);
- адресация в пределах всей КС;
- маршрутизация.

1.0.2.14

Подумайте, зачем нужен транспортный уровень?
Какую область очерчивает транспортный уровень?

1.0.2.15

Транспортный уровень позволяет перейти от оборудования к программам.

На *транспортном* (transport) уровне формализуют использование программным обеспечением сетевого оборудования, то есть как отдельно взятым программам предоставляется «транспорт».

1.0.2.16

Специфическими понятиями транспортного уровня являются:

- пакет (сегмент сообщения);
- программный порт;
- логическое соединение;
- надежность доставки;
- алгоритм борьбы с заторами в СПД.

1.0.2.17

Подумайте, зачем нужен сеансовый уровень?

1.0.2.18

Сеансовый или *сессионный* (session) уровень позволяет предоставить доступ к транспорту всем программам в многозадачном окружении.

1.0.2.19

Кроме собственно сессии, имеются еще два основных специфических понятия сеансового уровня:

- программный порт;
- алгоритм мультиплексирования программ.

В практических реализациях сеансовый уровень выражен слабо и обычно его совмещают с транспортным.

1.0.2.20

Подумайте, зачем нужен прикладной уровень?

1.0.2.21

Прикладной (application) уровень призван решать конкретные пользовательские задачи с помощью КС.

1.0.2.22

Приведите примеры прикладных задач?

1.0.2.23

Примерами прикладных задач могут служить:

- пересылка файлов между компьютерами;
 - пересылка электронных писем;
 - поддержка удаленных текстовых и графических терминалов, в том числе для администрирования;
 - пересылка мультимедийных документов;
 - обмен «мгновенными» сообщениями;
 - совместная разработка чего-либо;
- и другие.

Плюс, выделяемые особо, как несвойственные традиционным компьютерным сетям, задачи пересылки голоса и видео в реальном времени. При этом, качество обслуживания «возникает» и на всех нижестоящих уровнях.

Специфических понятий прикладного уровня великое множество и они зависят от решаемых задач.

1.0.2.24

Наконец, подумайте, зачем нужен уровень представления?

1.0.2.25

Уровень *представления* (presentation) позволяет адаптировать прикладную информацию в форму, приемлемую для передачи по КС, то есть является прослойкой между программами и транспортом.

1.0.2.26

Назовите основные задачи, решаемые на уровне представления (их две)?

1.0.2.27

Основными задачами уровня представления являются:

- кодирование информации (включая возможное сжатие) с целью обеспечения ее правильной интерпретации в последующем;
- шифрование информации с целью обеспечения ее защиты при пересылке по открытым для прослушивания сетям.

Поскольку обычно уровень представления «привязан» к прикладному уровню, в реализациях эти уровни часто совмещают.

1.0.2.28

Взаимодействие в рамках модели OSI может быть «вертикальным» и «горизонтальным»:

1. *Интерфейс* (interface) -- это правила взаимодействия между пространственно совмещенными соседними уровнями модели OSI.

2. *Протокол* (protocol) -- правила взаимодействия между пространственно разнесенными одинаковыми уровнями модели OSI.

И в том, и в другом случае предполагают определенную абстракцию.

1.0.3.1

Исторически сложились два основных семейства протоколов:

1. TCP/IP.
2. IPX/SPX.

В настоящее время TCP/IP полностью доминирует. IPX/SPX почти не используют, но вкратце будет рассмотрен позже.

1.0.3.2

Application	FTP	SSH	SMTP	IMAP	HTTP	..
Presentation						
Session	TCP			UDP		
Transport						
Network	ICMP	OSPF	BGP	...		
	IP					
	ARP			InARP		
Datalink						
Physical	Ethernet	Wi-Fi	...	ATM	...	

Описания протоколов будут введены в дальнейшем по мере необходимости.

Семейство протоколов TCP/IP

1.0.3.3

Семейство протоколов TCP/IP описано в стандартах RFC (Request For Comments).

1.0.3.4

The screenshot shows the 'Internet Protocol, Version 6 (IPv6) Specification' (RFC 2460) page on the Data Tracker website. The page includes a navigation bar with 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Tools', and 'Help'. The address bar shows the URL 'https://datatracker.ietf.org/doc/rfc2460/'. The page title is 'Internet Protocol, Version 6 (IPv6) Specification RFC 2460'. Below the title, there are tabs for 'Status', 'IESG evaluation record', 'IESG writeups', 'Email expansions', and 'History'. The 'Status' tab is selected. The 'Versions' section shows a timeline of the document's evolution, with '01' (RFC 2460) highlighted. The 'Document' section provides details about the document, including its type (RFC - Draft Standard), authors (Bob Hinden, Dr. Steve E. Deering), and last updated date (2013-03-02). The 'Stream' section shows the document's state (None) and shepherd (None). The 'IESG' section shows the document's state (Draft Standard) and responsible AD (None). The 'Send notices to' section shows (None). The 'Additional resources' section includes links to the mailing list discussion, errata, and other related documents. The 'Status of this Memo' section provides a summary of the document's status, including the network working group, request for comments, obsoletes, category, and the date of the specification.

Internet Protocol, Version 6 (IPv6) Specification
RFC 2460

Status IESG evaluation record IESG writeups Email expansions History

Versions:

01 RFC 2460

draft-ietf-ipv6-spec-v2
rfc2460

Jul 1997 Apr 1998 Dec 1998

Document Type RFC - Draft Standard (December 1998) **Errata**
Obsoleted by RFC 8200
Updated by RFC 5095, RFC 5722, RFC 5871, RFC 6437, RFC 6564, RFC 6935, RFC 6946, RFC 7112, RFC 7045
Obsoletes RFC 1883
Was draft-ietf-ipv6-spec-v2 (Ipnwg WG)
Authors Bob Hinden, Dr. Steve E. Deering
Last updated 2013-03-02
RFC stream Internet Engineering Task Force (IETF)
Formats [txt](#) [html](#) [pdf](#) [htmlized](#) [bibtex](#)
Additional resources [Mailing list discussion](#)

Stream [WG state](#) (None)
Document shepherd (None)

IESG [IESG state](#) RFC 2460 (Draft Standard)
Consensus boilerplate Unknown
Telechat date (None)
Responsible AD (None)
Send notices to (None)

[Email authors](#) [Email WG](#) [IPR](#) [References](#) [Referenced by](#) [Nits](#) [Search lists](#)

RFC 2460

Network Working Group S. Deering
Request for Comments: 2468 Cisco
Obsoletes: 1883 R. Hinden
Category: Standards Track Nokia
December 1998

Internet Protocol, Version 6 (IPv6)
Specification

Status of this Memo

Пример RFC

1.0.3.5

С семейством протоколов TCP/IP связана одноименная модель.

1.0.3.6

OSI Model		TCP/IP Model	
L7. Application		Application	
L6. Presentation			
L5. Session			
L4. Transport		Transport	
L3. Network		Internet	
L2. Datalink		Network Access	
L1. Physical			

Сопоставление модели TCP/IP с моделью OSI

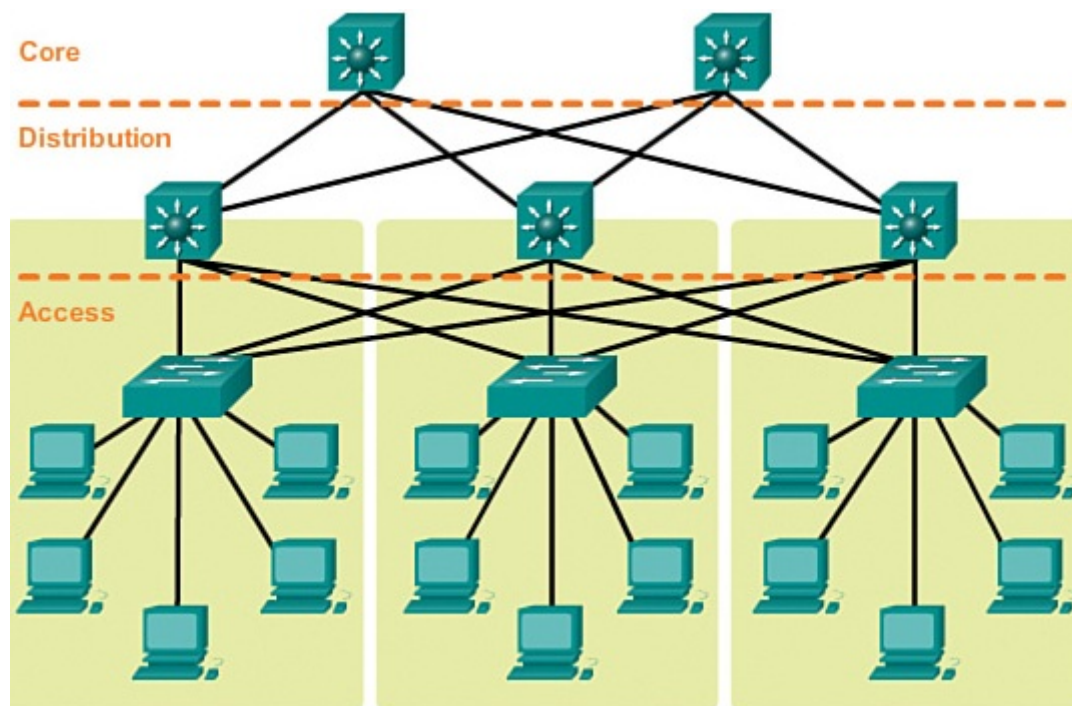
1.0.4.1

Компания Cisco на основе многолетнего опыта проектирования сетей разработала собственную иерархическую сетевую модель (Cisco hierarchical network model), которую рекомендует использовать в корпоративных (enterprise) сетях разного масштаба.

1.0.4.2a

Модель включает три уровня:

1. Access -- доступ.
2. Distribution (иногда aggregation) -- распределение.
3. Core -- ядро.



1.0.4.2b

Уровень доступа предназначен для обеспечения подключений к КС конечных пользователей. Особое внимание здесь уделяют предоставлению пользователям требующихся им ресурсов.

Уровень распределения предназначен для обеспечения взаимодействия в пределах групп пользователей. Особое внимание здесь уделяют резервированию соединений.

Уровень ядра предназначен для обеспечения высокоскоростной связи между относительно удаленными группами пользователей. Особое внимание здесь уделяют характеристикам трафика.

На всех уровнях значительное место отведено разграничению трафика с целями защиты пользователей друг от друга и защиты КС от пользователей.

При этом всем, технологии могут быть различными. Догм нет. Привязка конкретной технологии к тому или иному уровню требует ее понимания. Технологии Cisco будут рассматриваться в дальнейшем.

1.0.5.1

При разговоре о структурной и функциональной организации КС неизбежно возникает вопрос о *сетевой архитектуре* (network architecture).

В классическом представлении под архитектурой (в том числе сетевой) понимают «проекцию» вычислительной структуры на пользователя, то есть как пользователь «видит» оборудование.

1.0.6.1

Примеры сетевых архитектур Cisco:

1. Cisco SecureX.
2. Cisco Borderless Network (в рамках BYOD).
3. Cisco Collaboration.
4. Cisco Unified Data Center.

Конечно, архитектуры от Cisco ориентированы в первую очередь на собственные аппаратные и программные решения.

Просматривается, что сейчас все больший уклон делают в сторону защищенного подключения мобильных пользователей, виртуализации и облачных вычислений. Такие архитектуры позволяют строить так называемые КС с нечетко очерченной границей (borderless).

