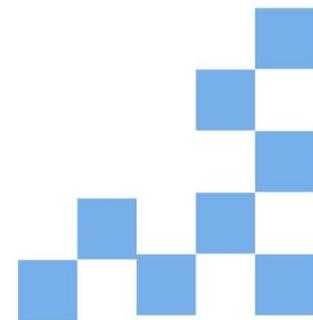
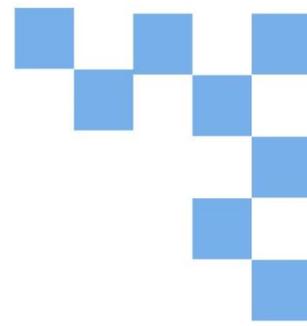


Многоуровневая модель и протоколы.

Владимир Борисович
Лебедев



Программа

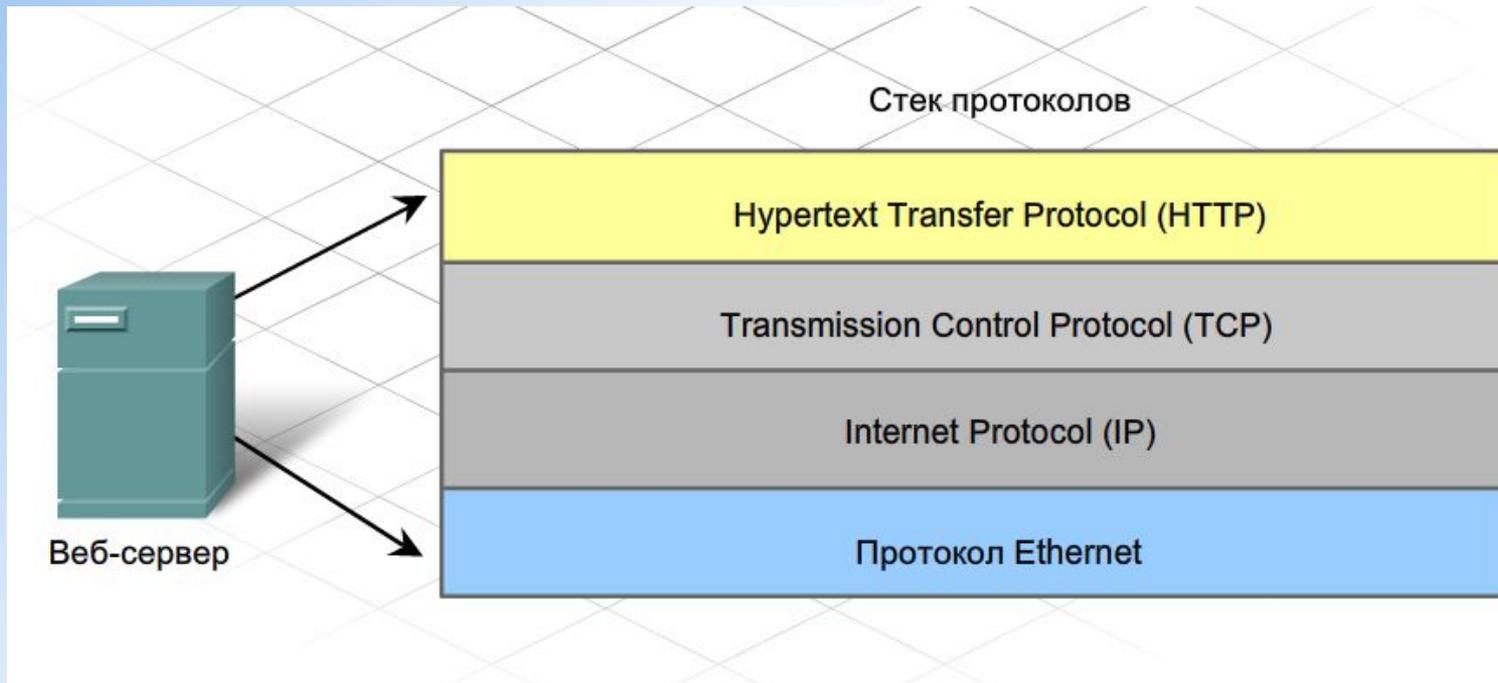


- Взаимодействие протоколов
- Взаимодействие протоколов при отправке и приеме сообщений
- Модель взаимодействия открытых систем

Взаимодействие протоколов

Взаимодействие между протоколами можно представить в виде стека протоколов. Протоколы в стеке представляют собой многоуровневую иерархию, в которой протокол верхнего уровня зависит от сервисов протоколов на более низких уровнях.

На этом графике показан стек протоколов с набором первичных протоколов, необходимых для запуска веб-сервера по сети Ethernet.



Взаимодействие

протоколов

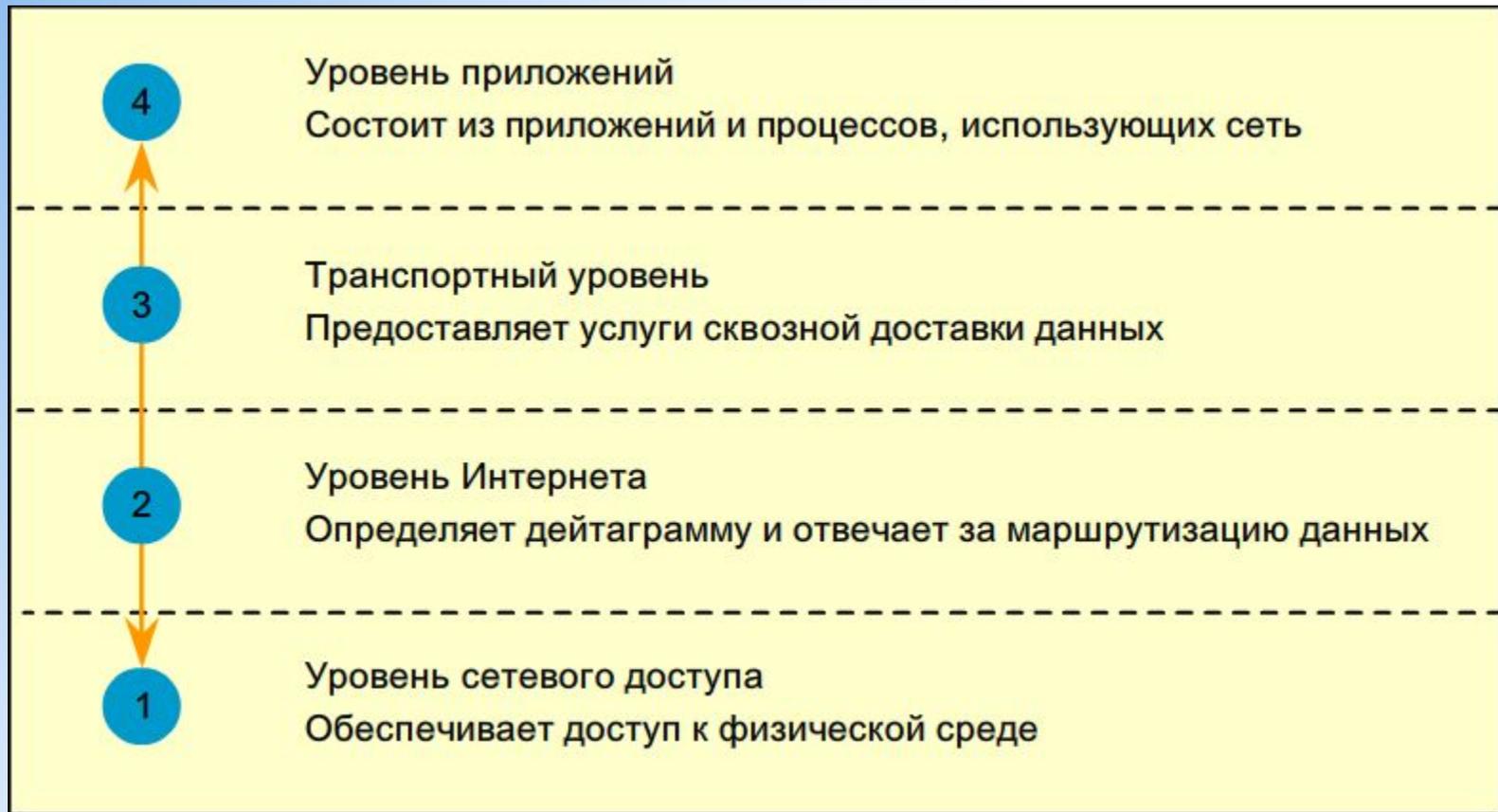
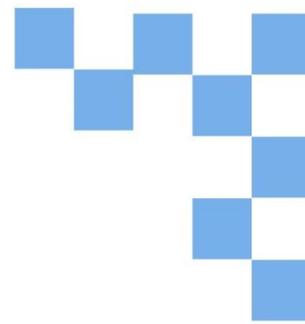
Многоуровневая модель описывает взаимодействие протоколов внутри каждого уровня, а также взаимодействие с верхними и нижними уровнями.

Многоуровневая модель имеет ряд преимуществ:

- она упрощает разработку протоколов, так как протоколы, работающие на определенном уровне, определяют формат обрабатываемых данных и предоставляют интерфейс к верхним и нижним уровням;
- заставляет поставщиков конкурирующих продуктов создавать унифицированные решения;
- исключает возможности изменения технологий или функций одного уровня без учета последствий для верхних и нижних уровней;
- предоставляет общий язык для описания функций сетевого взаимодействия.



Взаимодействие протоколов



Взаимодействие протоколов

при

отправке и приёме сообщения

При отправке сообщения по сети стек протоколов на узле работает от верхнего уровня к нижнему.

При отправке веб-страницы по стеку протоколов веб-сервера данные приложения разбиваются на TCP-сегменты.

Сегмент TCP инкапсулирует протокол HTTP и пользовательские данные веб-страниц в формате HTML и передает их на следующий уровень протоколов, то есть в IP.

Далее этот пакет IP передается протоколу Ethernet, где он инкапсулируется в заголовок кадра и в хвостовик. В каждом заголовке кадра Ethernet указываются исходный и конечный MAC-адреса.

В концевой метке указывается информация для проверки ошибок.

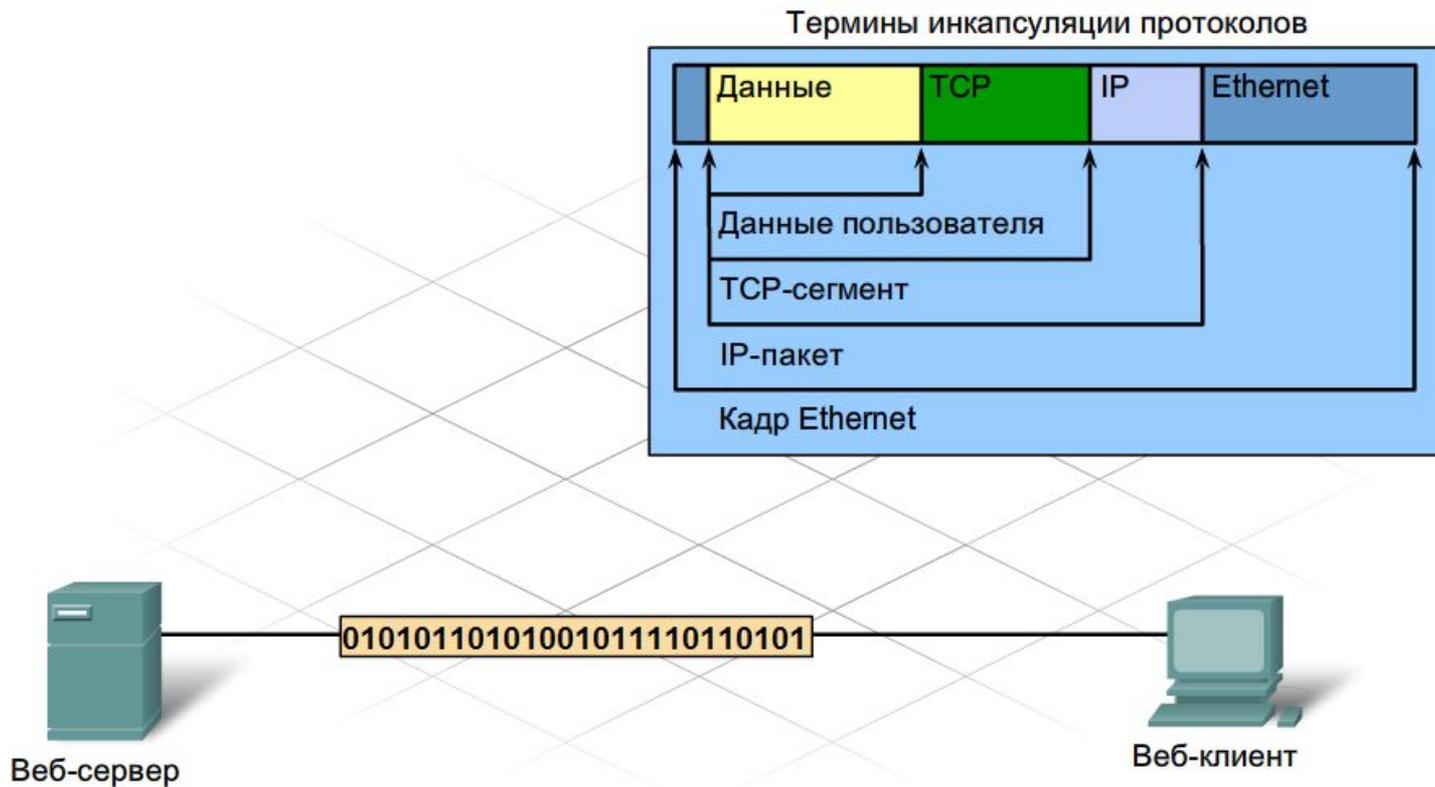


Взаимодействие протоколов

при

отп

соо



Процесс инкапсуляции при отправке сообщения другому узлу.

Взаимодействие протоколов

при

отправке и приёме

сообщения

При приеме сообщений из сети стек протоколов на узле работает с нижнего уровня к верхнему. При приеме веб-страницы начинается деинкапсуляция сообщения клиентом.

Выполняется декодирование битов интерфейсной платой на стороне клиента, а конечный MAC-адрес определяется клиентом как свой собственный.

Кадр передается по стеку протоколов веб-клиента, где заголовок Ethernet (исходный и конечный MAC-адреса) и конечная метка удаляются (деинкапсулируются).

На уровне IP заголовок IP (исходный и конечный IP-адреса) удаляется, и содержимое передается на уровень TCP.

На уровне TCP заголовок TCP (исходный и конечный порты) удаляется, а содержимое пользовательской веб-страницы передается в приложение Интернет-браузера по протоколу HTTP.

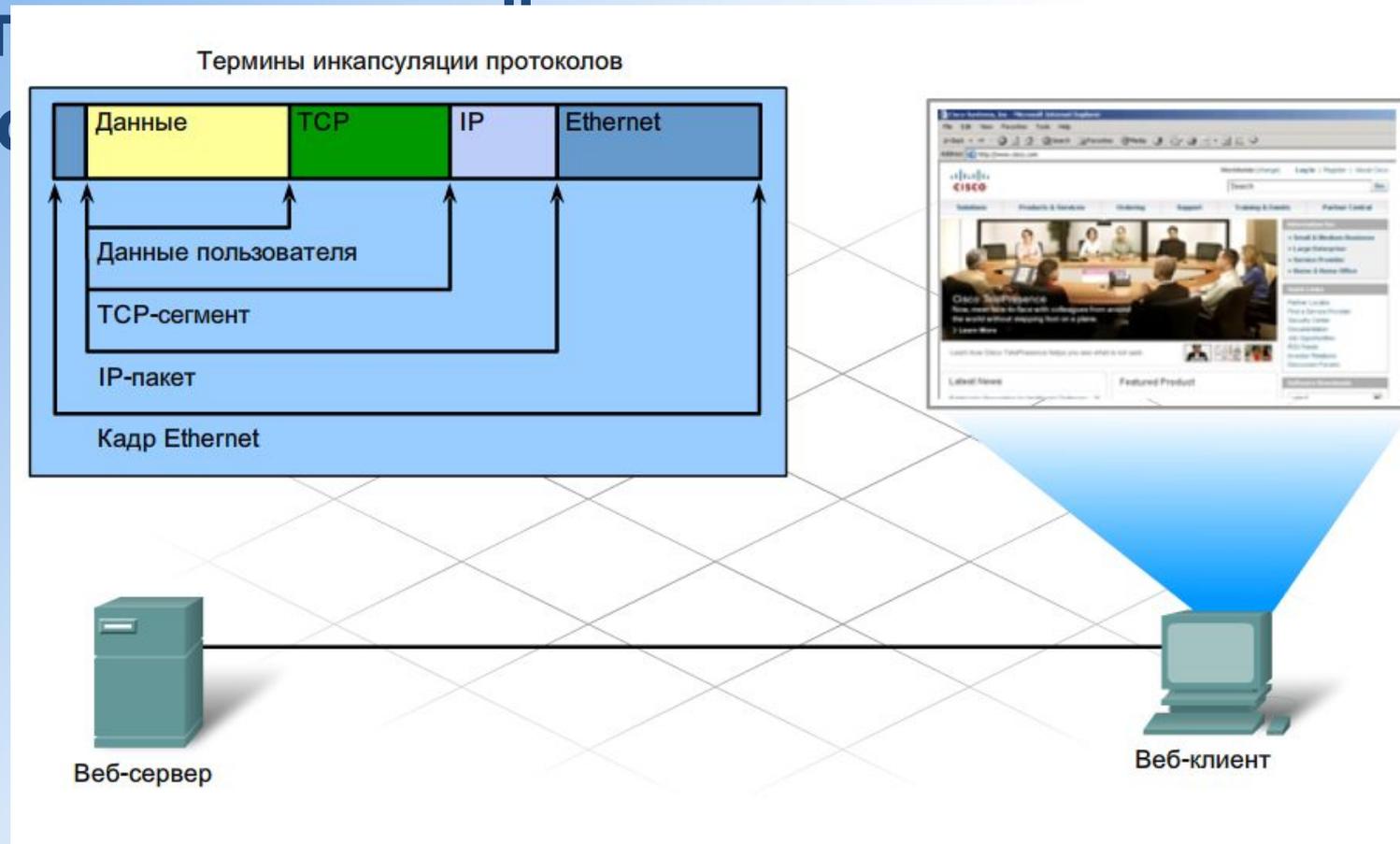


Взаимодействие протоколов

при

отг

соо



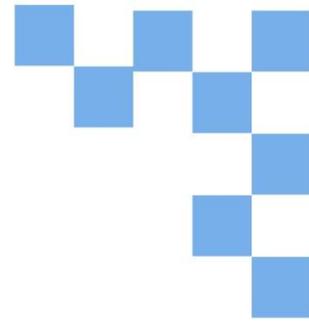
Процесс деинкапсуляции при получении сообщения от другого узла.

Модель взаимодействия открытых систем

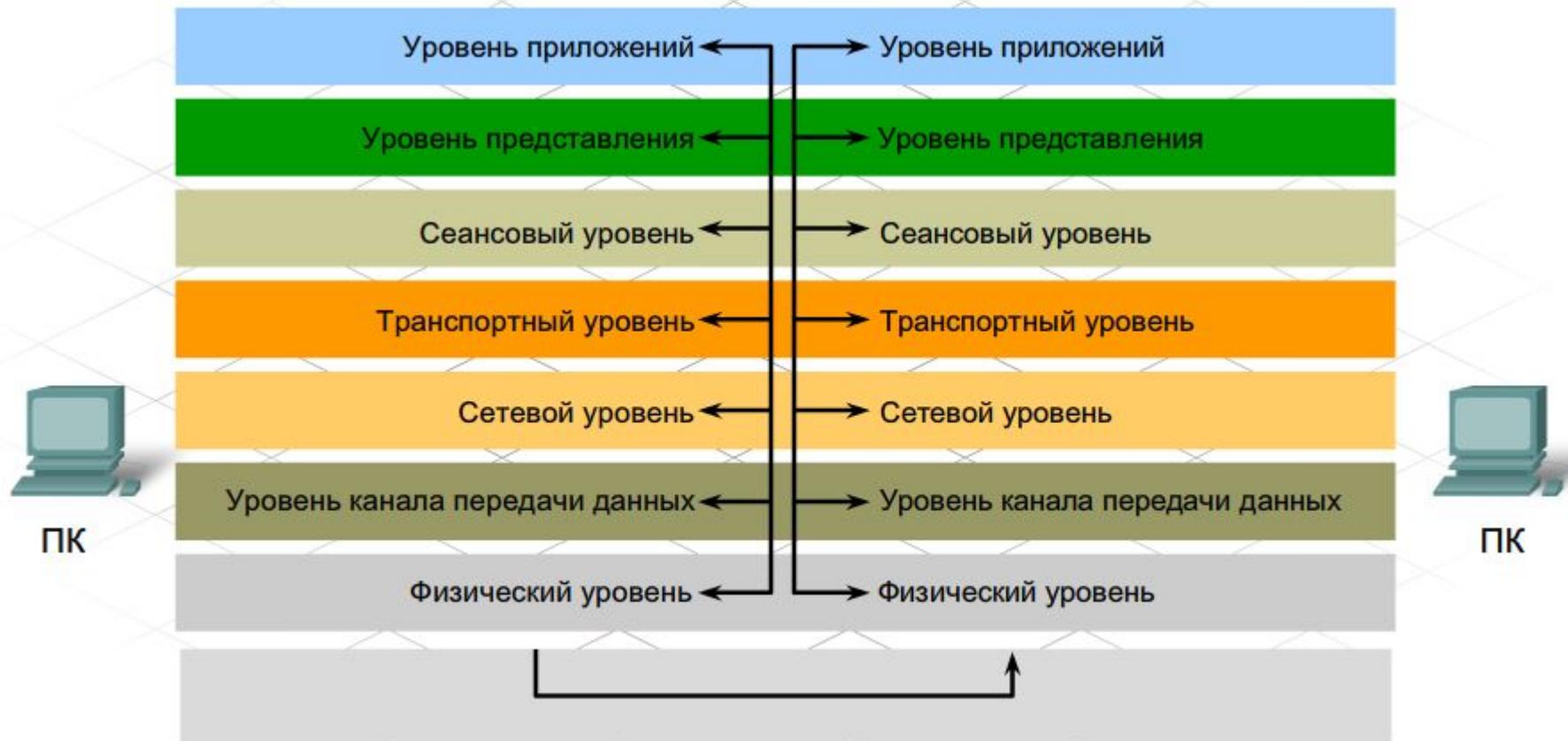
Модель взаимодействия открытых систем была разработана Международной Организацией по Стандартизации (ISO) в 1984 году. Она была создана в качестве базовой архитектуры, которую разработчики использовали для создания протоколов сетевого взаимодействия. На сегодня она считается эталонной моделью межкомпьютерных взаимодействий.

В модели OSI представлены все функции или задачи, ассоциированные с межсетевыми взаимодействиями, а не только с определенными протоколами TCP/IP.

Суть стека протоколов заключается в разделении и организации наиболее значимых функций. Разделение функций обеспечивает независимое функционирование каждого уровня в стеке.



Модель взаимодействия открытых систем



Модель взаимодействия



Уровни	Доступные функции
1. Уровень приложений	<ul style="list-style-type: none">- определяет интерфейсы между прикладными ПО и коммуникационными сетевыми функциями.- предоставляет стандартизированные услуги, такие как передача файлов между системами.
2. Уровень представления	<ul style="list-style-type: none">- осуществляет стандартизацию форматов данных пользователей для использования между различными типами систем.- кодирует и декодирует данные пользователей; шифрует и расшифровывает данные; сжимает и распаковывает данные.
3. Сеансовый уровень	<ul style="list-style-type: none">- управляет сеансами и диалогами пользователей.- поддерживает логические связи между системами.
4. Транспортный уровень	<ul style="list-style-type: none">- управляет сквозной доставкой сообщения через сети.- может обеспечивать надежную и последовательную доставку пакетов с помощью механизмов восстановления после ошибок и управления потоками.
5. Сетевой уровень	<ul style="list-style-type: none">- осуществляет маршрутизацию пакетов в соответствии с уникальными адресами сетевых устройств.
6. Уровень канала передачи данных	<ul style="list-style-type: none">- определяет процедуры для работы с каналом связи.- обнаруживает и исправляет кадры с ошибками передачи.
7. Физический уровень	<ul style="list-style-type: none">- определяет физические средства отправки данных через сетевые устройства.- обеспечивает интерфейс между сетевой средой и устройствами.- определяет оптические, электрические и механические характеристики.

Вопросы&Ответы

Многоуровневая модель и протоколы.

