

Модель OSI

JSP & Servlets
8 уровень, 1 лекция

ОТКРЫТА

Знакомство с OSI

Когда ARPA-сеть только разрабатывалась, очень хотелось сделать ее как можно более умной. Но чем сложнее сеть, тем сложнее ее развивать и поддерживать. В качестве решения было предложено разделить все функции сети на логические слои.

Модель функционирования сети называется сетевая модель базовой эталонной модели взаимодействия открытых систем ISO/OSI. Кратко — **модель OSI** (Open Systems Interconnection).

Модель OSI	
Данные	Прикладной доступ к сетевым службам
Данные	Представления представление и шифрование данных
Данные	Сеансовый управление сеансом связи
Блоки	Транспортный безопасное и надежное соединение точка-точка
Пакеты	Сетевой определение пути и IP (логическая адресация)
Кадры	Канальный MAC и LLC (физическая адресация)
Биты	Физический кабель сигналы бинарная передача данных

Всего в такой модели 7 уровней. Взаимодействие уровней жестко стандартизировано и сведено к минимуму. Нижний уровень понятия не имеет о наличии более высоких уровней и об их устройстве.

Самый нижний уровень умеет просто **посылать биты**. Даже не передавать, а именно посылать. Он понятия не имеет дойдут они или нет. Послал и забыл.

Более высокий уровень оперирует уже группами бит — **кадрами**, и немного знает об физическом устройстве сети, понимает MAC-адреса и тому подобное.

Следующий уровень — пакетный. Он еще более умный и умеет оперировать IP-адресами сети. И так далее.

Зачем это все нужно? Для максимизации гибкости.

Представь, что каждый уровень — это Java-интерфейс и у него может быть несколько различных реализаций. Так и тут. На физическом уровне вы можете пересылать биты по проводу, слать по воздуху (Wi-Fi), отправлять через спутник, а все остальные уровни даже знать об этом ничего не будут. И все будет работать, как и было задумано.

Стек протоколов OSI

Более детально стек протоколов ты можешь изучить [на картинке ниже](#):

Модель					
Уровень (layer)		Тип данных (PDU ^[15])	Функции	Примеры	Оборудование
Host layers	7. Прикладной (application)	Данные	Доступ к сетевым службам	HTTP, FTP, POP3, WebSocket	Хосты (клиенты сети), Межсетевой экран
	6. Представления (presentation)		Представление и шифрование данных	ASCII, EBCDIC, JPEG, MIDI	
	5. Сеансовый (session)		Управление сеансом связи	RPC, PAP, L2TP, gRPC	
	4. Транспортный (transport)	Сегменты (segment) / Датаграммы (datagram)	Прямая связь между конечными пунктами и надёжность	TCP, UDP, SCTP, Порты	
Media ^[16] layers	3. Сетевой (network)	Пакеты (packet)	Определение маршрута и логическая адресация	IPv4, IPv6, IPsec, AppleTalk, ICMP	Маршрутизатор, Сетевой шлюз, Межсетевой экран
	2. Канальный (data link)	Биты (bit)/ Кадры (frame)	Физическая адресация	PPP, IEEE 802.22, Ethernet, DSL, ARP, сетевая карта.	Сетевой мост, Коммутатор, точка доступа
	1. Физический (physical)	Биты (bit)	Работа со средой передачи, сигналами и двоичными данными	USB, RJ («витая пара», коаксиальный, оптоволоконный), радиоканал	Концентратор, Повторитель (сетевое оборудование)

Но если вы не системный администратор, то такая детализация протоколов вам ни к чему. Более интересным может быть изучение стека протоколов TCP (Transmission Control Protocol)/IP (Internet Protocol).

Три верхних уровня в модели OSI, то есть уровень приложения, уровень представления и уровень сеанса, отдельно не различаются в модели TCP/IP, которая имеет только прикладной уровень над транспортным уровнем:

Распределение протоколов по уровням модели OSI

TCP/IP	OSI	
Прикладной	Прикладной	HTTP, SMTP, SNMP, FTP, Telnet, SSH, SCP, SMB, NFS, RTSP, BGP
	Представления	XDR, AFP, TLS, SSL
	Сеансовый	ISO 8327 / CCITT X.225, RPC, NetBIOS, PPTP, L2TP, ASP
Транспортный	Транспортный	TCP, UDP, SCTP, SPX, ATP, DCCP, GRE
Сетевой	Сетевой	IP, ICMP, IGMP, CLNP, OSPF, RIP, IPX, DDP
Канальный	Канальный	Ethernet, Token ring, HDLC, PPP, X.25, Frame relay, ISDN, ATM, SPB, MPLS, ARP
	Физический	электрические провода, радиосвязь, волоконно-оптические провода, инфракрасное излучение

Стек протоколов TCP/IP

Стек протоколов TCP/IP включает в себя четыре уровня:

- **Прикладной уровень** (Application Layer)
- **Транспортный уровень** (Transport Layer)

- **Межсетевой уровень** (Сетевой уровень) (Internet Layer)
- **Канальный уровень** (Network Access Layer)

Протоколы этих уровней полностью реализуют все функциональные возможности модели OSI. На стеке протоколов TCP/IP построено все взаимодействие пользователей в IP-сетях.

Стек протоколов TCP/IP является независимым от физического оборудования благодаря чему, в частности, обеспечивается полностью прозрачное взаимодействие между проводными и беспроводными сетями.

На прикладном уровне (Application layer) работает большинство сетевых приложений.

Прикладной уровень

Для взаимодействия программ существуют высокоуровневые протоколы обмена информацией. Например, браузеры работают по протоколу HTTP, почта отправляется по протоколу SMTP, Телеграм работает по своему собственному зашифрованному протоколу.

Но приватные протоколы нам не очень интересны. Чаще всего ты будешь сталкиваться с массовыми протоколами, такими как ftp-клиент для протокола FTP (передача файлов), SSH (безопасное соединение с удаленной машиной), DNS (преобразование символьных имён в IP-адреса) и многие другие.

Почти все эти протоколы работают поверх протокола TCP, хотя некоторые для ускорения работы работают по UDP (User Datagram Protocol). Но, что важно, у этих протоколов есть порты по умолчанию. Пример:

- 20 FTP на TCP-порт 20 (для передачи данных) и 21 (для управляющих команд)
- 22 – SSH
- 23 – Telnet
- 53 – запросы DNS
- 80 – HTTP
- 443 – HTTPS

Эти порты определены Агентством по выделению имён и уникальных параметров протоколов (IANA).

Есть еще несколько популярных протоколов прикладного уровня: Echo, Finger, Gopher, HTTP, HTTPS, IMAP, IMAPS, IRC, NNTP, NTP, POP3, POPS, QOTD, RTSP, SNMP, SSH, Telnet, XDMCP.

Транспортный уровень

Протоколы транспортного уровня созданы для решения проблемы гарантированной доставки сообщений.

Сообщение (пакет данных) может быть послано и потеряно где-то в сети. В этом случае на плечи транспортного уровня ложится задача мониторинга таких ситуаций и повторной отсылки сообщения, если нужно.

Еще одна важная задача протокола транспортного уровня — это контроль за порядком прихода сообщений. Часто бывает так, что сообщения были посланы в одном порядке, а пришли в другом. И если собрать из таких кусочков изначально большое сообщение, то получится чепуха.

Чтобы такого не было, транспортный уровень или нумерует сообщения, или не отправляет новое, пока не получил подтверждение получения предыдущего. Протоколы автоматической маршрутизации, логически представленные на этом уровне (поскольку работают поверх IP), на самом деле являются частью протоколов сетевого уровня.

Протокол TCP — “гарантированный” транспортный механизм с предварительным установлением соединения, который предоставляет приложению надежный поток данных, дает уверенность в безошибочности получаемых данных, перезапрашивает данные в случае потери и устраняет дублирование данных.

TCP позволяет регулировать нагрузку на сеть, а также уменьшать время ожидания данных при передаче на большие расстояния. Более того, TCP гарантирует, что полученные данные были отправлены точно в такой же последовательности. В этом его главное отличие от UDP.

Протокол UDP — это протокол передачи датаграмм без установления соединения. Также его называют протоколом “ненадежной” передачи, в смысле невозможности удостовериться в доставке сообщения адресату, а также возможного перемешивания пакетов. В приложениях, требующих гарантированной передачи данных, используется протокол TCP.

UDP обычно используется в таких приложениях, как потоковое видео и компьютерные игры, где допускается потеря пакетов, а повторный запрос затруднен или не оправдан, либо в приложениях вида запрос-ответ (например, запросы к DNS), где создание соединения занимает больше ресурсов, чем повторная отправка.

И TCP, и UDP используют для определения протокола верхнего уровня число, называемое портом.

< Предыдущая лекция

Следующая лекция >

− +14 +

Комментарии (2)

популярные

новые

старые

JavaCoder

Введите текст комментария

Джама Уровень 70

18 сентября, 15:26 ⋮

Для лучшего понимания: http://infocisco.ru/network_model_osi.html
<https://javarush.com/groups/posts/2514-vvedenie-v-enterprise-razrabotku>

Ответить

− +2 +

Варвара Уровень 41

13 сентября, 18:59 ⋮

Думаю, хорошо бы было при первом ознакомлении с этими терминами увидеть их расшифровки:
OSI - Open Systems Interconnection
IP - Internet Protocol
TCP - Transmission Control Protocol
UDP - User Datagram Protocol

Ответить

− +3 +

ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ

