

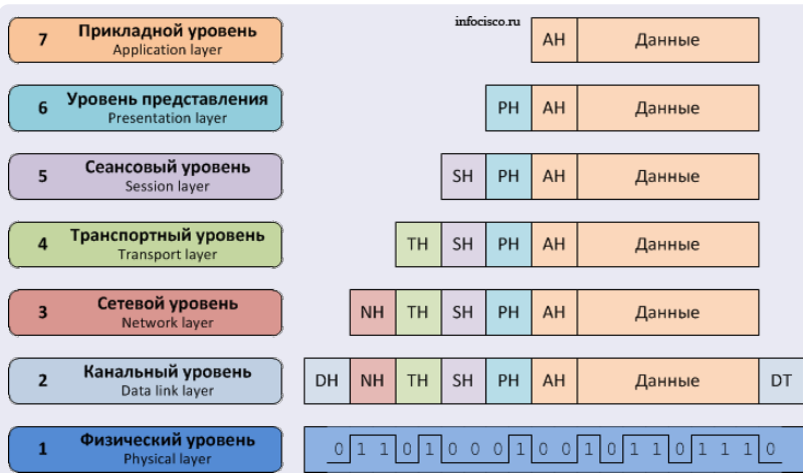
Лекция 6. Сетевое взаимодействие

ИУ8

December 6, 2017

Модель OSI/ISO

	Единица данных	Уровень	Функция	Примеры протоколов
ОС	Поток	Прикладной	Прикладная задача	HTTP, SMTP, DNS, etc.
		Представления	Представление данных, шифрование, etc.	MIME, SSL
		Сеансовый	Взаимодействие хостов (на уровне ОС)	NetBIOS, именов. пайпы
	Сегмент	Транспортный	Соединение конец-в-конец, контроль передачи данных	TCP, UDP
Сеть	Пакет	Сетевой	Логическая адресация и маршрутизация пакетов	IP, ICMP
	Фрейм	Канальный	Физическая адресация	IEEE 802.3, ARP, DHCP
	Бит	Физический	Кодирование и передача данных по физическому каналу	IEEE 802.3



Модель TCP/IP

TCP/IP	OSI Model	Protocols
Application Layer	Application Layer	DNS, DHCP, FTP, HTTPS, IMAP, LDAP, NTP, POP3, RTP, RTSP, SSH, SIP, SMTP, SNMP, Telnet, TFTP
	Presentation Layer	JPEG, MIDI, MPEG, PICT, TIFF
	Session Layer	NetBIOS, NFS, PAP, SCP, SQL, ZIP
Transport Layer	Transport Layer	TCP, UDP
Internet Layer	Network Layer	ICMP, IGMP, IPsec, IPv4, IPv6, IPX, RIP
Link Layer	Data Link Layer	ARP, ATM, CDP, FDDI, Frame Relay, HDLC, MPLS, PPP, STP, Token Ring
	Physical Layer	Bluetooth, Ethernet, DSL, ISDN, 802.11 Wi-Fi

Отвечает за взаимодействие между узлами сети.

IPv4

Адрес узла сети в IPv4 является 32-битным. Например, 192.168.0.10

127.0.0.1 - локальный IP-адрес

IPv6

Адрес узла сети в IPv6 является 128-битным. Например, 2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d

::1 - локальный IP-адрес

Транспортные протоколы

Отвечают за взаимодействие между отдельными процессами узлов сети

Адресация: 16-битное число - порт.

Пара "IP-адрес/порт" определяют конечную точку сетевого взаимодействия

Порты

0 - 1023 – хорошо известные службы. Например:

21 – SSH

80 – HTTP

443 – HTTPS

1024 - 49151 – зарегистрированные IANA службы

49152 - 65535 – любые цели

UDP

- проверяется только контрольная сумма пакета;
- при избытке пакетов в очереди пакеты отбрасываются;
- получатель никак не управляет входящим потоком данных.

Протокол UDP применяется в системах, где скорость важнее надежности. Например, в передаче голоса или видеопотока.

TCP

- протокол "с установлением соединения";
- согласование параметров соединения;
- проверка корректности получаемых данных;
- повторный запрос данных в случае некорректного получения.

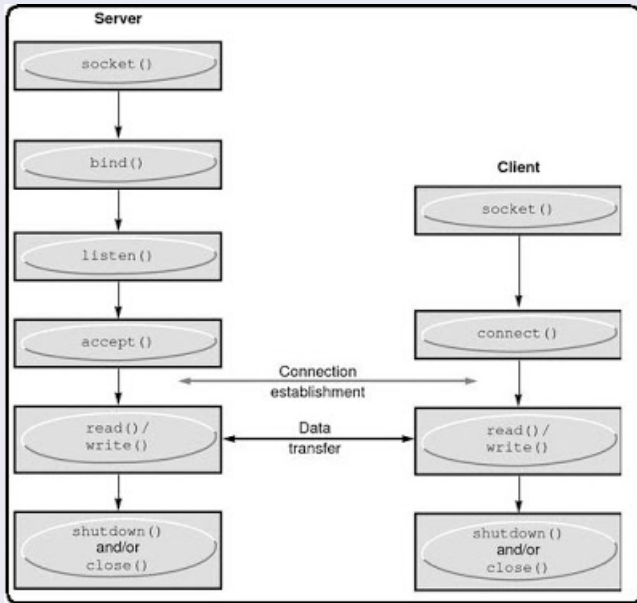
Протокол TCP применяется в системах, где безошибочность данных важнее, даже если приходится жертвовать скоростью.

Сервер

```
1 std::vector<Client> clients;
2
3 for (auto& client : clients) {
4     client << "some info from server";
5     std::string data;
6     client >> data;
7 }
```

Клиент

```
1 Server srv("192.41.1.1", 50123);
2 std::string data;
3 srv >> data;
4 srv << "send info to server";
```



bind - связывает сокет с конкретным адресом

listen - переводит сокет в пассивный режим.

accept - ожидание клиентский соединений.

accept

- функция accept блокирует выполнение потока;
- при подключении возвращает новый сокет;
- этот сокет используется для взаимодействия с конкретным подключенным клиентом.

Самостоятельное изучение

- асинхронное программирование
- boost.asio

Список литературы

- <https://code-live.ru/post/cpp-http-server-over-sockets/>
- Radchuk - Boost.Asio C++ Network Programming Cookbook
- Камер Д., Стивенс Д. - Сети TCP/IP, том 3. Разработка приложений типа клиент/сервер для Linux/POSIX
- <http://habrahabr.ru/post/192284>
- John Torjo - Boost.Asio C++ Network Programming
- Boost.Asio - 4. TCP Socket Programming