

## Семинар 19: Linux networking

30 апреля, 2020

# Протоколы

- ▶ Современные сети строятся на основе *протоколов*
- ▶ Каждый протокол – абстракция над другим протоколом (более низким) для решения какой-то проблемы
- ▶ Поэтому иногда говорят *стек протоколов*

# Модель OSI

- ▶ Разделяет современные протоколы на 7 уровней
- ▶ Physical layer, L1
- ▶ Data link layer, L2
- ▶ Network layer, L3
- ▶ Transport layer, L4
- ▶ Тут немного пропустим :D
- ▶ Application layer, L7

# Модель OSI: physical layer

- ▶ Самый нижний слой
- ▶ Описывает то, как данные будут переданы через физическую среду
- ▶ Протоколы:
- ▶ Bluetooth
- ▶ Ethernet physical layer: Ethernet over twisted pair, Fast Ethernet, Gigabit Ethernet, ...
- ▶ IEEE 802.11

## Модель OSI: link layer

- ▶ Строится поверх L1
- ▶ Обеспечивает обмен данными между узлами в одной сети LAN (local area network) или WAN (wide area network)
- ▶ На этом уровне появляется *канальный адрес (link address)*, например, MAC-адрес
- ▶ IEEE 802.11
- ▶ Ethernet
- ▶ На этом уровне работают, например, сетевые коммутаторы (свитчи)

## Модель OSI: network layer

- ▶ Обеспечивает связь между разными LAN и WAN
- ▶ На этом уровне появляется *сетевой адрес (network address)*, например, IP-адрес
- ▶ Появляется понятие *маршрутизации*
- ▶ Основной протокол – Internet Protocol (IP)

## IP: IP-адрес и маски подсетей

- ▶ IPv4 адрес состоит из 32 бит (для IPv6 – 128)
- ▶ Маска подсети – множество IP-адресов с одинаковым префиксом
- ▶ CIDR-нотация: 10.0.0.0/8 – общий префикс 8 бит
- ▶ Ещё иногда записывают так: 255.0.0.0
- ▶ Некоторые зарезервированные сети: 10.0.0.0/8, 127.0.0.1/8, 192.168.0.0./16

# IP: маршрутизация

- ▶ Теперь не все узлы сети связаны напрямую. Как передавать данные между ними?



# IP: маршрутизация

- ▶ Теперь не все узлы сети связаны напрямую. Как передавать данные между ними?
- ▶ Через другие узлы!
- ▶ Передача между соседними узлами в IP-сети называется *прыжком* или *хопом* (*hop*)
- ▶ Каждый узел сети имеет *таблицу маршрутизации* (иногда даже несколько)
- ▶ Эта таблица содержит маски подсетей и узел, куда нужно переслать данные
- ▶ AS – это система IP-сетей и маршрутизаторов, управляемых одним или несколькими операторами, имеющими единую политику маршрутизации с Интернетом (©Wikipedia)

# IP: маршрутизация

- ▶ AS – это система IP-сетей и маршрутизаторов, управляемых одним или несколькими операторами, имеющими единую политику маршрутизации с Интернетом (©Wikipedia)
- ▶ Точка обмена трафиком – точки обмена трафиком между разными AS :)
- ▶ Одна из самых крупных в Европе и России – MSK-IX

# IPv4: устройство пакета

IPv4 Header Format																																	
Offsets	Octet	0							1							2							3										
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP					ECN		Total Length																
4	32	Identification															Flags		Fragment Offset														
8	64	Time To Live							Protocol							Header Checksum																	
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															
24	192																																
28	224																																
32	256																																

# IP: мини-бонус про TTL

- ▶ TTL – time-to-live
- ▶ Байт, который описывает максимальное количество прыжков в сети
- ▶ Если очередной хост уменьшил TTL до нуля, то пакет просто дропается, а отправителю посылается специальное сообщение по протоколу ICMP (TTL exceeded)

# IP: проблемы

- ▶ Не гарантирует доставку данных (packet loss)
- ▶ Не гарантирует порядок доставки (packet reordering)
- ▶ Не гарантирует, что пакет будет отправлен лишь один раз (packet duplication)
- ▶ Непонятно как реализовывать multitenancy IP протокола – нельзя всем приложениям рассылать все IP-пакеты

# Модель OSI: transport layer

- ▶ Используется для передачи данных между различными приложениями на узлах сети
- ▶ Примеры: TCP, UDP, SCTP

# TCP

- ▶ Вводит понятие *порта* приложения – адрес получателя на IP-узле
- ▶ Обеспечивает надёжную доставку данных (reliable delivery)
- ▶ Обеспечивает порядок доставки и дедупликацию данных
- ▶ Connection-oriented – приложения должны установить полнодуплексное *соединение*
- ▶ Data stream – данные передаются не отдельными пакетами, а непрерывным потоком
- ▶ Также обеспечивает congestion control и flow control

## ТСР: устройство пакета

		TCP segment header																															
Offsets	Octet	0								1								2								3							
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 000		N S	C W R	E C E	U R G	A C K	P S H	R S T	S Y N	F I N	Window Size																
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if data offset > 5. Padded at the end with "0" bytes if necessary.)																															
...	...	...																															



# TCP: 3-way handshake

- ▶ Механизм установки соединения – «рукопожатие»
- ▶ Выполняется в три этапа:

# TCP: 3-way handshake

- ▶ Механизм установки соединения – «рукопожатие»
- ▶ Выполняется в три этапа:
  1. Инициатор соединения (клиент) посылает пакет с флагом **SYN** серверу

# TCP: 3-way handshake

- ▶ Механизм установки соединения – «рукопожатие»
- ▶ Выполняется в три этапа:
  1. Инициатор соединения (клиент) посылает пакет с флагом **SYN** серверу
  2. Сервер посылает пакет с флагами **SYN** и **ACK** клиенту, а также *sequence number*, с которого будут нумероваться все остальные байты

# TCP: 3-way handshake

- ▶ Механизм установки соединения – «рукопожатие»
- ▶ Выполняется в три этапа:
  1. Инициатор соединения (клиент) посылает пакет с флагом **SYN** серверу
  2. Сервер посылает пакет с флагами **SYN** и **ACK** клиенту, а также *sequence number*, с которого будут нумероваться все остальные байты
  3. Клиент посылает **ACK**

# UDP

- ▶ Не даёт никаких гарантий
- ▶ Пересылка осуществляется через пакеты (UDP datagrams)
- ▶ Используется там, где важна latency (например, онлайн игры)
- ▶ Или bandwidth (например, BitTorrent)

## Модель OSI: application layer

- ▶ Протоколы приложений (веб-браузеры, почтовые клиенты, игры)
- ▶ Один из самых известных – HyperText Transfer Protocol (HTTP)
- ▶ Почтовые протоколы – SMTP, POP3
- ▶ Secure Shell (SSH)

Теперь о том, как с этим  
работать в Linux

Спасибо!