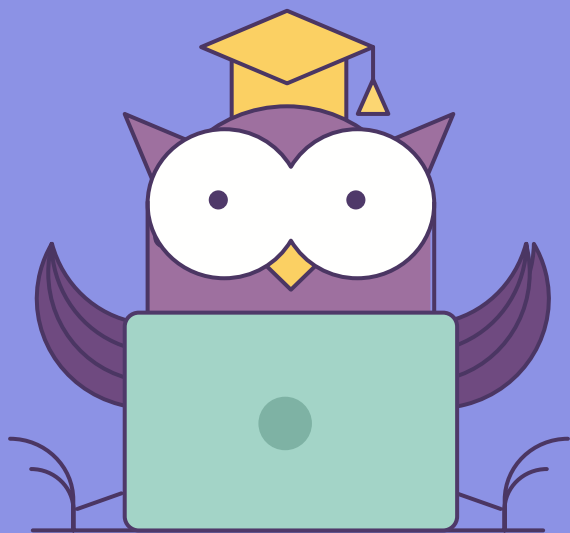




ОНЛАЙН-ОБРАЗОВАНИЕ

Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте  если все хорошо

Архитектура сетей

OSI

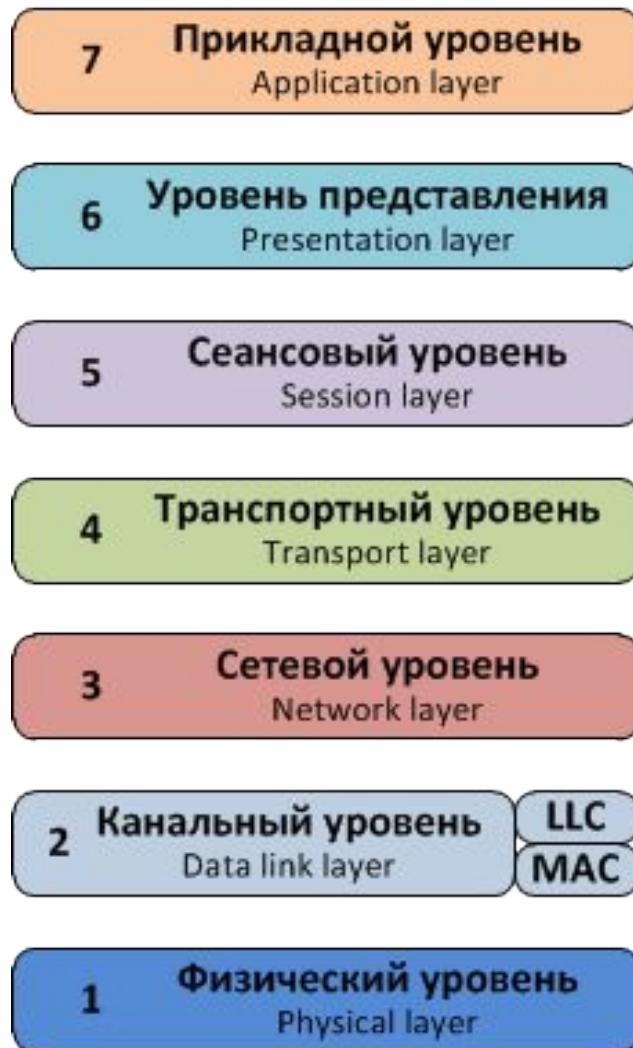
TCP/IP



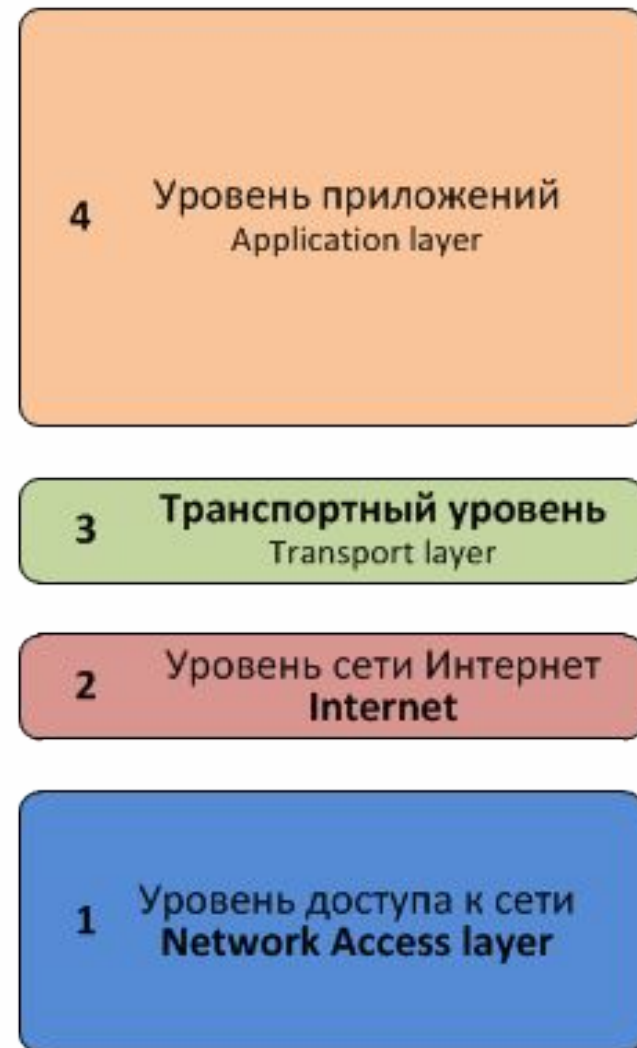
01

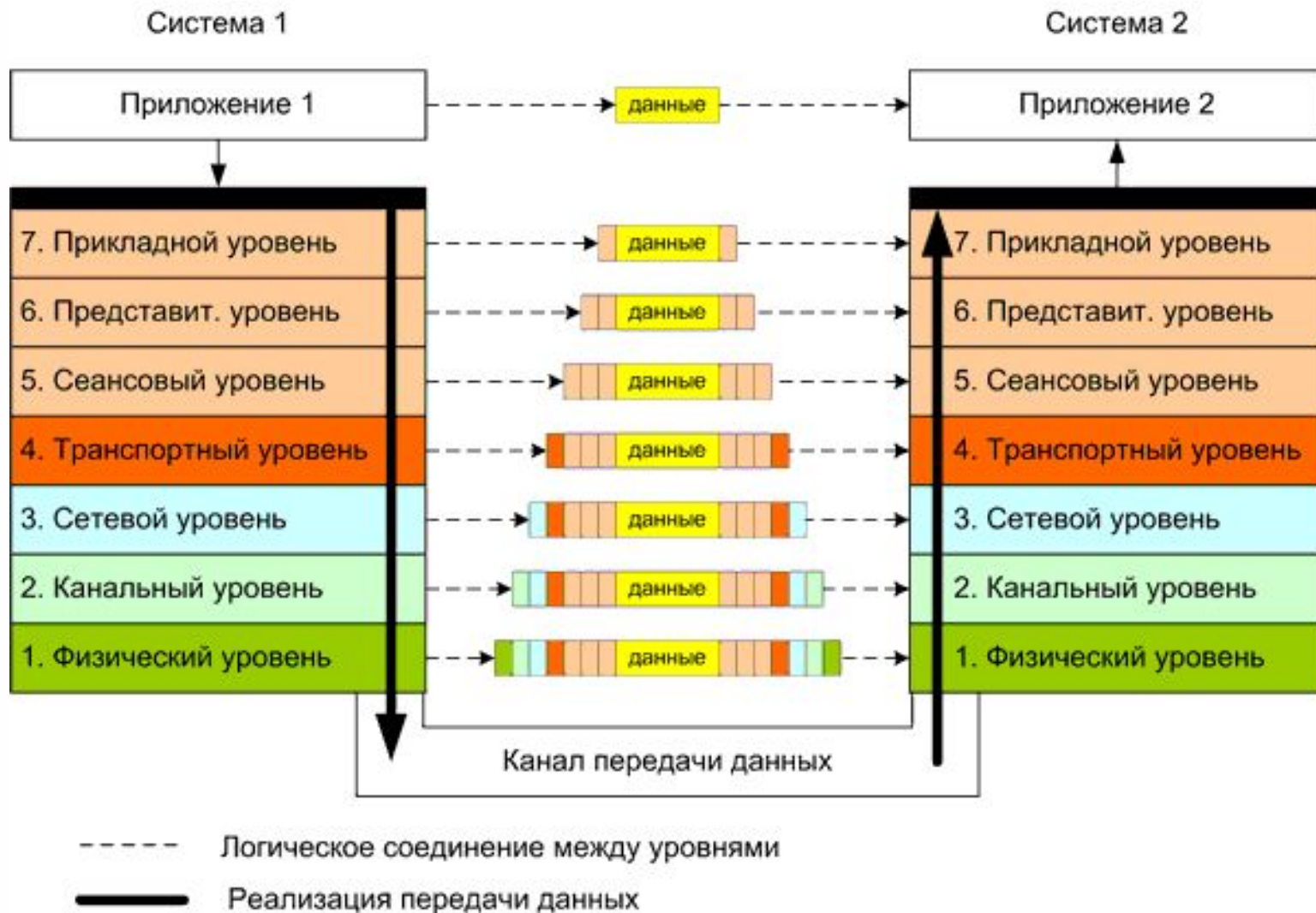
OSI

OSI

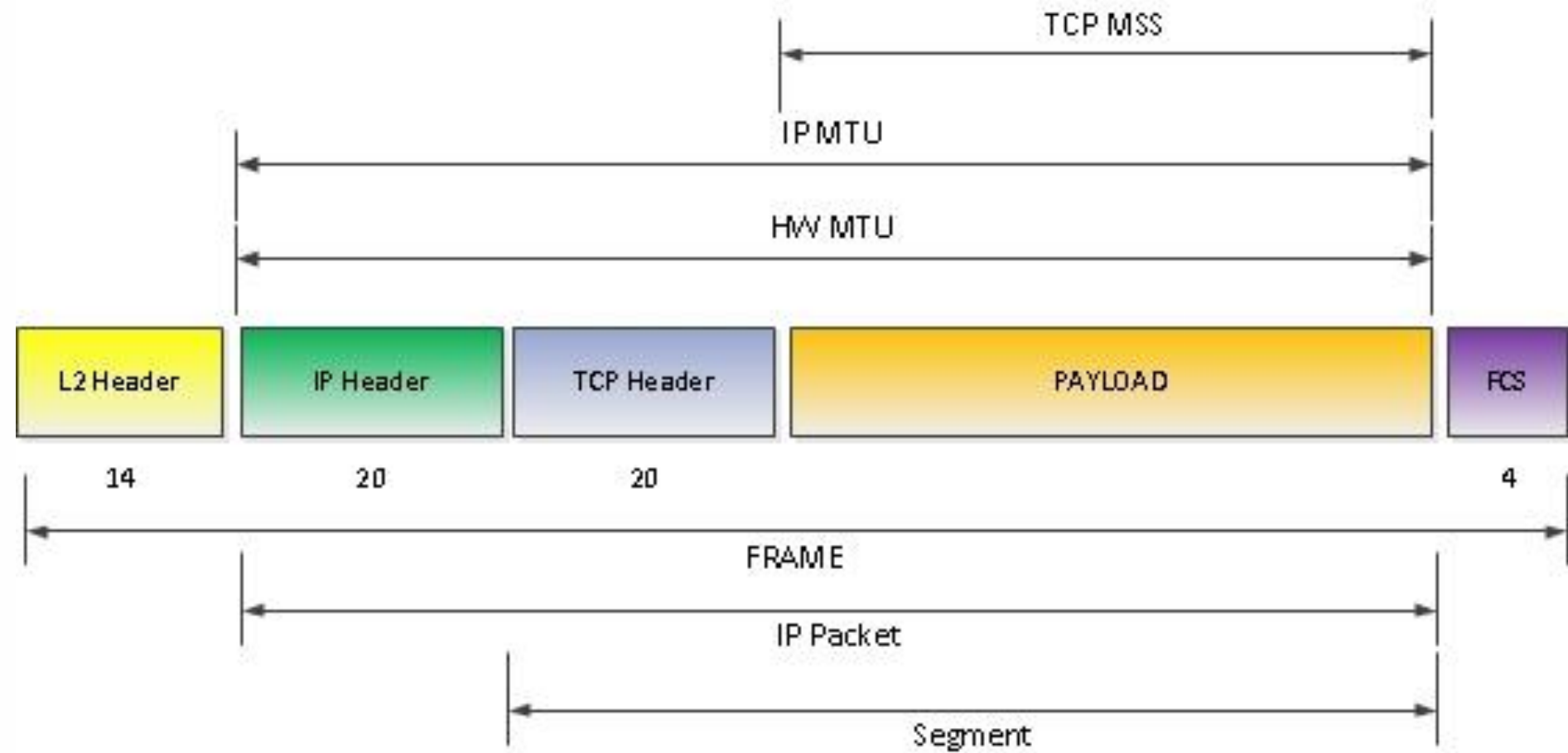


TCP/IP (DOD)

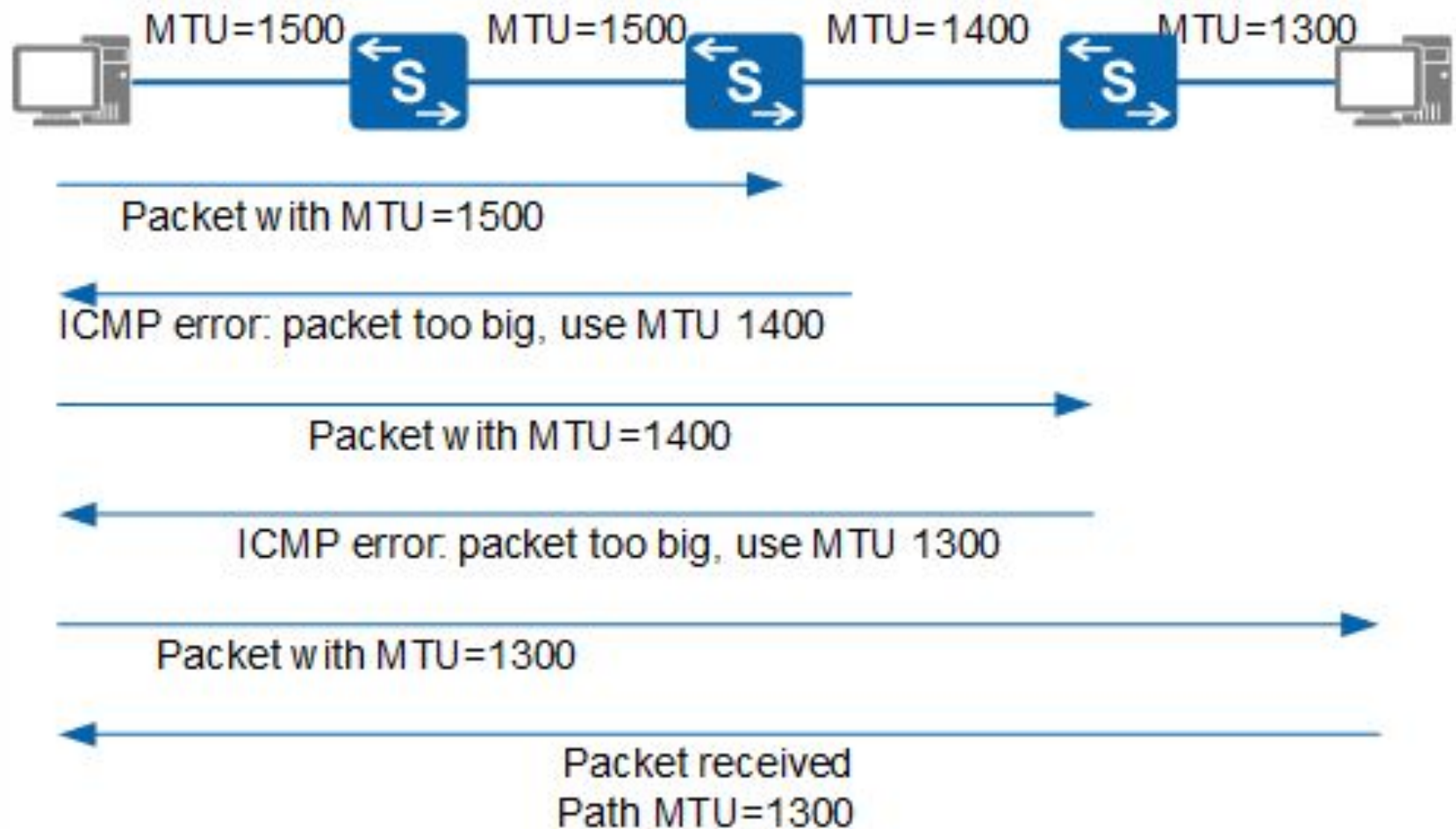


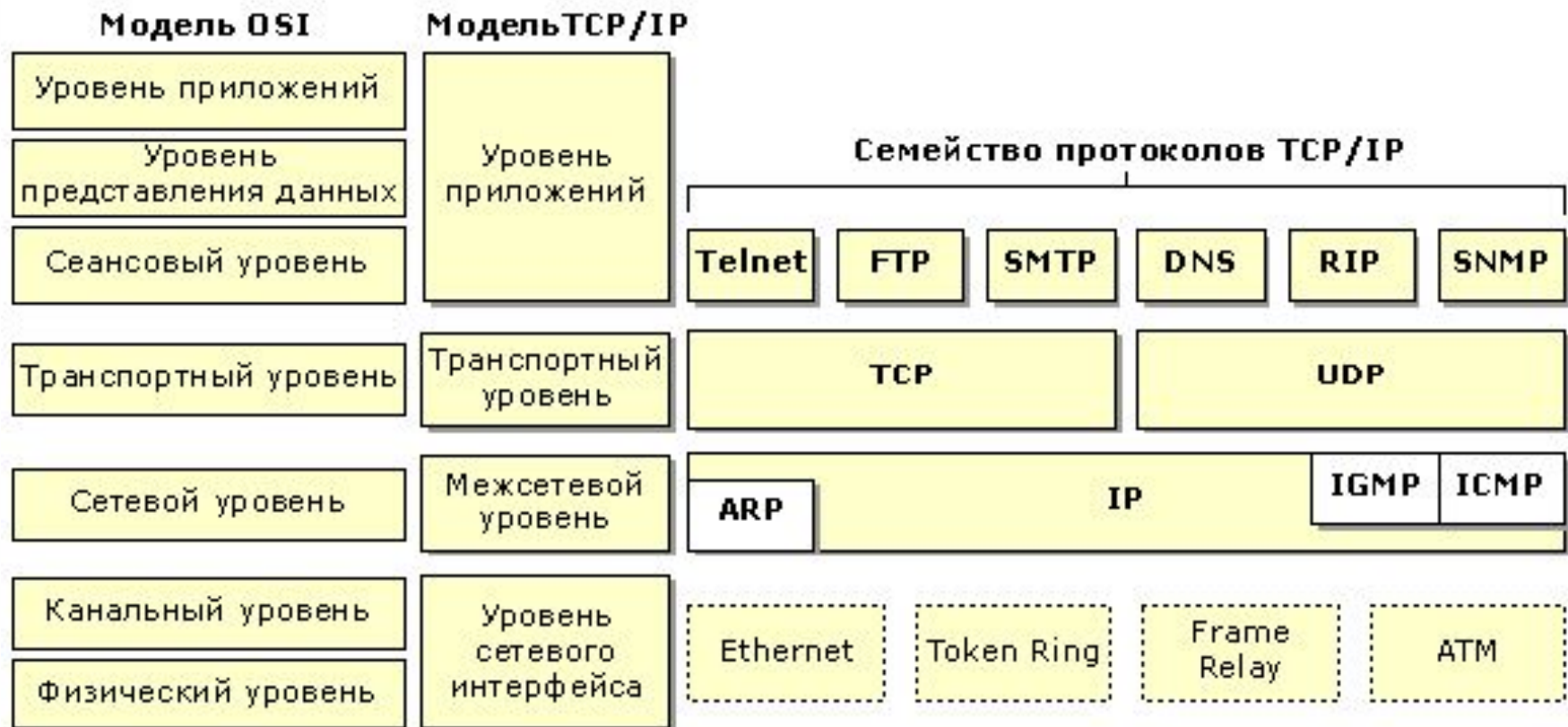


MTU & MSS

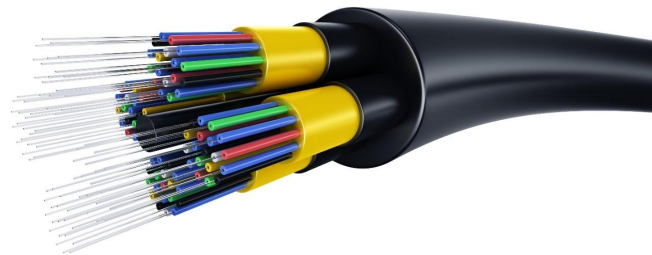
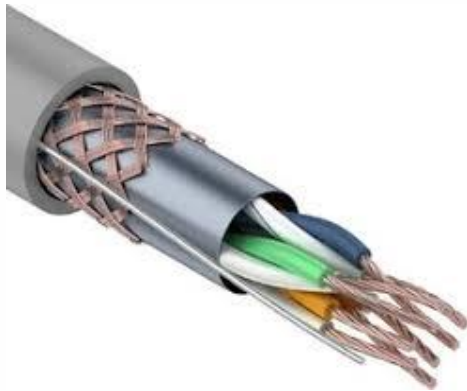


Path MTU Discovery





Физический уровень описывает способы передачи бит через физические среды линий связи, соединяющие сетевые устройства



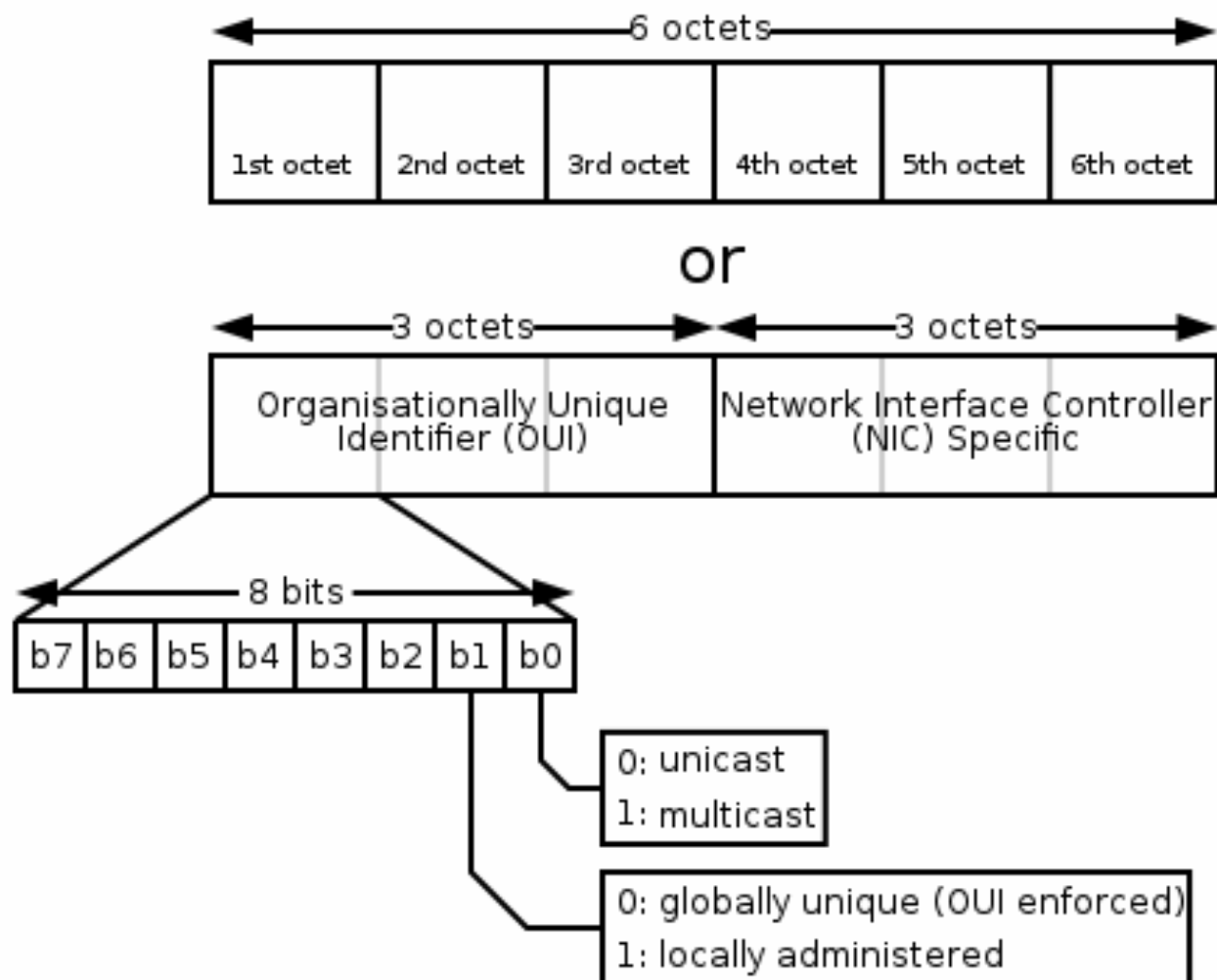
Канальный уровень(1-ый уровень модели TCP/IP) - описывает способ кодирования данных для передачи пакета данных на физическом уровне

MAC(Media Access Control, или Medium Access Control) - подуровень управления доступом к среде

LCC(Logical Link Control) - подуровень управления логической связью

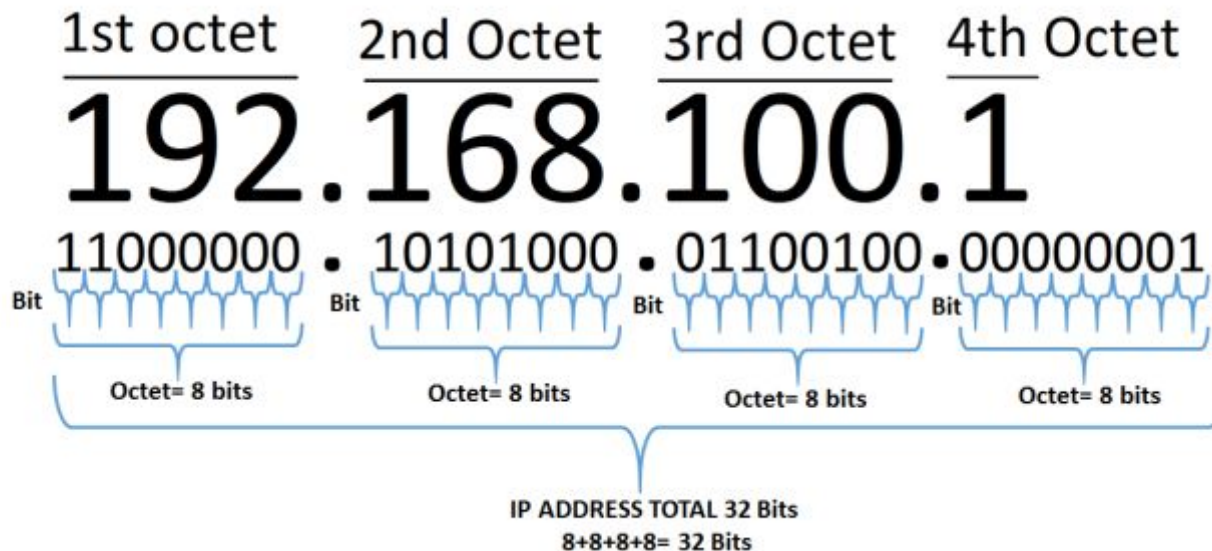
Задачи уровня 2:

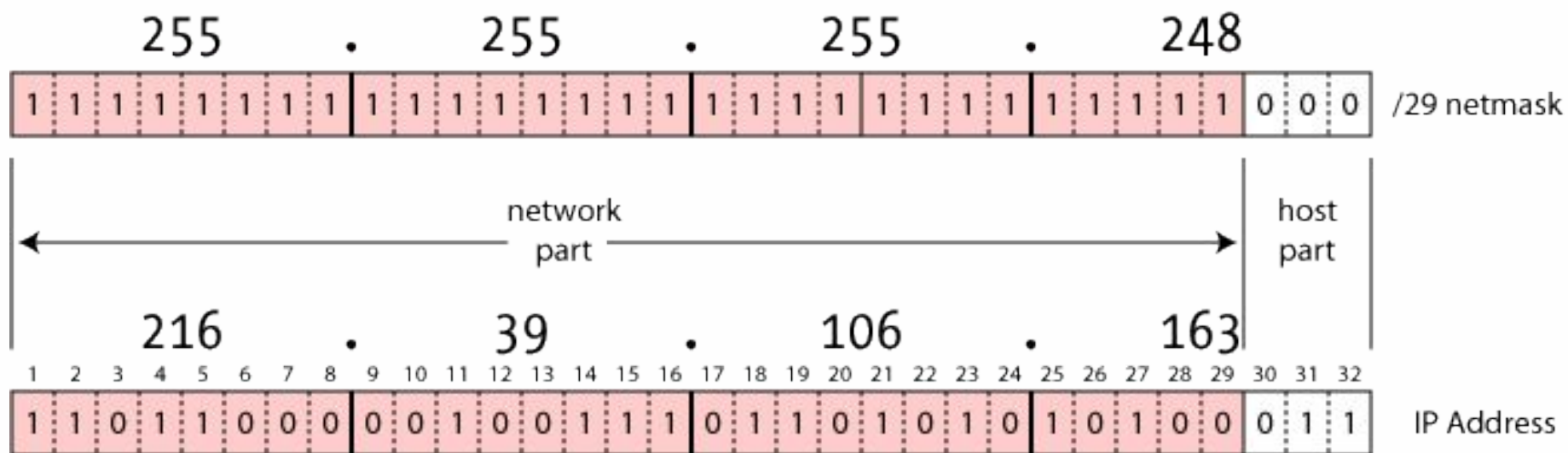
- Формирование / обработка сигнала
- Множественный доступ
- Выделение границ кадра
- Аппаратная адресация
- Контроль ошибок передачи



- Старая нотация `eth0`, `eth1`... *iftypeN*. Группировка интерфейсов по типу и сквозная нумерация. Из глобальных минусов - в качестве `eth0` может оказаться не тот интерфейс, что до перезагрузки, например, если вставить новую карточку в “младший” слот.
- Новая нотация (от `systemd`) - Predictable Network Interface Names. В своем виде по умолчанию использует форматы (упрощенно)
(en|wl)[P<domain>]p<bus>s<slot>[f<function>][n<phys_port_name>|d<dev_port>] -
PCI location
(en|wl)[P<domain>]o<bus>[f<function>][n<phys_port_name>|d<dev_port>] -
Onboard device
Таким образом `enp0s3` говорит нам о том, что мы имеем дело с Ethernet-адаптером подключенным к шине `pci` №0 в слот №3, а `eno1` говорит об onboard ethernet-адаптере с индексом 1.

Предназначается для определения пути передачи данных. Отвечает за определение кратчайших маршрутов и маршрутизацию, отслеживание неполадок и заторов в сети





4 бита Номер версии	4 бита Длина заголовка	8 бит Тип сервиса					16 бит Общая длина					
		PR	D	T	R							
16 бит Идентификатор пакета							3 бита Флаги	13 бит Смещение фрагмента				
								D	M			
8 бит Время жизни		8 бит Протокол верхнего уровня					16 бит Контрольная сумма					
32 бита IP-адрес источника												
32 бита IP-адрес назначения												
Параметры и выравнивание												

Bits 0-2: Precedence.

Bit 3: 0 = Normal Delay, 1 = Low Delay.

Bit 4: 0 = Normal Throughput, 1 = High Throughput.

Bit 5: 0 = Normal Reliability, 1 = High Reliability.

Bit 6-7: Reserved for Future Use.

Precedence

111 -	Network Control	011 -	Flash
110 -	Internetwork Control	010 -	Immediate
101 -	CRITIC/ECP	001 -	Priority
100 -	Flash Override	000 -	Routine

Flags

Bit 0: reserved, must be zero

Bit 1: (DF) 0 = May Fragment, 1 = Don't Fragment.

Bit 2: (MF) 0 = Last Fragment, 1 = More Fragments.

```
# ipcalc 195.239.108.7/26
```

```
Address: 195.239.108.7 11000011.11101111.01101100.00 000111
```

```
Netmask: 255.255.255.192 = 26 11111111.11111111.11111111.11 000000
```

```
Wildcard: 0.0.0.63 00000000.00000000.00000000.00 111111
```

```
=>
```

```
Network: 195.239.108.0/26 11000011.11101111.01101100.00 000000
```

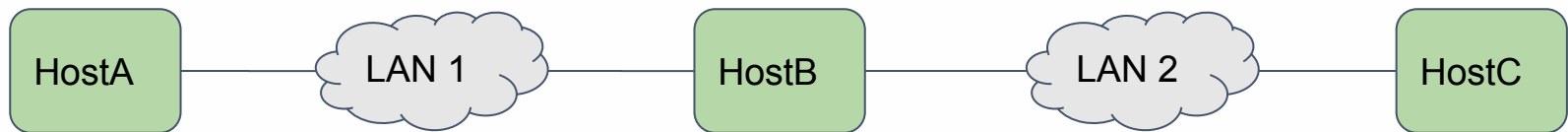
```
HostMin: 195.239.108.1 11000011.11101111.01101100.00 000001
```

```
HostMax: 195.239.108.62 11000011.11101111.01101100.00 111110
```

```
Broadcast: 195.239.108.63 11000011.11101111.01101100.00 111111
```

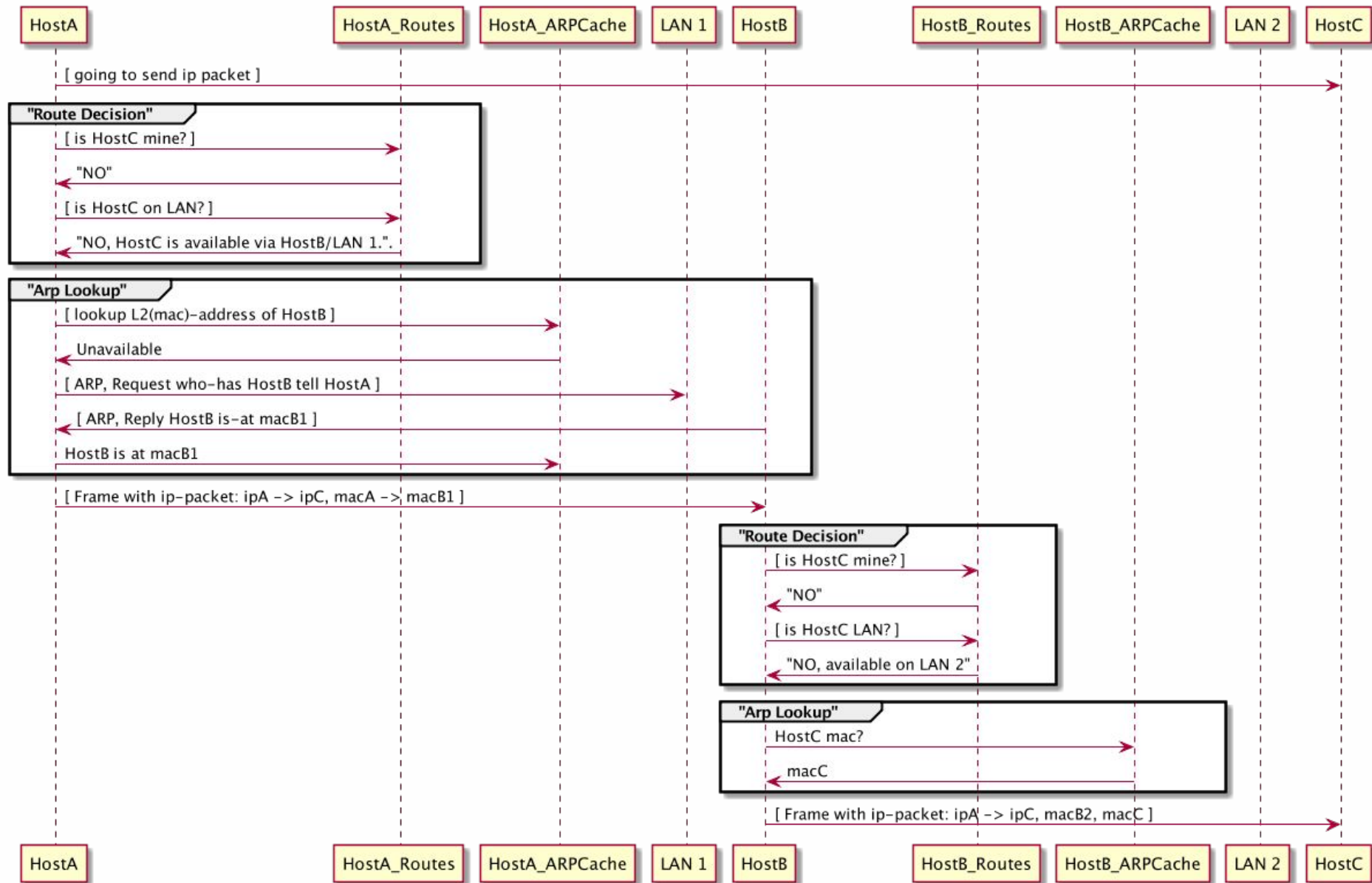
```
Hosts/Net: 62 Class C
```

Предположим у нас есть сеть:

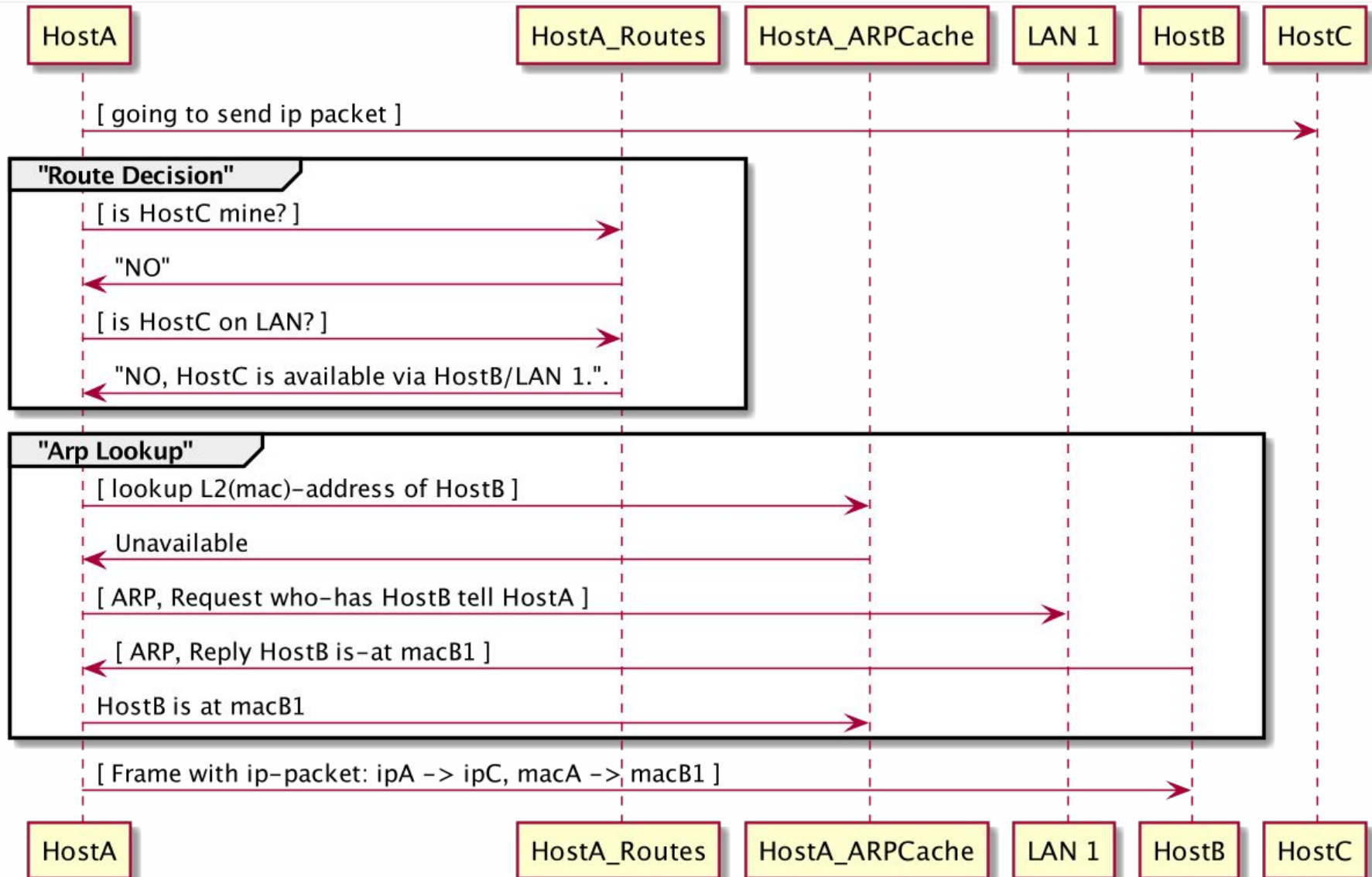


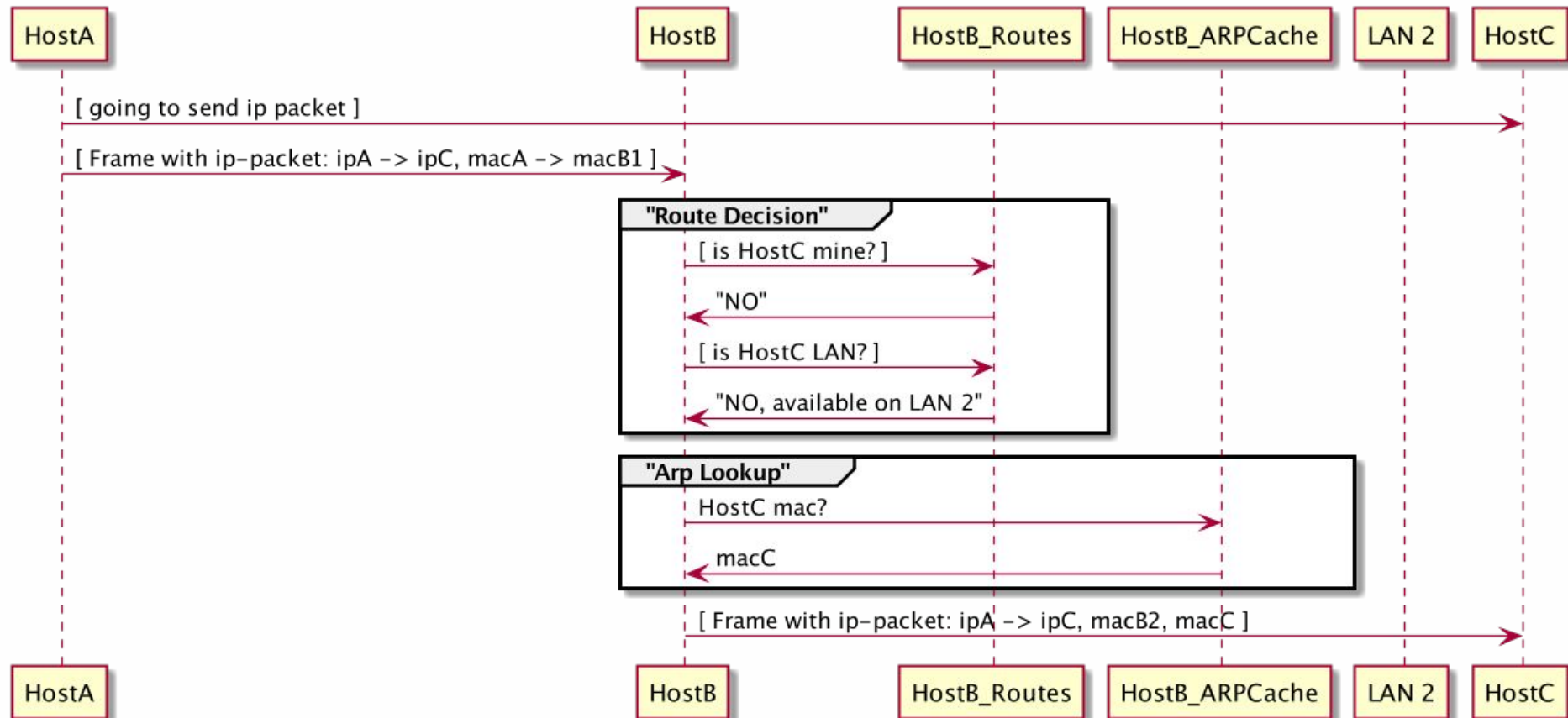
HostA посылает HostC ip-пакет

ARP



ARP





TCP



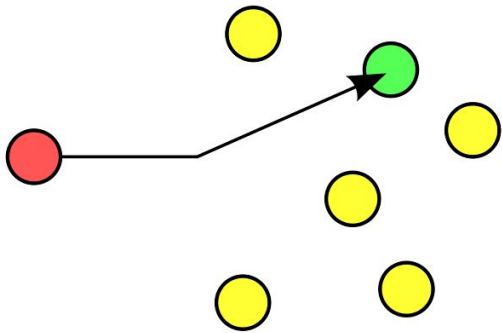
UDP



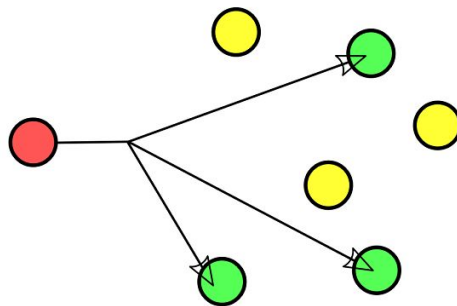
Transmission Control Protocol (TCP) Header

source port number 2 bytes				destination port number 2 bytes			
sequence number 4 bytes							
acknowledgement number 4 bytes							
data offset 4 bits		reserved 3 bits		control flags 9 bits		window size 2 bytes	
checksum 2 bytes				urgent pointer 2 bytes			
optional data 0-40 bytes							

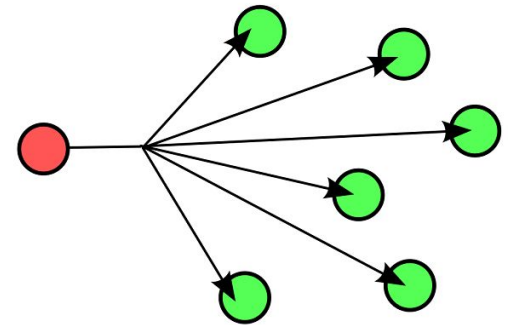




Unicast



Multicast



Broadcast

03

TCP

Transmission Control Protocol (TCP) Header

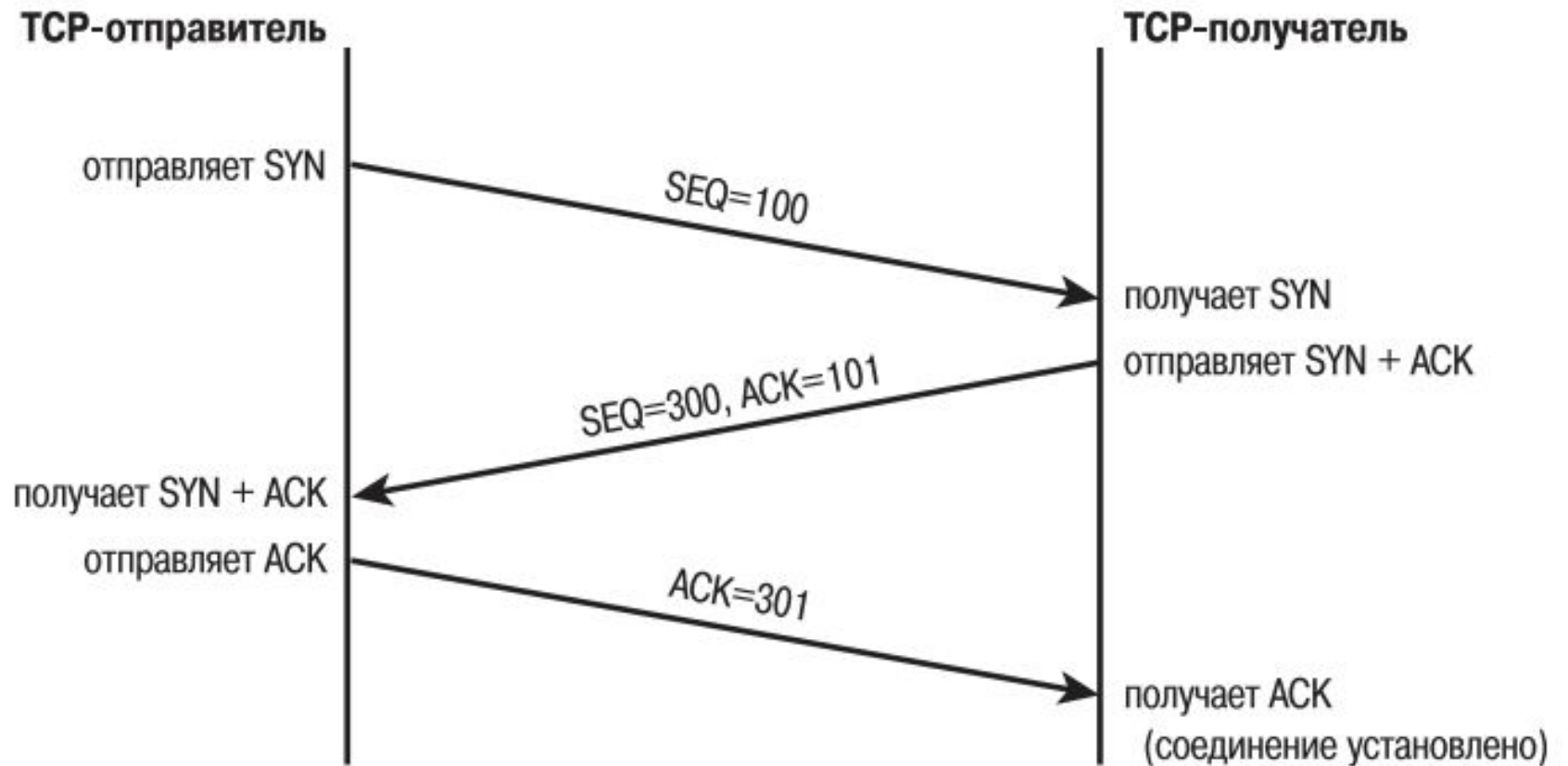
source port number 2 bytes				destination port number 2 bytes			
sequence number 4 bytes							
acknowledgement number 4 bytes							
data offset 4 bits		reserved 3 bits		control flags 9 bits		window size 2 bytes	
checksum 2 bytes				urgent pointer 2 bytes			
optional data 0-40 bytes							

- CWR (Congestion Window Reduced) — Поле «Окно перегрузки уменьшено» — флаг установлен отправителем, чтобы указать, что получен пакет с установленным флагом ECE (RFC 3168)
- ECE (ECN-Echo) — Поле «Эхо ECN» — указывает, что данный узел способен на ECN (явное уведомление перегрузки) и для указания отправителю о перегрузках в сети (RFC 3168)
- URG — поле «Указатель важности» задействовано (англ. Urgent pointer field is significant)
- ACK — поле «Номер подтверждения» задействовано (англ. Acknowledgement field is significant)

- PSH — (англ. Push function) инструктирует получателя протолкнуть данные, накопившиеся в приёмном буфере, в приложение пользователя
- RST — оборвать соединения, сбросить буфер (очистка буфера) (англ. Reset the connection)
- SYN — синхронизация номеров последовательности (англ. Synchronize sequence numbers)
- FIN (англ. final, бит) — флаг, будучи установлен, указывает на завершение соединения (англ. FIN bit used for connection termination).

TCP_Flags

[illegible]



- CLOSED - Начальное состояние узла. Фактически фиктивное
- LISTEN - Сервер ожидает запросов установления соединения от клиента
- SYN-SENT - Клиент отправил запрос серверу на установление соединения и ожидает ответа
- SYN-RECEIVED - Сервер получил запрос на соединение, отправил ответный запрос и ожидает подтверждения
- ESTABLISHED - Соединение установлено, идёт передача данных
- FIN-WAIT-1 - Одна из сторон (назовём её узел-1) завершает соединение, отправив сегмент с флагом FIN

- CLOSE-WAIT - Другая сторона (узел-2) переходит в это состояние, отправив, в свою очередь сегмент ACK и продолжает одностороннюю передачу
- FIN-WAIT-2 - Узел-1 получает ACK, продолжает чтение и ждёт получения сегмента с флагом FIN
- LAST-ACK - Узел-2 заканчивает передачу и отправляет сегмент с флагом FIN
- TIME-WAIT - Узел-1 получил сегмент с флагом FIN, отправил сегмент с флагом ACK и ждёт $2 \times \text{MSL}$ секунд, перед окончательным закрытием соединения
- CLOSING - Обе стороны инициировали закрытие соединения одновременно: после отправки сегмента с флагом FIN узел-1 также получает сегмент FIN, отправляет ACK и находится в ожидании сегмента ACK (подтверждения на свой запрос о разъединении)

04

Управляющие пакеты

- Net-tools (arp, ifconfig, netstat, route) - deprecated
- Iproute2 (ip, ss, tc, nstat)
- NetworkManager (nmcli)

iproute2

- ip - управление маршрутизацией, интерфейсами, arp-таблицами
- tc - traffic control - управлением приоритезацией трафика
- ss - sockstat - информация о socket'ах (одна из сторон netstat)
- nstat - информация о сетевых каунтерах

- ip link list
- ip addr show
- ip route show
- ip route ls
- ip neigh show
- ip rule list
- cat /etc/iproute2/route/route
- ip route list table <main|local|default>
- echo 200 Otus >> /etc/iproute2/route/route
- ip rule add from 10.0.0.10 table Otus
- ip route add default via 195.96.98.253 dev ppp2 table Otus

включение форвардинга

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

выключение фильтрации асинхронной маршрутизации

```
#!/bin/bash
for DEV in /proc/sys/net/ipv4/conf/*/rp_filter
do
    echo 0 > $DEV
done
```

- tcpdump - информация о сетевой активности. Работает максимально близко к “проводу”
- ngrep - утилита для поиска пакетов по содержимому, Network grep. По смыслу схожа с tcpdump.
- Wireshark (tshark)

05

Questions?

**Спасибо
за внимание!**

