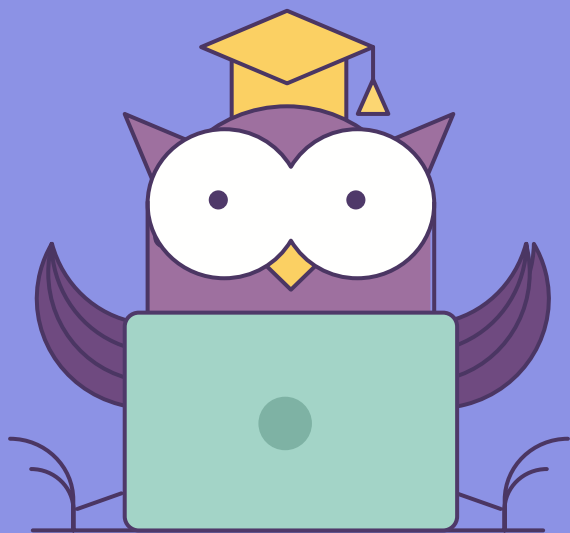




ОНЛАЙН-ОБРАЗОВАНИЕ

Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте  если все хорошо

Архитектура сетей

OSI

TCP/IP

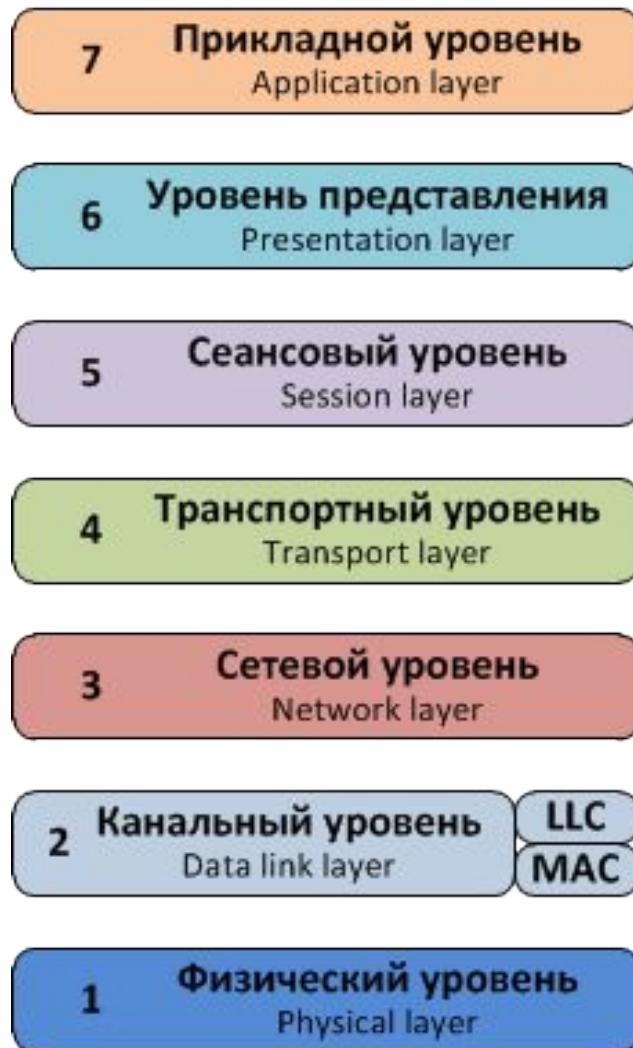




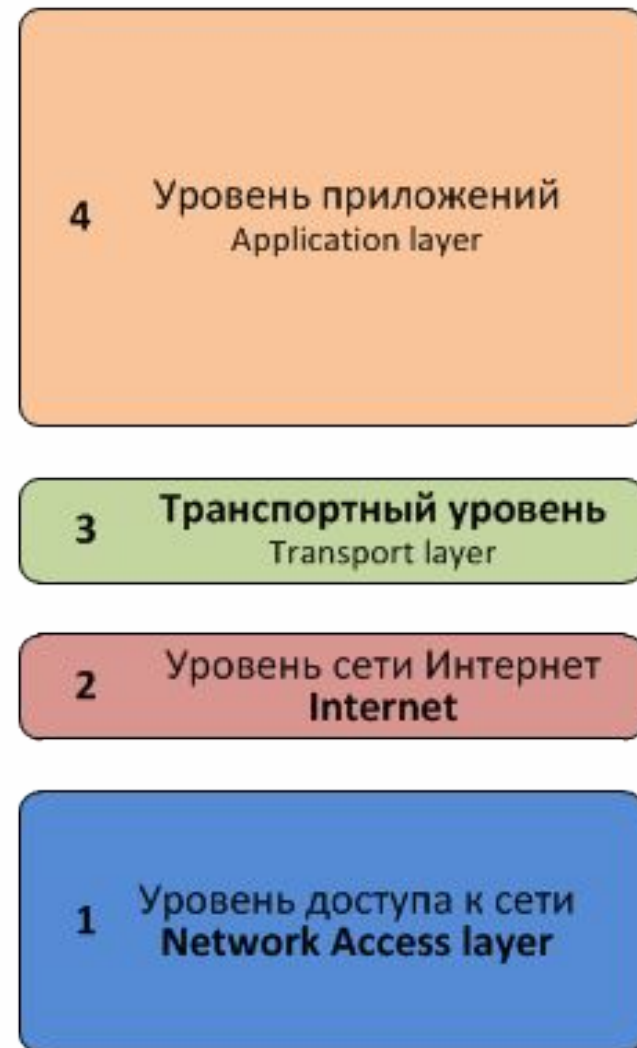
01

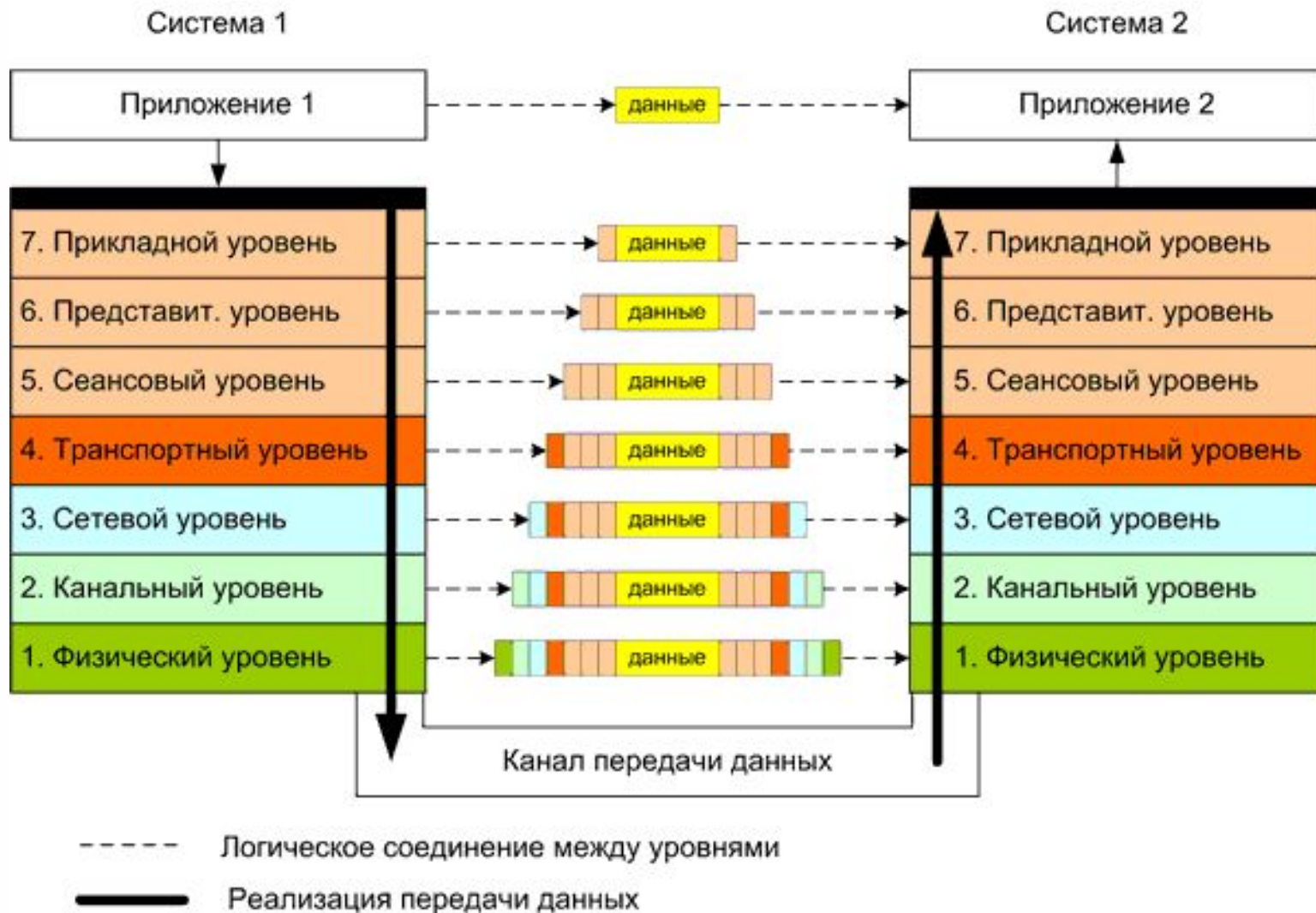
OSI

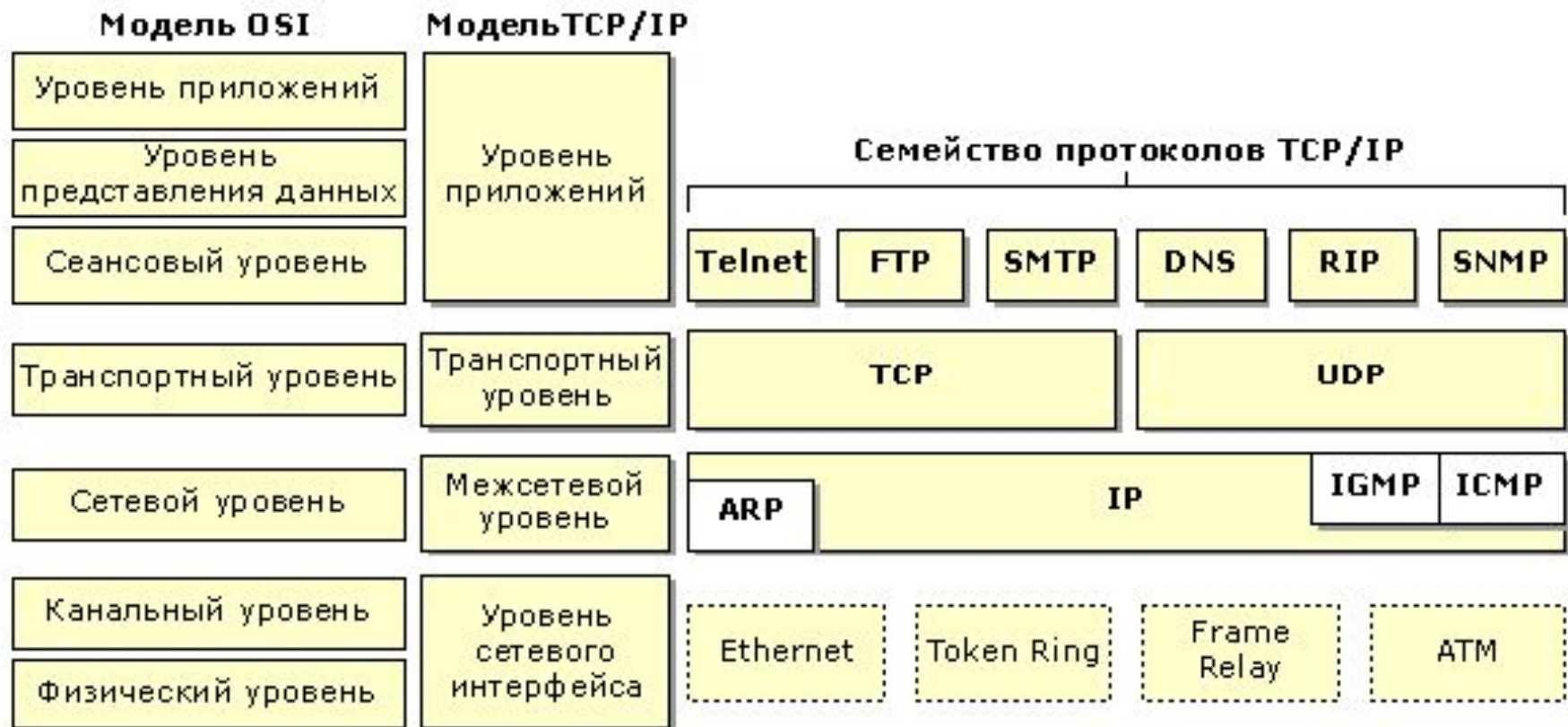
OSI



TCP/IP (DOD)



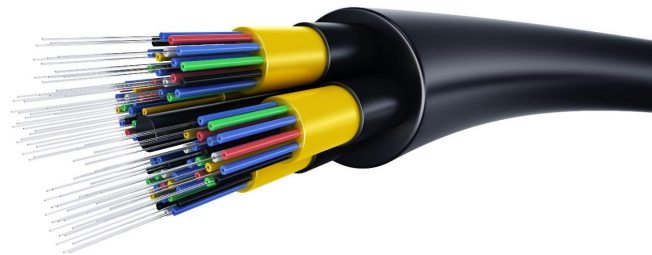
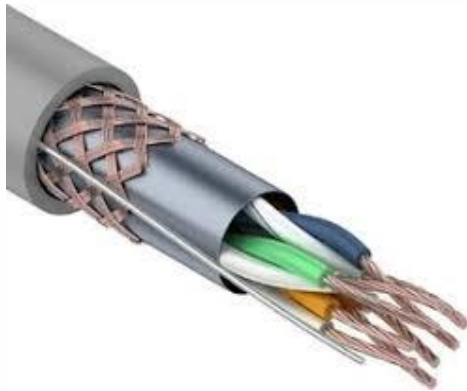




02

уровни OSI и
протоколы

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



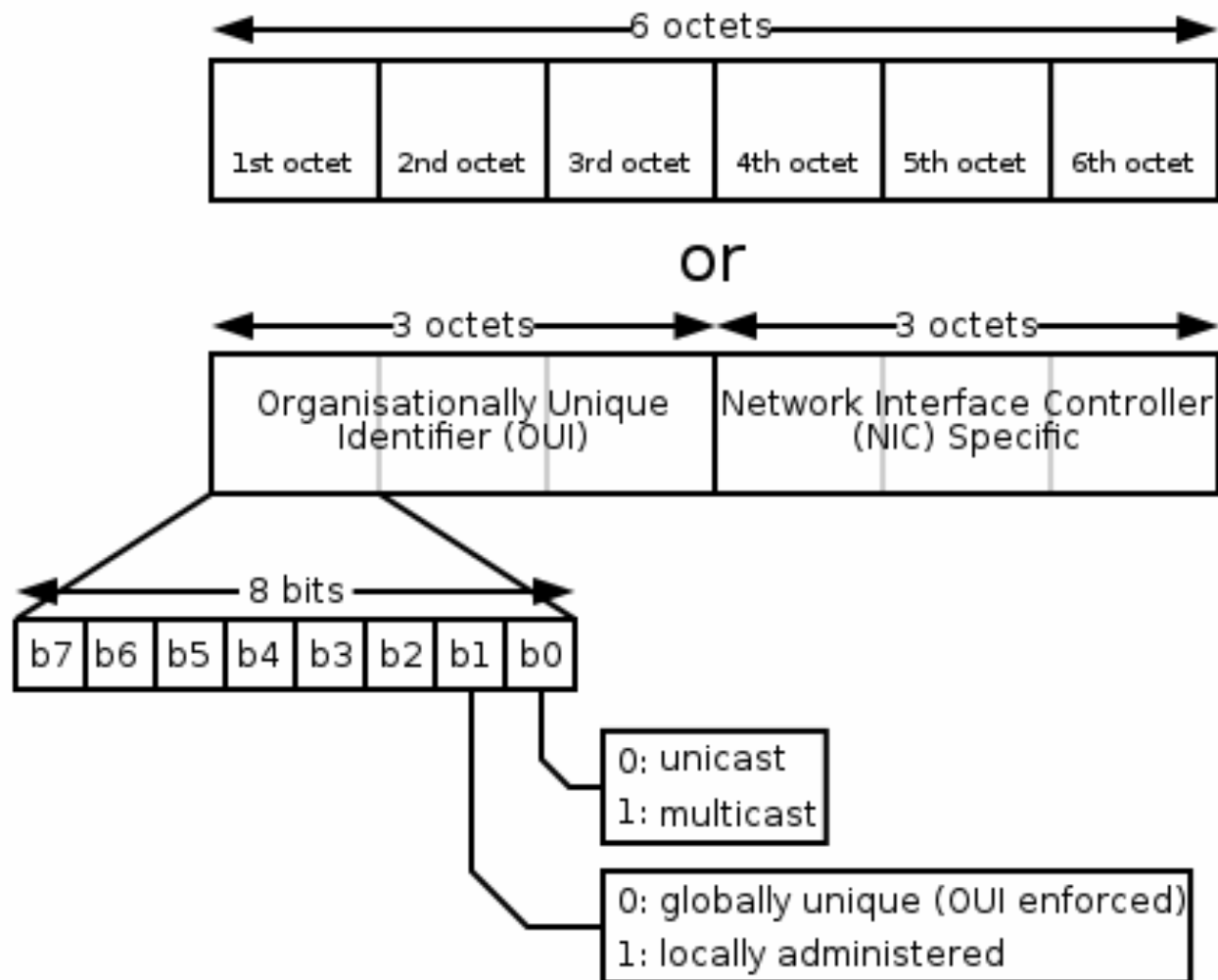
Канальный уровень(1-ый уровень модели TCP/IP) - описывает способ кодирования данных для передачи пакета данных на физическом уровне

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

LCC(Logical Link Control) - подуровень управления логической связью

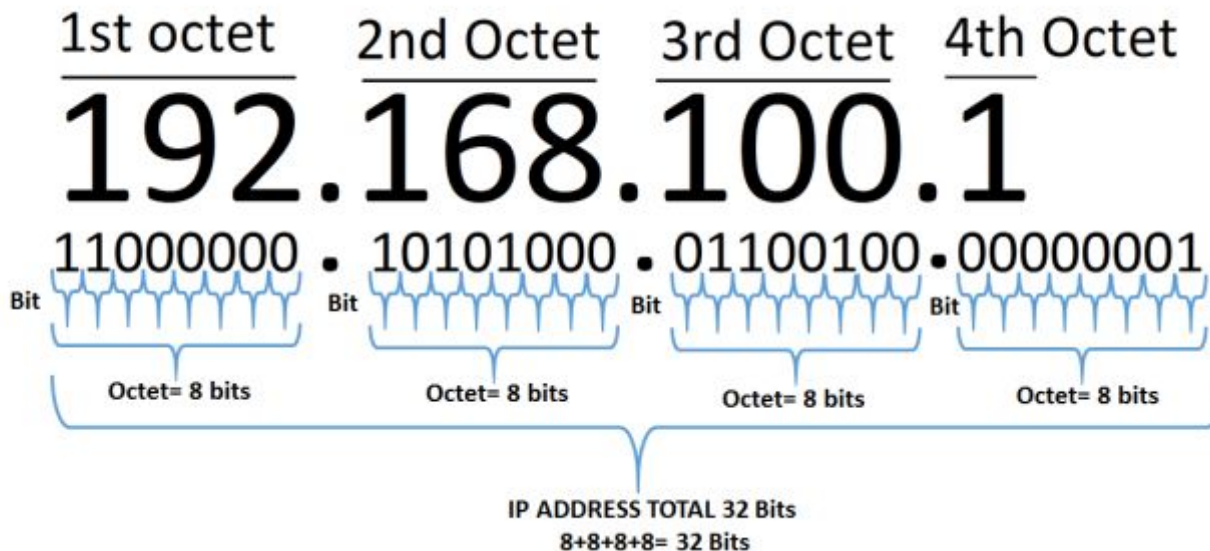
Задачи уровня 2:

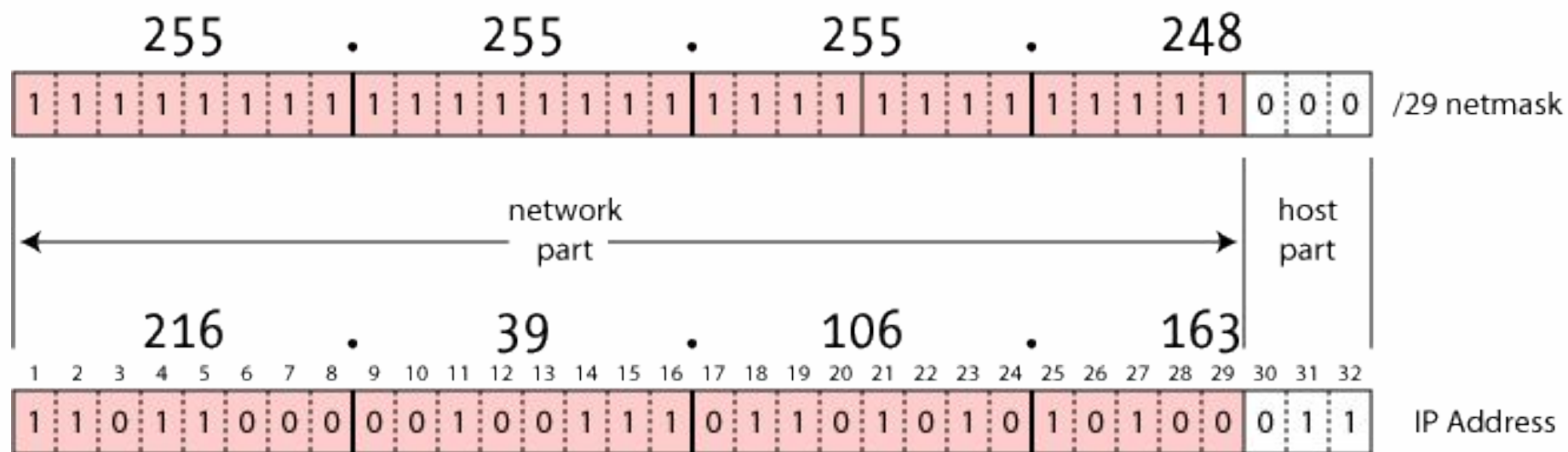
- Формирование / обработка сигнала
- Множественный доступ
- Выделение границ кадра
- Аппаратная адресация
- Контроль ошибок передачи



- 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														

Тип обслуживания (Type of Service, ToS)

- **0-2** — приоритет (precedence) данного IP-сегмента
- **3** — требование ко времени задержки (delay) передачи IP-сегмента (0 — нормальная, 1 — низкая задержка)
- **4** — требование к пропускной способности (throughput) маршрута, по которому должен отправляться IP-сегмент (0 — низкая, 1 — высокая пропускная способность)
- **5** — требование к надежности (reliability) передачи IP-сегмента (0 — нормальная, 1 — высокая надежность)
- **6-7** — ECN — явное сообщение о задержке (управление IP-потокотом).

Биты флагов

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
```

```
=>
```

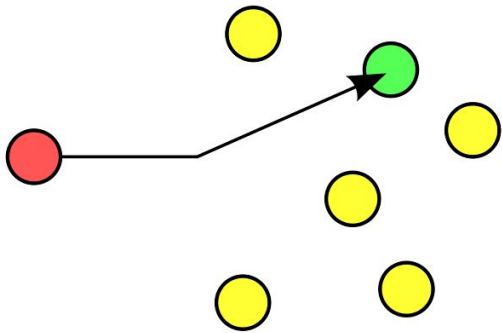
```
Network: 195.239.108.0/26 11000011.11101111.01101100.00 000000
```

```
HostMin: 195.239.108.1 11000011.11101111.01101100.00 000001
```

```
HostMax: 195.239.108.62 11000011.11101111.01101100.00 111110
```

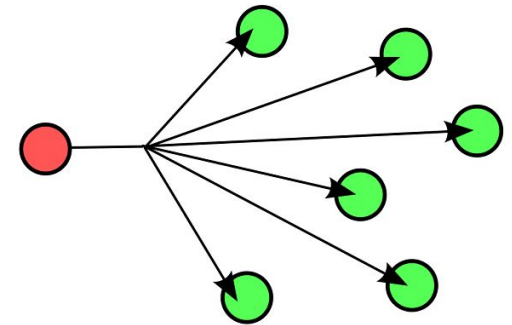
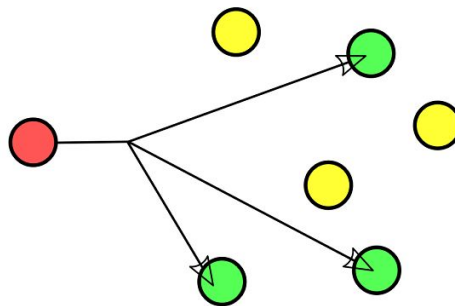
```
Broadcast: 195.239.108.63 11000011.11101111.01101100.00 111111
```

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



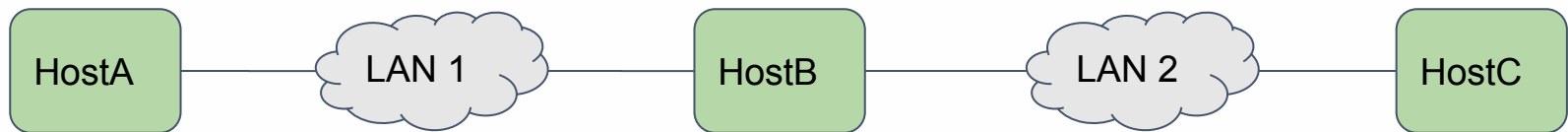
Unicast

Multicast



Broadcast

Предположим у нас есть сеть:

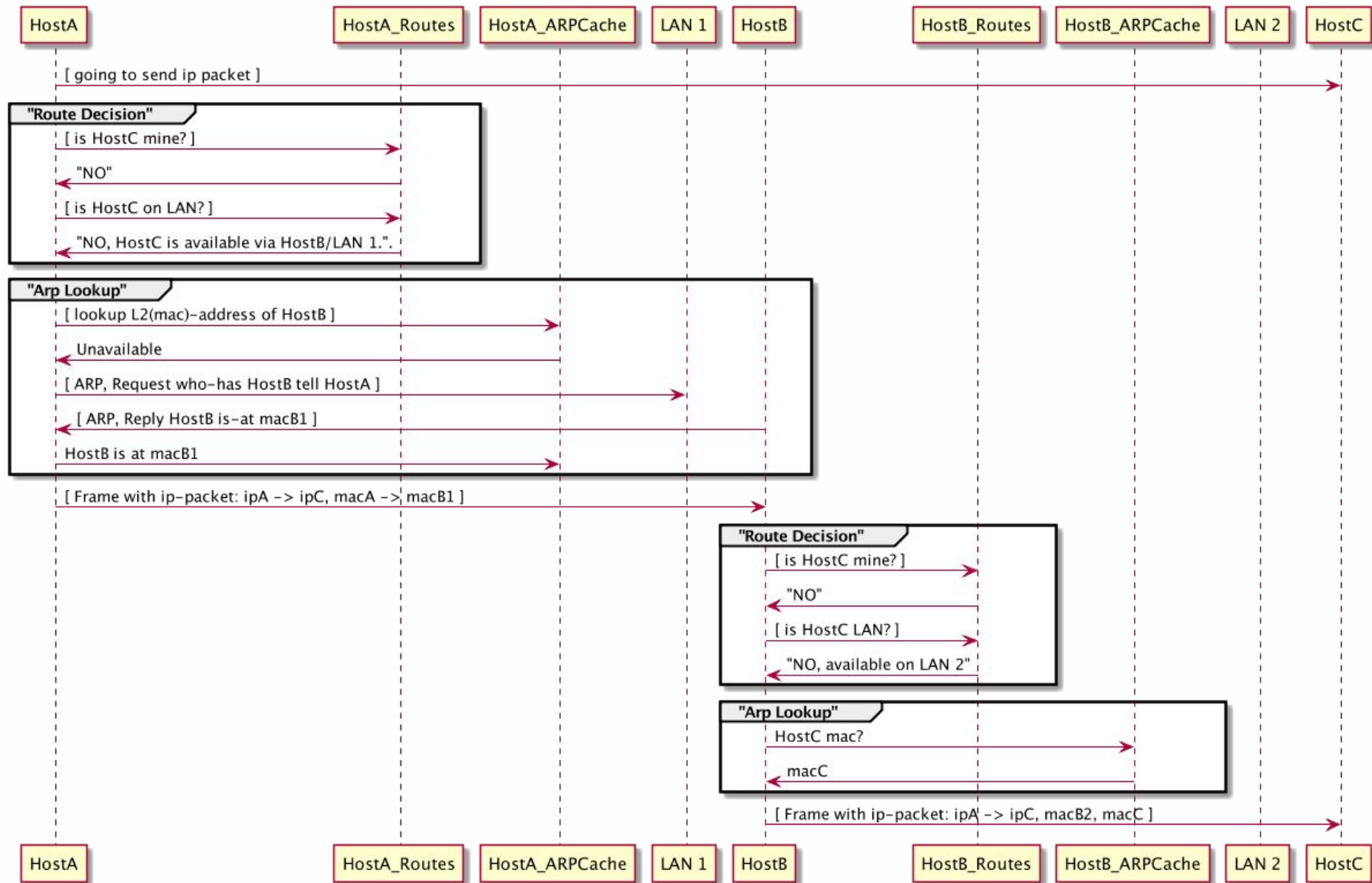


HostA посылает HostC ip-пакет

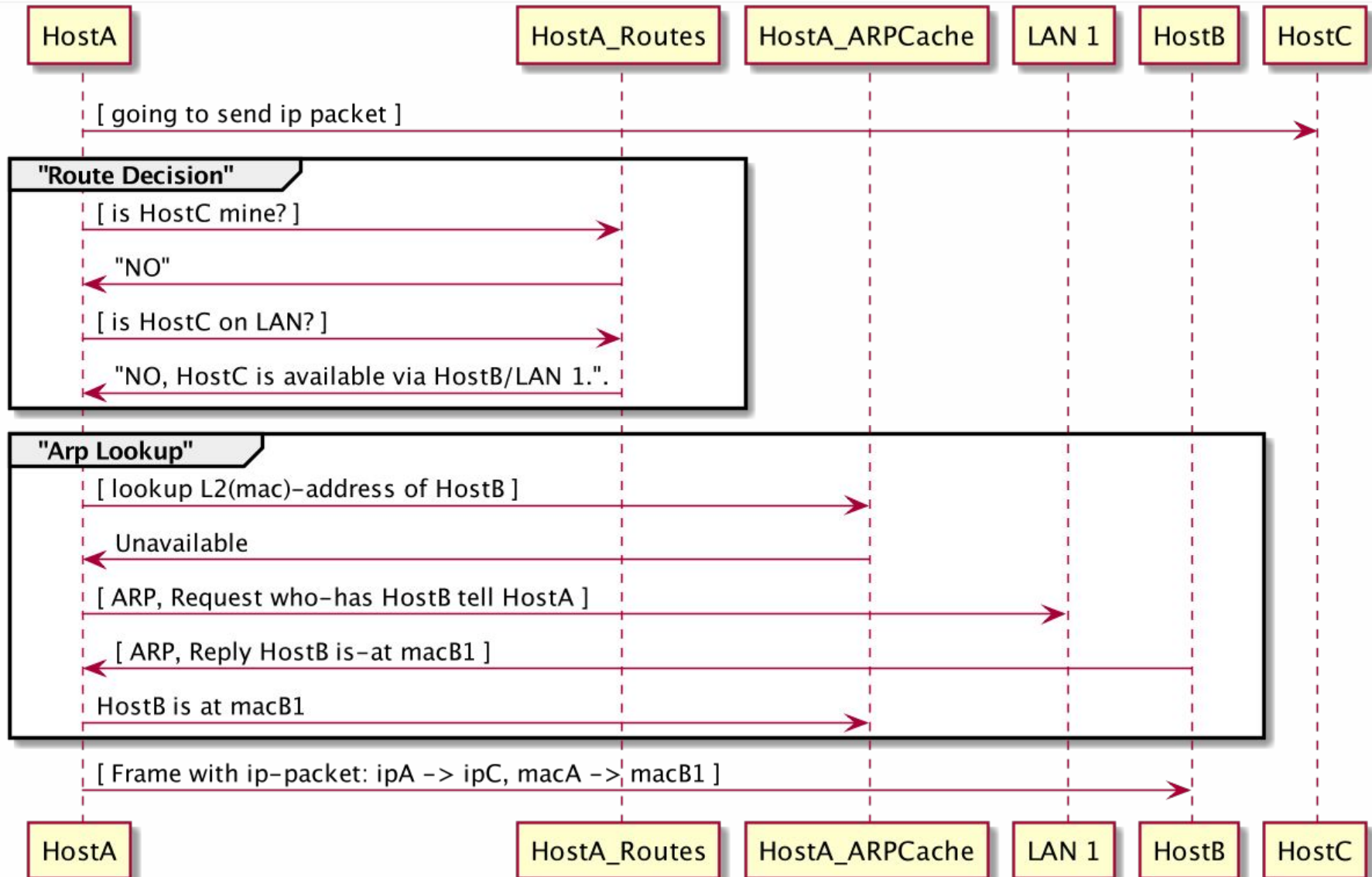
Протокол ARP (Address Resolution Protocol)

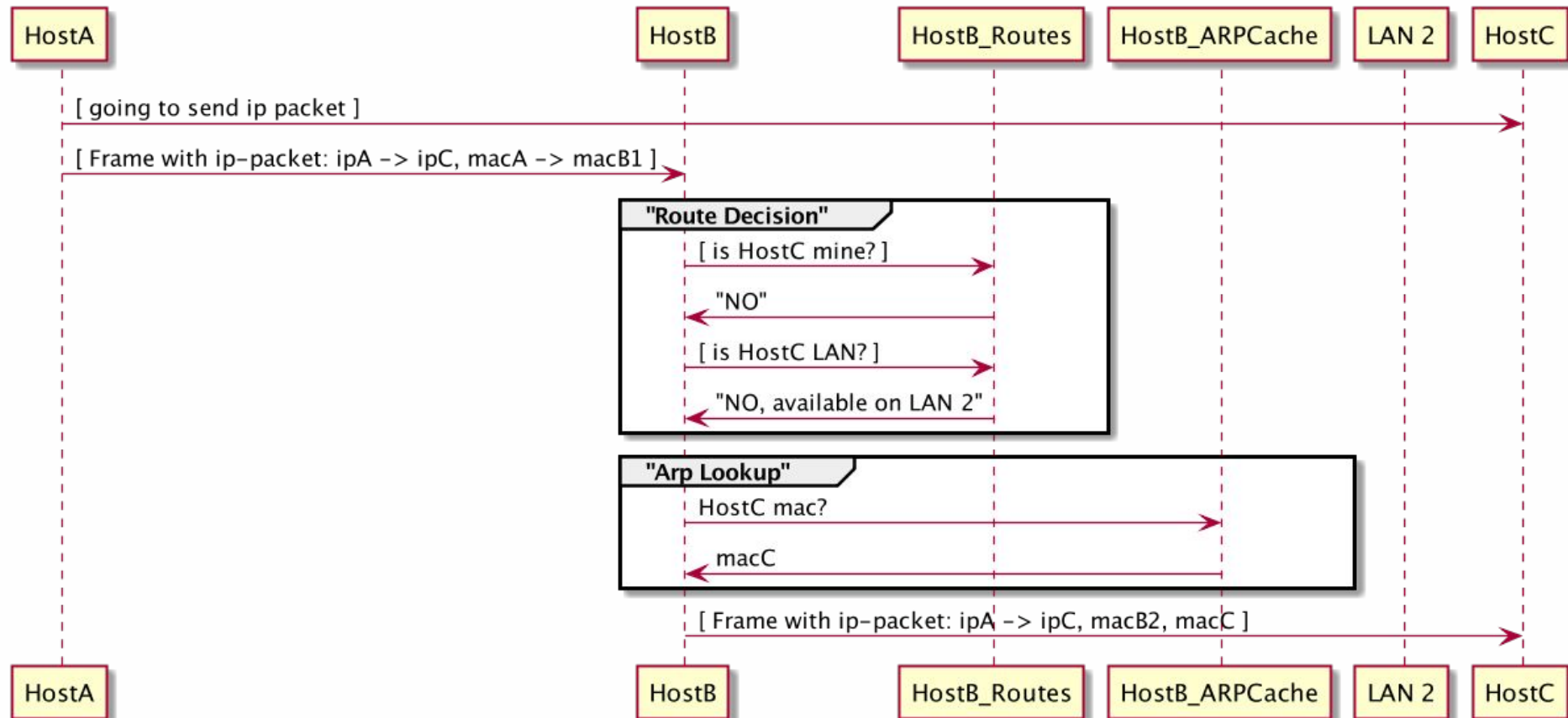
служит для разрешения адресов (поиска соответствия мас-адресов и IP-адресов) в пределах одного L2-сегмента (L2-домена).

ARP

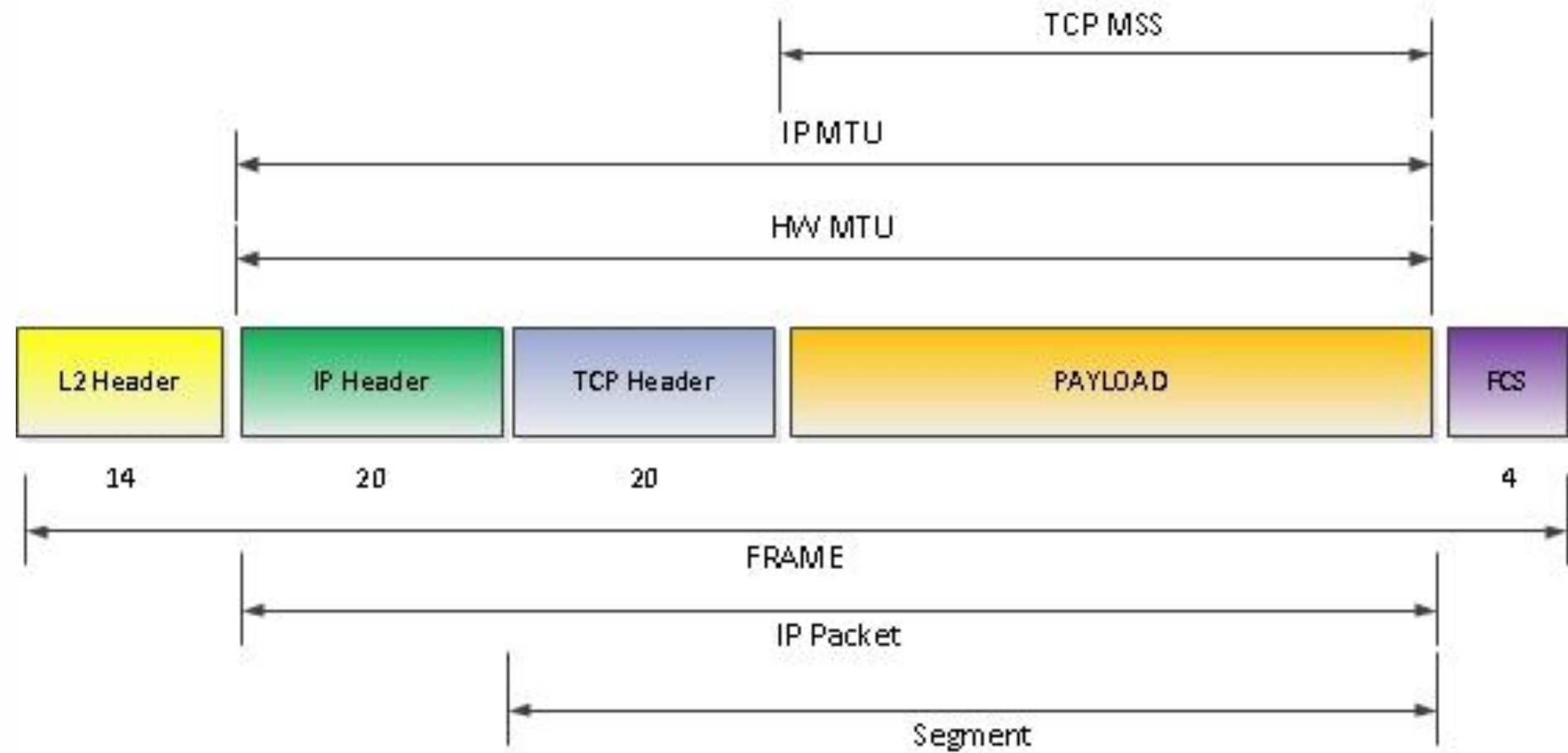


ARP

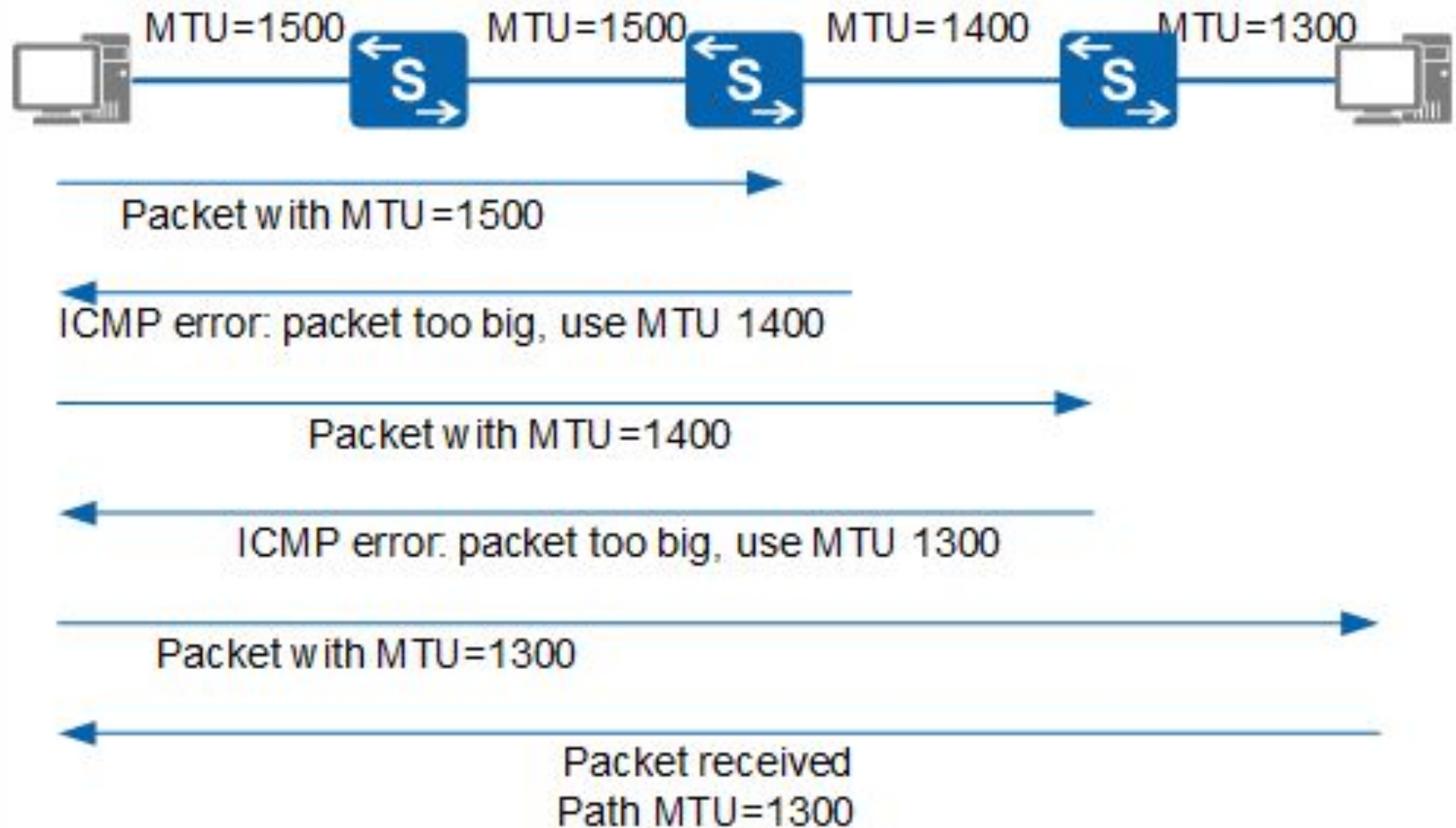




MTU & MSS



Path MTU Discovery



TCP



UDP





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							

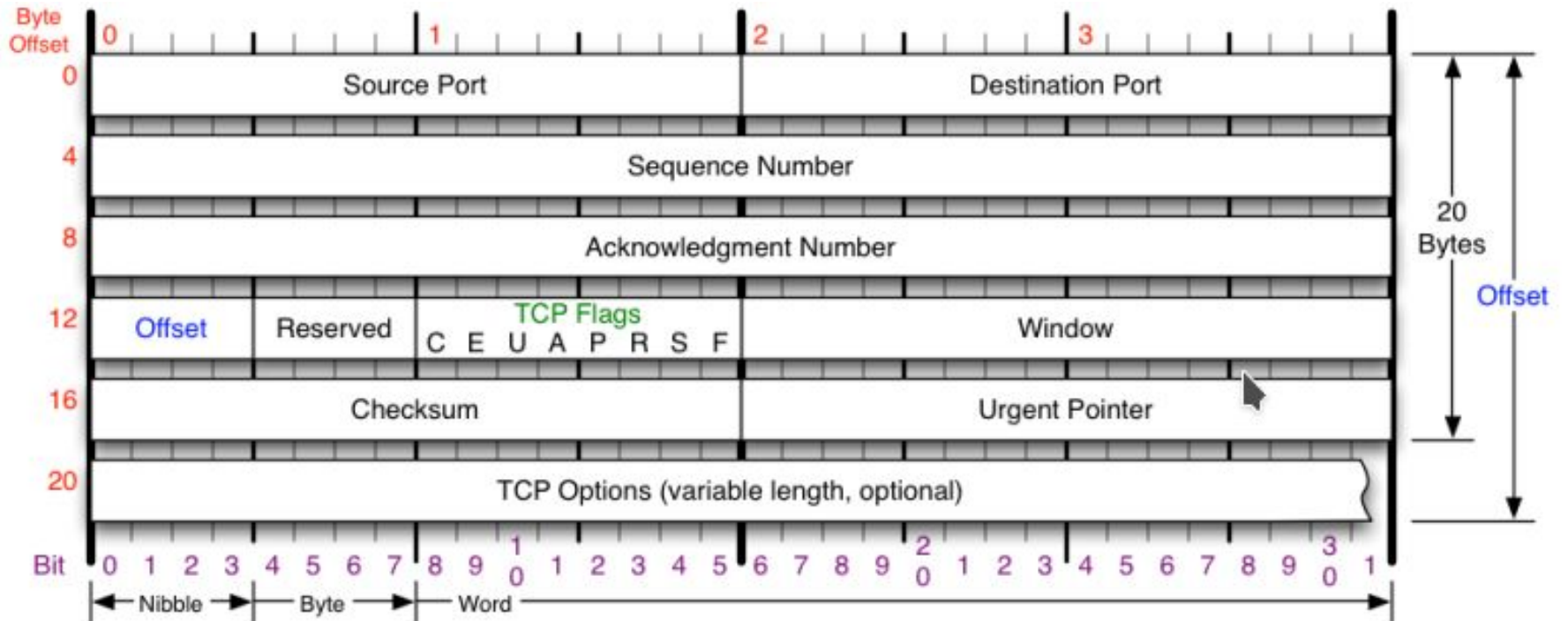
Протокол TCP (Transmission Control Protocol) - протокол с контролем передачи. Предназначен для надежной передачи данных. Это реализуется с помощью механизмов:

- Установки соединения
- Подтверждения передачи
- Закрытия соединения

Как результат протокол TCP обеспечивает:

- “контроль доставки”
- Сохранение очередности пакетов

TCP



Номер в последовательности (sequence number)

- 32-битовое поле
- содержимое определяет (косвенно) положение данных TCP-пакета внутри исходящего потока данных, существующего в рамках текущего соединения.
- В момент установления соединения каждая сторона генерирует свой начальный "номер в последовательности"
- основное требование- не повторяться в промежутке времени, в течение которого TCP-пакет может находиться в сети
- Партнеры обмениваются этими начальными номерами и подтверждают их получение. Во время отправления TCP-пакетов с данными поле "номер в последовательности" содержит сумму начального номера и количества байт ранее переданных данных.

Номер подтверждения (acknowledgement number)

- 32-битовое поле, содержимое которого определяет (косвенно) количество принятых данных из входящего потока к TCP-модулю, формирующему TCP-пакет.

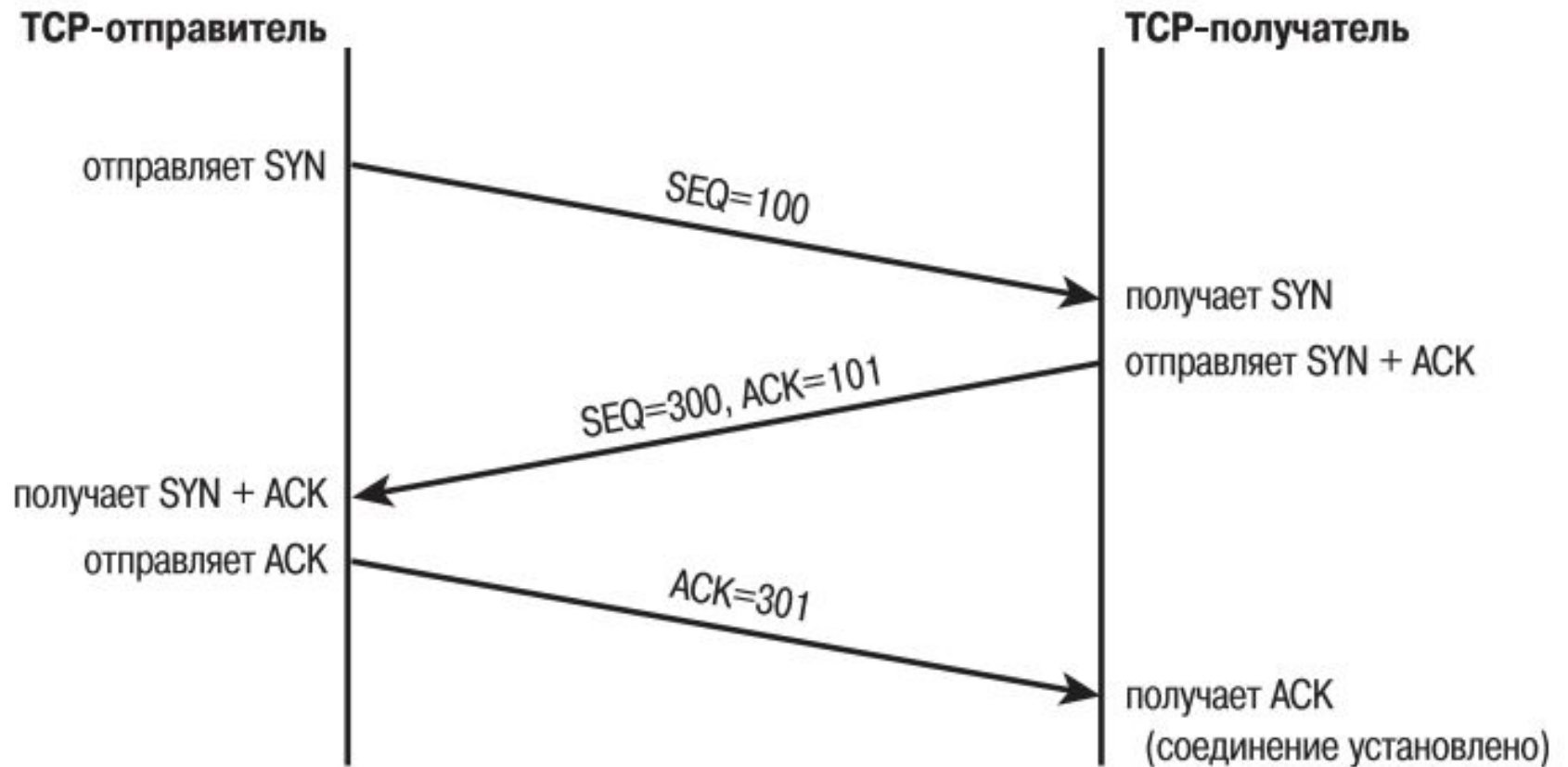
- CWR (Congestion Window Reduced) — Поле «Окно перегрузки уменьшено» — флаг установлен отправителем, чтобы указать, что получен пакет с установленным флагом ECE (RFC 3168)
- ECE (ECN-Echo) — Поле «Эхо ECN» — указывает, что данный узел способен на ECN (явное уведомление перегрузки) и для указания отправителю 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

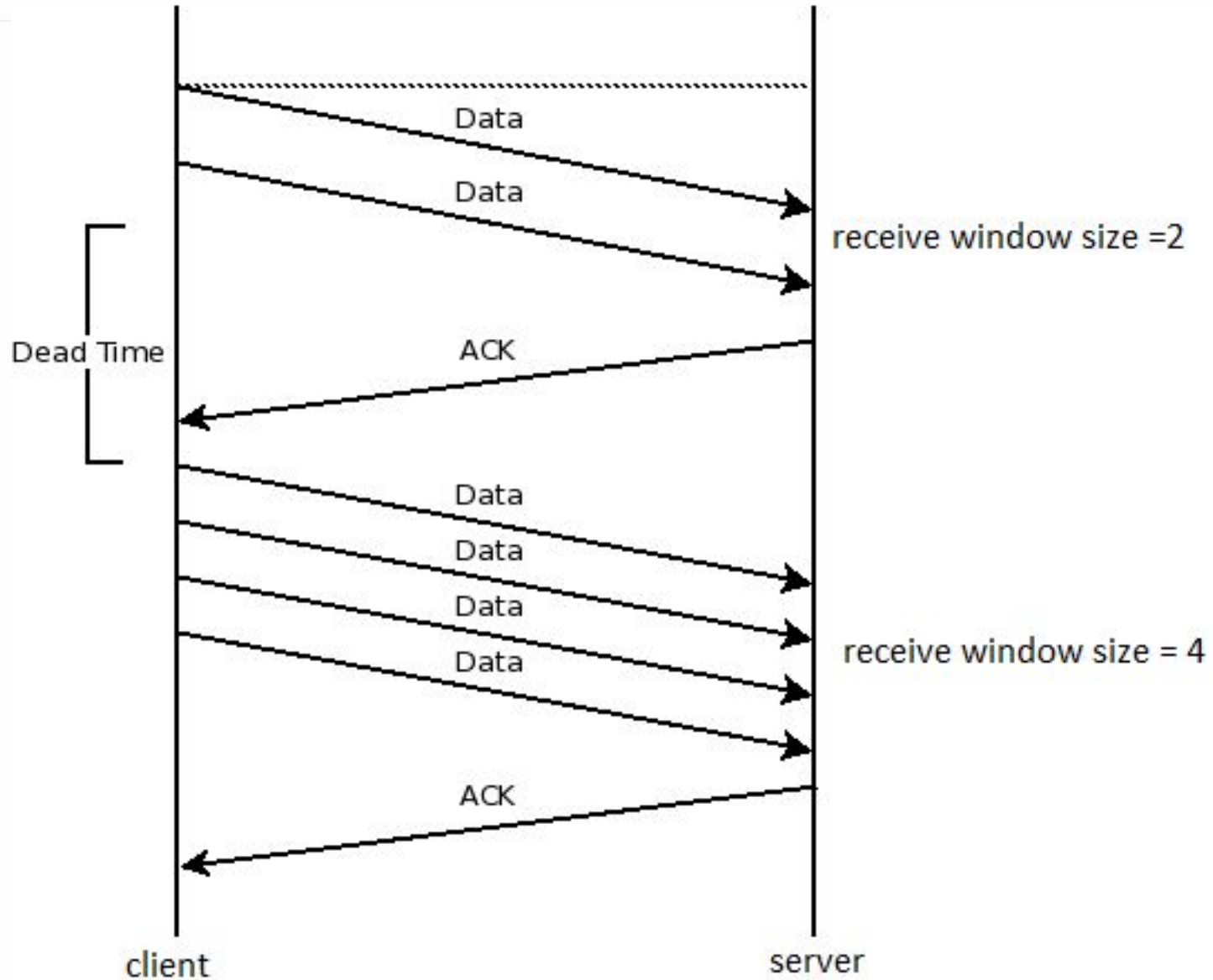
[illegible]

TCP handshake

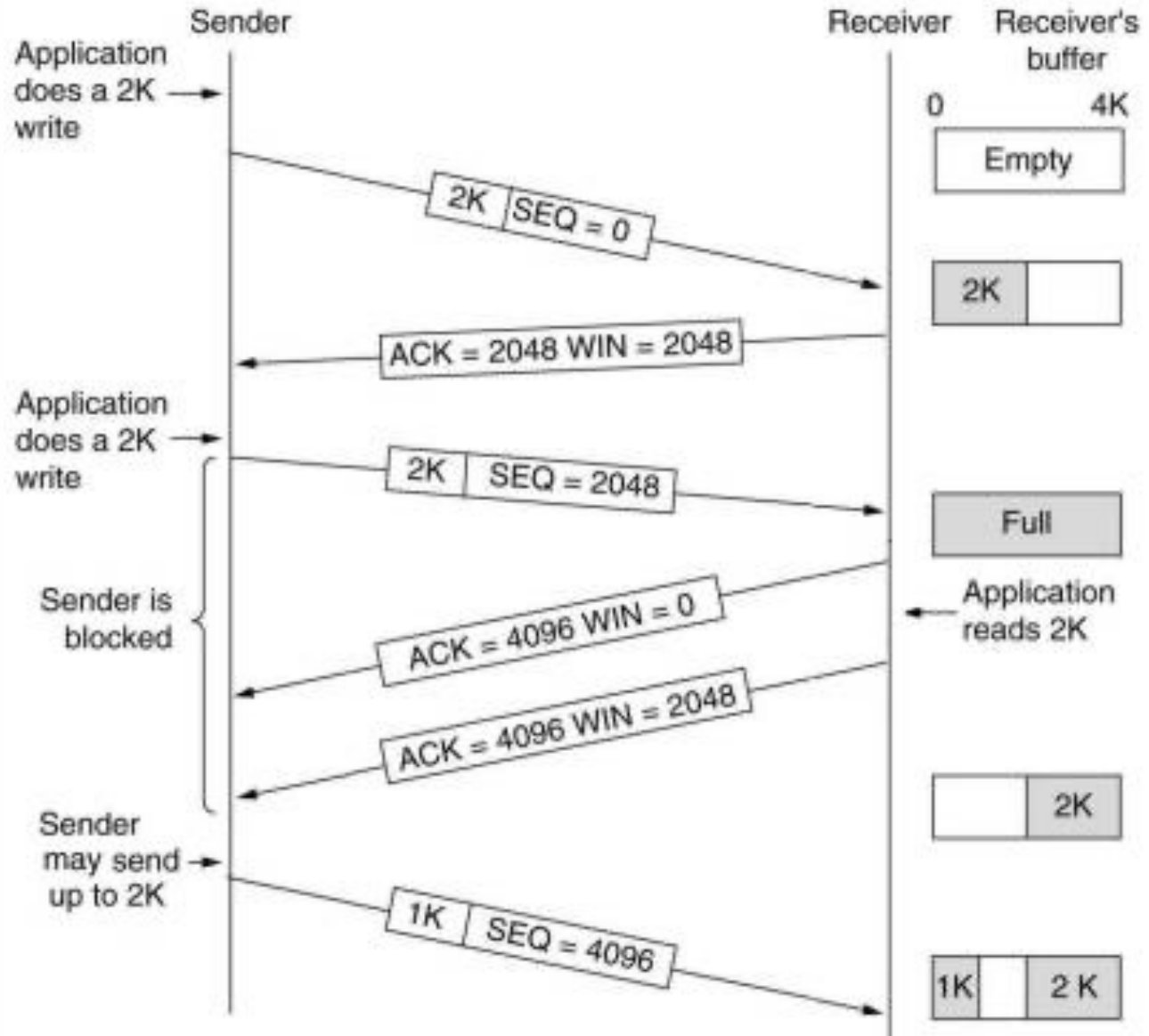


Window Size определяет количество байт данных (payload), после передачи которых ожидается подтверждение от получателя.

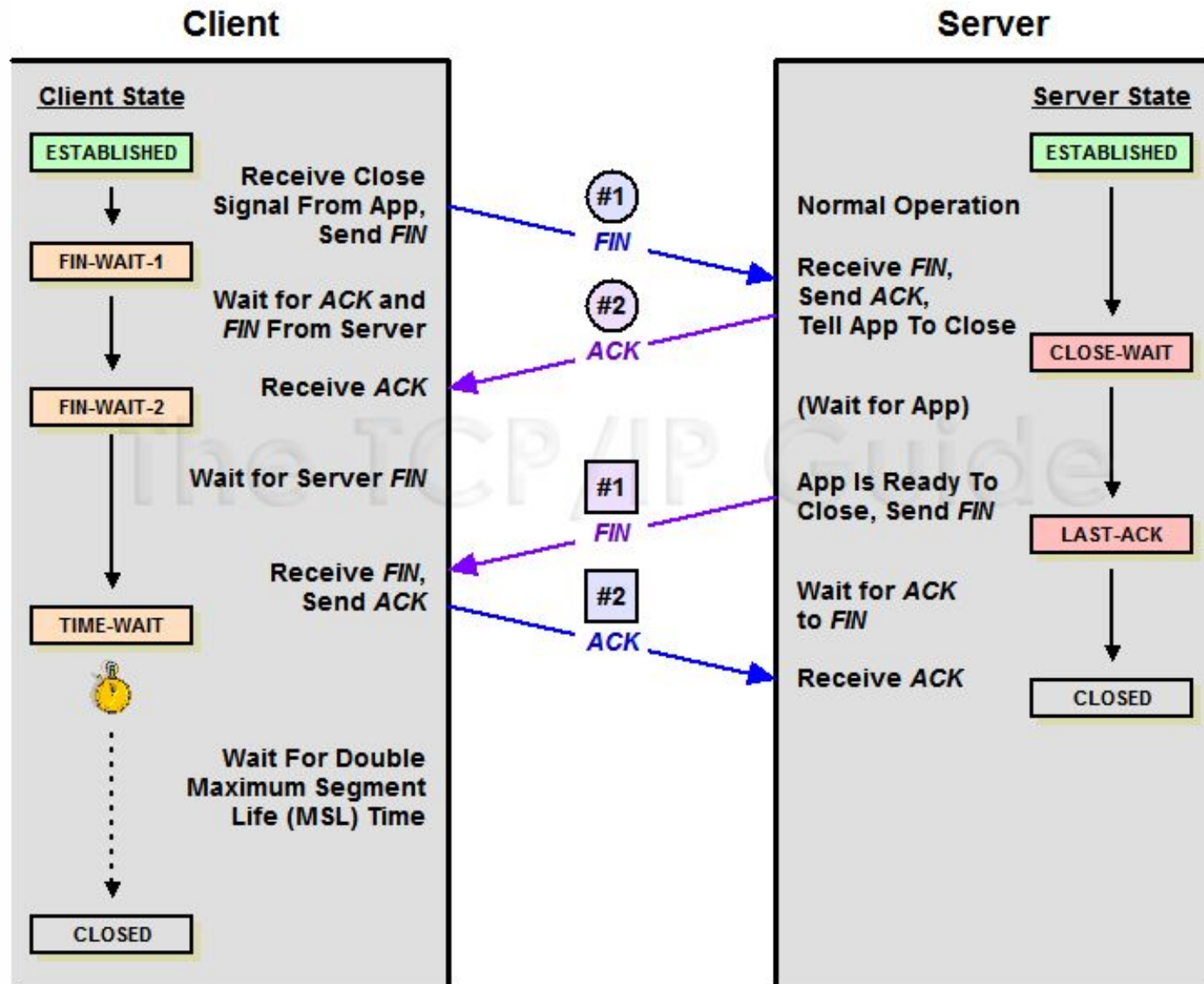
TCP data sending



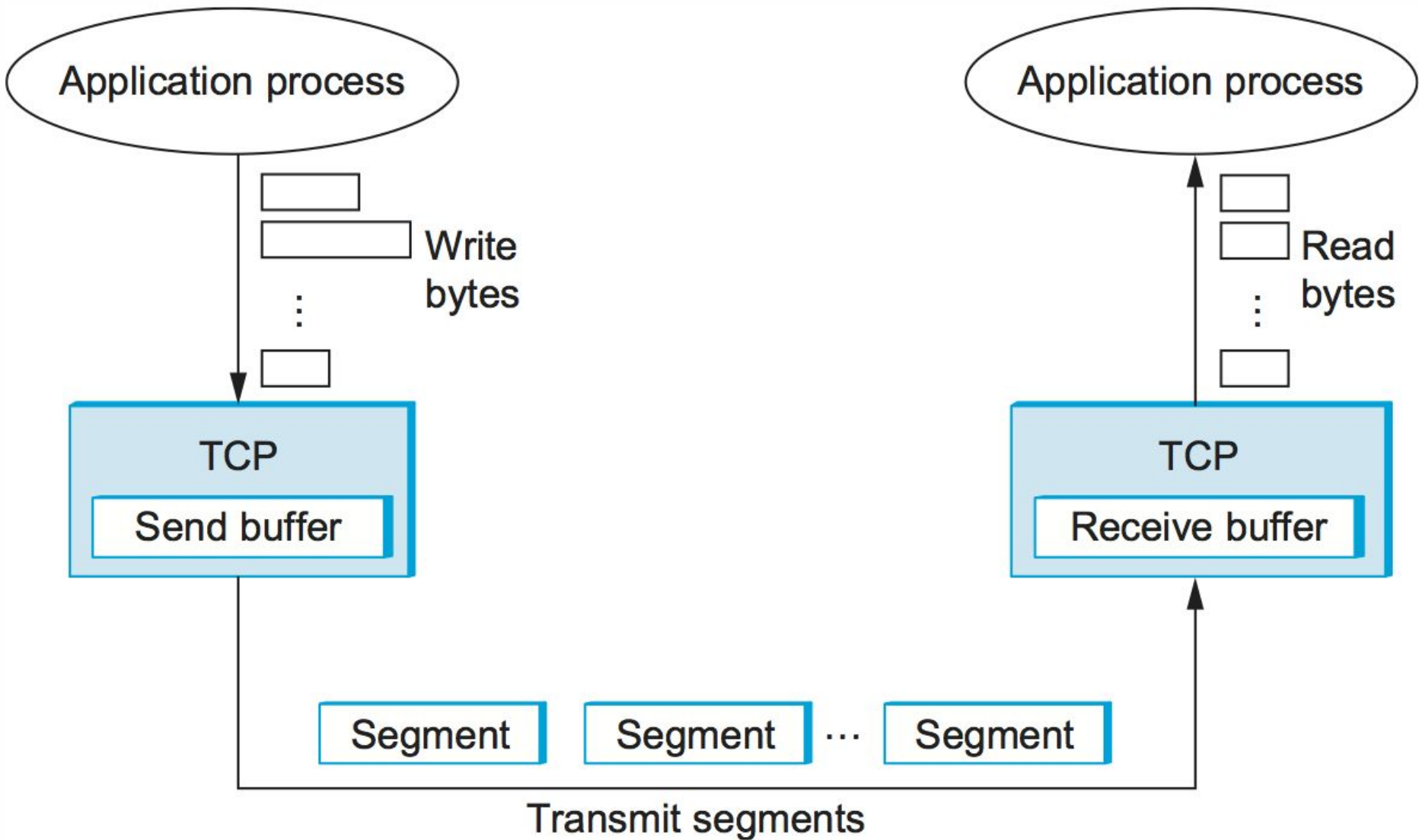
TCP window size = 0



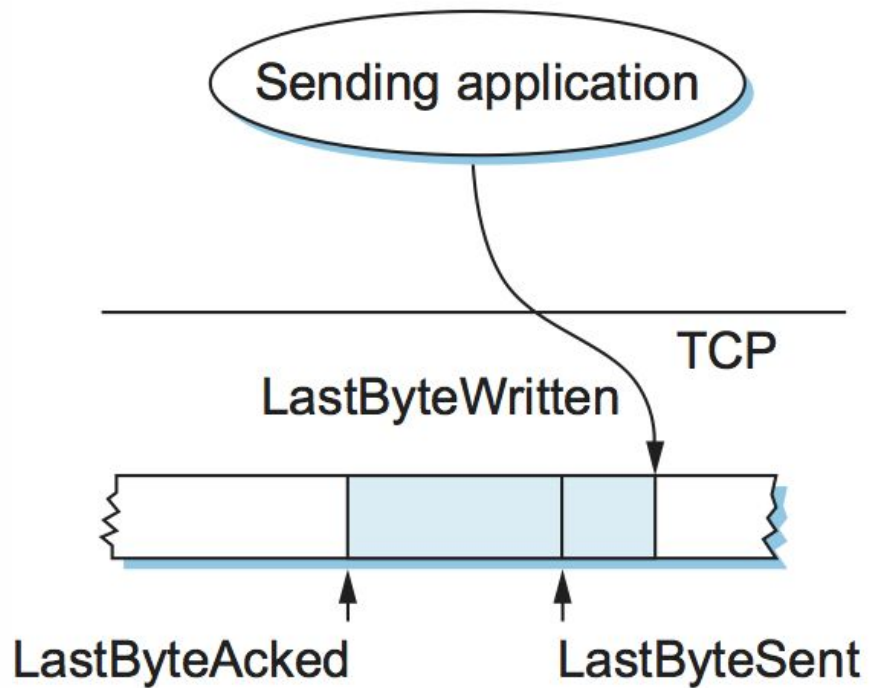
TCP closing connection



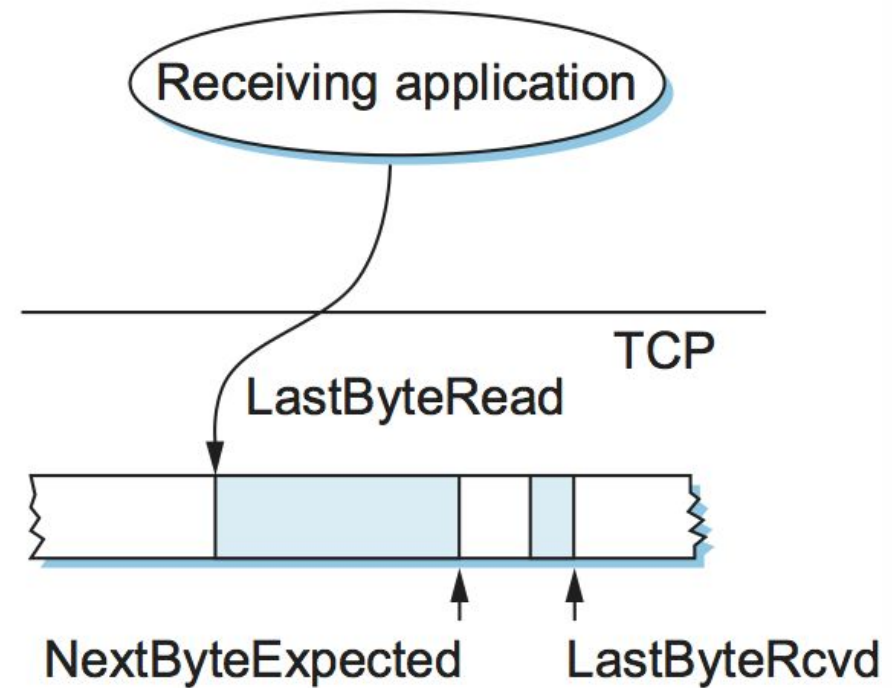
TCP buffers



(a)



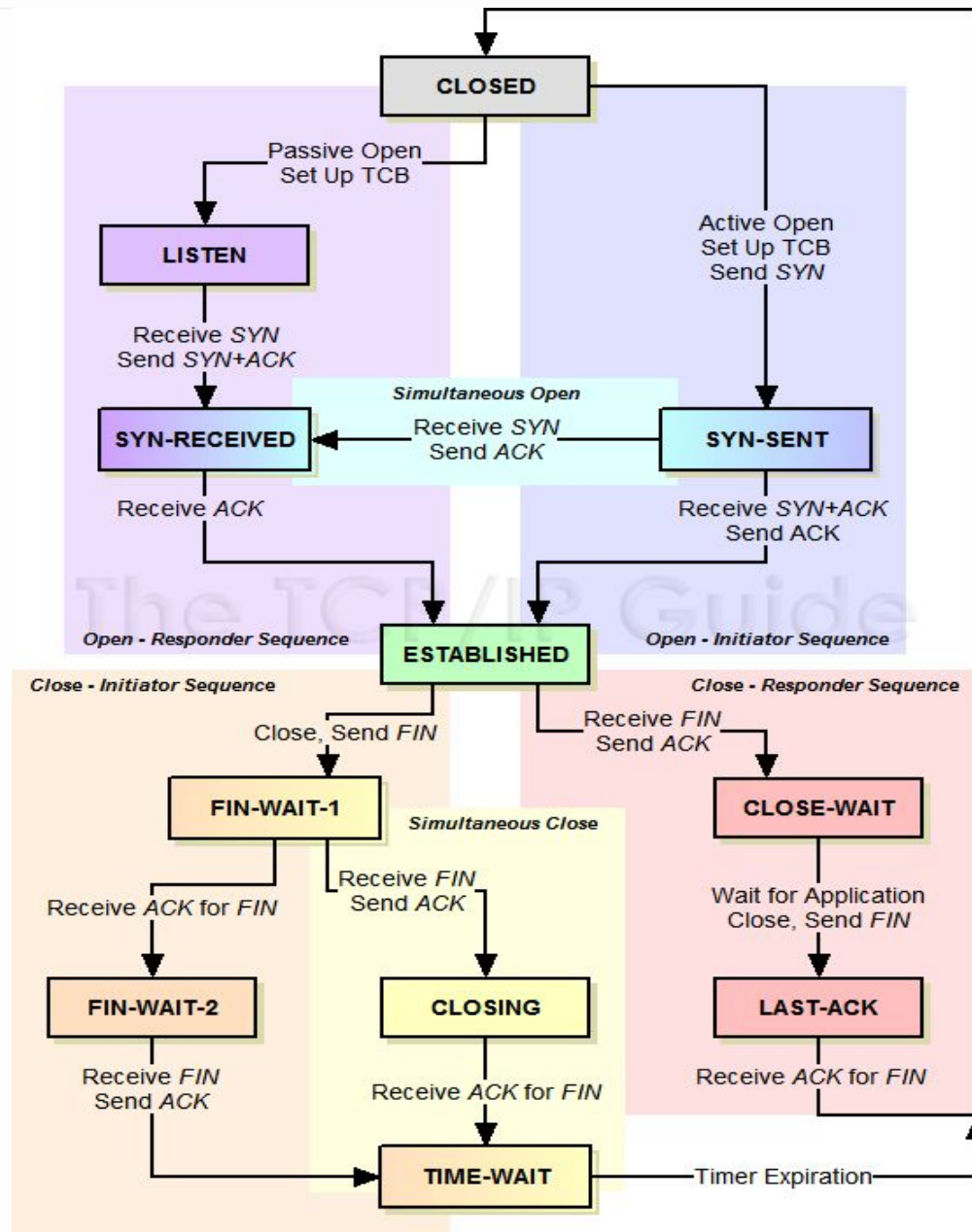
(b)



- 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)

TCP state diagram



[документация по параметрам](#)

- net.ipv4.conf.all.accept_redirects = 0
- net.ipv4.conf.all.secure_redirects = 0
- net.ipv4.conf.all.send_redirects = 0
- net.ipv4.tcp_max_orphans = 65536
- net.ipv4.tcp_orphan_retries = 0
- net.ipv4.conf.all.rp_filter = 1
- net.ipv4.conf.all.accept_source_route = 0
- net.ipv4.tcp_rfc1337 = 1
- net.ipv4.tcp_max_tw_buckets = 720000
- net.ipv4.ip_forward = 0
- net.ipv4.icmp_echo_ignore_broadcasts = 1
- net.ipv4.icmp_echo_ignore_all = 1

- - net.ipv4.tcp_fin_timeout = 10
- - net.ipv4.tcp_keepalive_time = 1800
- - net.ipv4.tcp_keepalive_intvl = 15
- - net.ipv4.tcp_keepalive_probes = 5
- - net.ipv4.tcp_max_syn_backlog = 4096
- - net.ipv4.tcp_synack_retries = 1
- - net.ipv4.netfilter.ip_conntrack_max = 16777216
- - net.ipv4.tcp_timestamps = 1
- - net.ipv4.tcp_sack = 1
- - net.ipv4.tcp_fastopen = 1
- - net.ipv4.tcp_slow_start_after_idle = 1
- - net.ipv4.tcp_congestion_control = htcp

- net.ipv4.tcp_no_metrics_save = 1
- net.ipv4.ip_local_port_range = 1024 65535
- net.ipv4.tcp_window_scaling = 1
- net.core.somaxconn = 65535
- net.core.netdev_max_backlog = 1000
- fs.file-max = 64000
- net.ipv4.tcp_mem = 50576 64768 98152
- net.ipv4.tcp_rmem = 4096 87380 16777216
- net.ipv4.tcp_wmem = 4096 65536 16777216

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!**

