

Инфраструктура вычислений в биоинформатике

Лекция 4. Компьютерные сети. Сетевой и
канальный уровни. Интерфейсы, local host.

Алгоритмы сетевой безопасности.

Сетевой уровень

Протоколы этого уровня во всех устройствах по сети.

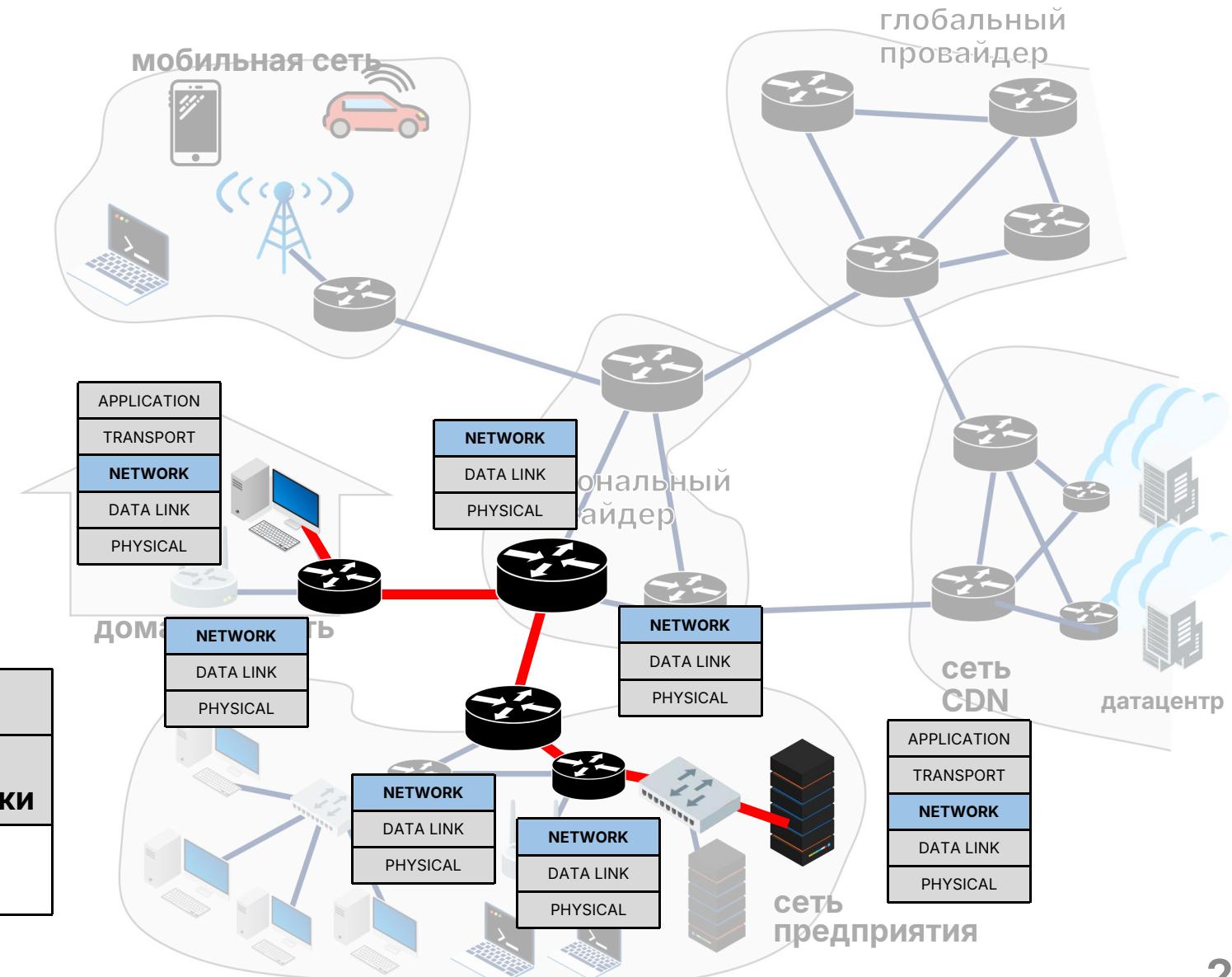
Основная масса устройств - маршрутизаторы (роутеры).

Функции: перенаправление (local) и маршрутизация (global).

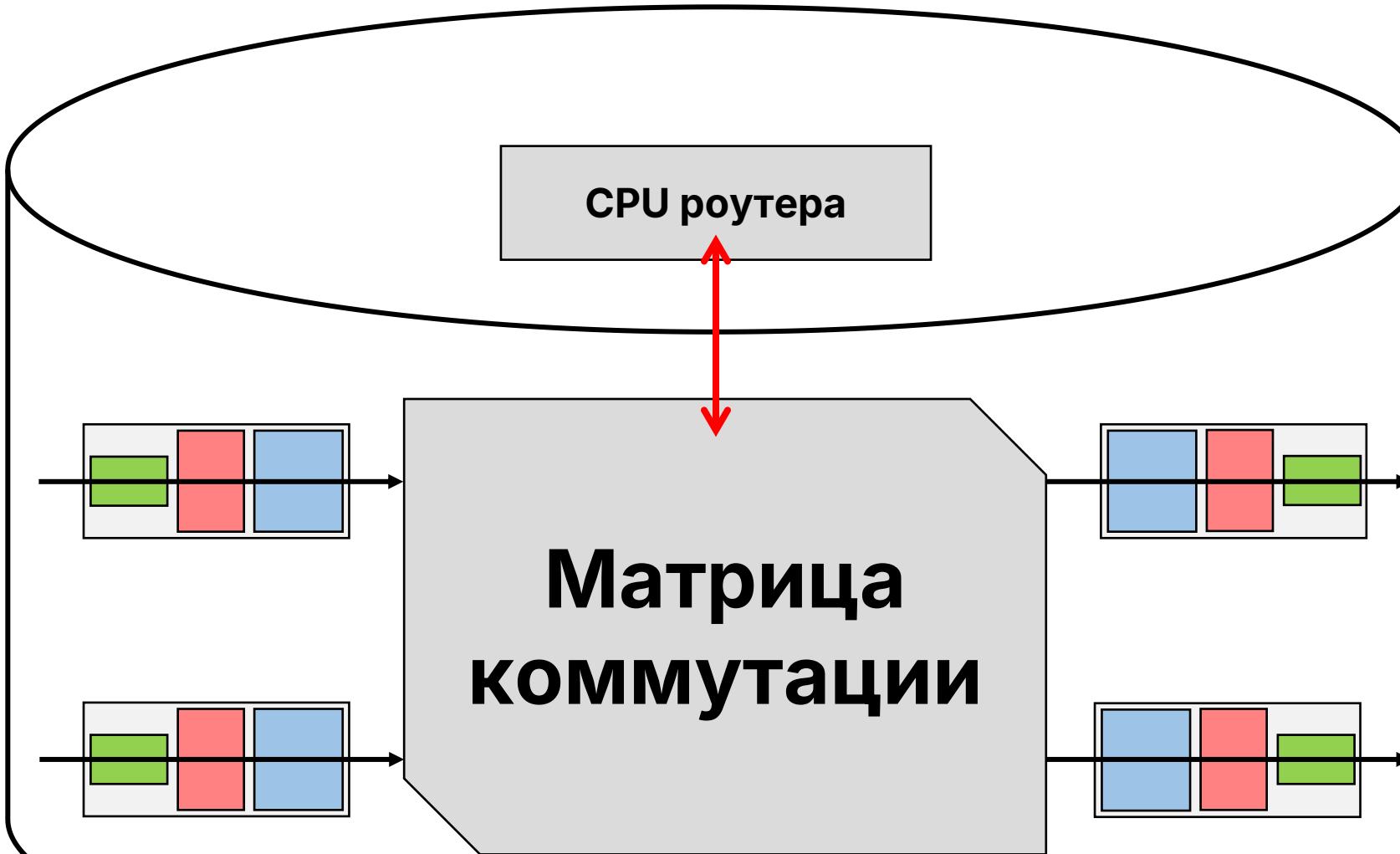
Модель "best-effort":

Модель	Гарантии обслуживания			
	ширина канала	отсутствие потерь	порядок доставки	срок доставки
best effort	Нет	Нет	Нет	Нет

Другие модели: ATM, Diffserv, ...



Что внутри роутера?



- Конец/начало линии (физический уровень)
- Приемник/передатчик (канальный уровень)
- Входная/выходная очередь коммутации

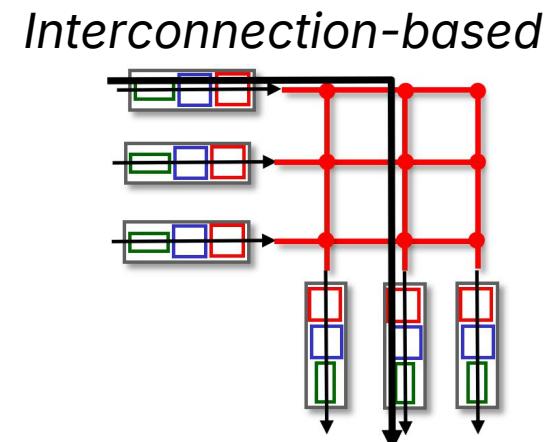
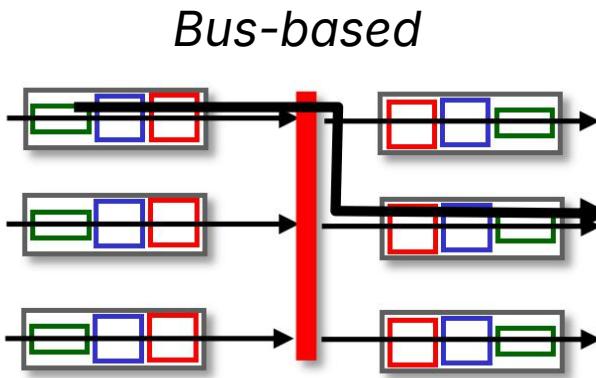
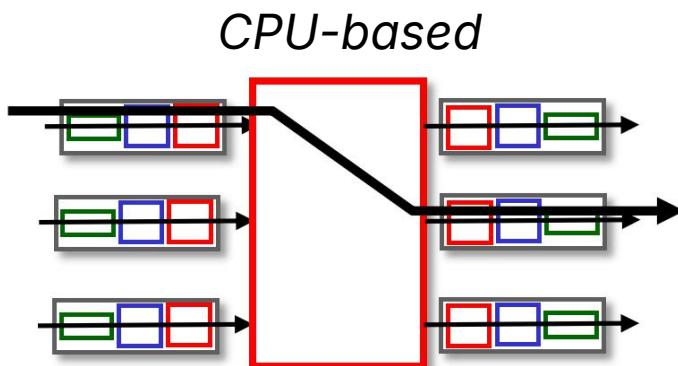
Внутри входной очереди коммутации процесс поиска выходного порта роутера, на который надо направить пакет. Подбор по наибольшему префиксу

Матрица и таблица коммутации

Внутри входной очереди коммутации процесс поиска выходного порта роутера, на который надо направить пакет. Подбор по наибольшему префиксу

IP range	Out port interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2

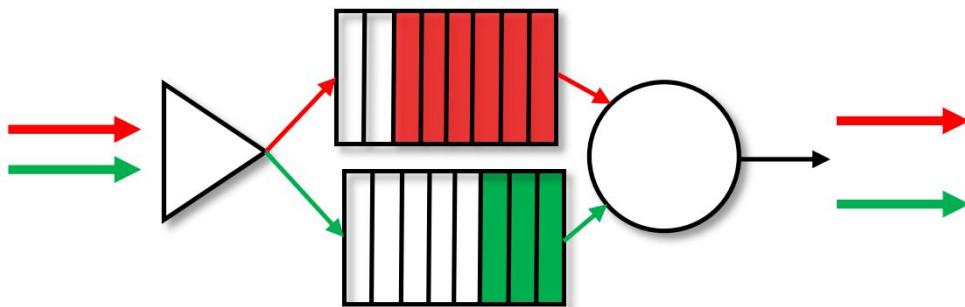
Таргетный IP-адрес на железе сравнивается с матрицей и за один такт находится наибольший префикс (TCAM-интерфейс)



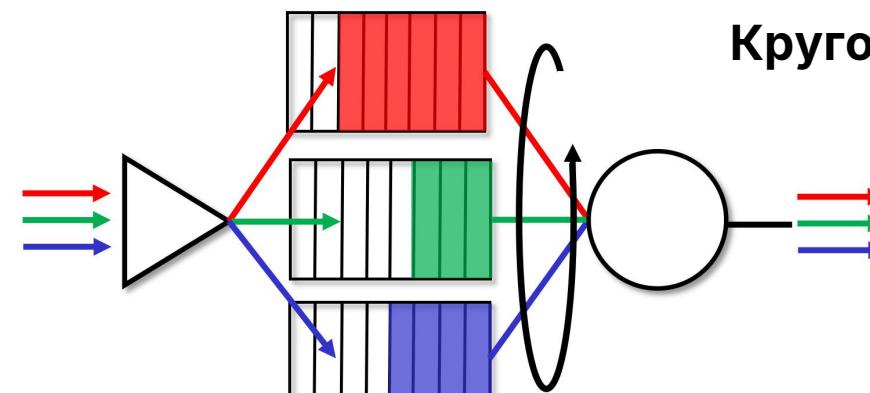
Задержки в буфере

На выходном интерфейсе скорость м.б. ниже скорости матрицы коммутации. Возникают задержки и буферизация. Решения:

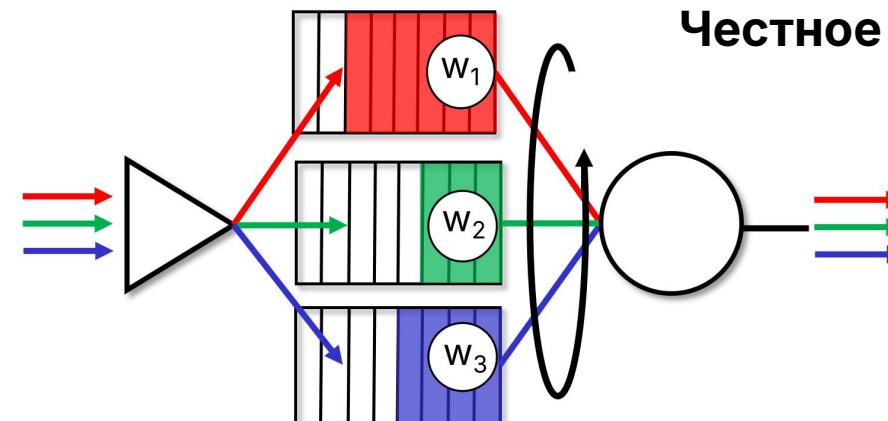
Приоритет одних пакетов над другими



Круговое распределение



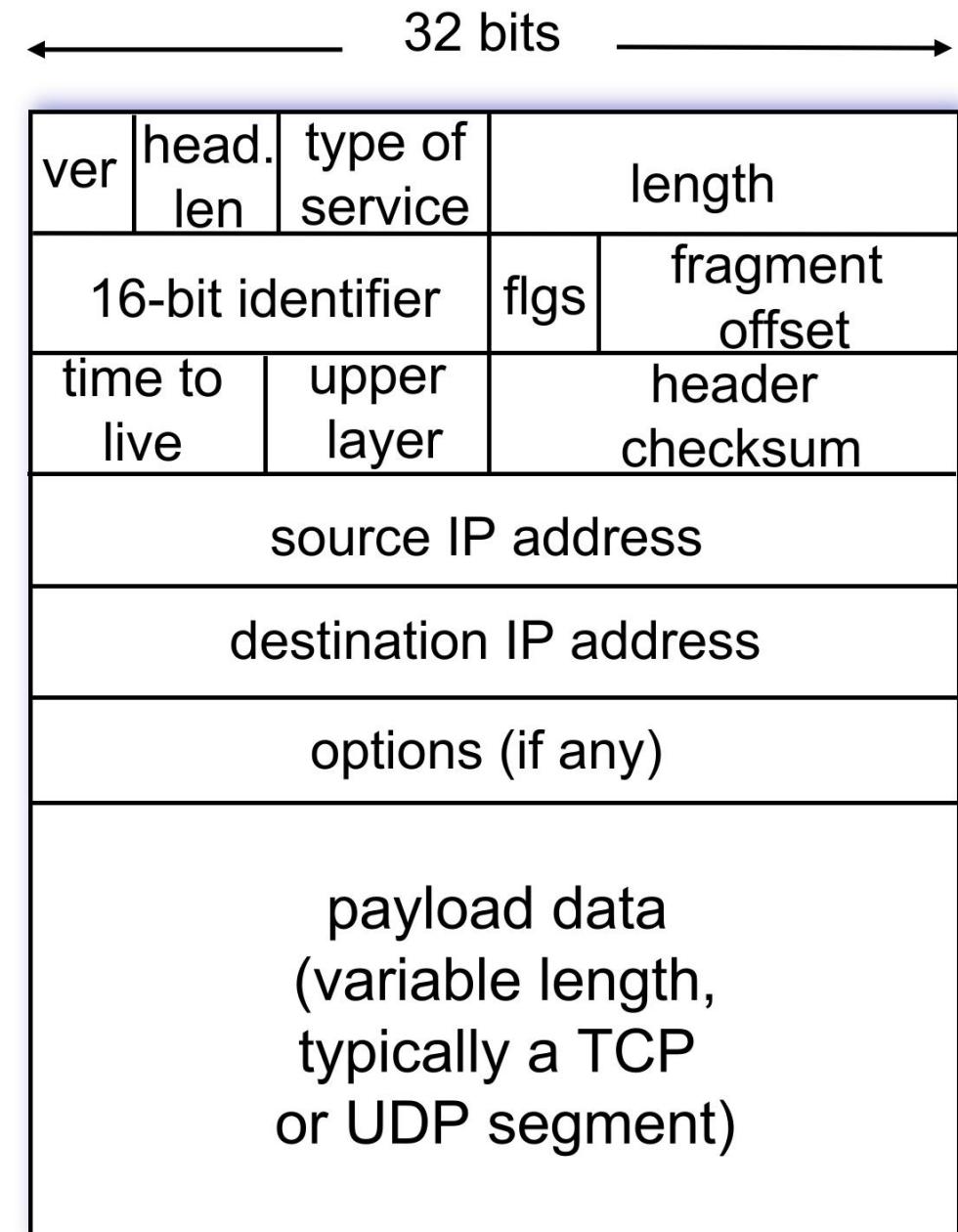
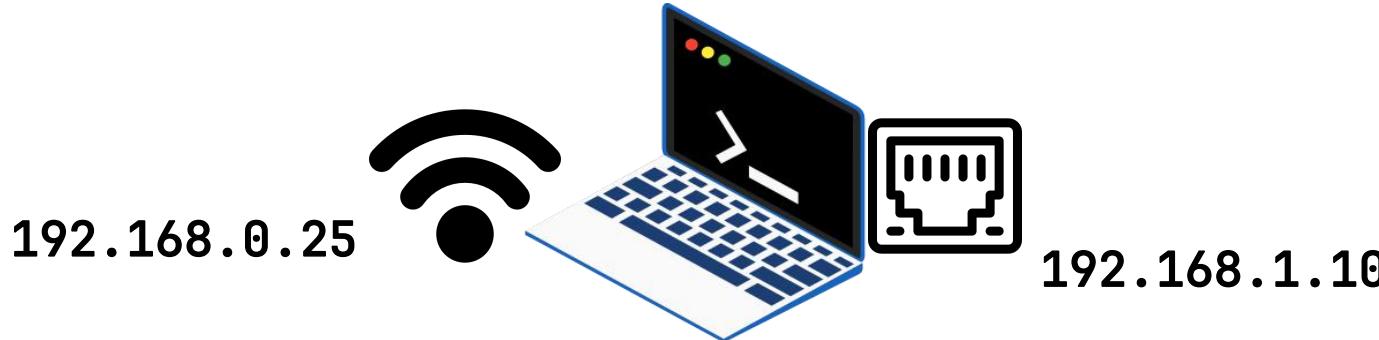
Честное взвешенное (WFQ)



Протокол IP

Internet Protocol. Обычно датаграмма ~ 1500 байтов, из которых 40 - заголовки TCP + IP. TTL - время жизни, контрольная сумма может отсутствовать в версии IPv6.

IP-адрес определяет не устройство, а физ. интерфейс передачи данных (NIC, WiFi модуль). У одного хоста может быть несколько интерфейсов → IP-адресов.



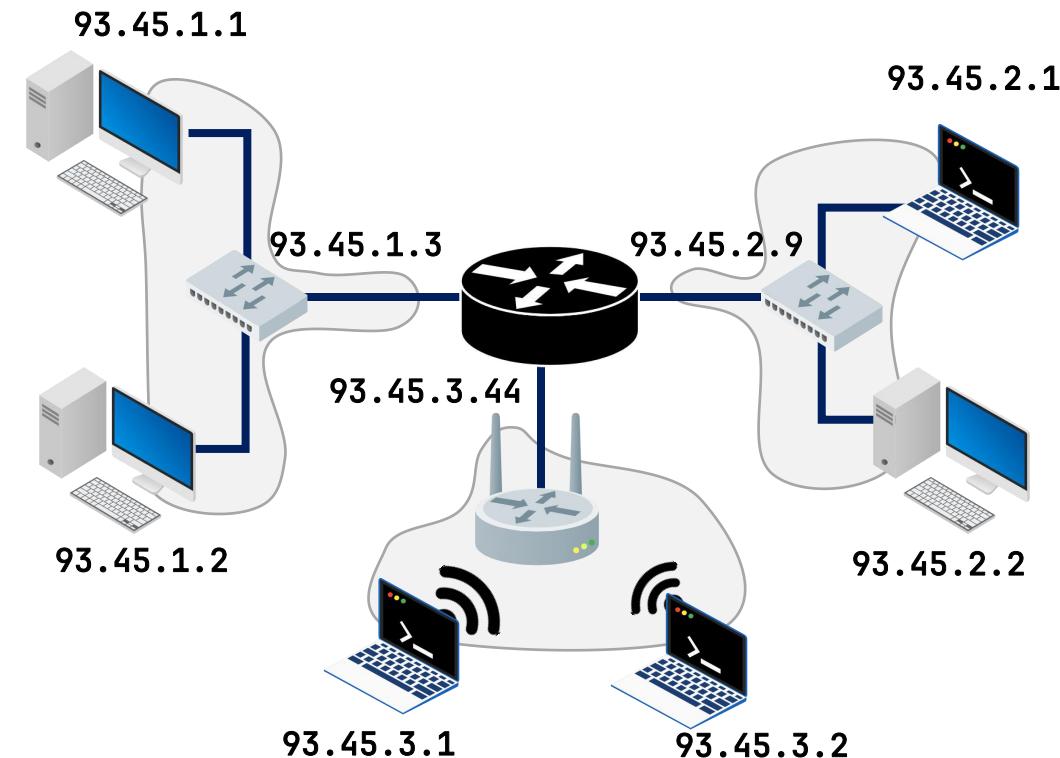
Подсеть и маски подсети

Связанные интерфейсы имеют похожие IP. **Подсеть** - интерфейсы, маршрутизация между которыми не требует прохода по роутеру.

На картинке 3 подсети:

- 93.45.1.0/24
- 93.45.2.0/24
- 93.45.3.0/24

Форма записи - Classless InterDomain Routing (CIDR). Записывается IP адрес, зануляются различающиеся части, через "/" пишут количество общих битов префикса



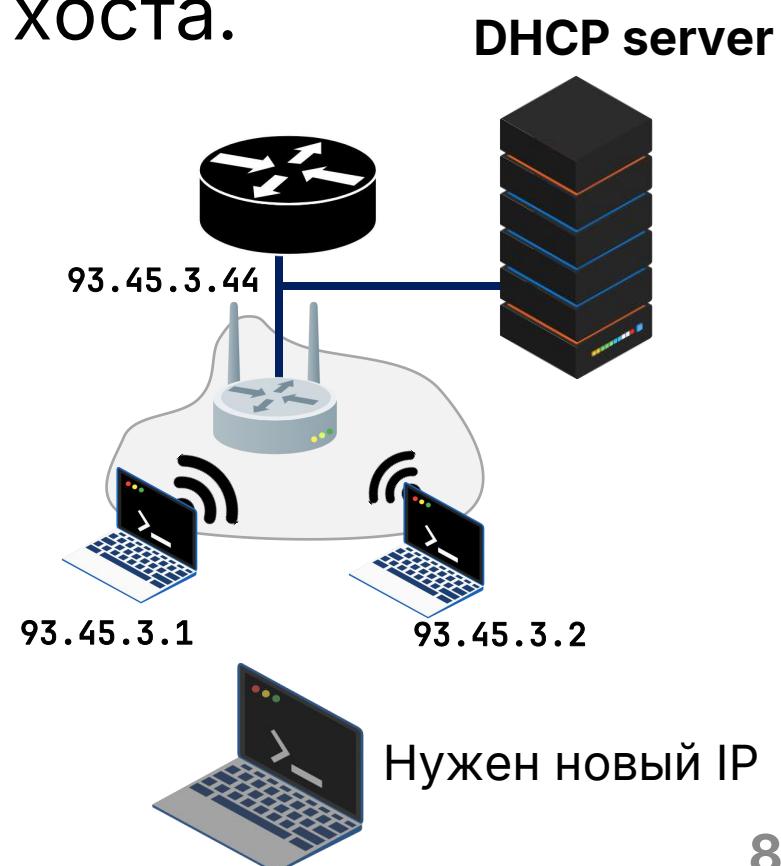
Как получить IP-адрес?

Подсеть получает доступные адреса от провайдера (ISP от ICANN).

А как хосту получить IP-адрес? Протокол **DHCP**!

Dynamic Host Configuration Pосвобождение IP-адреса для (не)постоянного хоста.

1. **DHCP discover** - широковещательная отправка UDP-сообщения "поиск DHCP сервера" на порт 67 всем машинам в подсети
2. **DHCP offer** - сервер широковещательно отправляет свободный IP-адрес клиенту на порт 68 (м.б. несколько)
3. **DHCP request** - клиент широковещательно отправляет свой предполагаемый адрес на порт 67
4. **DHCP ACK** - DHCP-сервер широковещательно соглашается на адрес клиента



DHCP

Первые два шага могут и не происходить, если клиент просто обновляет использование своего IP.

255.255.255.255 - адрес широкого вещания. Такие пакеты принимаются всеми машинами, но реагируют на них только те, у которых на нужном порте стоит сервис.

DHCP сервер: 223.1.2.5

DHCP discover

src: 0.0.0.0, 68
dest: 255.255.255.255, 67
yiaddr: 0.0.0.0
transaction ID: 654

DHCP offer

src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddr: 223.1.2.4
transaction ID: 654
lifetime: 3600 secs

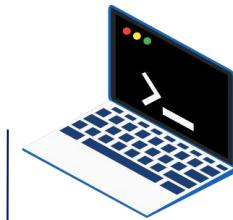
DHCP request

src: 0.0.0.0, 68
dest: 255.255.255.255, 67
yiaddr: 223.1.2.4
transaction ID: 655
lifetime: 3600 secs

DHCP ACK

src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddr: 223.1.2.4
transaction ID: 655
lifetime: 3600 secs

Клиент



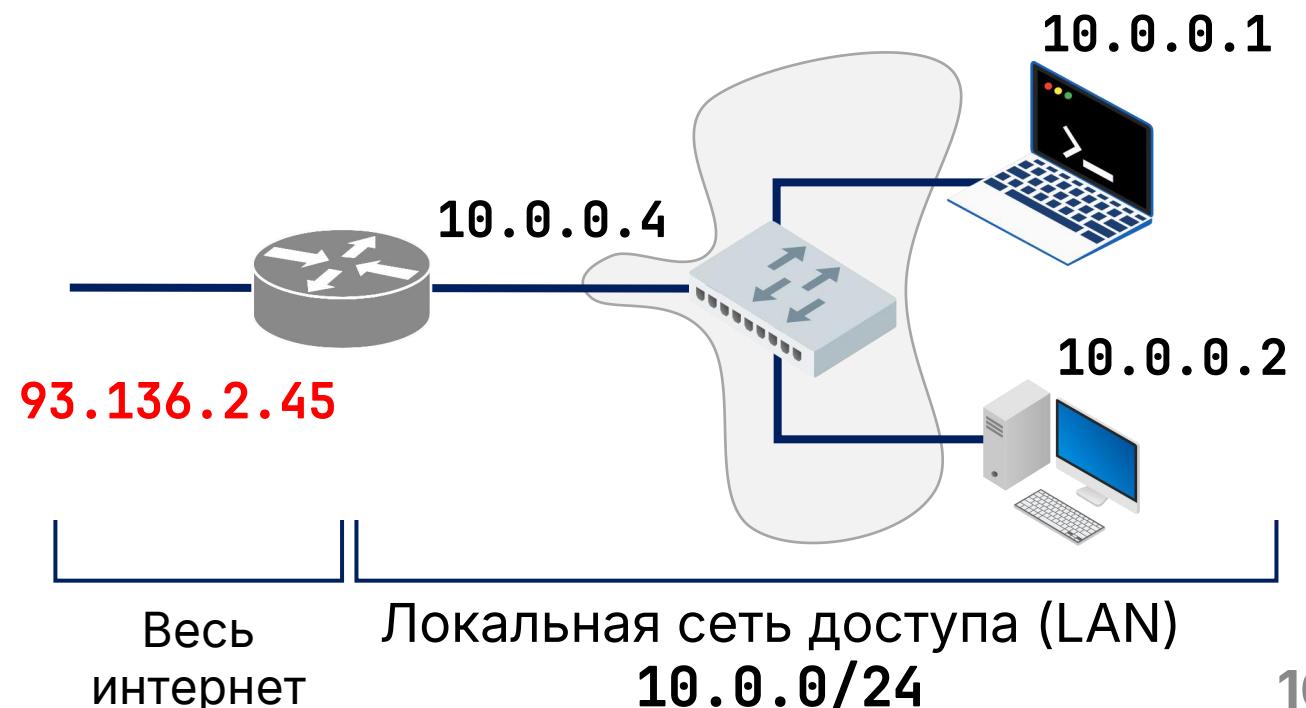
DHCP-сервер так же может возвращать IP-адрес DNS-сервера, первого граничного роутера, маску подсети

IP-адресация: NAT

Адреса иерархичны - распространение изменений быстрое - route aggregation.

В 2011 году IPv4-адреса кончились. Решения: IPv6 (128-битный адрес) и NAT (Network Address Translate)

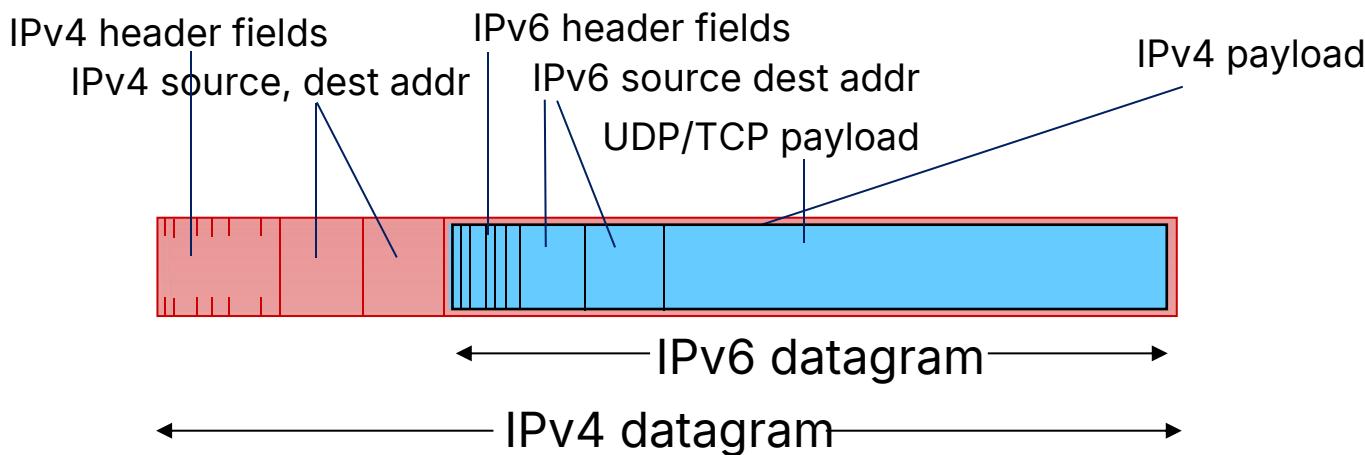
У всех исходящих датаграмм один и тот же IP (93.136.2.45), но разные номера портов.
Внутри локальной сети у всех устройств свой 32-битный адрес (e.g., 192.168/16).
Плюсы очевидны. Минусы?



IPv6

Нет контрольных сумм (и пересчёта),
фрагментации, опций. Фиксированный
хедер.

IPv6 туннелируется с помощью IPv4.



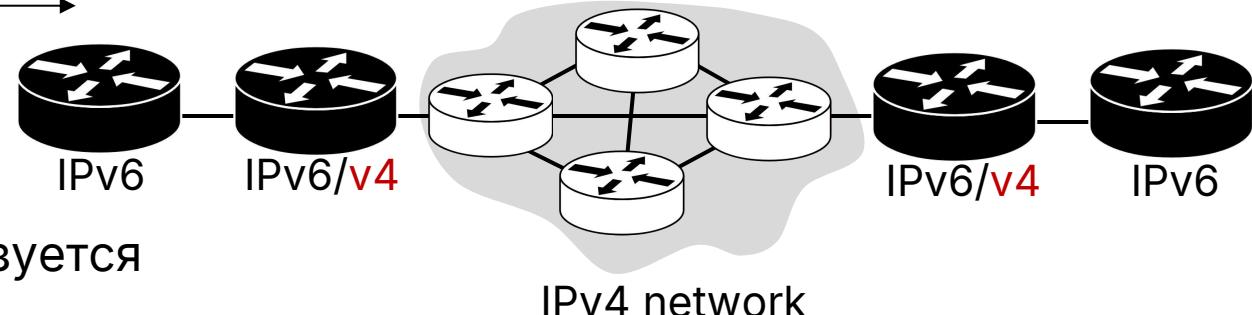
IPv6 ~45% устройств

SDN OpenFlow: любое поле из хедеров используется
(firewalls, forward, other actions, ...)

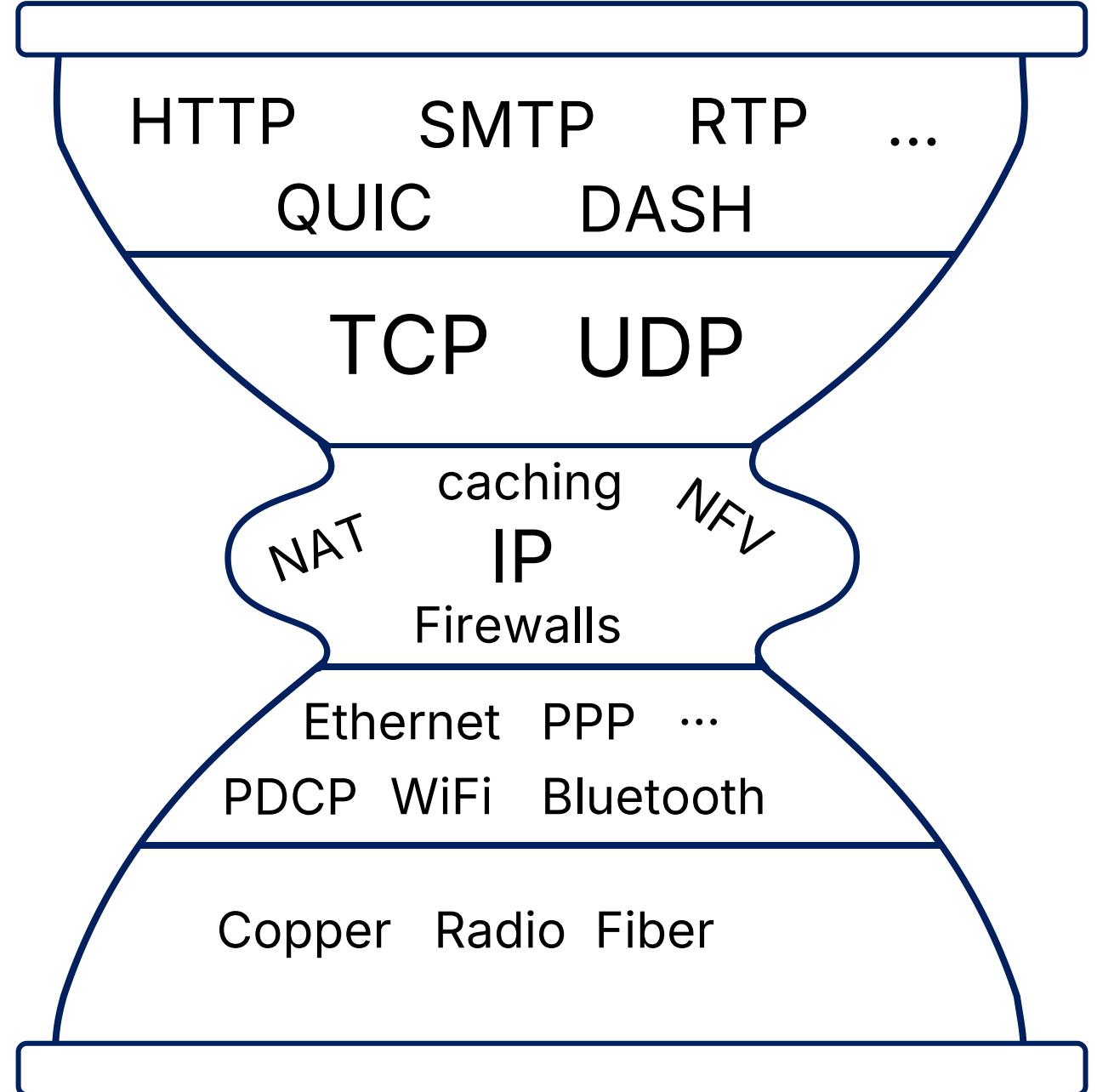
32 bits				
ver	pri	flow label		
		payload len	next hdr	hop limit
		source address (128 bits)		
		destination address (128 bits)		
payload (data)				

IPv6 дешевле IPv4, наводнён ботами

Сосуществование: роутеры с
двойной поддержкой туннелируют
датаграммы через IPv4 подсети

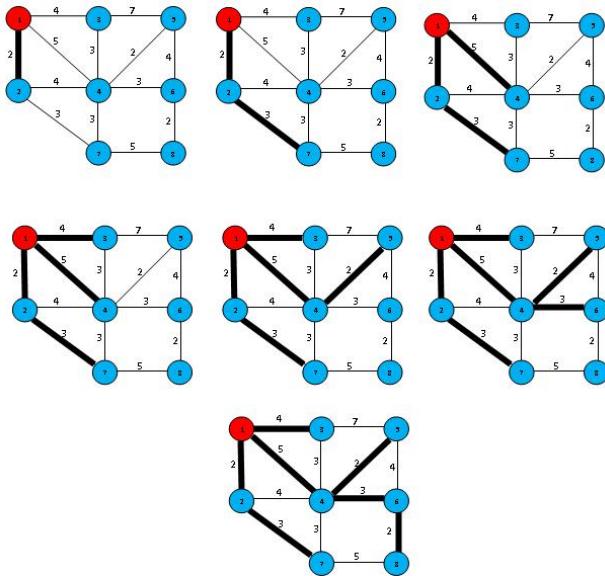


IP hourglass

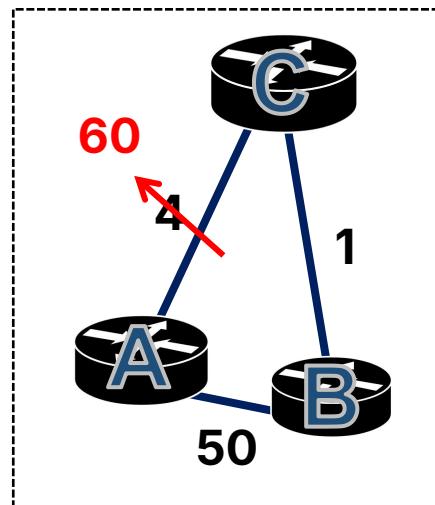


Алгоритмы маршрутизации

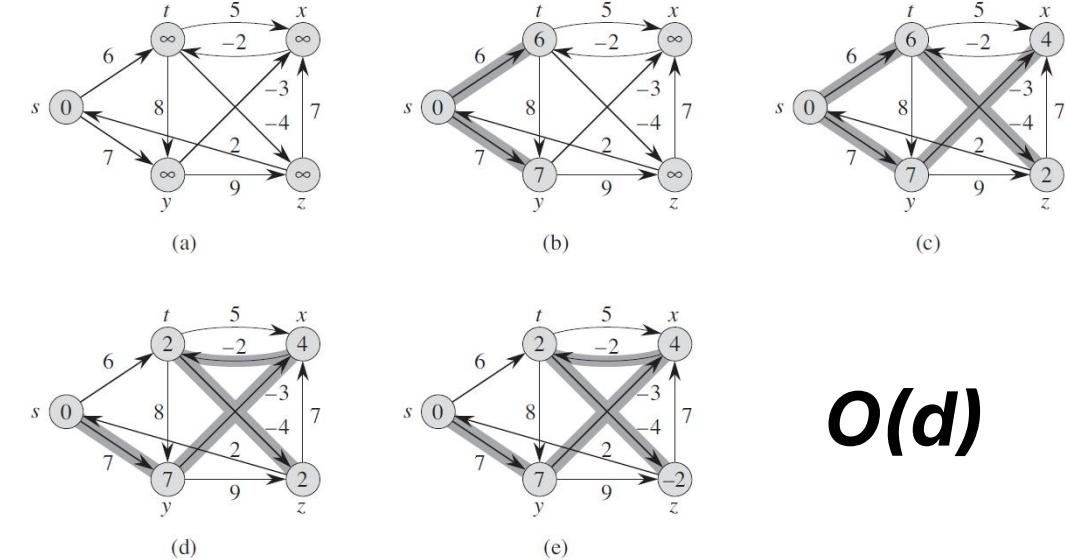
Link-state (OSPF) (алг. Дейкстры)



$O(V^2)$



Distance-vector (RIP, EIGRP, BGP) (алг. Беллмана-Форда)



$O(d)$

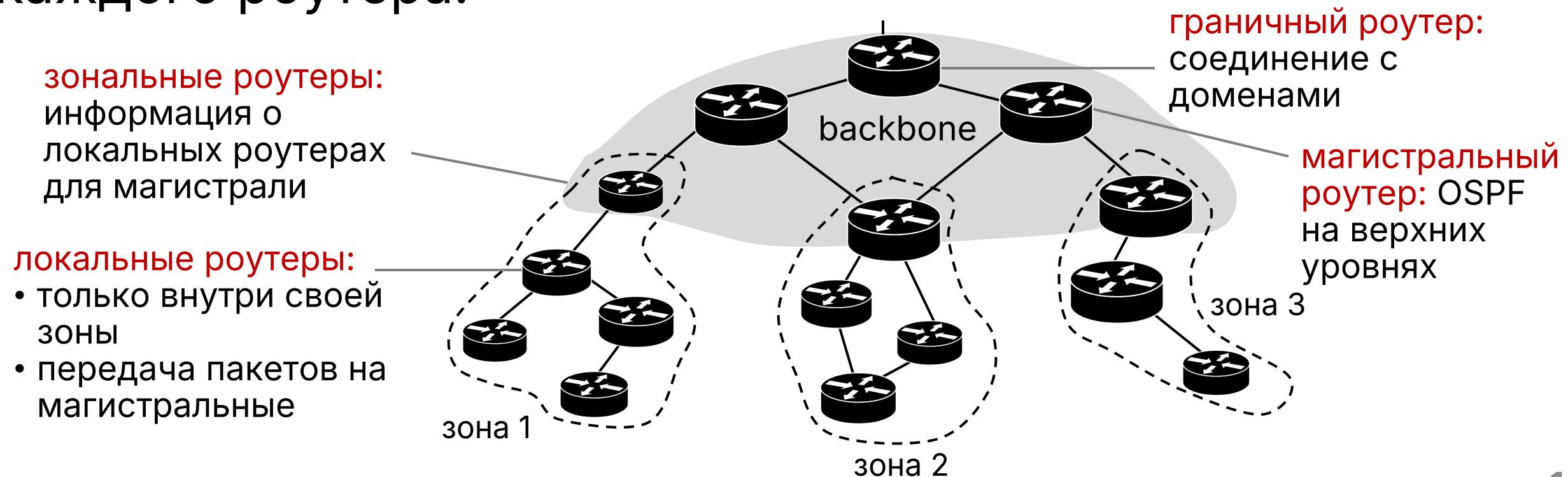
Плюсы: жадный, быстрее динамики
Минусы: централизованный, нужно
полное состояние сети,
осцилирование при изменении весов

Плюсы: распределённый, "good
news travel fast"
Минусы: плохо реагирует на
увеличение веса (count-to-infinity)

Intra-domain маршруты: OSPF

Open Shortest Path First

По IP каждый роутер передаёт информацию о своих связях так часто, что вся информация о домене есть у каждого роутера.



Inter-domain маршруты: BGP

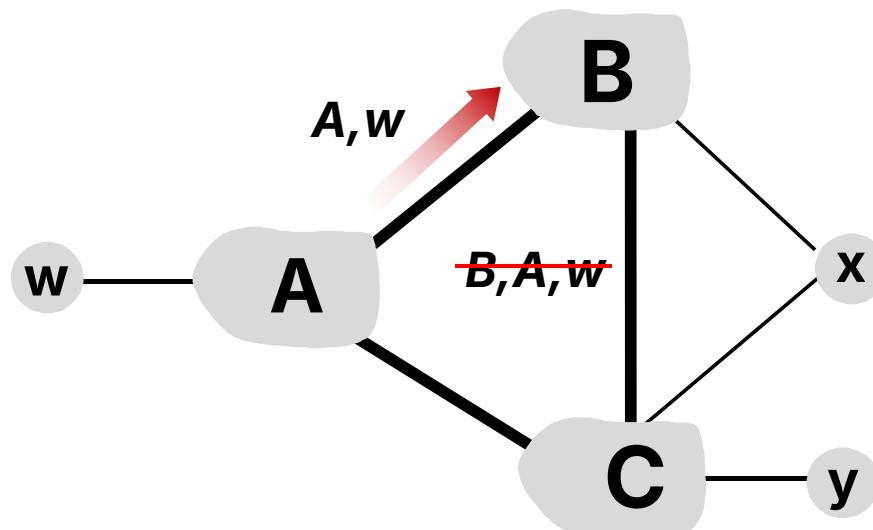
Border Gateway Protocol

Распределён. Каждый граничный роутер распространяет информацию о том, куда может привести. TCP по 179 порту.

eBGP - между доменами, **iBGP** - между граничными одного домена. Информация: путь между доменами.

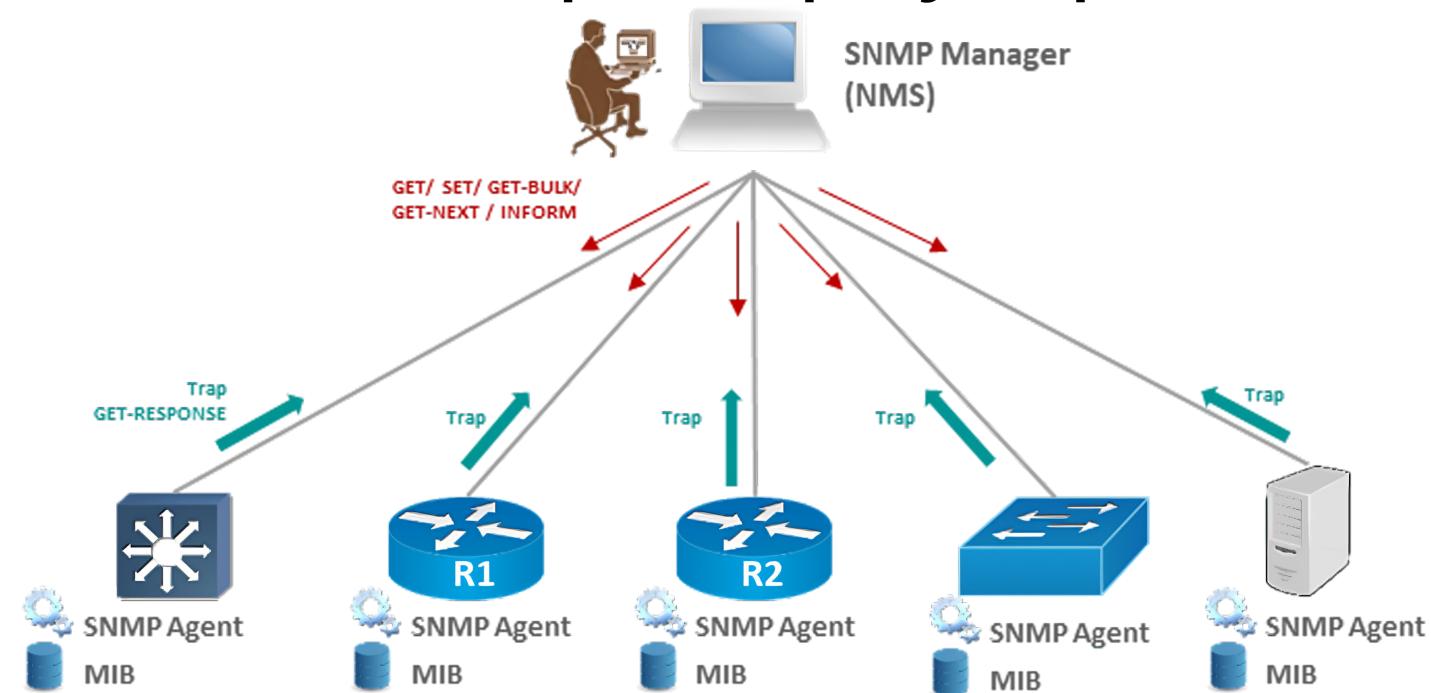
Policy-based: провайдерам нет нужды гонять в своей сети транзитный трафик.

Hot-potato routing: выбор того граничного роутера, до которого путь короче внутри домена.



ICMP и SNMP: управление сетью

- Internet Control Message Protocol - обмен информацией между роутерами (traceroute!)
- Simple Network Management Protocol - управление и мониторинг роутеров в сети



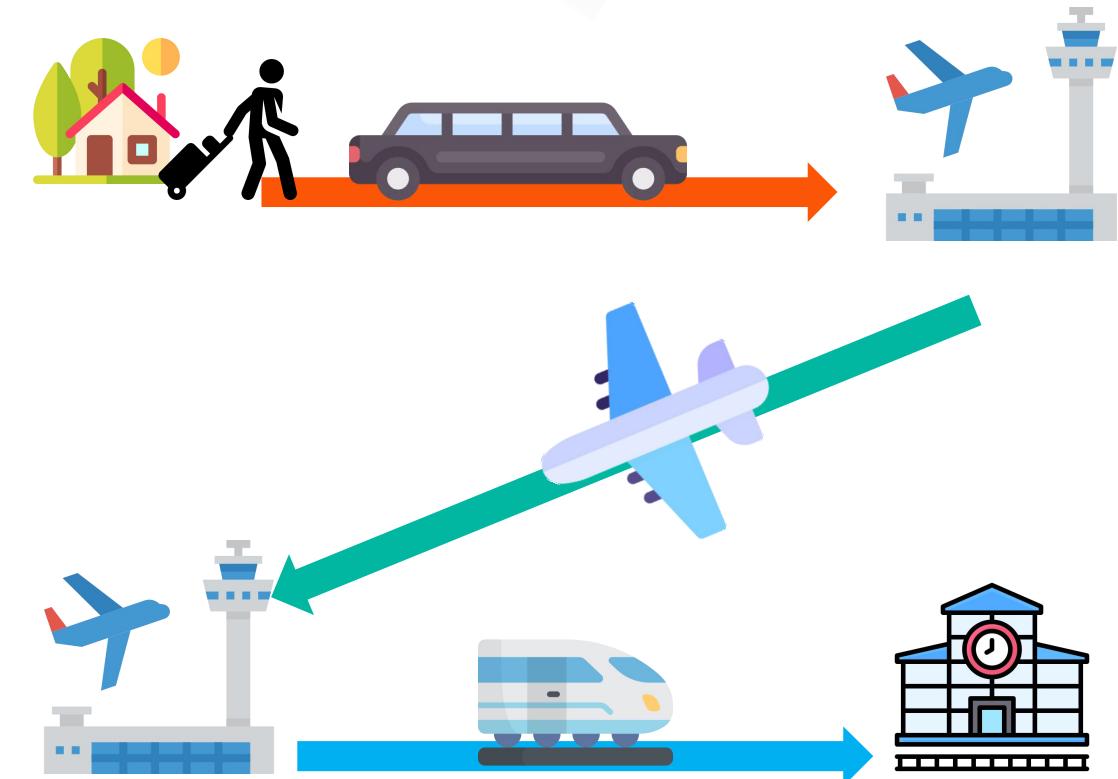
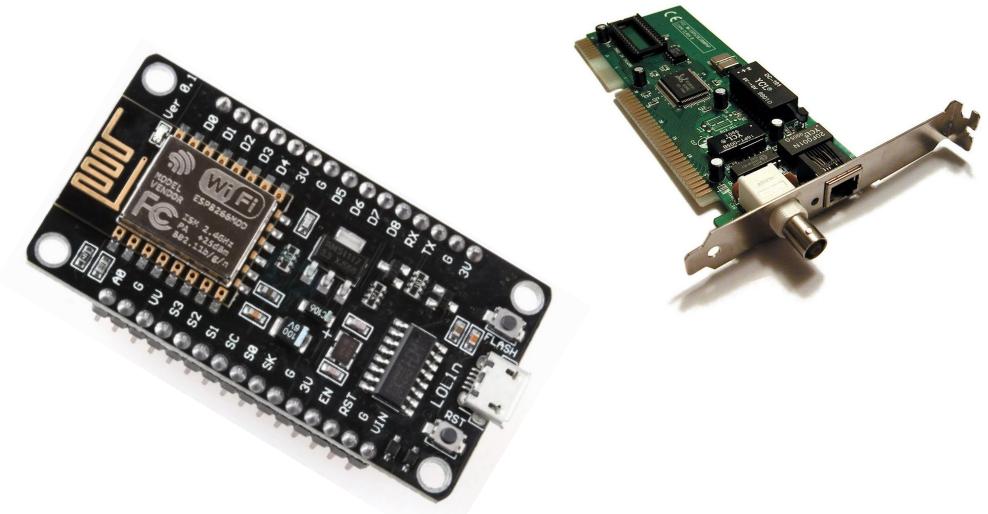
Канальный уровень

Функция: связь двух **соседних** устройств сети (роутеры, хосты)

Связь внутри LAN (по проводу или без, главное - без роутера!)

Разные протоколы (в т.ч. по среде передачи): WiFi (802.11), Ethernet, Bluetooth

Сервисы: инкапсуляция, множественный доступ, (надежная передача данных), (контроль потока), **MAC-адреса (Media ACcess)** интерфейсов - 48-битный адрес интерфейса



Error detection & Correction

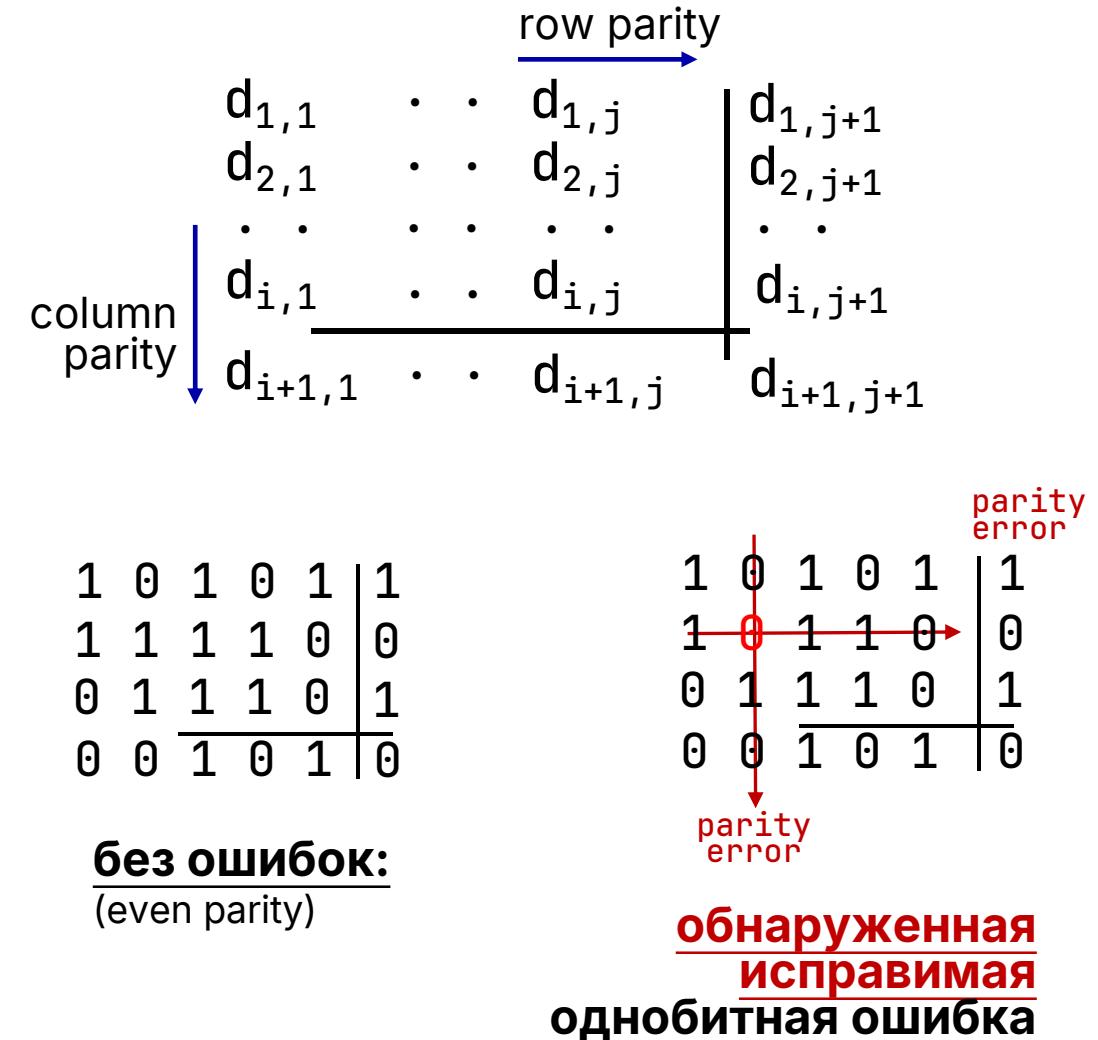
- 2D Parity check
- Internet checksums
- Cyclic Redundancy Check (CRC)

Выбирается некоторое G из $r+1$ бита. Для данных рассчитывается R такое, что $\langle Data, R \rangle$ делится на G нацело.

$$\langle Data, R \rangle = D * 2^r \text{ XOR } R$$

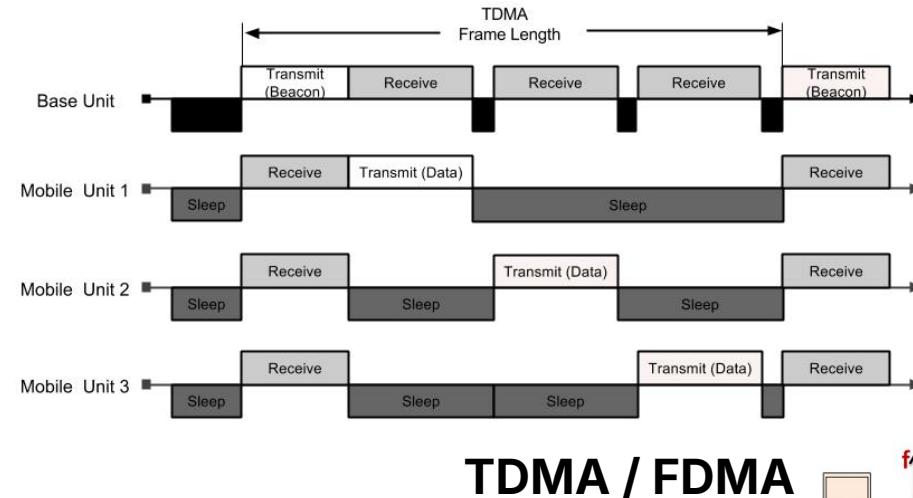
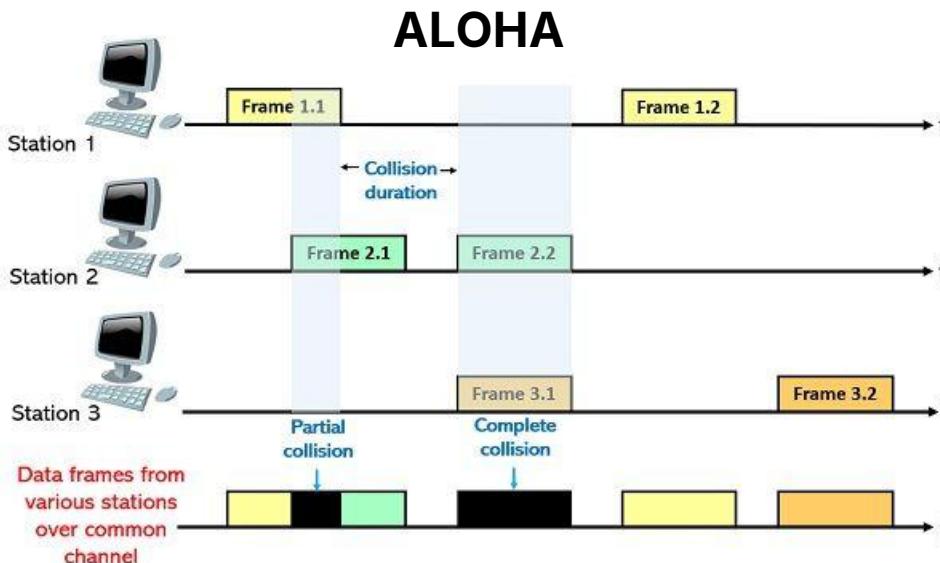
Получатель делит $\langle Data, R \rangle$ на G . Если ненулевой остаток, то кадр повреждён.

$$R = D * 2^r \bmod G$$

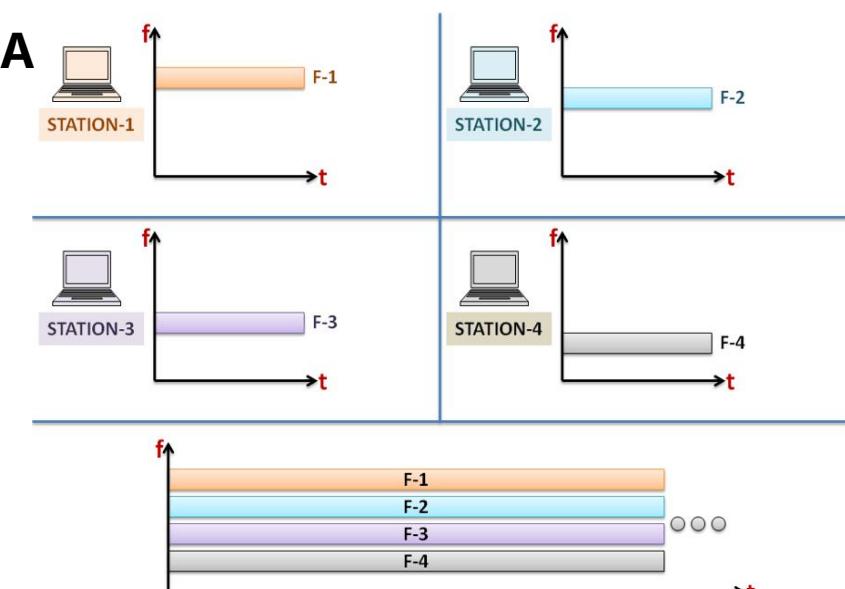


Множественный доступ

- Разделение канала
- Случайный доступ
- Поочередный доступ
- Прослушивание несущей



TDMA / FDMA



MAC-адрес

Пример:
1A-2F-BB-76-09-AD

У каждого интерфейса есть MAC-адрес и IP-адрес.

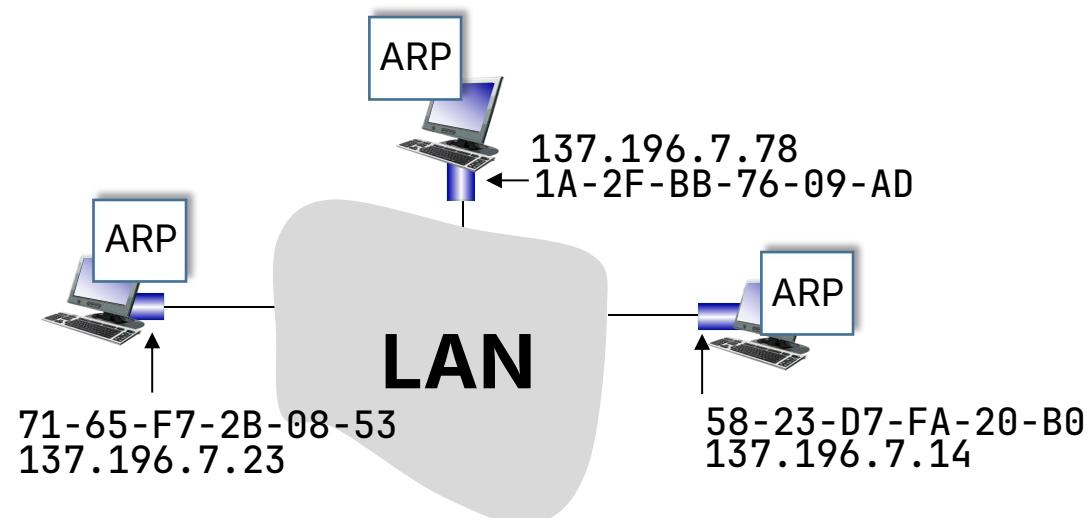
Нужны ОБА: IP для эффективной маршрутизации, MAC для уникальной идентификации.

MAC-адреса выдаются производителям в IEEE.

Широковещательный адрес: FF-FF-FF-FF-FF-FF.

Внутри подсети работает протокол ARP

(Address Resolution Protocol): у каждого адаптера интерфейса есть таблица



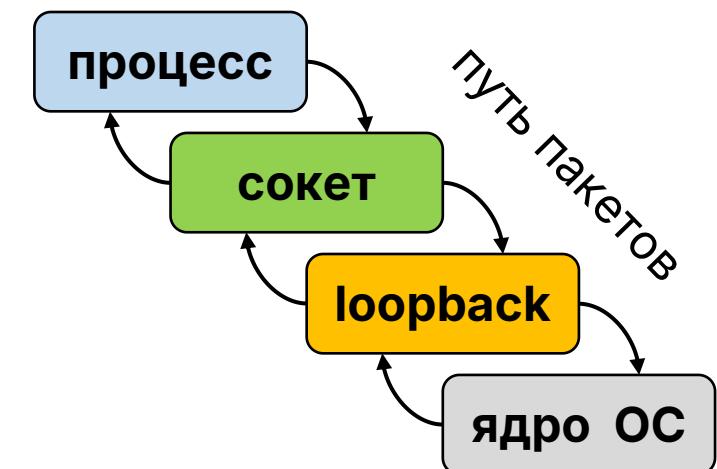
Виртуальный сетевой интерфейс

Для удобства в компьютерах абстракция, не имеющая физических адаптеров, но работающая как сетевой интерфейс

Примеры:

- **loopback**-интерфейс (localhost)
- **docker0** (дефолтный бридж docker)
- **veth0** (виртуальный ethernet между созданными сетями)

127.0.0.1 на любом ПК



localhost - DNS-имя вашего хоста

В UNIX общая сущность:

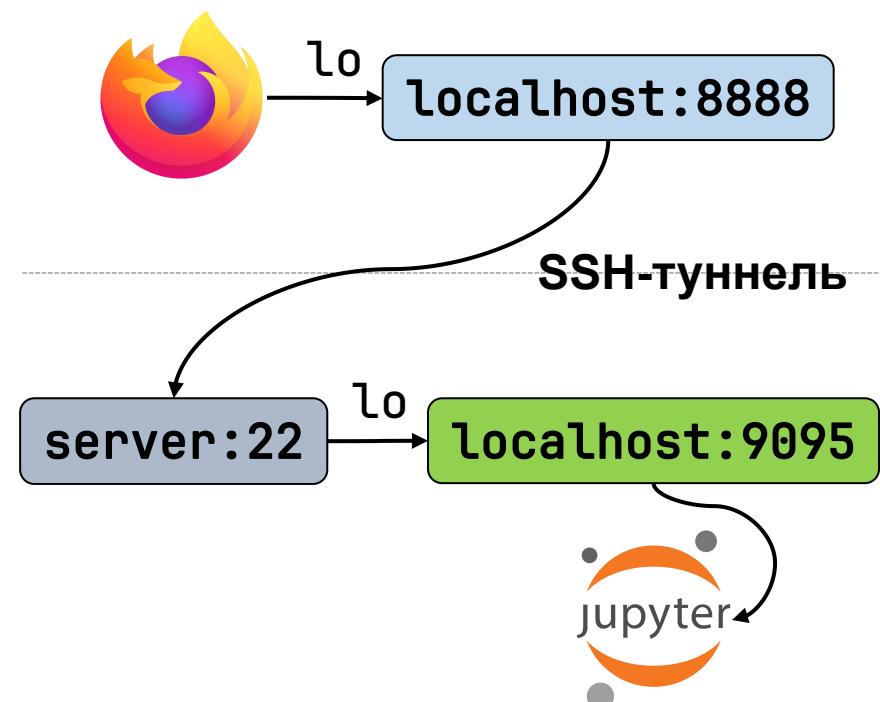
имя: localhost
адрес: 127.0.0.1
интерфейс: lo

Нет MAC-адреса, т.к.
это виртуализация
L3 (сетевого уровня)

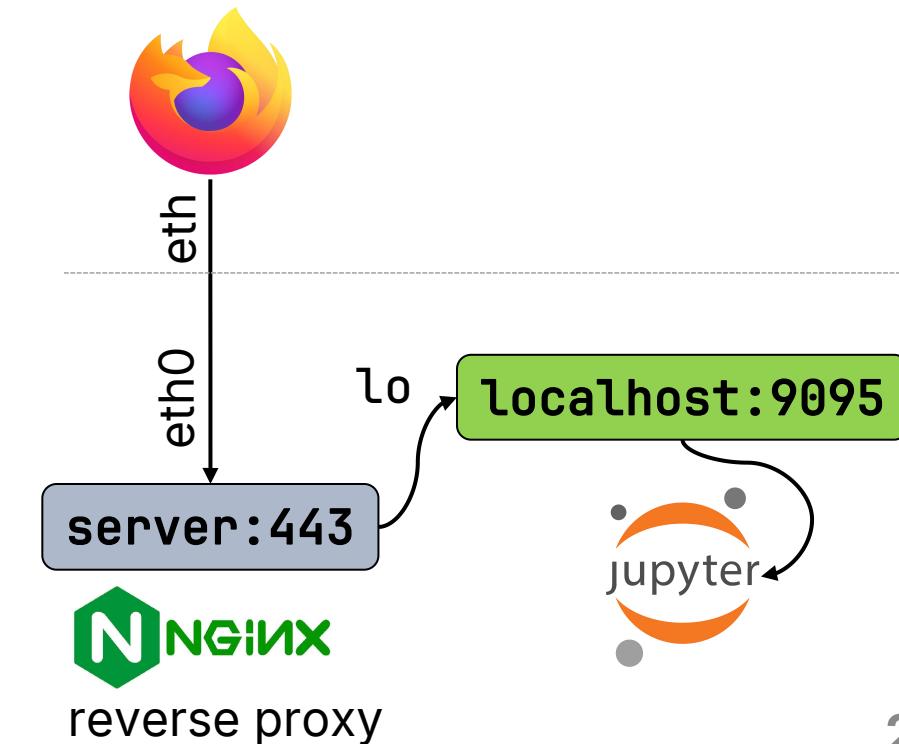
У docker0 эмуляция
L2, поэтому MAC-
адреса есть!

loopback нужен только **для отладки**. Экспонировать его во внешний интернет опасно кроме как для личного пользования

Потоки по интерфейсам
до Jupyter на localhost
сервера:

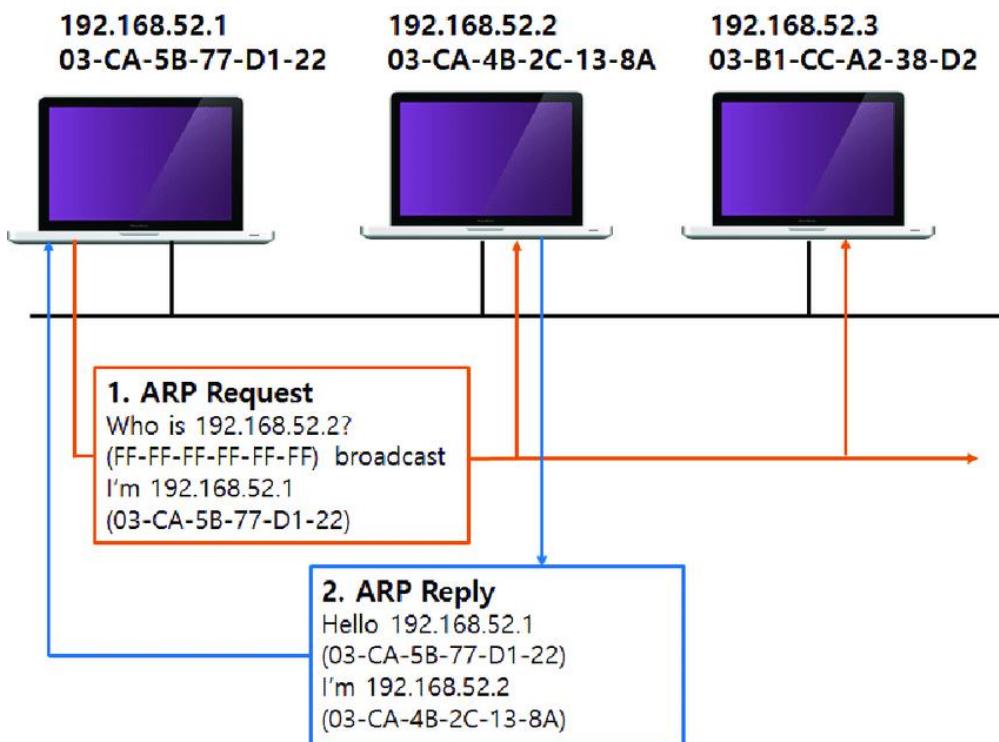


Потоки по интерфейсам,
если бы Jupyter был веб-
сервером на сервере:

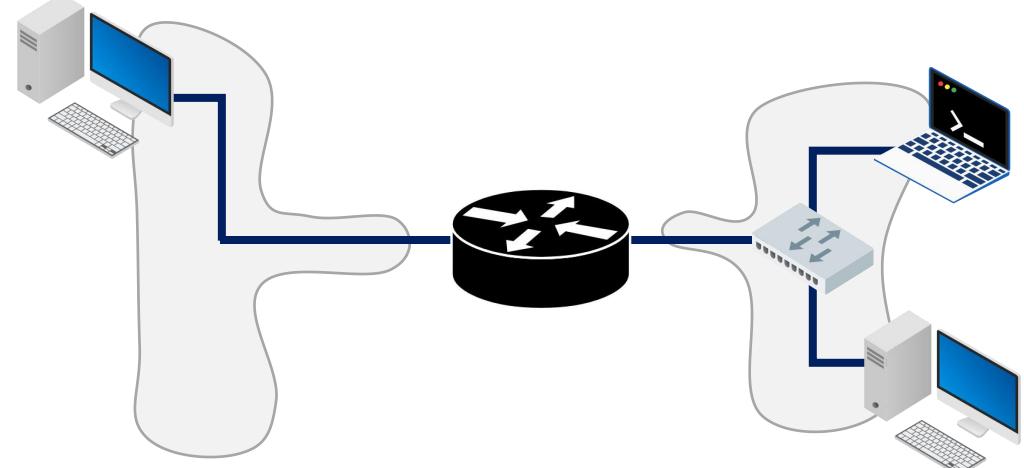


ARP

Если адреса в таблице нет, широкое вещание на всю LAN. Устройство с указанным IP-адресом отправляет на (IP, MAC) свой MAC-адрес в ARP-кадре.



ARP также используется в роутерах при пересылке между подсетями. Кадр отправляется на MAC-адрес роутера, на роутере меняется на таргетный MAC-адрес



Ethernet

Превалирующий протокол проводного интернета

Без хендшейка, ненадёжный, имеет очень много стандартов в зависимости от требуемой скорости



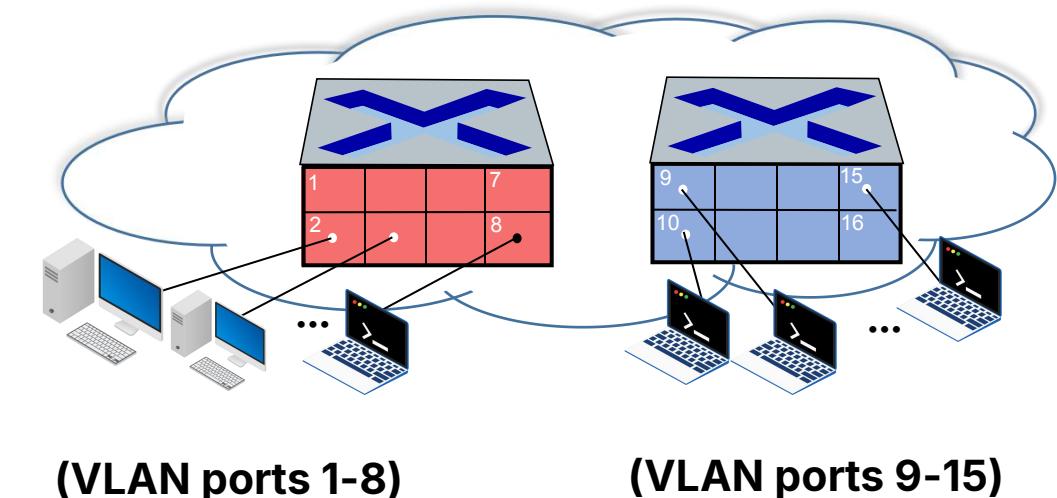
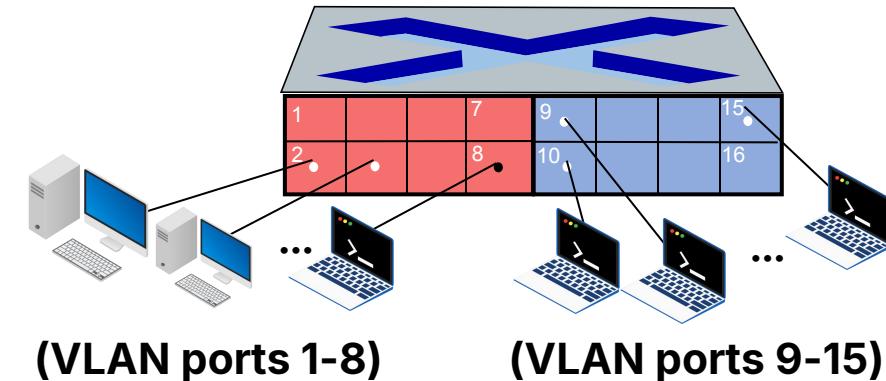
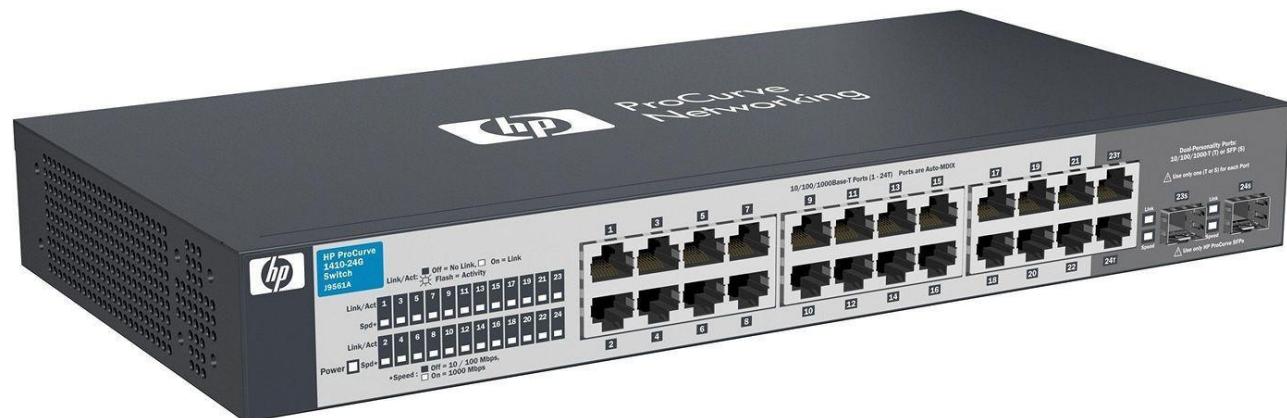
Сетевой свитч (коммутатор)

Увеличение эффективности обработки канальных кадров

Таблицы коммутации (MAC-адреса вместо IP-адресов), обучаются сами.

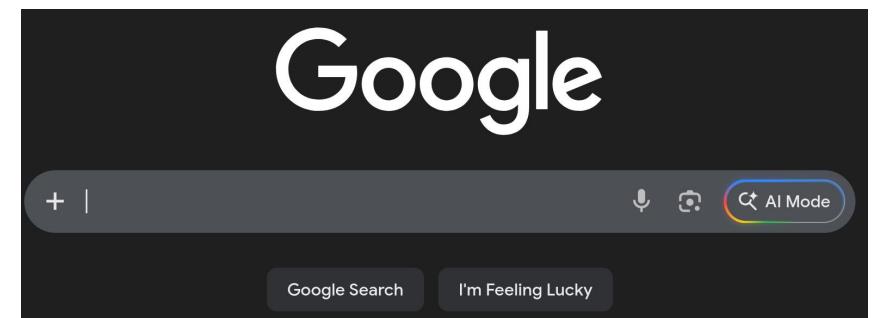
Позволяют реализацию VLAN:

Virtual Local Area Network



Итог: что происходит при запросе веб-страницы?

1. Подключение к сети и выдача IP-адреса: **DHCP**
2. Определение IP-адреса сайта с помощью **DNS**
3. Поиск интерфейса роутера для отправки HTTP-запроса: **ARP**
4. Отправка датаграммы на сеть: **OSPF + BGP**
5. Установка **TCP** соединения с сервером сайта, создание сокета
6. Передача всех сегментов веб-страницы
7. Завершение TCP-сессии



Информационная безопасность

Для передачи данных в интернете нужна:

- **конфиденциальность** (только отправитель и получатель понимают сообщение)
- **аутентификация** (подтверждение личности)
- **целостность** (сообщения не подменялись)

Злоумышленник может:

- **подслушивать**
- **вставлять сообщения** в канал соединения
 - выдавать себя за другого (**спуфинг**)
- **устраивать атаки DoS** (denial-of-service)

Криптография

С симметричным ключом

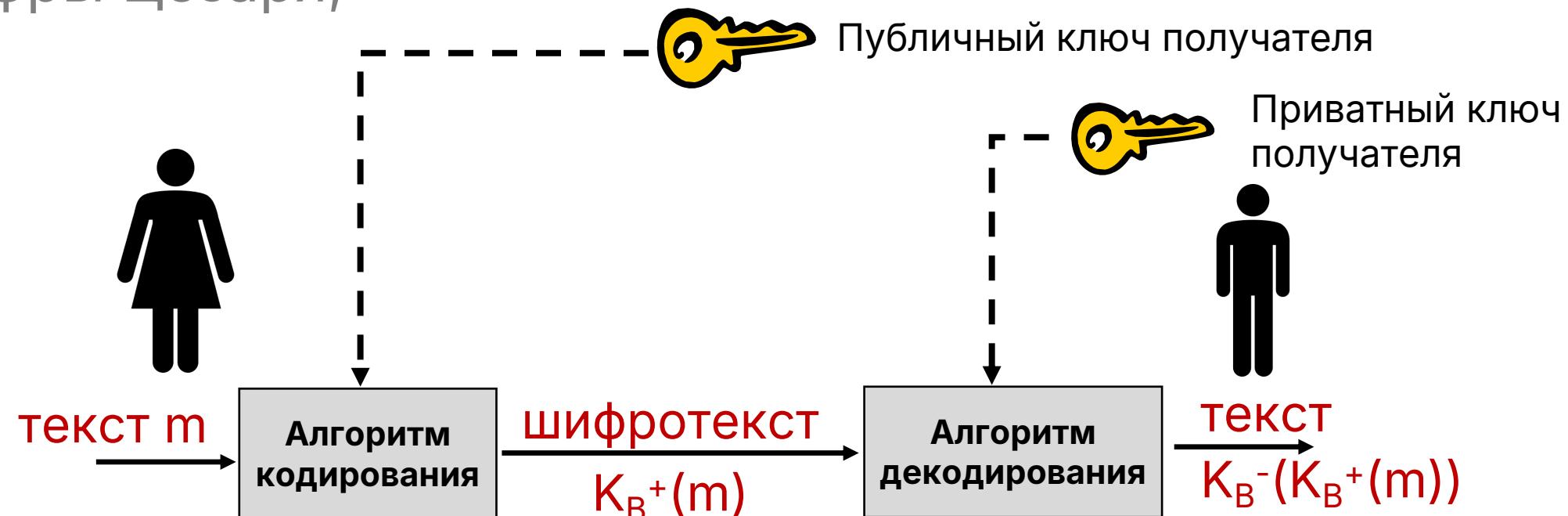
Отправитель и получатель
как-то договариваются и оба
знают шифр

Пример: шифры Цезаря,
Виженера

С открытым ключом

Получатель и отправитель
оба знают публичный ключ.

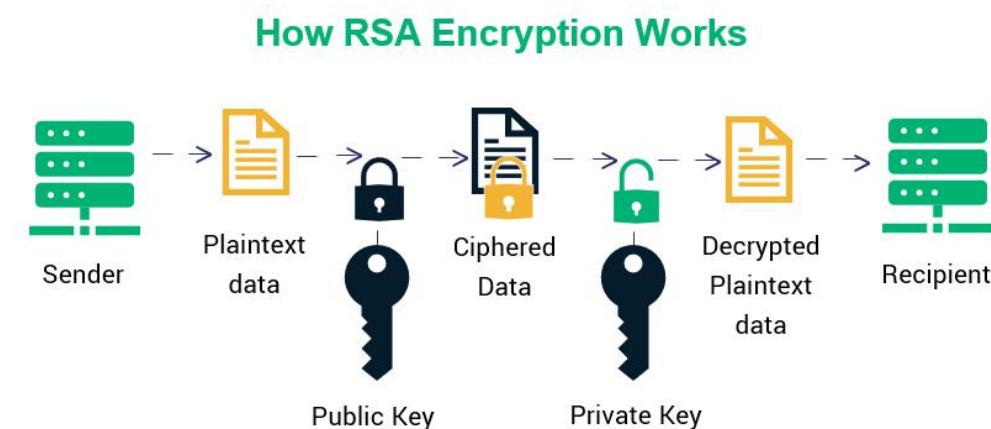
Получатель знает также
приватный ключ.



RSA

Rivest, Shamir, Adelson. Подготовка:

1. Выбираются 2 больших простых числа p, q
2. $n=pq, z=(p-1)(q-1)$
3. Выбирается e такое, что e и z взаимно простые
4. Выбирается d такое, что $ed-1$ делится на z
5. Публичный ключ - пара (n, e) . Приватный - (n, d)



RSA - (де)кодирование

$$1. \text{cipher} = m^e \bmod n$$

$$2. m = \text{cipher}^d \bmod n$$

$$\begin{aligned}m &= (m^e \bmod n)^d \bmod n = \\&= m^{ed} \bmod n = \\&= m^{(ed \bmod z)} \bmod n = \\&= m^1 \bmod n = m\end{aligned}$$

Более того, в RSA верно, что:

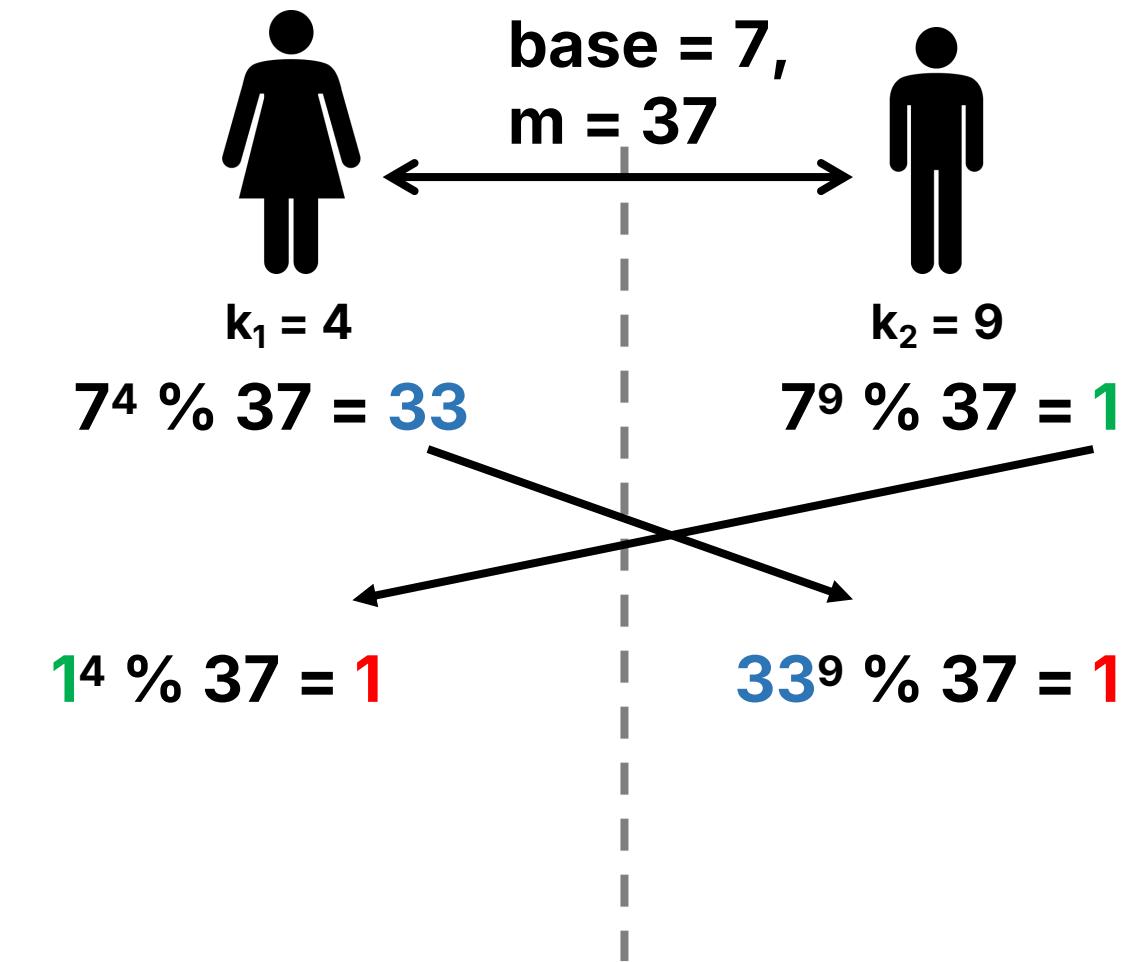
$$K_B^{-1}(K_B(K_B(m))) = K_B(K_B^{-1}(m))$$

На практике шифрование с открытым ключом используют для установления соединения и передачи симметричного ключа сессии

Безопасный договор о симметричном ключе

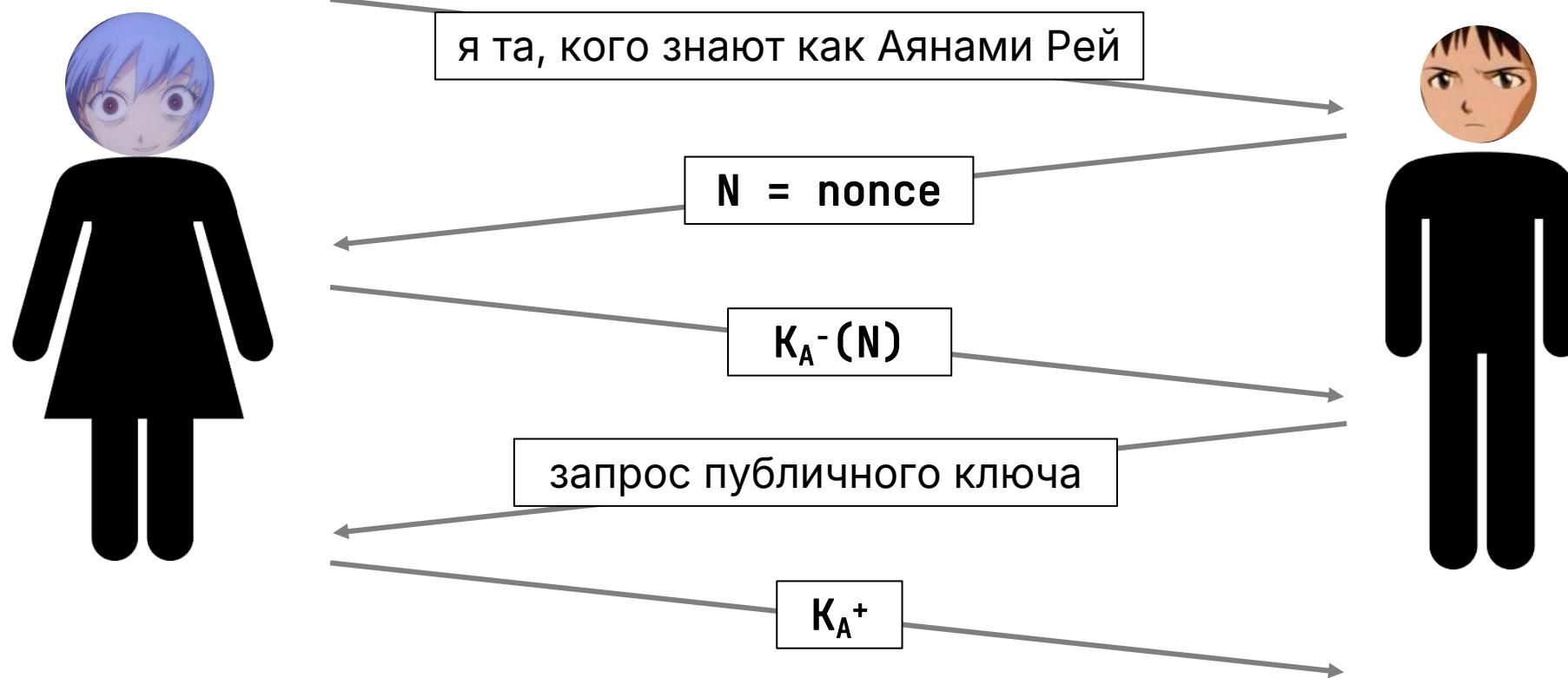
Алгоритм Диффи-Хелмана

1. Договариваются о едином ключе публично: основание и модуль
2. Каждый с секретным ключом считает $b^k \bmod m$ и отправляет друг другу
3. С полученным значением делается то же самое. В результате одно и то же значение у обоих.



Аутентификация

Используется одноразовое число - nonce.



Проблема: man-in-the-middle

Решение: дополнение с
шифрованием открытым ключом

Перешифрование:
 $K_A^+ (K_A^- (N)) = N$
успешно, значит
общение идёт с
Аянами Рей

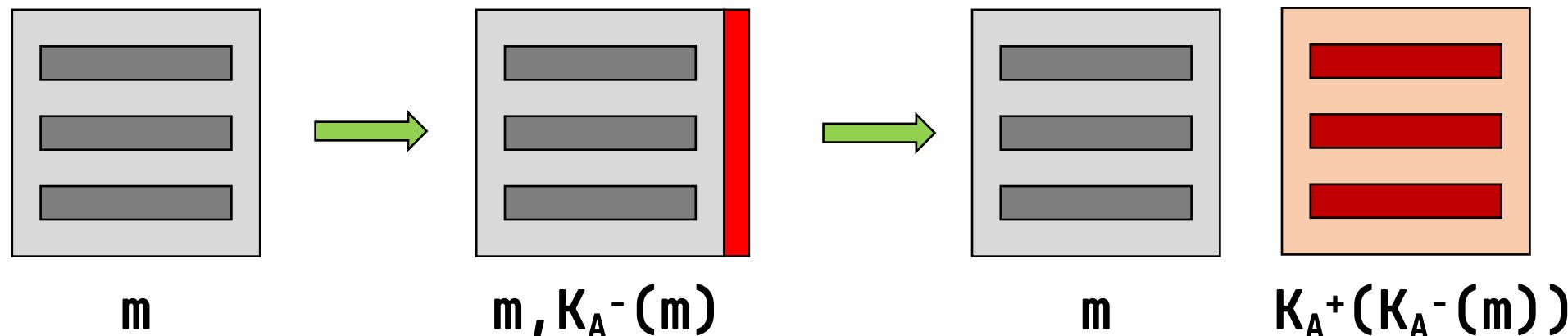
Цифровая подпись

Подпись: сообщение шифруется с помощью приватного ключа. Вместе с исходным текстом отправляется подпись.

Если публичный ключ декодирует сообщение в то же самое, значит тот, кто его отправил, владеет приватным ключом.

Шифрование обычно применяют к хешу сообщения

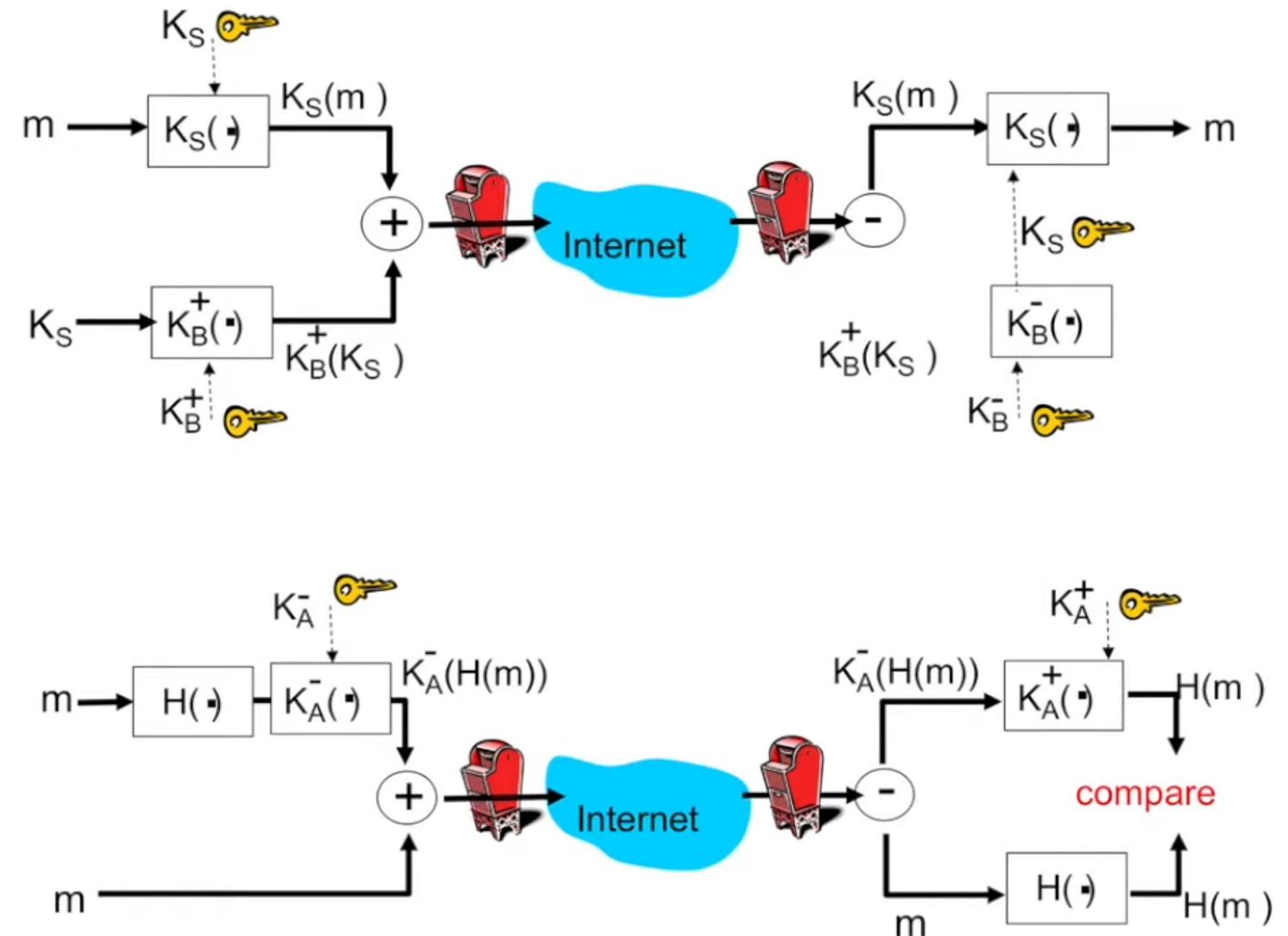
Достоверность пары публичного и приватного ключей сертифицируется в РКСА



отдельная тема:
секьюрное
общение по
e-mail. Не
трагаем

Запугивание безопасным e-mail

Благо, всё это
реализовано хорошими
инструментами из
коробки. Можно просто
развернуть контейнер
на сервере.

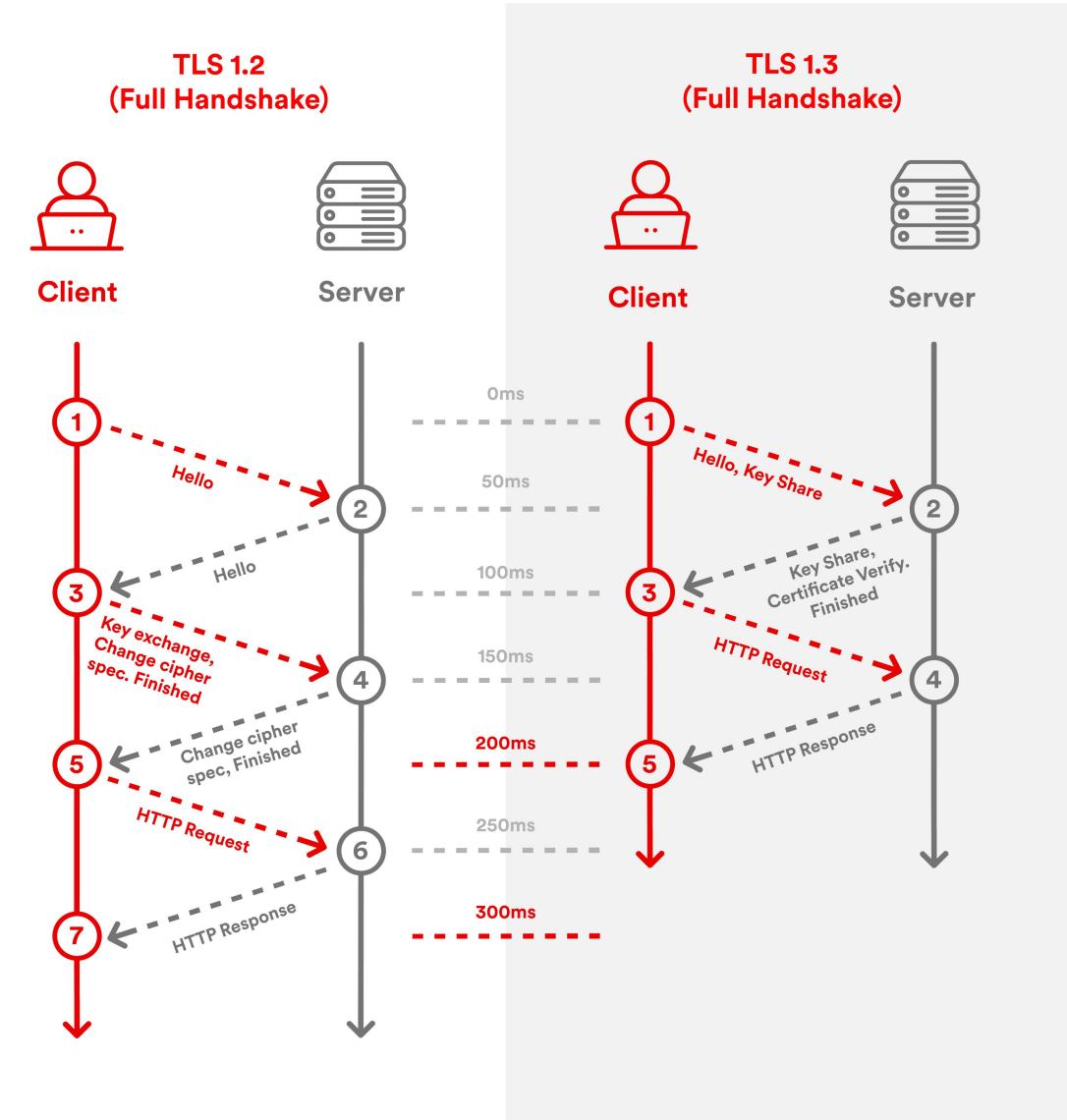


Безопасное TCP-соединение: TLS

Transport Layer Security over Secure Socket Layer (TLS/SSL)

Обеспечивает **конфиденциальность** (симметричное шифрование), **целостность** (хеширование) и **аутентификацию** (шифрование с открытым ключом)

1. Установление TLS-соединения (хендшейк как в TCP)
2. Удостоверенность в айдентике устройства собеседника
3. Отправка Мастер-Ключа
4. Сессия



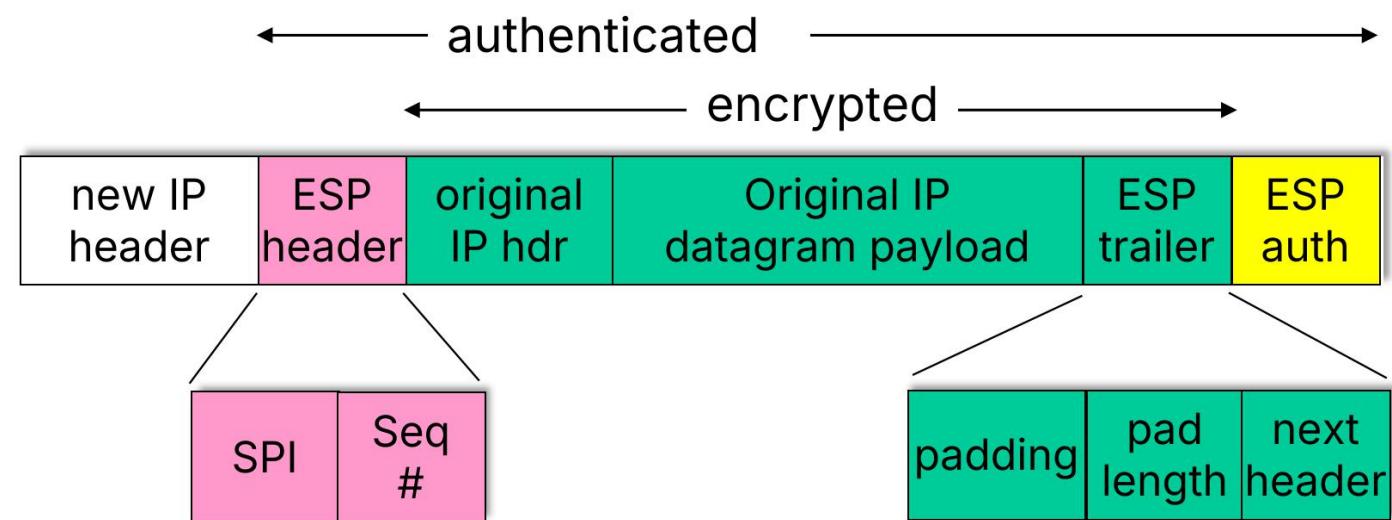
IPsec: безопасность датаграмм

Два режима:

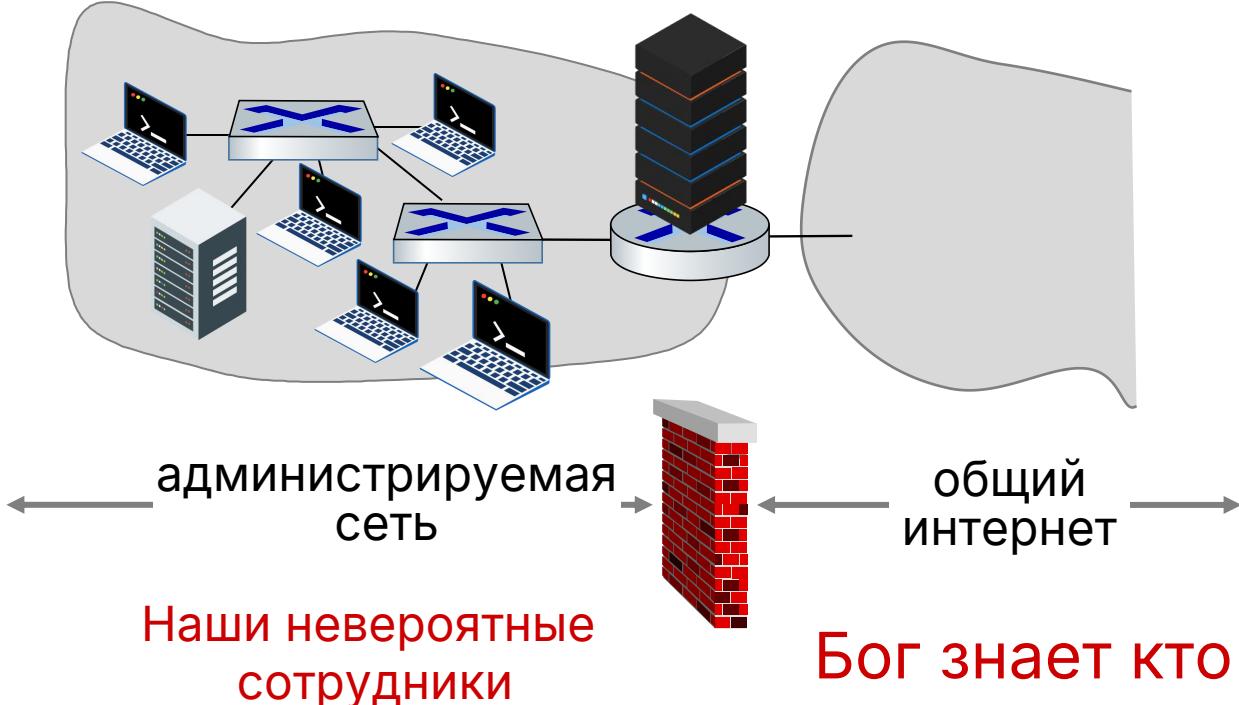
- **транспортный** (шифруются только данные)
- **тоннельный** (датаграмма шифруется целиком и инкапсулируется в новую)

IPsec сохраняет состояние как TCP, а также ориентирован на соединение

На основе IPsec-тоннелирования фактически работает VPN



Файерволл



Сервер или граничный маршрутизатор в организации, позволяет действия со входящим или исходящим трафиком

Три типа:

- stateless packet filters - ежепакетная проверка
- stateful packet filters - проверка установленных TCP-соединений
- application gateways - проверка пакетов определённых приложений