

# **Практическое занятие по курсу СПО**

## **Принципы построения сетей ТСР/ІР**

# Рассматриваемые темы

- › История развития Internet
- › Стандартизация Internet
- › Семейство протоколов TCP/IP
- › Адресация в протоколе IP
- › Механизмы конфигурирования сетевых интерфейсов
- › Маршрутизация пакетов TCP/IP
- › Протоколы ARP, ICMP, TCP, UDP
- › Система доменных имён DNS
  
- › Задачи лабораторной работы № 3
  
- › Межсетевые экраны
- › Проксирование соединений
- › Виртуальные сетевые интерфейсы и сети
- › Виртуальные частные сети
- › Оверлейные сети
- › Фрагментация сетевых пакетов
- › Вопросы отладки работы сетей

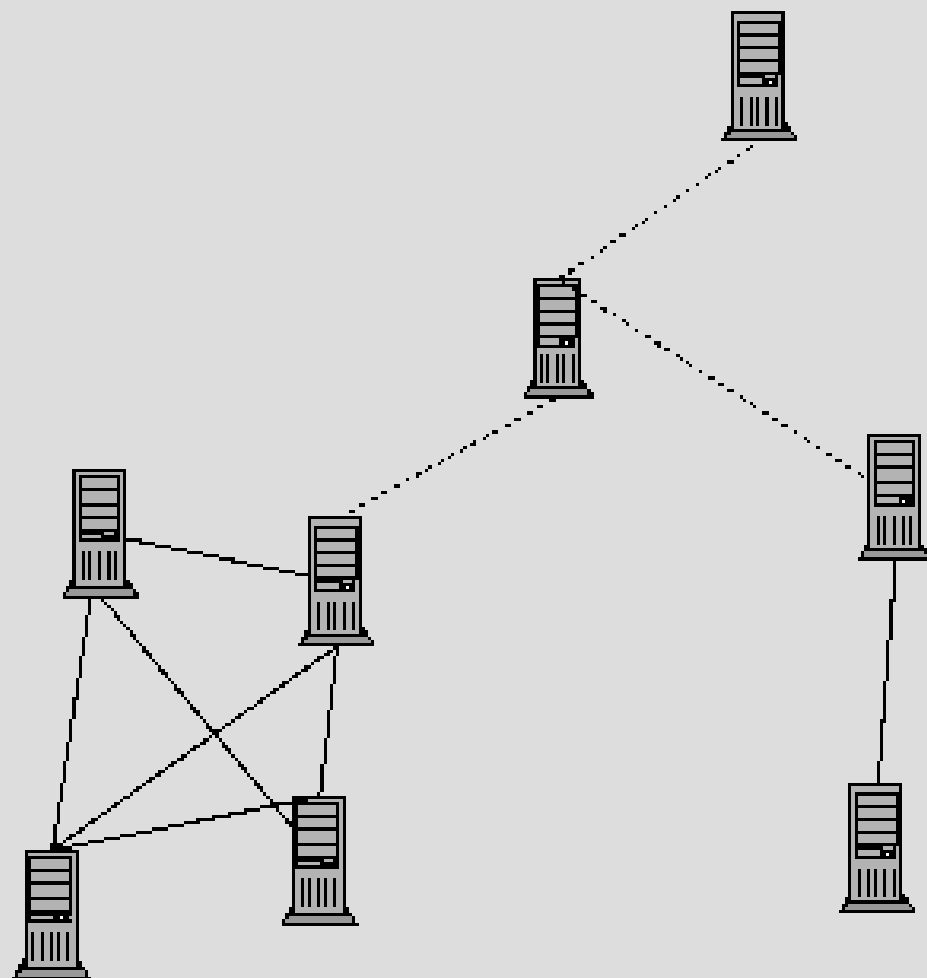
# Основы построения сетей

По каналам связи:

- коммутируемые каналы передачи данных
- выделенные каналы передачи данных

По организации связи:

- одноранговые
- маршрутизируемые



# Основы построения сетей

Сети:

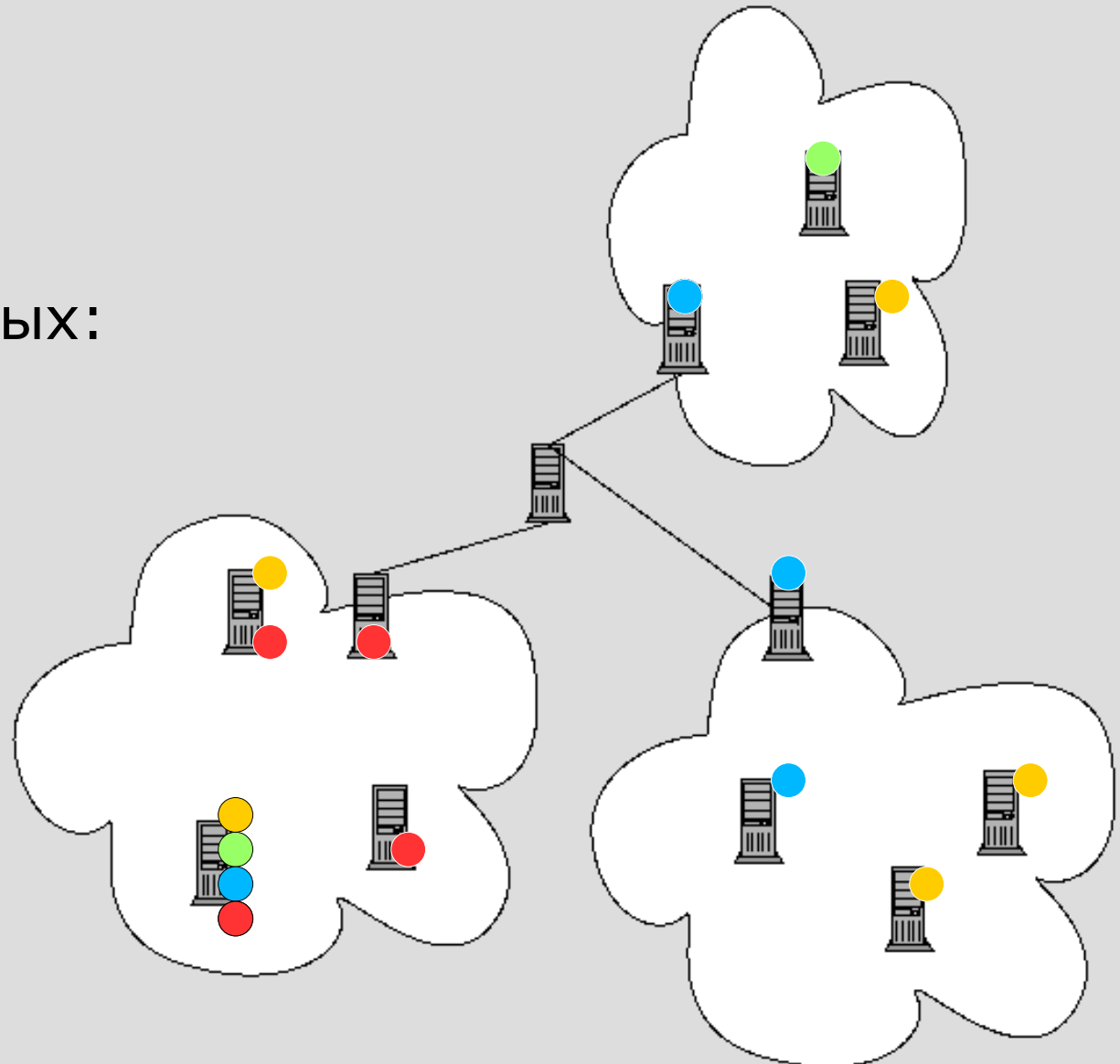
- локальные
- глобальные

Для передачи данных:

- уникальные адреса хостов

Адресация в сетях

- unicast
- multicast
- anycast
- broadcast



# Сетевая модель OSI ISO



# Основы построения сетей

Сети:

- потоковая передача данных
- пакетная передача данных

Пакет данных:

- уровни вложенности



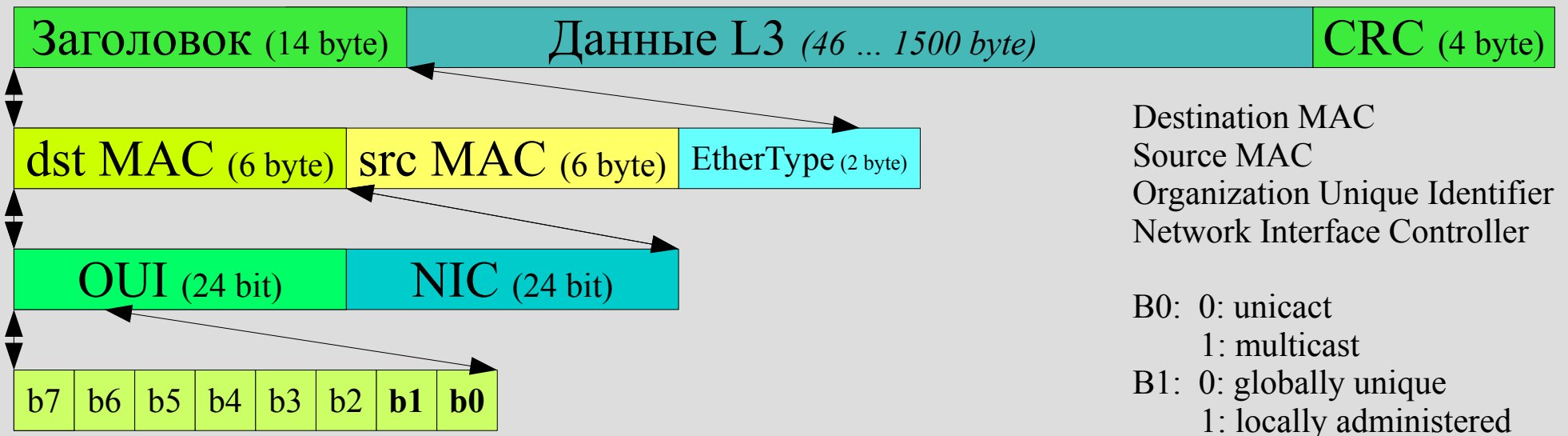
# Сети Ethernet

Ethernet (IEEE-802.x, начало разработки 1980 г.):

- протокол L2;
- объединяет peer'ы;
- уникальный адрес каждого peer'а;
- связи: peer-peer, peer-hub, hub-hub.

Адреса: MAC (Media Access Control), IEEE (Institute of Electrical and Electronics Engineers). Сейчас – MAC-48.

Есть также EUI (Extended Unique Identifier): EUI-48, EUI-64.



# История глобальных сетей обработки информации

- 1956-59 гг. – проект ЕГСВЦ (Единой государственной сети вычислительных центров), Китов А. И.
- 1962-64 гг. – проект ОГАС (Общегосударственной автоматизированной системы учёта и обработки информации), Глушков В. М.
  
- 1966 г. – эскизный проект ARPANET
- 1969 г. – запуск сети ARPANET
- 1971 г. – электронная почта
- 1973 г. – начало разработки TCP/IP
- 1975 г. – первая сеть TCP/IP
- 1.1.1983 – переход ARPANET на TCP/IPv4
- 1984 г. – NSFNet
- 1985-1994 гг. – коммерциализация Internet



# Стандартизация Internet

- ISOC – Internet Society, 1992 г.
- IETF – Internet Engineering Task Force, 1986 г.
- IAB – Internet Architecture Board
- ICANN – Internet Corporation for Assigned Names and Numbers
- IANA – Internet Assigned Numbers Authority

## RFC – Requests For Comments

1969-04-07: RFC 1 «Host Software»

Март 1992: RFC 1310 «The Internet Standards Process»

Март 1994: RFC 1602 «The Internet Standards Process -- Revision 2»

Октябрь 1996: RFC 2026 «The Internet Standards Process -- Revision 3»

2008-04-01: RFC 5241 «Naming Rights in IETF Protocols»

2018-04-01: RFC 8367 «Wrongful Termination of Internet Protocol (IP) Packets»

2024-04-01: RFC 9564 «Faster Than Light Speed Protocol (FLIP)»

# Семейство протоколов TCP/IP

	Модель OSI	Семейство протоколов TCP/IP
7	Прикладной	FTP, HTTP, Telnet SMTP, POP3, IMAP, XMPP, OSCAR, SSH, CIFS
6	Представления	NFS
5	Сеанса	XDR
4	Транспортный	RPC
3	Сетевой	TCP, UDP
2	Канальный	IP, ICMP, протоколы маршрутизации
1	Физический	ARP, RARP
		Физический

# Адресация в сетях IP

## Протокол IP:

- Сетевой уровень модели OSI (L3)
- Передаёт данные через сетевые интерфейсы от хоста к хосту
- Каждый хост имеет уникальный адрес
- Длина адреса IPv4 – 32 бита, IPv6 – 128 бит

## Сетевые пакеты IP:

- Заголовок
  - версия протокола (4/6)
  - длина заголовка,
  - идентификатор пакета, флаги фрагментации
  - время жизни
  - адрес IP отправителя
  - адрес IP получателя
  - идентификатор протокола следующего уровня
  - контрольная сумма заголовка
- Данные. Максимальная длина пакета IP — 65 535 байт.



# Адреса протокола IPv4

Адрес IP v4:

- целое беззнаковое число
- длина адреса 32 бита, или 4 байта
- записывается по-байтно через точку
- состоит из адреса сети и адреса хоста

Маска сети:

- служит для выделения адреса сети
- длина маски равна длине адреса
- начинается с последовательности единиц
- кончается нулями

Адрес IP – 193.233.68.72/255.255.255.0

11000001 11101001 01000100

01001000

11111111 11111111 11111111

00000000

# Классы сетей IPv4

Класс	Маска	Диапазон адресов
A (0...)	255.0.0.0	1.0.0.0 - 127.255.255.255
B (10...)	255.255.0.0	128.0.0.0 - 191.255.255.255
C (110...)	255.255.255.0	192.0.0.0 - 223.255.255.255
D (1110...)	-	224.0.0.0 - 239.255.255.255
E (1111...)	?	240.0.0.0 - 255.255.255.255

D — адреса multicast; E — зарезервировано

127.0.0.0/8 – сеть loopback-интерфейса, IP 127.0.0.1/255.0.0.0

# Адресация в сетях IPv4

Проблемы классовой адресации:

- Ограниченное количество сетей

  - Класс А — 126 сетей по 16 777 214 хостов,

  - Класс В — 16 386 сетей по 65 534 хостов,

  - Класс С — 2 097 154 сетей по 254 хоста,

  - Следствие: исчерпание адресного пространства

- Использование адресного пространства внутри сетей организаций

  - Проблемы с маршрутизацией, с выделением новых сетей при росте организации

- Отсутствие группировки сетей

  - Следствие: размеры таблиц маршрутизации на магистральном оборудовании

# Адресация в сетях IPv4

Бесклассовая адресация (CIDR, Classless Inter-Domain Routing):

- отказ от выровненных по границам байта масок сетей
- произвольная длина маски сети
- агрегация сетей

Частные сети (RFC 1918, 1996 г.):

1 сеть A: 10.0.0.0/8

16 сетей B: 172.16.0.0/16 – 172.31.0.0/16

256 сетей C: 192.168.0.0/24 – 192.168.255.0/24

Частные сети уровня провайдеров (RFC 6598, 2012 г.):

Диапазон адресов 100.64.0.0/10

Доступ из частных сетей в Internet:

- прокси-серверы
- трансляция адресов



# Адреса протокола IPv6

Адрес IP v6:

- целое беззнаковое число
- длина адреса 128 бит, или 16 байт
- записывается группами по 4 16-ричных цифры
- группы разделяются двоеточиями
- ведущие нули групп можно опускать
- самую большую группу нулей можно опускать
- маска записывается в бесклассовой нотации

2001:0db8:0000:0064:0000:0000:aa72:0004/64

2001:db8:0:64:0:0:aa72:4/64

2001:db8:0:64::aa72:4/64

Адрес локального интерфейса - ::1/128

# Конфигурация сетевых интерфейсов

Физический и канальный уровень:

- наиболее распространённый протокол – Ethernet
- имеются уникальные сетевые адреса канального уровня (MAC-адреса)

Сетевой уровень – IP:

- необходимо назначить сетевому интерфейсу адрес IP и указать маску сети

Конфигурация сетевых интерфейсов IP:

- статическая
- автоматическая
- динамическая

# Конфигурация сетевых интерфейсов

Статическая конфигурация:

- ручная настройка адресов IP
- сохраняется в настройках операционной системы

Автоматическая конфигурация:

- адреса назначаются операционной системой
- адреса создаются на базе MAC-адреса
  
- IPv4 : сеть 169.254.0.0/16  
адрес хоста – случайное число
  
- IPv6 : сеть fe80::/64  
адрес хоста – на базе идентификатора EUI-64

# Конфигурация сетевых интерфейсов

Автоматические адреса IPv6:

MAC: 00:18:51:61:49:5a

IPv6: fe80::0**2**18:51**ff:fe**61:495a/64

Динамическая конфигурация:

- информация о сети получается от других хостов
- возможно получение информации о маршрутах, доступных в сети ресурсах, и т.п.

IPv4: протокол DHCP

IPv6: протоколы динамической конфигурации IPv6,  
протокол DHCPv6

# Конфигурация сетевых интерфейсов

## DHCP:

- требуется сервер DHCP
- хост ищет сервер DHCP через широковещательные запросы
- адрес хосту выделяется на определённое время
- по истечению аренды хост повторно запрашивает адрес
- хосту также сообщаются маршруты, серверы DNS, пр. данные
- в сети может работать только один сервер DHCP
- сервер DHCP не может инициировать смену адресов клиентов

# Конфигурация сетевых интерфейсов

Динамическая конфигурация IPv6:

- информацию рассылают маршрутизаторы сети
- хостам сообщаются префиксы сетей и маршруты
- хосты назначают адреса IPv6 в полученных сетях с использованием EUI-64
- адреса и маршруты назначаются для каждой из анонсированных маршрутизаторами сетей
- устаревшая конфигурация автоматически удаляется

По сравнению с DHCPv4:

- маршрутизаторы не ведут учёт адресов хостов
- нельзя передать адреса серверов DNS и пр.
- потенциальное отслеживание пользователей по EUI-64

# Конфигурация сетевых интерфейсов

## IPv4:

- обычно 1 IP на интерфейсе
- используется **один из** механизмов конфигурации
- автоматической конфигурации обычно нет

## IPv6:

- обычно много IP на интерфейсе
- используются **все доступные** механизмы конфигурации
- автоматическая конфигурация используется почти всегда
- минимальная выделяемая сеть – /64.  
Рекомендовано выделение сети /56 каждому подключённому к Internet пользователю.

# Маршрутизация в сетях TCP/IP

- Internet – объединение сетей TCP/IP
- Каждая сеть имеет свой адрес сети
- В каждой сети есть один или несколько шлюзов
- Шлюз - сетевое устройство, входящее сразу в несколько сетей
- Шлюз может передавать пакеты IP из одной сети в другую
- Маршрут - путь пакета от хоста-источника до хоста-приёмника
- Возможных маршрутов может быть несколько
- Выбор маршрута – задача шлюза
- Выбор осуществляется по таблице маршрутизации
- Маршрутизация бывает статическая и динамическая



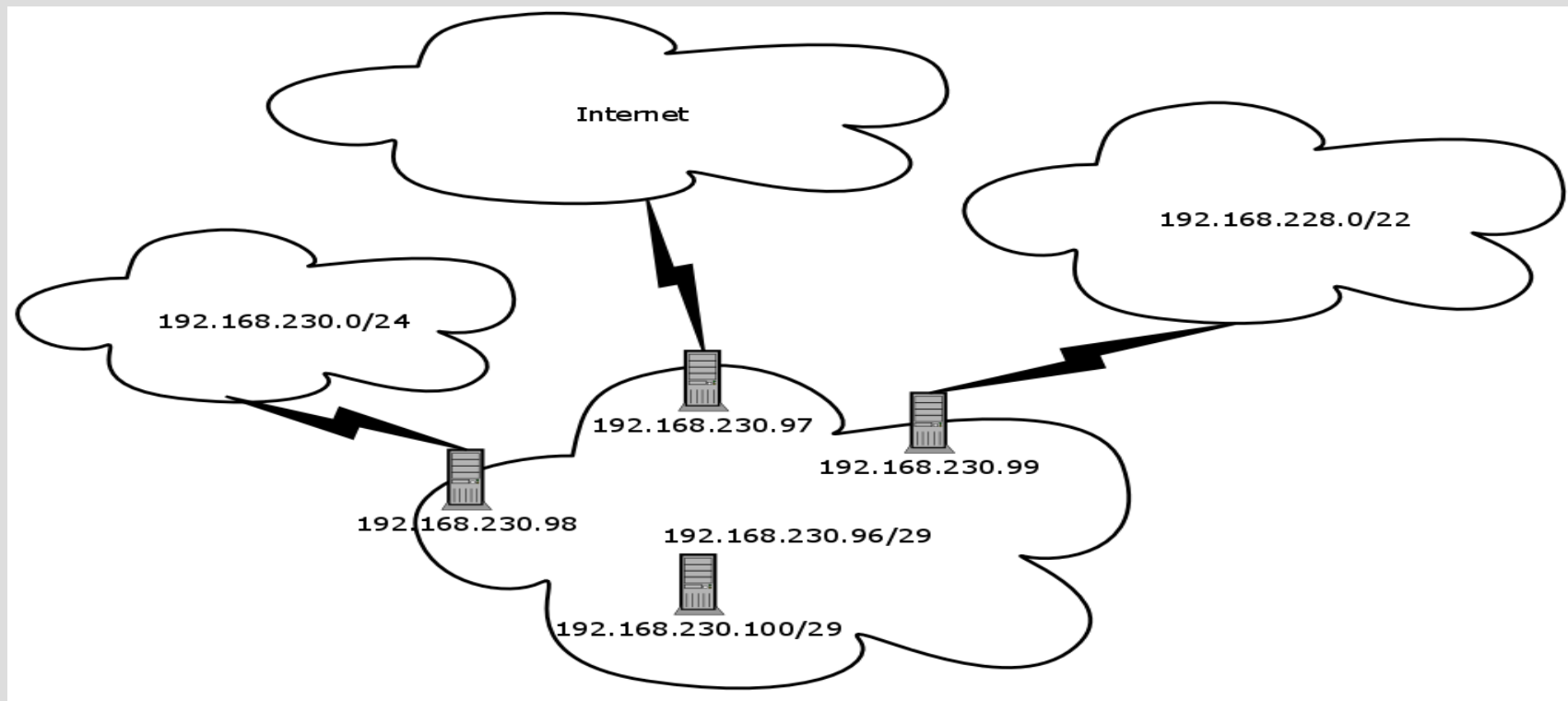
# Маршрутизация в сетях TCP/IP

- Как правило, статические маршруты прописываются для:
- сетей, к которым непосредственно подключён хост – без указания шлюза;
  - сетей, маршруты к которым должны проходить через известные шлюзы – с указанием шлюзов;
  - для всех остальных сетей – через шлюз по-умолчанию.

Все шлюзы должны быть в одной сети / сетях с хостом – т.е., статически прописать проходящий через несколько шлюзов маршрут нельзя.

Динамическая маршрутизация применяется в-основном на центральных шлюзах в крупных сетях и на магистральных маршрутизаторах Internet.

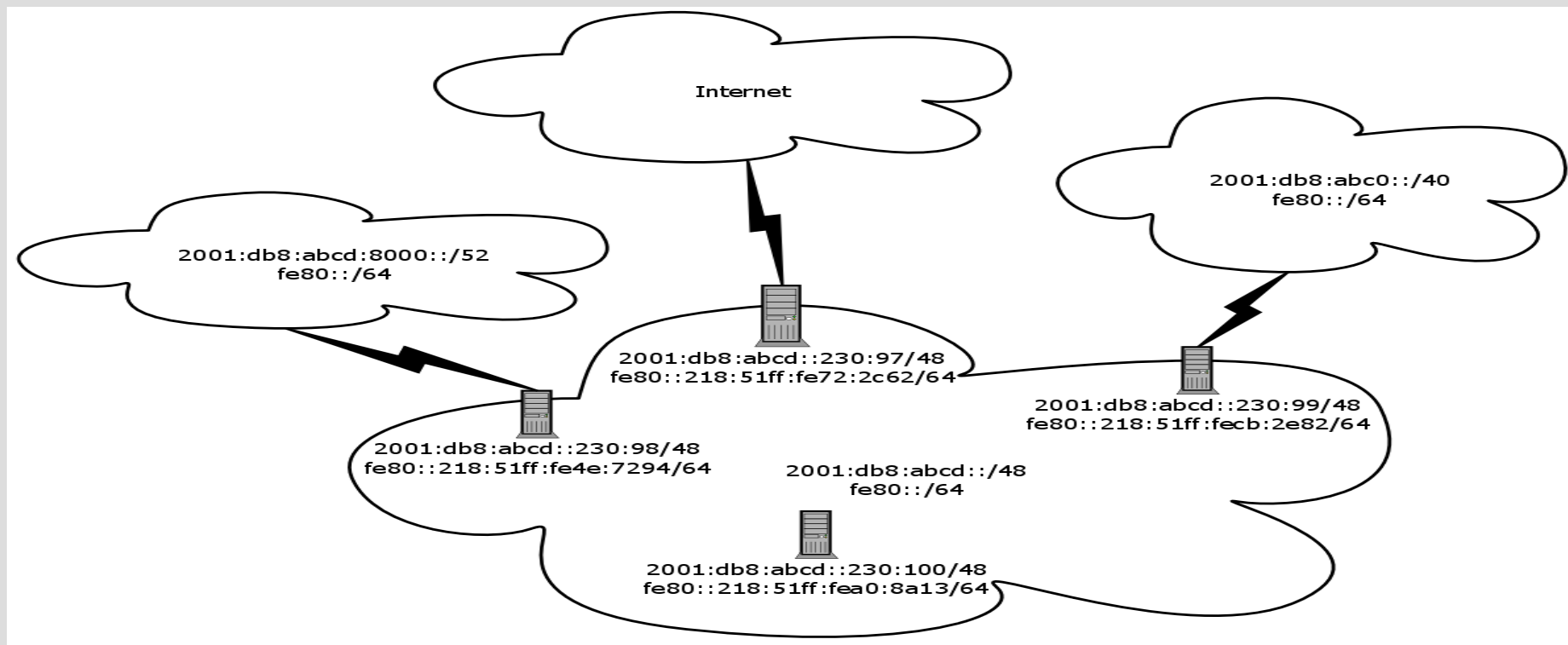
# Маршрутизация в сетях IPv4



```
# ip route show
192.168.230.96/29 dev eth0 src 192.168.230.100
192.168.230.0/24 via 192.168.230.98 dev eth0
192.168.228.0/22 via 192.168.230.99 dev eth0
default via 192.168.230.97 dev eth0
```

```
(default => 0.0.0.0/0)
```

# Маршрутизация в сетях IPv6



```
# ip -6 route show | sed -e 's/metric.*//'
```

```
2001:db8:abcd:8000::/52 via 2001:db8:abcd::230:98 dev eth0
2001:db8:abc0::/40 via 2001:db8:abcd::230:99 dev eth0
2001:db8:abcd::/48 dev eth0 proto kernel
fe80::/64 dev eth0 proto kernel
default via 2001:db8:abcd::230:97 dev eth0
```

# Протоколы ARP, ICMP

Протокол ARP (Address Resolution Protocol, RFC 826):

- протокол семейства IPv4,
- определение MAC-адреса хоста по известному адресу IPv4,
- запрос MAC-адреса идёт отправкой широковещательного пакета,
- ответы кешируются (и в т.ч. на *прочих* узлах локальной сети),
- есть таблица ARP с известными парами адресов IP — MAC.

Есть протокол RARP (Reverse Address Resolution Protocol).

Для IPv6 протокола ARP нет.

Протоколы ICMP (Internet Control Message Protocol):

- протокол семейства IP;
- работают поверх протоколов IPv4 (ICMPv4, RFC 792) и IPv6 (ICMPv6, RFC 4443);
- позволяют передавать управляющие сообщения между хостами;
- в частности, есть управляющие сообщения Echo Request / Echo Reply, см. утилиту ping;
- для IPv6 обеспечивают в т.ч. определение MAC-адреса хоста по IP.

# Протоколы TCP, UDP

## Протоколы TCP и UDP:

- протоколы семейства IP;
- протоколы транспортного уровня (L4);
- работают поверх протоколов IPv4, IPv6;
- к адресам хоста добавляется номер порта: целое 16-битное число.

Порты 0...1000 — зарезервированы для использования администратором системы (приложения должны работать с UID=0).  
Порты 10000...50000 — обычно используются клиентскими приложениями.

Есть стандартные порты приложений, выделяются IANA.  
Перечень стандартных портов — /etc/services .

22/tcp — SSH,  
53/udp, 53/tcp — DNS,  
80/tcp — HTTP,  
443/tcp — HTTPS,

...

# Протокол ТСР

## Протокол ТСР (Transmission Control Protocol):

- позволяет отправлять/получать поток данных,
- в заголовках пакета ТСР указывается:
  - номер порта получателя,
  - номер порта отправителя,
  - порядковый номер пакета (SN, Sequence Number),
  - номер подтверждения (ACK SN, Acknowledgment Number)
  - длина заголовка пакета,
  - флаги (управляющие биты) – управление соединением,
  - различные опции,
  - контрольная сумма.

## При использовании ТСР:

- устанавливаются долговременные соединения между приложениями,
  - требуется передача служебных пакетов ТСР для установления соединения и его закрытия,
- данные получаются в том же порядке, как были отправлены,
- потери пакетов данных отслеживаются,
- потерянные пакеты отправляются повторно,
- есть возможность подстройки работы соединения под характеристики канала связи.

# Протокол UDP

## Протокол UDP (User Datagram Protocol):

- позволяет отправлять/получать отдельные пакеты данных,
- в заголовках пакета UDP указывается:
  - номер порта получателя,
  - номер порта отправителя,
  - длина пакета и контрольная сумма.

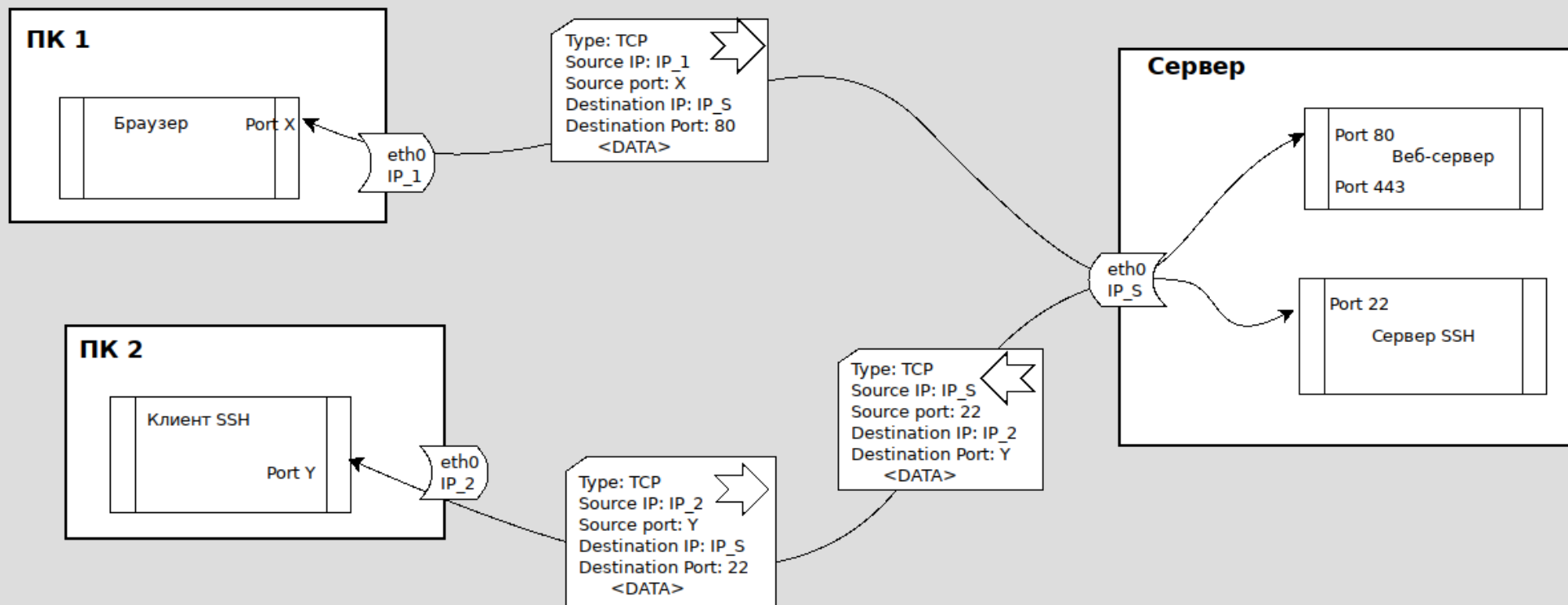
## Пакеты UDP:

- отправляются получателю без явного установления соединения
  - нет передачи служебных пакетов для установления/завершения соединения, т.к. нет соединения,
- могут быть потеряны при передаче
  - и отправитель об этом не узнает,
- могут быть получены в произвольном порядке
  - и порядка опрвления пакетов получатель не знает
- ответы также могут быть потеряны, получены в другом порядке и т. п.

# Протоколы TCP, UDP

## Протоколы TCP и UDP:

- используются на уровне L5-L7 приложениями для передачи данных,
- работают поверх протокола IP (уровень L3).





# Система доменных имён DNS

Соответствие адреса хоста и символического имени:

- /etc/hosts

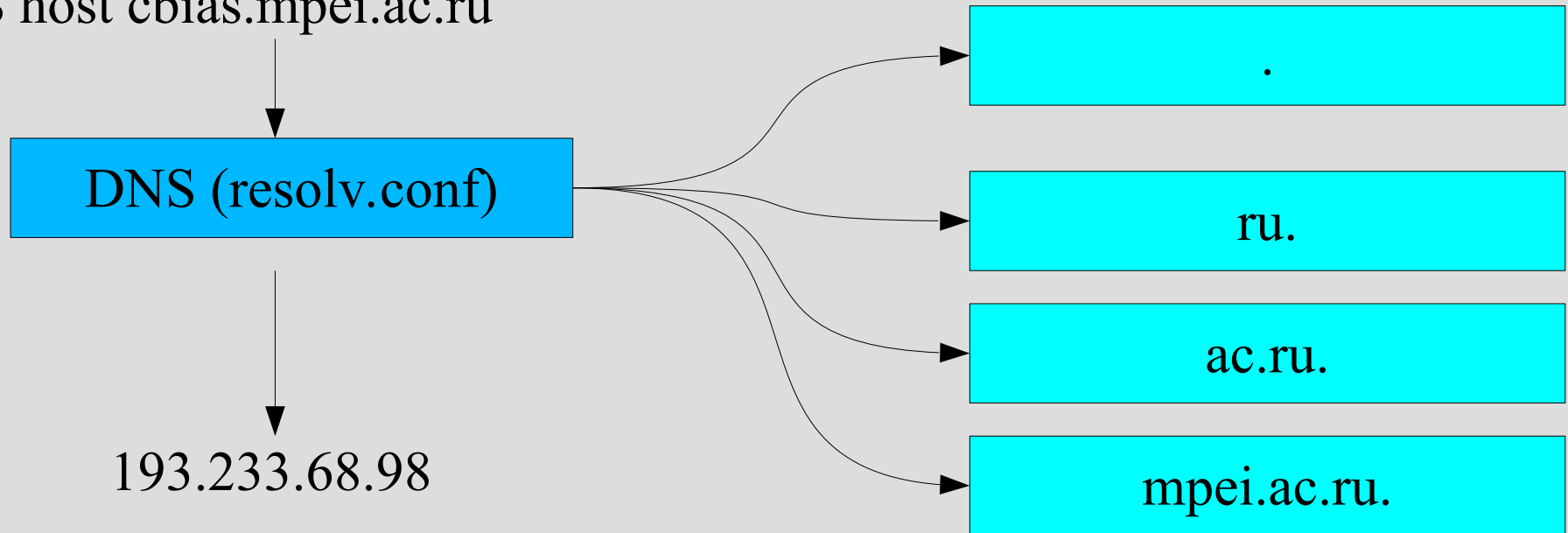
```
$ cat /etc/hosts
127.0.0.1      localhost.localdomain localhost
::1           localhost6.localdomain6 localhost6
```

- DNS

```
$ cat /etc/resolv.conf
search example.com
nameserver 192.0.2.123
nameserver 2001:db8::1234
```

# Система доменных имён DNS

```
$ host cbias.mpei.ac.ru
```



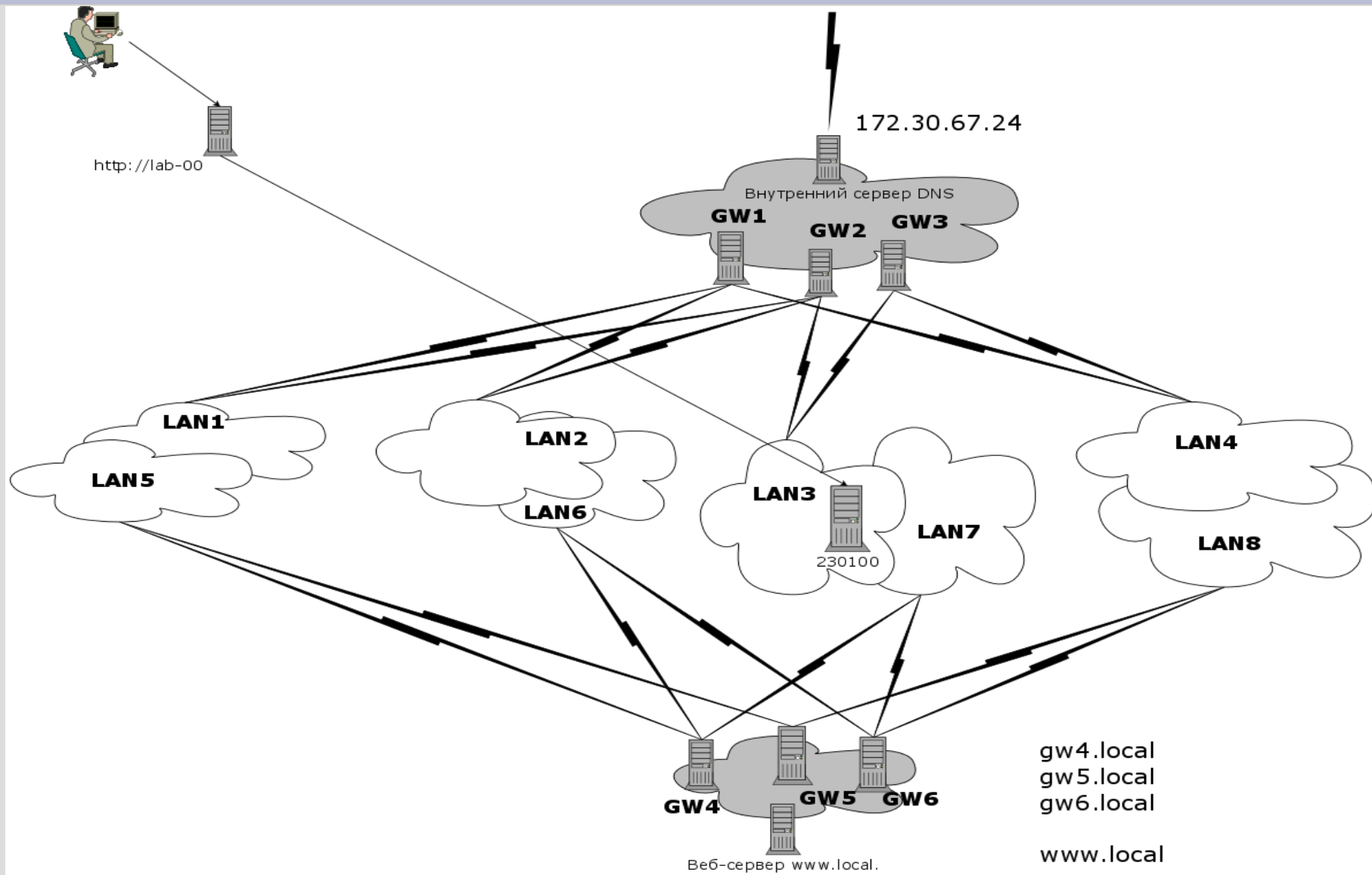
Серверы DNS:

- авторитетные (содержащие записи о доменных зонах)
- рекурсивные (позволяющие выполнить запрос информации из DNS)
  - серверы DNS провайдеров
  - публичные серверы DNS

Публичные рекурсивные серверы NSDI:

- a.res-nsdi.ru (195.208.4.1; 2a0c:a9c7:8::1)
- a.res-nsdi.ru (195.208.5.1; 2a0c:a9c7:9::1)

# Задание на лабораторную работу



# Задание на лабораторную работу

В каждой из сетей 3го уровня – 2 маршрутизатора.

Сети LAN1-LAN4 – IPv4

- адрес первого маршрутизатора – один из списка, получаемого с тестового сервера SSH
- адрес второго маршрутизатора – первый или последний IP соответствующей сети.

Сети LAN5-LAN8 – IPv6.

Адреса маршрутизаторов IPv6:

gw4.local, gw5.local, gw6.local

Адрес сервера DNS: 172.30.67.24

Адрес внутреннего веб-сервера: www.local

# Задание на лабораторную работу

- проверить наличие обновлений и обновить систему
- определить, в какую сеть IPv4 входит интерфейс eth0;
- назначить интерфейсу адрес IPv4 из этой сети;
- найти и задать статические маршруты до каждой из сетей IPv4 и до внутреннего сервера DNS;
- настроить службу DNS;
- определить, в какую сеть IPv6 входит интерфейс eth0;
- назначить интерфейсу адрес IPv6 из этой сети;
- найти и задать статические маршруты до каждой из сетей IPv6 и до внутреннего веб-сервера;
- задать маршрут по-умолчанию для IPv6;
- проверить доступность веб-сервера `www.local`, внешних веб-серверов сети IPv6;
- настроить проксирование запросов к веб-серверу `www.local`;
- обеспечить восстановление сделанных настроек сети при перезагрузках системы,
- не мешать коллегам выполнять данную работу:
  - автоматические проверки и удаление:
    - ошибочно назначенных адресов маршрутизаторов (-0.1 за адрес)
    - дублирующихся адресов, использующихся кем-то ещё.

# Межсетевые экраны

Задача межсетевых экранов: фильтрация и обработка сетевого трафика.

Могут работать на разных уровнях сетевой модели:

- L2: фильтрация пакетов Ethernet:
  - source MAC-адрес и EtherType;
- L3+L4: фильтрация пакетов TCP/IP:
  - тип протокола IP;
  - source IP, source IP + source port;
  - destination IP, destination IP + destination port;
  - обработка сессий TCP
- L5..L7: фильтрация по приложениям, пользователям, данным и т. п.

Могут быть stateless и statefull:

- stateless: фильтрация пакетов идёт без учёта активных соединений (TCP/UDP),
- statefull: при фильтрации учитываются соединения TCP и ответы на пакеты UDP.

# Межсетевые экраны

## Фильтрация трафика:

- разрешить дальнейшую обработку сетевого пакета
- отбросить сетевой пакет
  - возможно, с отправкой соответствующего пакета ICMP

## Обработка сетевого трафика межсетевыми экранами:

- подмена source/destination IP/port (SNAT/DNAT)
- что-либо ещё

## Межсетевые экраны в Linux:

- реализованы в ядре операционной системы (NetFilter),
- уровень L2 ... L4.

## Реализации межсетевых экранов в Linux:

- ipchains (ядра 2.4 включительно, устарело)
- iptables (iptables, ip6tables, arptables, ebtables)
- nftables (nft)
- bpfiler (начиная с ядер 4.8, основан на eBPF, extended Berkeley Packet Filter)

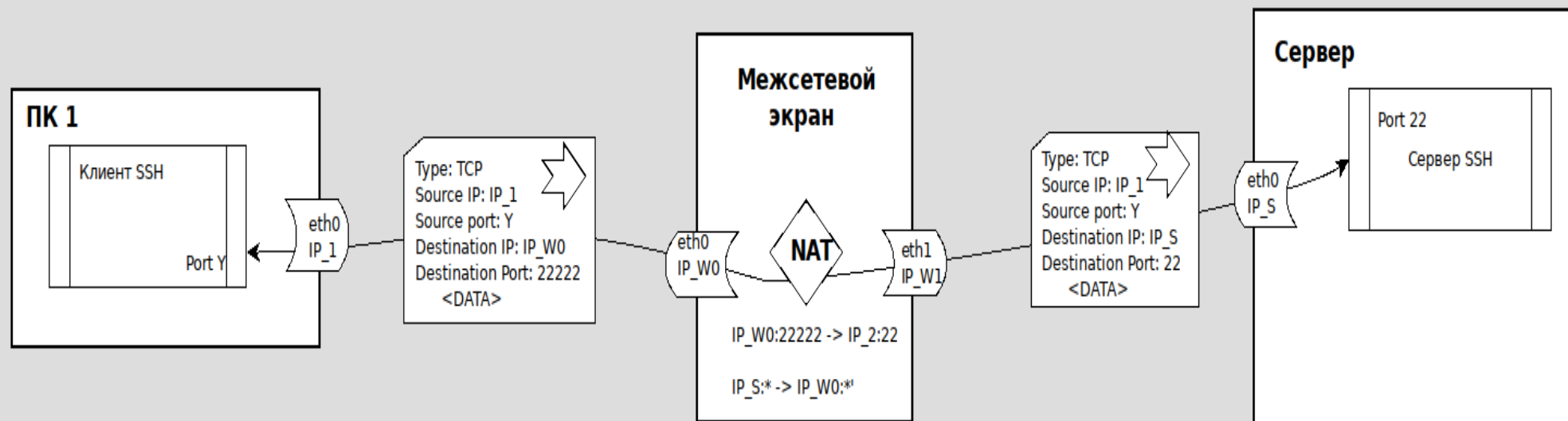
Для построение сложных фильтров используются программные надстройки, общего назначения или специфичные для дистрибутивов.

# Межсетевые экраны

Правила обработки трафика межсетевыми экранами:

- таблицы с последовательностями правил для разных этапов обработки трафика: filter, nat, mangle, raw, security;
- в таблицах есть стандартные цепочки правил,
- можно создавать свои цепочки правил и направлять определённые пакеты в них,
- у стандартных цепочек правил есть политики по-умолчанию: ACCEPT, DROP;
- правила обработки трафика могут расширяться модулями ядра;
- фильтрация трафика: таблица filter, цепочки INPUT, OUTPUT, FORWARD;
- изменение пакетов: таблица nat, цепочки PREROUTING, POSTROUTING, INPUT, OUTPUT.

Трансляция адресов:



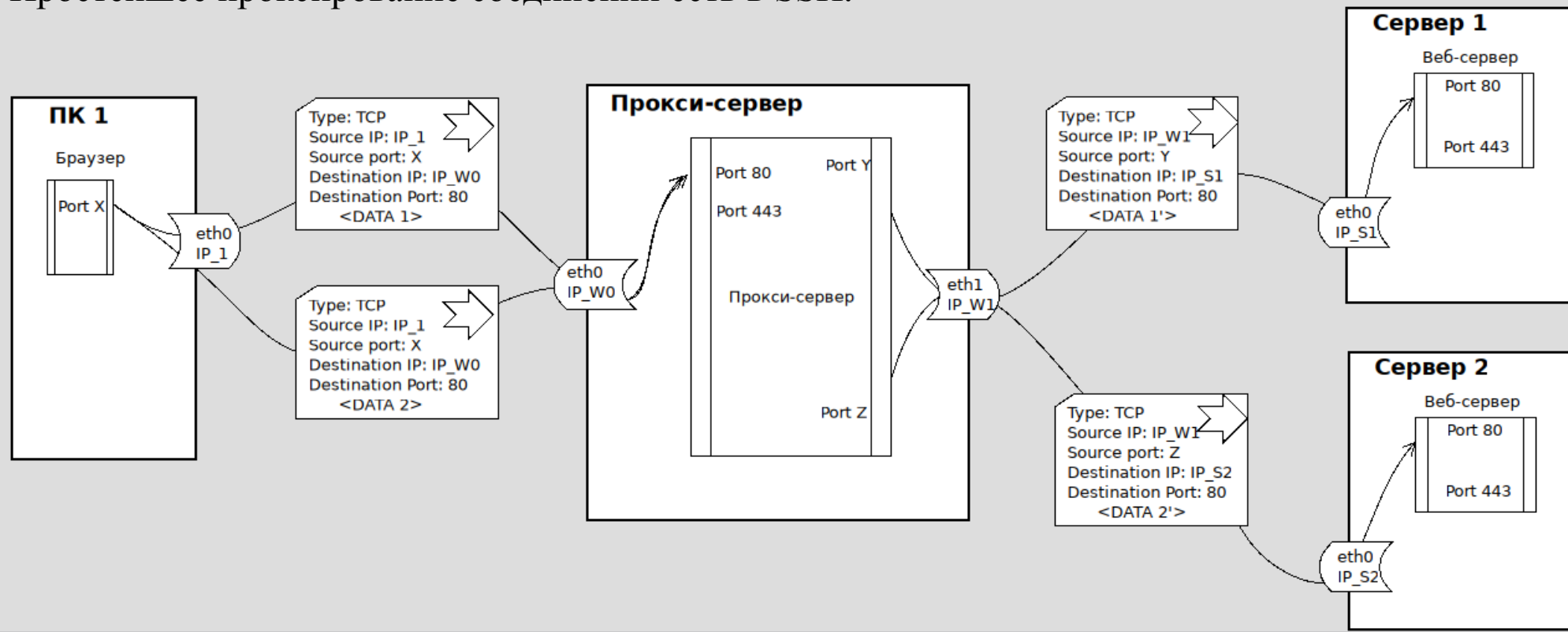


# Проксирование соединений

Проксирование соединений:

- запрос от клиента принимается прокси-сервером,
- прокси-сервер запрашивает целевой сервер,
- полученный ответ прокси-сервер возвращает клиенту,
- возможна фильтрация и изменение запросов и ответов на прокси-сервере.

Простейшее проксирование соединений есть в SSH.



# Виртуальные сетевые интерфейсы

## Сетевые интерфейсы:

- Физические – соответствуют аппаратному обеспечению
  - уровень L1
- Виртуальные – эмулируются средствами операционной системы
  - уровень L2 (работают DHCP, ARP, и другие протоколы L2)
  - уровень L3 (работает IP).
- Виртуальные сетевые интерфейсы в Linux:
  - пары виртуальных интерфейсов veth, уровень L2:  

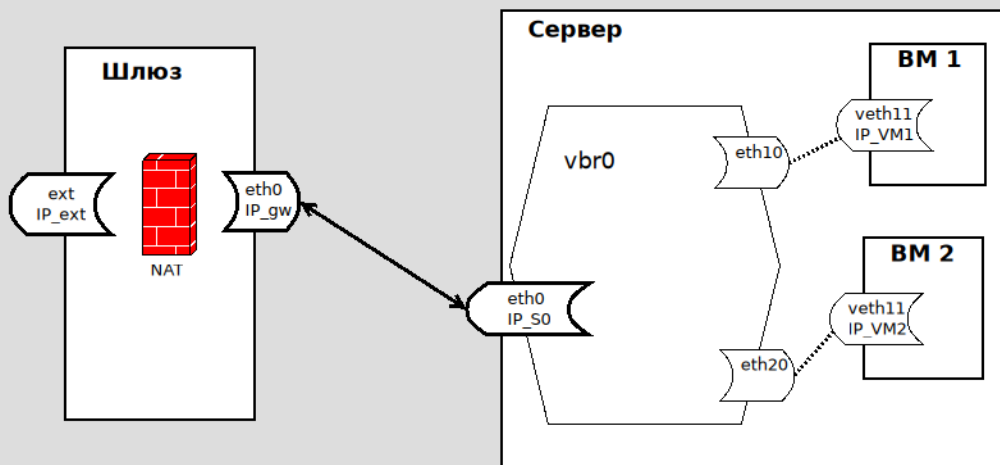
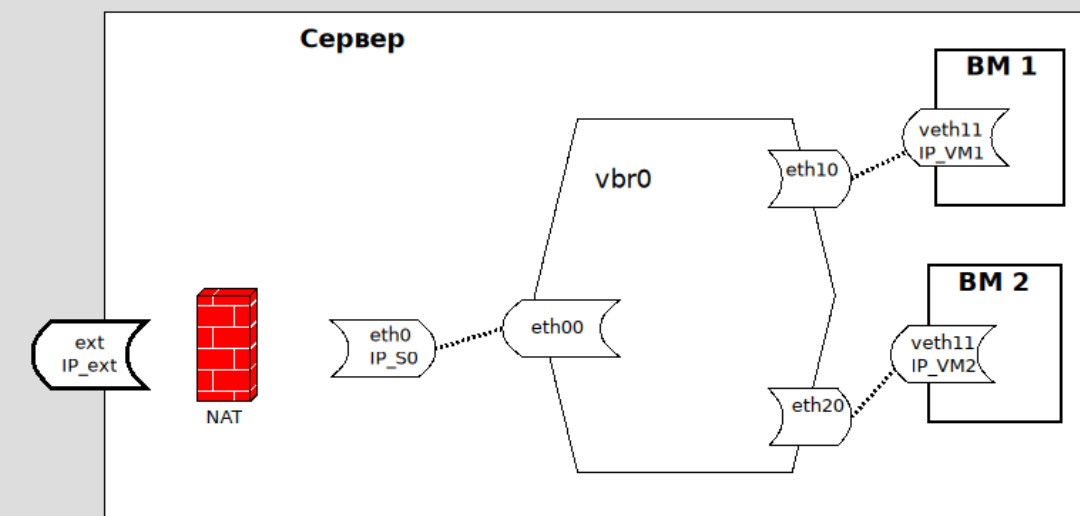
```
ip link add name veth00 type veth peer name veth01
```
  - /dev/tap – API для приложений, сетевой интерфейс уровня L2;
  - /dev/tun – API для приложений, сетевой интерфейс уровня L3;
  - разделение трафика на уровне сетевого интерфейса (MACVLAN, IPVLAN);
  - виртуальные мосты, уровень L2:  

```
ip link add name br0 type bridge
```
  - туннели (инкапсуляция пакетов IP в пакеты IP), уровень L3:  

```
ip link add name ipip0 type ipip local LOCAL_IP remote REMOTE_IP
```
  - ...

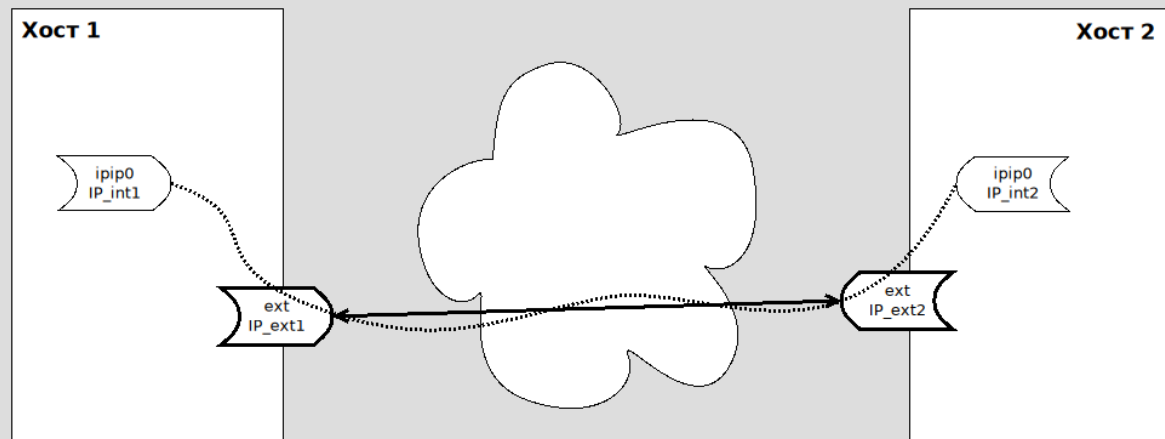
# Виртуальные сети

Сети виртуальных машин:

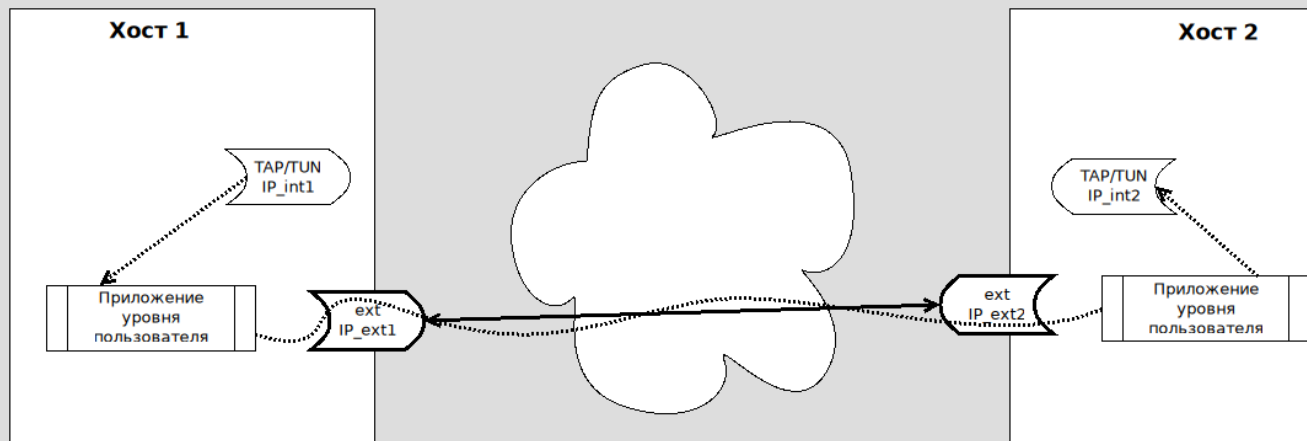


# Виртуальные сети

Виртуальные сети с инкапсуляцией пакетов:



Виртуальные сети через приложения уровня пользователя:



# Виртуальные частные сети

Виртуальные частные сети (VPN, Virtual Private Network):

- позволяют организовывать сети и передавать данные поверх других сетей,
- как правило, обеспечивают защиту передаваемого трафика через публичные сети с неизвестным уровнем доверия,
- как правило, обеспечивают авторизованный доступ.

Типы VPN:

- по способу реализации:
  - аппаратные (на уровне сетевого оборудования)
  - на уровне маршрутизаторов
  - отдельные криптошлюзы
  - программные (на уровне операционной системы)
- по сетевому уровню:
  - уровень L2
  - уровень L3
- по архитектуре:
  - с выделенным сервером VPN
  - без выделенного сервера (mesh-сеть)

Могут быть сертифицированы под законодательные требования по защите данных.

# Виртуальные частные сети

Типы виртуальных частных сетей:

- IPSec (IP Security):
  - расширение для IPv4;
  - встроенное решение для IPv6;
  - симметричное шифрование данных пакетов IP между двумя хостами или двумя сетями через два шлюза;
  - работает на уровне ядра;
  - авторизация и обмен ключами – отдельный протокол IKE уровня пользователя;
  - стандартное решение; есть поддержка в ряде аппаратных маршрутизаторов;
  - есть много разных реализаций IKE;
  - для объединения сетей чаще используется для защиты внутреннего трафика виртуальной сети PPP / IPsec.
- WireGuard
  - более простой по сравнению с IPSec;
  - обмен ключами на уровне ядра;
  - реализация для Linux.

# Виртуальные частные сети

## Программные решения:

- PPTP
  - встроенная поддержка в Microsoft Windows;
  - уровень L3;
  - есть проблемы с производительностью и безопасностью;
- OpenVPN
  - межплатформенный;
  - использует SSL/TLS;
  - поддержка уровней L3 и L2;
  - работает поверх UDP или TCP (стандартный порт — 1194/udp, 1194/tcp)
- ИнфоТеКС ViPNet
  - аппаратные сертифицированные крипто-шлюзы
  - программные клиенты
  - использует SSL/TLS с алгоритмами ГОСТ
- КриптоПро CryptPro NGate
- Cisco AnyConnect
- Cisco 3000 VPN
- Tinc, GVPE, пр.

# Оверлейные сети

Оверлейные сети (Overlay Network):

- виртуальная сеть, организованная других сетей,
- общий случай виртуальных частных сетей (VPN),
- работают поверх TCP/IP и публичных сетей,
- для виртуальной сети используют или TCP/IP, или специфичные для конкретной оверлейной сети протоколы;
- могут как иметь шлюзы в Internet, так и быть изолированными от глобальной сети.

Примеры:

- Freenet
- IPFS (InterPlanetary File System)
- Tor
- I2P
- Yggdrasil



# Размеры сетевых пакетов, их фрагментация и MTU

Размер сетевых пакетов:

- определяется физическими ограничениями: длиной кабелей, скоростью света;
- задаёт MTU (Maximum transmission unit) – максимальный размер пакета данных;
- для физических сетей Ethernet 10/100TX:  
MTU=1500 байт данных (1520 байт с заголовком)
- для физических сетей Ethernet 1000TX и выше:  
возможны Jumbo-пакеты до 9000 байт данных;
- для виртуальных сетей Ethernet – уменьшается на размеры заголовков вложенных пакетов;
- минимальный допустимый MTU – 1300 байт (ограничение, накладываемое IPv6).

Пакеты L3 с данными более MTU разбиваются на части:

- фрагментация пакетов IP

Для TCP возможно динамическое определение MTU конкретного соединения (PMTU, Path MTU) в каждую из сторон

- при работающих соответствующих запросах ICMP.

# Вопросы отладки работы сетей

## Доступность узлов:

ICMP Echo Request/Reply; оно же ping

```
ping [-c N] <IP>
```

## Возможность подключения и передачи данных

```
netcat [-v] <IP> <port>
```

```
netcat [-v] -l <IP> <port>
```

```
telnet <IP> <port>
```

## Путь передачи пакетов:

```
tracpath <IP>
```

```
tracroute <IP>
```

## Определение размера MTU:

```
ping -s $((1500-28)) -M do <IP>
```

## Захват и отображение пакетов

```
tcpdump [-n] [-i <IFACE>] <условие выбора трафика>
```

```
tcpdump -n -i eth0 host 195.208.4.1 and port domain
```

## Сканирование портов удалённых хостов:

```
nmap [-sP] <IP>
```