



**CSMA/CD** - Carrier Sense Multiple Access / Collision Detection      Ethernet has random collisions!

применяется в локальных сетях.

max frame size = 1500B (если не захватывают канал)

min frame size = 64B - чтобы не было конфликтов на небольших расстояниях. Если < 64B - padding (расширение)

**Модернизация Ethernet**

**Мост**: запоминает: интерфейс  $\rightarrow$  MAC  $\rightarrow$  время t - таблица хранит адреса (запоминание - learning)

No new frame  $\rightarrow$  не изменяется значение в таблице; разбивает в отдельные ячейки, т.к.

работает только то, что нужно: фильтрация. Есть I+ collision domain - коллизия не проходит через bridge

**Коммутатор (switch)** - многопортовый мост: интерфейс не знает о существовании 2, 4, 8, 16, 24 или 48 портов

какого коммутатора сети (всю информацию о коммутаторе передает в 1 интерфейс) Мультиплексирование - разделение коммутатором одного домена на несколько.

CAM Table Overflows: адрес на switch: он вынужден пересыпать дальше переходами

Полный буфер - оптимальное решение. Побочное значение switch - интерфейс. Устройство с большим буфером

Мультиплексирование + полный буфер  $\rightarrow$  коммутация

Коллизионный домен - есть сети, в которых возникают коллизии. domain (ограниченный радиусом)

Маршрутизация - есть сети, в которых возникают маршруты. Протоколы (ограниченные радиусом и принципом)

Рекомендации работы switch:

- Store and Forward: сохраняет и пересыпает: получает сокращенные проверки FCS и принимает решение о коммутации. Программируется, что есть конфликт: блокирует возможные ошибки.

- Cut through: предполагает, что есть конфликт: то switch принимает любую мин. возможную frame, которая там DA и сразу начинает пересыпать (если там одна ошибка, то коммутация). Но если ошибки  $\rightarrow$  > склонны  $\rightarrow$  то это приведет к более frame (из-за ошибок передачи и приема)

**Fragment free**: если: принимает первые 64 bytes frame и если выше нет пакета, коммутирует и забирает от исключения конфликта, и передает с min. задержкой

Обработка frame на коммутаторе

### Сетевой уровень

**Адресация**

**IP** - универсальный логический адрес: если мы сидим не в одной сегменте LAN, Bridgeает IP адресов в FDDI. Например. Задаваемые числа группы IP address.

IP address: A : 0... - классовая  
B : 10... - межсетевая  
C : 110... - сетевая

8	1	24
16	1	16
24	1	8

Устройство сопрягает сетевые адреса  $\rightarrow$  устройство прикрепляет один адрес  $\rightarrow$  много разных групп настроек - без повторов, только коммутаторами / hub-ами

Потом убирают что это конкретно: службами назначения хостов

**Сетевая маска**: 32 бита: определяет количество IP address на сетевом и хостовом уровнях:  $\rightarrow$  1 сет  $\rightarrow$  2<sup>n</sup> - 2 IP address

Но не засчитан IP address в группах не 2<sup>n</sup> - не по принципу классовых сетей.

Т.е. требует некоторой конфигурации IP address, или определить группу масок, т.к. они имеют разные количество сетей. IP address = 10.1.2.3  $\rightarrow$  00001000.00000001.00000000.00000000  
mask = 255.255.255.255  $\rightarrow$  11111111.11111111.11111111.11111111

Деление не по принципу битов! В классовом выражении это невозможно

**VLSM** - Variable length Subnet masks: фиксированы сетевые маски  $\rightarrow$  различные хосты.

2 резерв. адреса: адрес сети (subnet): все хосты, кроме = 0  
广播地址: адрес broadcast: все хосты = 1

Хостовыe маски = n бит, n = 32 - [32 - сетевые маски]  $\rightarrow$  1 сет  $\rightarrow$  2<sup>n</sup> - 2 IP address

2 способа: сетевые маски =  $\rightarrow$  можно через оборудование 1-2 уровня (hub, switch, bridge...)  $\neq \rightarrow$  вручную подсчитать, это для router

Max количество сетей в локальной сети  $\rightarrow$  IP  
 $2^{n-2} - 2 \rightarrow n = 3 - \text{хостов.} \rightarrow \text{сетей} = 2^3 - 2 \rightarrow \text{маска} = 255.255.255.248$

**CIDR** формат: Classless Inter-Domain Routing: делит маски через точку: /29

**Зарубежный IPv4**

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	32
Version	IHL	DSCP	TOS	Total Length																											
Identification	Flags	Fragment offset																													
TTL	Protocol	Header checksum																													
Source IP address																															
Destination IP address																															
Options (if IHL > 5)																															

Flags и Fragment offset - срежущие аппаратные. Не фрагментацию пакетов

TTL - Time to Live - время жизни в хостах: при прокладке пути по маршруту TTL  $\downarrow$  на 1. TTL = 0  $\rightarrow$  не живет

**Version** - IPv4/IPv6

**IHL** = Internet Header length  
- гипотетическая

**DSCP** = Different Services Code Point  
- Коды приоритета в различных сетях

**ECN** = Explicit Congestion Notification  
- Гипотетическое сообщение о перегрузке

**ID** - Identifier - идентификатор

Фрагментация - деление исходного IP пакета на 2 или >, когда они были < MTU  
 ID - идентификатор пакета - он всегда один. При этом генерируются несколько фрагментов - так как у ID нет хеша,  
 но есть номер, что это это фрагмент.

3 флаги определяют header IP пакета: DF | MF

PPS - packets per second

DF = don't fragment = 1 => не разбивать пакеты. Если не разбиваем - отправляем. Число пакетов

MF = more fragments = 0 => это последний фрагмент. В исходном пакете он PSH = 1

Каждый фрагмент - полноданный IP-пакет с заголовком и данными.

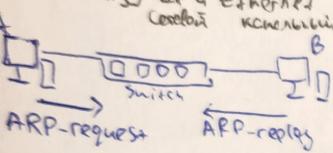
Fragments offset - как отмечать следующий фрагмент исходного

Соответствует IP-пакету из фрагментов: 1й - у него Fragment offset = 0, последний MF = 0. Если конца - то ppfragment не пришло - пакет отрывается.

- Сборка пакета производится в page - получателе (фрагменты, которые пропускаются первым page)
- если определены - то если определены - то Fragment offset = \sum следующий при котором фрагментацию
- заголовок из transport layer будет поменян в 3м отрывке
- Фрагментация - не не делится каждое по MAX MTU, потому что это основное
- иногда может быть больше составных пакетов (или внутренних пакетов), но исходный пакет отдельный пакет (исходные составные пакеты логически).

## (ARP) (Address Resolution Protocol)

61...36 МБУП IP и Ethernet: позволяет в локальном сегменте (где не нужны полярные) избрать IPv4-адрес, иметь MAC.  
 Перед отправкой IP пакета, не зная его MAC, отправляет ARP-request и получает MAC.



ARP-сообщение:

01...785...	15167...	38...31
HTYPE	PTYPE	
HLEN	PLEN	OPER
SMA = sender hardware address		
SPA = sender protocol address		
TMA = target hardware address		
TPA = target protocol address		

HTYPE = Hardware Type - какая технология коммутации используется

PTYPE = Protocol Type - какой протокол используется

(T-o. использует IPv4 <-> Ethernet)

HLEN - сумма адреса протокола + IP. (исп. MAC для Ethernet)

PLEN - сумма адреса протокола + TCP. (исп. IP для IPv4)

OPER - определенные операции: request/replay ...

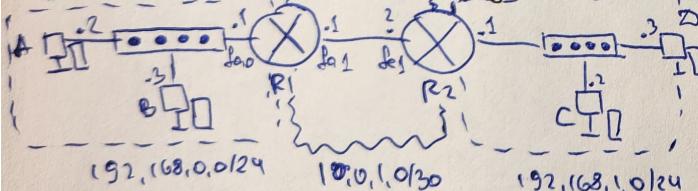
SMA - оригинальный адрес отправителя, TMA - получатель; канабл. IP

SPA - оригинал. адрес отправителя, TPA - получатель. : canabll. IP

TMA при запросе не известен, его мы и будем

ARP - протокол уровня 2

ARP сообщение не является частью мертвого цикла; его неиск. знает MAC устроиство не из локального сегмента



на R1:

C 192.168.0.0/24 fa0

C 10.0.1.0/30 fa1

S 192.168.1.0/24 10.0.1.2

на R2:

C 192.168.0.0/24 -

S 10.0.0.0/0 192.168.0.1

default gateway

- 1) А хочет определить IP B: знает IPаг. B и не знает MAC, с чем мы в локальном B => 99 => узнает ARP-ом MAC узла B
- 2) А хочет определить MAC C: локальный, что C-е в локал. не знает и ARP запрос отправлено бессмыслица

Поскольку А не знает default gateway = спецификация на R1. Для этого использует ARP-request от MAC agree

этого узла по локальному => R1 ему отвечает и А в свою очередь ARP-cache занесет IP-адрес конкретно R1. R1 отвечает MAC agree

ARP cache - таблица из IPагре - MACагре - IPагре и "занесено"

Теперь когда А определил что C-е в локал. он определил ARP-request от MAC agree

узнано IP-пакет - пакеты, что он не знает и отвечает, что это где-то C => узнает MAC и FCS

frame R1 идет MAC R2: он отвечает C => узнает MAC и ARP-reply. И R2 в свою очередь ARP-request с ID = 10.0.1.2.

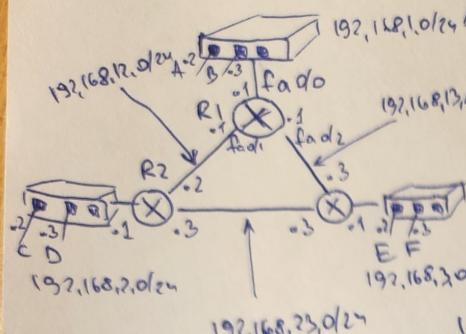
MAC узла C

А ARP сообщение Padding'ется до минимальной Ethernet-frame = 64 байта.

После выполнения ARP reply значение хранится в ARP cache которое будет: если окончание узла  
помечено как одиничный, то уникальный и уникальный (или же занят MAC). Если же помечено как  
использование этого таймера узнает узел использование использование использование.

Protocol - процесс, который передает информацию в логике TCP/UDP/ICMP.  
Основы маршрутизации. Рассмотрим пакет

3 routers, 3 Switch



Классовый маршрут входит в таблицу маршрутизации в формате 192.168.1.0/24  
Маршрутный путь: 192.168.1.1. Адреса узлов и сетей в таблице маршрутизации указываются через -, где X - номерная часть IP-адреса; маска битов /24 в таблице маршрутизации указывает количество битов маски, поэтому адрес 192.168.1.1 имеет значение 1 для R1.  
Как понять, какой маршрут принадлежит конкретному адресу? Если адрес совпадает с адресом сети в таблице маршрутизации, то это маршрут - для этого адреса:  
- правильный адрес 192.168.1.2 - правильный адрес 192.168.1.0/24 - маска 255.255.255.0:

192.168.1.2	(11 000000)	10101000	00 000000	00000000
192.168.1.0	(11 000000)	10101000	00 000000	00000000
255.255.255.0	11111111	11111111	11111111	00000000

192.168.1.2 - маршрут в таблице маршрутизации. Пример: узел A: 192.168.1.2/24 и C 192.168.2.2/24 - правильный маршрут! А если маршрут 192.168.2.0/24, то правильный адрес.

A → R1 → R2 → C.  
Узел A имеет два маршрута в таблице маршрутизации, один из которых имеет маску на один адрес - адрес C: оба маршрута в таблице маршрутизации имеют одинаковую маску, но они в таблице маршрутизации имеют разные маски.

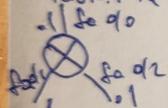
На холостой базе маршрутизации - всего 2 пункта:

- Запись о сети, куда не может пройти
- Маршрут по умолчанию - куда передавать, если путь не в таблице маршрутизации.

IP-адрес соединения R1 канал:

Правильный маршрут - когда путь проходит через один или более посередине.

Тест. маршрут: R1:



C 192.168.1.0/24 fa0/0  
C 192.168.12.0/24 fa0/1  
C 192.168.13.0/24 fa0/2

Положение в таблице маршрутизации  
после включения, но до конфигурации IP  
настройки масок

S 192.168.2.0/24 192.168.12.2  
S 192.168.3.0/24 192.168.13.3

Была введена администратором: Задан маршрут по сетям  
3A R2 и R3. В них нет - не используется, адреса используются,  
которые могут передаваться из-за них звонок звонят

S-static - записи сетей, в которых есть путь для передачи информации  
C-connected - записи сетей, в которых есть путь для передачи информации

Быстро можно сказать что путь не используется, R1 их не знает!

Нашим маршрутом является запись C - 192.168.2.2 => R1 получает, что отработало на R2

Чтобы маршрут не работал - это просто: не использовать fa0/1. R2 не будет генерировать слоты, что узел C - не его интерфейс: он этого события не знает и не на C.

Если в таблице 2 строки - передает, то выбираем с MAX. длинной маски V S 192.168.2.0/24 192.168.12.2  
X S 192.168.0.0/16 192.168.13.3

Испр.: сконфигурированы строки, в которых есть несколько строк с одинаковыми в DA, а значит  
берутся первые строки с самой длинной маской.

AD - administrative distance - уровень, который определяет приоритетность маршрутов (чем меньше, тем выше).  
C-connected - то, что можно настроить: AD=0.

Будут выбраны строки с минимальным AD

S-static - вручную администратор: AD=1

OSPF - распределенный протокол маршрутизации: AD=110 OPEN Shortest Path First

RIP - распределенный протокол маршрутизации: AD=120 Routing Information Protocol

BGP - Path-vector протокол маршрутизации: AD=20 или AD=200 Border Gateway Protocol

Админ. оконце: на R1 находит static на R2, на R2 → R3, но R3 → R1. Но AD у S < AD у C => все будет OK  
но! Если у R2 будет нет связи с R3:

Будет маршрутизация от R1 к R2, R2 к R3, R3 к R1. Тогда маршруты A & C: R1 → R2 → R3 → R1 - первая

route. При котором проходит TTL = 1. Изначально ходят 1 раз и дальше - пересыпается header с учетом TTL

После, на котором TTL = 0 отправляется ICMP-сообщение: TTL exceeded: пакет не может быть доставлен

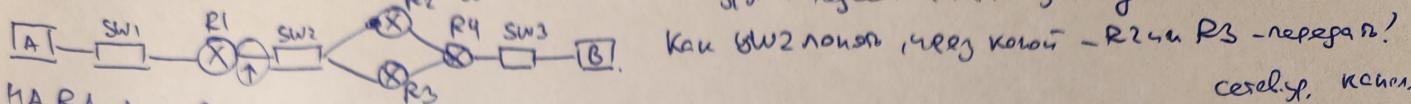
пакета в сетях. ICMP = Internet Control Message Protocol

MTU = Maximum Transmission Unit - максимальный размер пакета который может быть отдан на сеть без ошибок

**Proxy ARP** - (не используете ч3-зя блокица тарбузки)

Роут С и D-клиент, A и D-получатель. Путь на С подключен к узелку IP с 2 на 4 с НЕПРАВИЛЬНОЙ маской 124 → 116. Тогда A → C: он уже помнит его MAC: A → R1 → R2 → C.  
но трафик C Syntex, что A сидит воронийским чуже неизвестной маски и поэтому узла C: C ≠ A.

Proxy несет! Proxy будет у себя ARP request от IP, которого в этом случае быть не может - C знает только IP адрес A и MAC адрес R2(клиента) - он этого не знает, то frame go A going.  
C получает в таблицу IP адрес A и MAC адрес R2(клиента) - он этого не знает, то frame go A going.  
T.p. так же, как это было C Syntex, что A сидит в другом месте (default gateway = MAC R2 (client))



на R1 есть правило что по умолчанию = IP адрес лежит на R2. Т.е. правило IP A MAC - R1 получает IP B MAC - R2

**ICMP** - Internet Control Message Protocol (не использует)

аналогичный протокол Unix (err.) или же несет роль гамбургеров

- **Ping** - команда, отправляем ICMP echo-request: проверка работоспособности узла: полученный это уже gotten отвечает echo-reply

- Отправляем багажные IP-packets с DF=1, не проходя через МГЧ. Мы их обрабатываем, но не передаем ICMP-сообщение, с указанием что пакеты через МГЧ = ... не прошли packets too big

- **Path MTU discovery** - аналогичное определение максимального значения MTU сколько можно: отправлен IP-packet не проходит его, отправляем ICMP-сообщение с указанием MTU, которое не прошло.

- **Trace route** - алгоритм определение IP всех промежуточных, через которых проходит. При TTL=1 пройдет ближайший ICMP TTL exceeded со своим IP. Потом TTL+1 и узнаем все IP адреса промежуточных

- **Сообщение о недостижимости хоста/сервера**: no route to host - пакет пойдет по роутеру, который не знает куда их привести
- ICMP м.б. используется для проверки: кто в IP-адресе является сервером ICMP, потому что IPV4 (они IP-адрес и порт, см. IP-порт). Их можно блокировать для изоляции на промежуточном маршрутизаторе

18.	19.	20.
Time	log	config.
...	...	...
DATA		

### Протоколы передачи

### TCP и UDP

**UDP** - User Datagram Protocol

connectionless: не гарантирует доставку, не организовано исключительное соглашение, засо засоры

**UDP Datagram**

Source port	Destination port	Length	Checksum
DATA			

Source port - порт отправителя  
Destination port - порт получателя

Length - длина заголовка: 16 бит  $\rightarrow 2^{16}$  = max длина.

Порт UDP - от 1 до 65535 - уникальныйидентификатор процесса.

Порты  $\rightarrow$  клиентские - с конечных, получателей

$\rightarrow$  серверные - к нему ОС назначает номер

Первые 1024 порта - well known: DNS = 53, ECHO = 1.

Порт Идентифицирует процесс на хосте, IP-адрес - конкретизирует хост.

В момент порт м.б. назначен только К одному процессу, но это не обязательно

может относится к другому процессу, но это не обязательно

**Sockets** = IP-адрес + Port - идентифицирует процесс глобально. Возможные методы сокетов, не используя, то source port = 10500 (лучшее число из числа свободных, кроме DNS-сервера).

**TCP** - Transmission Control Protocol

протокол управления передачей. В отличие от UDP: передает последовательно спирально (не нужны накопчиатели буфер.ур-и)

Reliable, connection oriented. В отличие от UDP, IP, Ethernet передает

ГАРАНТИРОВАННАЯ ДОСТАВКА + в конце

**TCP Segment**

0-3	4-6	7-15	16-31
source port		destination port	
Sequence number (SN)			
Data offset	reserved	Flags	Window size
			urgentic point
			options
			DATA

И TCP и UDP - мультиплексирование

по номеру порта: т.к. один и тот же IP-адрес

отправляет различные приложения

на узле по номеру порта назначения

приложения, которые генерят

такие номера порта - well known

SP / DP - порядок отправленных байтами: 2 байта  $\Rightarrow 65536 \text{ байт} = 2^{16}$   
Sequence number - номер 1го байта в segment, Acknowledged number - 30-й байт момента приема пакета  
DATA offset - смещение данных; где начинается информационное поле segmentа данных  
 Т.к. TCP зеркальный протокол передачи данных (TCP like IPv4), data offset < size of header  
 TCP-segments начинаются пакетом данных

Flags:

C	E	U	A	P	R	S	F
W	C	R	C	S	S	X	I
R	F	G	K	H	T	M	N

URG = Urgent: важность данных

. Если  $=1$ , то это означает, что есть Ургент-пойнт — границы для этих данных

ACK = Acknowledgement

означает, что пришло подтверждение о приеме данных, следующие данные не должны быть

PSH = Push

предыдущие данные не хранятся в буфере, а сразу переданы пользователю

RST, FIN

— срочные TCP сокеты (RST — нарушение последовательности засыпалось)

SYN — синхронизация Sequence Number и последовательных сегментов

установление соединения.

Three-way handshake: установление соединения в TCP

Classic.

### TCP-сессия

TCP — connection-oriented (отлично от UDP): передача данных лишь после установления соединения.

A

B

1) A  $\rightarrow$  B: SYN=1: пункт B получает Seq. Number: "нечто" от предыдущего пакета начиная с байта NFX"

2) B  $\rightarrow$  A: SYN=1, ACK=1:

последние Seq. Num.: и сразу же подтверждает данные с байта Y

TCP:  $\Rightarrow$  пункт Seq. N. с любых сторон.

Acked: подтверждение насе Acknow. Num. у B: B подтверждает получение пакета.

Ack. Num. = номер байта в пакете, который B отдает подтверждение (seq.  $\Rightarrow$ )

$\Rightarrow$  то есть все получено без ошибок

Seq, ack — номера пакетов

seq = X  $\Rightarrow$  ack = X + 1 — при установлении соединения

если есть число от 0 до d, то ack = ((x-1)+d)+1 = x+d

T.е. B получила x+d-1 пакет, начиная с байта x+d является пакетом.

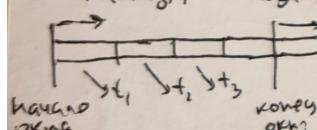
3) A  $\rightarrow$  B: Ack = Y+1: A: я засыпал Seq. Num. и X-1о данных

(?)

Приемные окна Sliding window — скользящее окно:

Определение окна по пакетам: пока не пришло подтверждение с Ack=1, ничего не передано — какая разница!

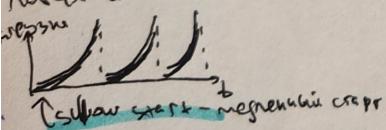
Механизм скользящего окна — определение нескольких сегментов пакетов по получению подтверждения о приеме



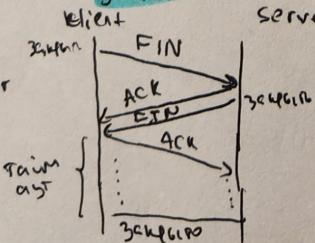
Определение получаемого пакета и сегмента в пакете после этого момента не подтверждение подтверждение о приеме пункта приема определение его сегмента. Пункт приема также может также t1. Это означает, что окно не 1, иначе все подтверждение из него было бы. Если т.е. не пришло, то недавно отдано, где сейчас окно, и сегмент определен заново  $\Rightarrow$  Retransmission — гарантия доставки безошибочно.

T.е. прием окна, определение новых сегментов и прием новых контролируемых расстояний безошибочно

На принимающей стороне — слайдер: то есть прием, и это касается в буферах, т.е. подтверждение про них можно определить только после получения. Принимающая сторона может не засыпать обратного приема, т.е. приема от предыдущего буфера, но не засыпать. Время приема  $\Rightarrow$  window size. window-size — промежуток открытия, в котором нет, то подтверждение  $\Rightarrow$  window size. Кто раньше приводит



Задержка TCP:



Оптимизированное TCP-соединение

1) Задержка подтверждения (Delayed Acknowledgment):  
 Точка подтверждения о получении определена не на каком-либо сегменте, а позже. Пункт приема получает пакеты позже, чем предыдущие. Поэтому

2) Алгоритм Нагла (Nagle)

Если от блочных протоколов приходит пакет с большим количеством информации, т.е. TCP header + Ethernet header + IP header, то количество байтов меньше, чем максимальные. Поэтому

### 3) QoS - Quality of Service

QoS-2 опции: CWR — Congestion Window Reduced

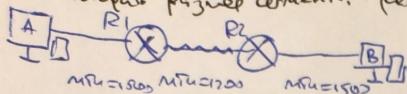
ECN — Explicit Congestion Notification

увеличение пакетов в зависимости от большего количества пакетов.

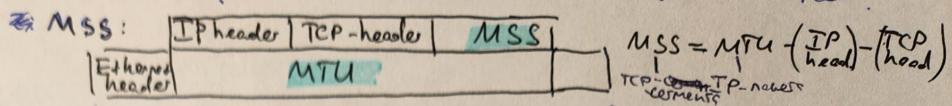
In Payload, window size изменяется больше, т.е. это возможно для

## 4) MSS (Maximum Segment Size), Options фрагменты TCP segments

Как бывает разные сегменты? (секрет IP header & Ethernet опции) = MTU



R1 неизвестно о параметрах: для нагрузки  $\Rightarrow$  1 кадр в секунду. 0 pps  
1: Статус DF=1, R1 отвечает на A ICMP что MSZ = 1200.



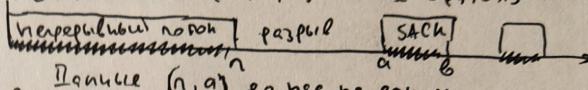
- MSS - максимальный размер полезного блока  
занят в байтах

2: А при этом получает в 1м кадре от A он & none Option содержит один MSS. В ответе в 2м кадре.  
После этого уже можно пропустить неизвестные в параметры, это и есть правильное правило: он уже ставит  
предположение о том что MSS от B один. MSS = MTU - R1 может изменить MSS, только в  
последующем кадре.

## 5) SACK/NACK (Selective ACK, Negative ACK) Options

Важно! TCP can retransmit segment, но может не использовать гамиль, которые присыпали посы.

SACK: подтверждение пакетов, пропущенных с разрывами: когда ходят по пакетам. Т.о. & none Options работают,  
но получают (0,0) (ACK) и (q, l) (SACK) одинаковые пакеты



NACK (TCP не кон. NACK если баг) - указывает что, какие пакеты не получены. Пакеты (q, l) не все не получены.

Важно! TCP определен только отправка (n, l) U[q, l]. С SACK/NACK - меньше перегораживаний

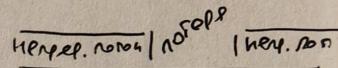
## 6) Scaling Factor Options

Для широких/gigabit каналов: горячий запас полезного времени: канал просачивается

Scaling Factor = n (8 TCP options) : Window size \* =  $2^n$ . Window size = окно: больше гигабитов просачиваются  
из-за перегораживания. Все использование TCP-options гарантирует коррекцию в момент установления соединения

## 7) Fast Retransmission (Быстрая повторная отправка) Options!

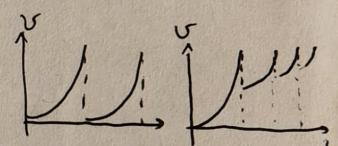
не используют TCP Options, лишь механизм classic TCP. Текущийтаймер не тики: увеличение времени ожидания, сразу отправляет ЗАК и не ждет все время.



Одновременно могут быть отправлены сразу несколько TCP-segments с ACK=1

8) не удаляется при потерях пакетов до min.

При потере не сбрасывается до 0, а в 2 раза: Потом в среднем скорость будет >. Окно не сокращается, а уменьшается до MAX скорости.



## 9) SYN cookie

Проблема: механизм залоготактинга затягивает с TCP-соединением!

Заваливают сервер запросами с SYN, ему нужно выделить память, отвечать ACK+SYN... - SYN Flood

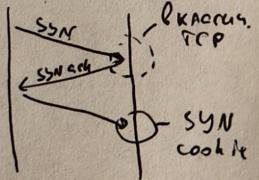
Сервер может забыть адрес: Куда SYN, но отдал ACK нет, так что хост не уходит.

SYN cookie: сервер получает SYN, но не отдает никаких данных. Потом отвечает SYN+ACK

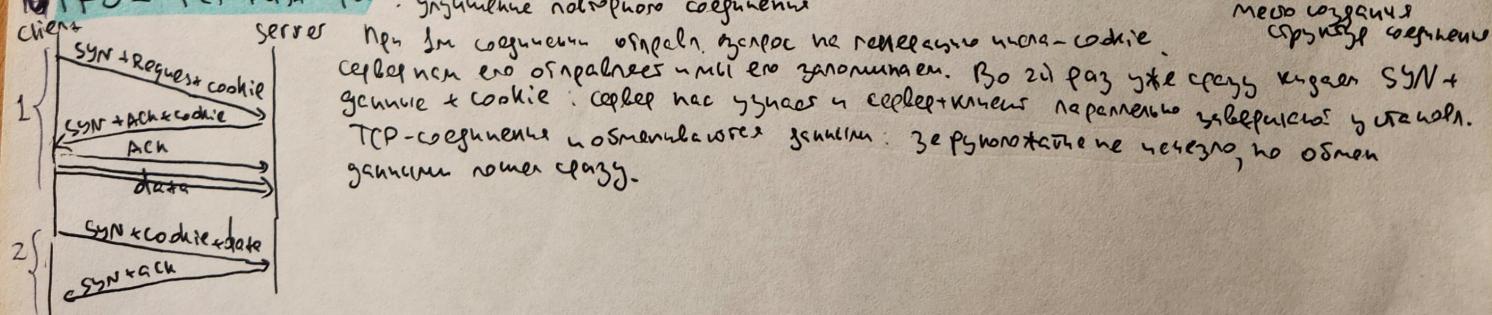
из-за отсутствия ответа. И лишь если клиент примет обещание ACK = Seq + 1, сервер

отправит с тем Seq. и это он отвечает и снова он уже больше никогда не получит.

“+” блокировка “-”: не может использовать больше options - MSS, Scaling Factor, SACK/NACK - из-за соединения клиент



## 10) TFO - TCP Fast Open: Унифицированное соединение



Но для соединения нужно указать число cookie. Сервер не отвечает пока не получит число cookie. Сервер не отвечает и мы его запоминаем. Во 2м разе уже сразу можно SYN+данные + cookie. Сервер не уходит и сервер-клиент параллельно завершают установку. TCP-соединение идентифицируется cookie. Залоготактинг не используется, но обмен данными может быть быстрее.

## DHCP (Dynamic Host Configuration Protocol)

Автоматическое назначение IP адресов на основе динамического.

4 стадии - DORA:

Discover: клиент пишет в лог DHCP-сервер - определяет широковещательный адрес 255.255.255.255, для IP и его же - отвечает на него с адресом 0.0.0.0

Offer: сервер определяет IP параметры клиенту + инф. о времени аренды - lease time

Request: клиент принимает данные и подтверждает, что сервер он принял (также 0.0.0.0 на 255.255.255.255)

Acknowledgement: если сервер не передумал, он пишет клиенту подтвержд. + IP адрес, маску, default gateway

Когда клиент получает 1/2 т аренду он отправляет request, что говорит о необходимости аренды. Сервер пишет магн. сообщение, либо отказ. Если он отказал, то нет т request в 2-м шаге.

**Release:** Клиент отбирает IP-адрес; сервер возвращает адрес в том же самом виде

**Request:** Клиент берет адрес по своему отображению и передает запросы на сервер

**Decline:** Сервер берет клиенту на самом деле занесенный IP-адрес; отказывает

**Inform:** Уведомление о том, что клиент уже имеет IP-адрес

**Бесплатный =**  
**= DHCP over ICMP**

**IPAM.** IP address management - система хранение в одном месте конфигурации IP

**DHCP Relay:** Роли "приводителя" DHCP сервером: на самом деле все запросы Discover на DHCP сервер

**DHCP Option 82**

Все онущие DHCP барьеры в соединении клиент/сервер. Option 82 содержит "уникальный идентификатор" который определяет IP к MAC, чтобы знать откуда/кто клиент не выдавал ему IP-адрес?

Но, как это сделать если нет? Тогда придется IP к конкретному порту - порт коммутатора

**RIP** - Routing Information Protocol - протокол динамической маршрутизации

**Broadcast suppression** - более ранний, чем Router на изменении сети: здесь это больше.

**RIP-DVA** - дистанционно-декомпьютерный алгоритм: router, на котором используется RIP отправляет соседям

метрику + маршрут. Маршрут - узелейная проприетарная

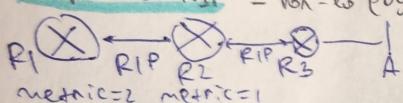
В link-state протоколах router извещает только о том, что есть или нет маршрута, а в DVA протоколах router знает многое о соседях и насколько далеко от них до целевой IP-адреса. RIP использует UDP

**RIP v1:** Классовая маршрутизация  $\rightarrow$  это не корректно. Адрес получателя = Broadcast = 255.255.255.255.

**RIP v2:** IPv4. Адрес получателя = Broadcast или multicast.

**RIP v3:** IPv6. Адрес получателя = multicast (Broadcast)

Маршрут в RIP - non-local router по пути между отправителем и получателем = количество промежуточных

 MAX метрика = 15. Если = 16, то сеть недостижима

Маршрут состоящий из  $\sum_{i=1}^n$  одна единица

Расчет метрики - не оптимальный! Не учитывает пути. ↗

**Таймеры RIP**

- **Update timer (30с)** - частота обновления состояния субнейта: 40с. RIP работает обычно, это всех проприетарных

таймеров это в его базе данных

- **Invalid timer (180с)** - если обнаружено что маршрут не получен за 180с, то маршрут помечен как invalid (метрика = 16)

Если получено update с этого обновления проприетарной. При исчезновении invalid маршрут = 16 и используется

→ **Hold down timer (180с)** - ждет после получения маршрута как временного

Если в это время снова приходит update, то маршрута не будет, но если нет, то маршрута не будет и не будет использоваться.

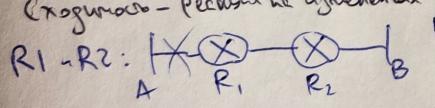
- **Flush timer (240с)** - перезапуск после update. По исчезновению маршрута о проприетарном

Если в это время не получено update, то маршрут станет static. Используется RIP

или используется для очистки маршрутов

**Проблема неприменимости RIP**

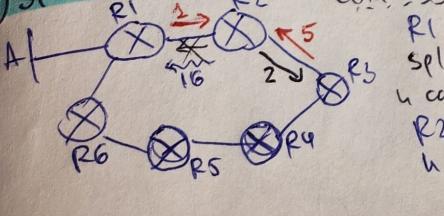
Проблема - because не учитывает в сети.

 R1 получает update от R2: что A доступна через него с метрикой 2. Тогда R2 получает update от R1, что A доступна через него с метрикой 2. R1 знает о том, что A доступна через R2, но не знает что A доступна через него сам, потому что R2 не сообщает о том, что A доступна через него.

Итак, получив update от R2, R1 знает, что A доступна с метрикой 2, R2 знает, что A доступна через R1, но не знает, что A доступна через него сам. R1 знает, что A доступна с метрикой 3, R2 знает, что A доступна с метрикой 2. Кто больше метрика = 16, то это конфликт RIP converge: все роутеры знают, что A недостижима: получили conflict to infinity.

**Оптимизация RIP**

1) **Split horizon**: Тогда router не отправляет обновленную информацию тот, кто которой он не знает, он не может знать.

 R1 получает R2, что A доступна с метрикой 1, а R3 доступна R2, что A доступна с метрикой 5. Split horizon: R2 не отправляет обновленную информацию через интерфейс, через который он это получил и сам неизвестен. R2 использует буфер обновления с метрикой = 1 (min. метрика). R2 заменяет в своем базе. что A метрика = 1 через next hop = R3 или interface R1. И R2 в свою очередь R3 отправляет обновление о том, что A с метрикой 2, а в свою очередь R1 одинаковый маршрут не будет.

2) **Poison reverse**: установление split horizon: в своем базе, от которого получены путь к маршруту в обновленном состоянии метрика = 16 со своим предиктом: он недостижим.

В своем базе, R2 отсылает, что R1 содержит путь метрике со A = 16

**3. Route poisoning:** если какой-то маршрут недоступен, то не отсылаем этот маршрут из update, а instead update с меткой =16 т.к. R1 отключил связь с A, то он не убирает A из update и продолжает отдавать.

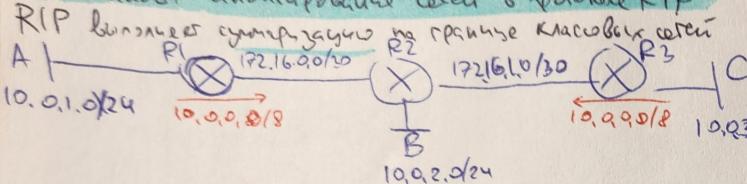
**4. Triggered update:** обновление отсылается сразу после непрерывного invalid timer.

У R1 включена одна цель C; R1 не отсылает обновление update, т.к. не получает обратного invalid timer.  
- рассыпается моментально update о новых и улучшенных маршрутах > update  
быстро реагирует на сеть. Но при пропаже сети - те же гейммы, только медленнее.

**5. Triggered extensions:** На R1 и R2 работает RIP, и засчитывается обновление.  
Несколько маршрутов R1-R2 сортируются такими же критериями, как и для A-B, т.к. обновление на конечном маршруте, но для RIP это конец разницы между!

**triggered extensions:** в зоне обновления передаются все маршруты, иначе, и засчитывается нет. R1 R2  
update отсылаются если что-то произошло: изменение сети.

### Особенности анонирования сетей в протоколе RIP



Сети 10.0.1.0/24, 10.0.2.0/24, 10.0.3.0/24 – класс A,  
172.16.0.0/30 – класс B. > R1 сортирует  
разные классы сетей: RIP автоматически суммирует.  
то есть класс: R1 в своем R2 будет отсылать  
уведомление о 10.0.1.0/24, а о классах 10.0.0/8

т.е. при отправке data по B в A, R2 не будет знать, куда отсыпало! R1-R2 оба делают одинаковые преобразования  
выполнил балансировку: **ECMP (Equal Cost MultiPath)** – балансировка по тому пути с одинаковыми метриками  
т.е. правило 50% пакетов по B пойдет в A > потому отсылают одинаковую суммарную стоимость.  
Тогда R1-R2 отсылают в R2 таблицу маршрутов 10.0.1.0/24 и 10.0.3.0/24 и все ОК.

**Режим суммаризации:** если хотим отсыпать все адреса по одному маршруту. Например: вместо 10.0.1.0/24 отсыпать  
маршрут по 10.0.0.0/16, который будет, чем раньше есть. Автомат. суммаризация – только по классам сетей  
режим – это когда, что! RIP не отсылает более суммаризацию по супerset: с меньшей классом сетей  
Например: 10.0.1.0/24 Ещё A, а класс A маска = /8 > есть 10.0.1.0/24 потому суммаризация  
суммаризация необходима чтобы уменьшить количество передаваемых по сети и ↓ размер табл. маршрута  
(RIB – Routing Information Base)

### NAT / PAT

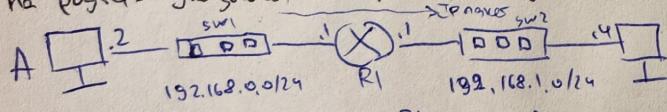
Задача

Что такое NAT и зачем используется IPv4.

**NAT (Network Address Translation)** – трансляция сетевых адресов: переводит в заголовке IP пакета внутренний  
IP-адрес DA и SA: в связи с ограниченностью, не хватает свободных адресов, но их хватает.

### Статический NAT:

На роутере указывается что привязана определенная конкретная машина.



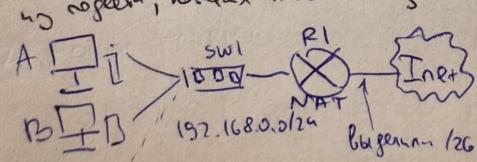
Пусть узел А подключен к узлу В и выше  
изменен. В его конфигурации: основного по В узел маркируется  
по умолчанию: тут же будет написано этим узлом, т.к. по  
его трафик доходит, а обратно – нет.

Технология NAT помогает: на R1 настроен static NAT, при прохождении data от A по B R1 будет подменять  
IP-адрес узла A на адрес 192.168.1.100: Тот же узел в синтаксисе маршрутизации называется его портом  
и узел B не отвечает на default gateway: этот адрес получает напрямую. Он отсылает ARP-request  
по его IP и своим MAC-адресом R1 и узел B формирует ответ с адресом порта 192.168.1.100  
R1 получает на данный порт запрос и находит NAT-таблицу, помогает IP обратно на 192.168.1.2, этот  
адрес – обязан по своим правилам. Круглосуточно работает.

т.е. Задача NAT: b → меняет IP-адрес отображения, b ← меняет обратный IP-адрес назначения

### Динамический NAT

Две трансляции групповых адресов: первый номер группы, который имеет LAN трансляцию 1 адреса,  
и второй, который имеет группу 2 адресов



В LAN все узлы имеют 192.168.0.0/24: первые (глобально не маркируются) адреса  
также называются в глобальном сенсе не назначаются: они уникальны только  
в нашем LAN. Необходимо номер который нам /26: создавая динамический NAT  
всюду нужно помнить: нет однозначного соответствие между сетью  
и 128 и 126: 2^8=256. Но! В нашем LAN однозначно, кроме Гер.

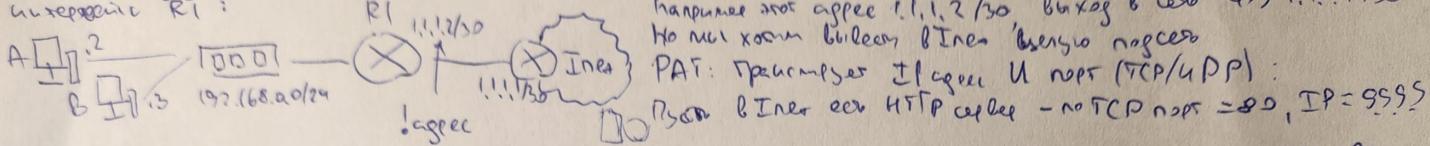
Работает не все время

Ноутбук подключен к сети из диапазона 192 и имеет адреса 192.168.0.2/24, а его порт 80 открыт. Динамический NAT - это когда адреса из 192 и 192.168.0.2/24 не скрыты за маршрутизатором, пока порт разбирается и отвечает запросу, R1 заменяет его IP, когда ответ обрывается - меняется на его IP. Важно. Ходят в NAT - правила засекают, "засекают" адрес из 192 не подключены.

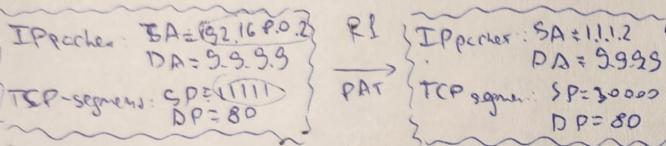
**PAT (Port Address Translation)** - приведение адресов с использованием порта приват. службы, с которых (3,4) в них не используется услуга, соответствующая процессору на устройстве.

### Динамический PAT

Схема из предыдущего примера: Пробайкер выделил только 1 адрес для выхода в сеть, это настроено на маршрутизаторе R1:



Ходят в сети A: маршрут к сайту: основной, народ 99.99.99.99: SA IPad = 192.168.0.2, DA = 99.99.99.99. Порт наружный = 80, отправителя (Random > 1000) = 11111: SP = 11111, DP = 80. Тогда R1 DA IP не меняется, SAIP = 1.1.1.2, SP = 80, а DP = порт 11111 (или порт 11111 в PAT-таблице уже занят). Порт DP один = 30000, т.к. PAT содержит таблицу: что на каком было засекено + какой там порт (TCP/UDP).



При этом порт 80 может быть занят другим сайтом на разных сетях. L-local, G-global

**Inside Local** - адрес хоста в LAN, когда номер в LAN **Outside Local** - адрес хоста в GAN когда номер в LAN  
**Inside Global** - адрес хоста в LAN, когда номер в GAN **Outside Global** - адрес хоста в GAN, когда номер в GAN  
 Если в сети A есть IP 192.168.0.2/24, то IP меняется на 1.1.1.2, а порт - номер 30000  
 Все это существо, что с этим бывает связано, что 1.1.1.2 с разными портами

### Статический PAT

Вручную можно занести все PAT-таблицы приведения.

Пример 1. Определяем сервис на хосте A (нужен HTTP-server). Тогда хосту надо присвоить не занятой int. R1 назначит хосту A → не R1 адреса 192.168.0.2:80 → 1.1.1.2:80 TCP

Пример 2. на хосте A HTTP server → 2 процесса: 1) обрабатывают клиентов из LAN на 80 порту, 2) - клиентов из Интернета на 8080 порту: они получают одинаковый контент. Но порт 80 занятому не используется! на R1 нужно внести: PAT-правила: 192.168.0.2:8080 → 1.1.1.2:80 TCP. Не нужно никаких переключений в Inet на порт 8080, все будет работать как надо и на порту 80.

### Режимы работы NAT

- **Symmetric**. Самый распространенный. MAX. ограничение в работе приведено: адрес Inside Local → Inside Global неизменяется и адрес Outside Global. Впринципе R1 обрабатывает несколько трафика, т.к. есть один и тот же адрес - отображается. +: более легкая реализация -: некоторые протоколы работают не хотят

- **Full Cone (point-to-house)**: полное соответствие Symmetric.

Ноутбук имеет TEP номер, АРН DA, SP+SA значение не меняется!

Неопределенные режимы: Restricted (ограниченный) cone

Address Restricted Cone: неизменяются DP, DA и SA, SP-изменяется

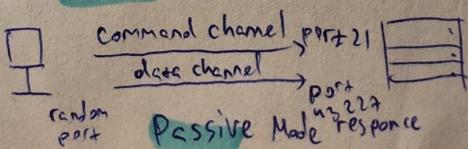
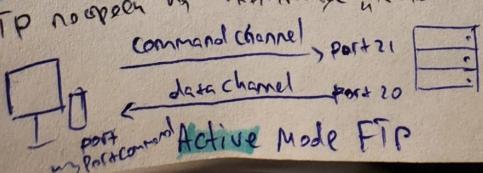
Port Restricted Cone: неизменяются DP, DA и SP, SA-изменяется.

Часто используемое NAT может помешать некоторым софтверным приложениям. FTP, SIP помешают многое используемое услуги - ALG - Application Layer Gateway

### FTP и NAT

**FTP - File Transfer Protocol** - несет два канала на сеть. Установлено несколько TCP/UDP-сессий в FTP-протоколе: 1) установление control-session → есть подключение, но который передает данные 2) устанавливается data-session впереди или позади, что хост спрашивает 2) устанавливается 2-й (рабочий) session. Использует для этого механизм ALG.

FTP поддерживает "command-response": 2 режима активный/пассивный.



- A Cecue: (control-session) и B Active, B Passive: устанавливается от клиента к серверу.
- B Passive: сеанс симметричный - от клиента к серверу, нет разницы с NAT/PAT.
- B Active: один клиентом есть порт с NAT/PAT-проксируется, но сервер получает IP от клиента - это IP порта и т.д. Результат: сервер получает устанавливается с результатом соединения, но на порте этого соединения никого не знает: соединение отбрасывается. FTP не работает, скажи что ничего не можете

### ALG (Application Layer Gateway)

назначение ALG

Работает на порте, анализирует протокол TCP/UDP (приложения), какими FTP-командами обменялся клиент и сервер. Видит, что клиент в отдаленном порте и сервер подключился непосредственно по 2-му сессии не этот порт, и этот порт на себе может создать динамичное NAT/PAT правило для подключения. Со временем сервера непривязаны не первый порт второго клиента. Он сам создает для NAT/PAT правило!

ALG:

- 1) Упрощено (порт откладывает и читает правила TCP)
- 2) Не совсем корректно работает

+ клиент все время занимает один и тот же порт

назначение UPnP

### UPnP (Universal plug-and-play)

База для занимает место. Использует функциональные клиенты и служебные программы.

Клиентские протоколы: открытое порт → UPnP на порт откладывает правило. → порт получает управление от UPnP-устройства и firewall

- UPnP:
- 1) Не работает наслаждаемых за границами (границы за границы)
  - 2) Не на всех портах

### IPv6

В IPv6 уже получили блок IPv6 - нехватка адресов. Временное решение - блоки от классов масок (VLSM + CIDR), NAT/PAT и DHCP v6, постоянное решение - IPv6.

Принцип IPv6 128бит - биты группируют. Пример: 2001:0000:0db8:0000:0000:0000:0000:0000

128бит в 16бит. деление на 2 байта = 416-битовыми. Пример: 2001:0000:0db8:0000:0000:0000:0000:0000

Сокращение записи IPv6 адреса

/64 - стандартные маски

- 1) группы подобъектов группируются в каждую группу узла
- 2) узел имеет формат группу головокузы /0

2001:0:db8:0:0:790:765d

2001:0:db8::790:765d

:/0 - маркирует то что осталось (запись в виде /0).

2001:2345:6789::/64 адрес стандартной головки-роутера

↑ Interface ID (хорошо в IPv4) - всегда скопируется

### Типы IPv6 адресов:

unicast	multicast	anycast
1 конкретный узел	Все в борадка	Несколько близлежащих узлов с таким IP

Broadcast!

Группы Unicast адресов:

- Глобальные (2000::/3) - глобально маршрутизируемые : от 2000 до 3 FFFF
- Unique Local (FC00::/7) - LAN или шире в Internet
  - 8 бит = L=local flag показывает, что предыдущий октет (L=1, FC00::/7) или адрес порт. Все биты (L=0, FE00::/8)
- Link Local (FE80::/10) - не маршрутизируемые, назначаются автоматически, идент. в локальной сети
  - то же принципиально можно в одном L2 сегменте сначала через порт не пройдет к адресу как называет адрес из этой группы на один интерфейс.
  - на нескольких интерфейсах устройство может быть адресовано в одном портни.

### Распределение IPv6 адресов

Первый распределение получает в IANA большие блоки адресов ~ /12.

Представляем - это блоки ~ /32, конеч. пользователем - блоки 1/48 - 1/56.

Получатель делит свой блок на блоки /64

Обычно маски /64, т.к. тоже Interface ID будет меняться часто, адреса - 48 бит.

## Применение IPv6

1) Большое адресное пространство

2) Сокращение времени настройки: Сокращение DHCP (как в IPv4) и + SLAAC (хорошее для быстрой конфигурации)

3) Интернет-безопасность (за счет более сложной настройки)

4) Больше возможностей для маршрутизации (Больше блоков /64, логика на /96 → 32 биты для маршрутизации).

5) Нет необходимости в NAT/PAT

6) Broadcast → широковещательный broadcast. Потому? →

BIPv6 использует 243 узла для ARP, в IPv6 - ICMPv6.

Multicast → Solicited - запросы - определяются по IPv6 Unicast address

Multicast address → assigned - назначенные

В ARP пакете есть поле на Broadcast адрес называется LTA, а здесь в ICMPv6 - destination port → для чего?

Задача ICMPv6, MAC адреса передаются не multicast, но broadcast. IPv6 Unicast

Multicast адреса группам - может содержать в "таблицах" IPv6 Unicast адресами.

7) оптимизация & заоноление

### Заоноление IPv6

ver	4	8	12	16	...	32	361	...	48	52	56	60	64
traffic class							Flow Label		Payload Length	Next Header	Hop Limit		

Source address  
Destination address

Flow Label - идентификатор потока: чтобы пакет оказался определенным  
избыточные bytes для этого не нужны

поток = flowlabel + IPv6 address

- 1) Для IPv6 функционирует лучше! В IPv4 более Options, что затрудняет маршрутизацию
- 2) нет checksum: одна из причин в Ethernet (256) и TCP/UDP (456)
- 3) нет MF, DF, fragment offset: фрагментация возможна только определенным образом, он всегда фрагментируется, но не может быть разбит на несколько фрагментов
- 4) none options - route record: отсутствует. Фрагментации по пути пакета нет
- 5) next header - не всегда супер классический протокол, но иногда какая-либо специальная информация

Ver - версия = 6

traffic class - класс для сервиса  
(DSCP и ECN в IPv4)

Payload Length - сумма длины

Next Header - тип следующего (Protocol & IPv6)

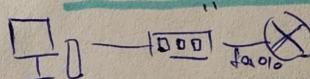
Hop Limit = TTL

### Методы назначения IPv6 адресов

1) статическое

2) статичный DHCP = DHCPv6

3) SLAAC: Router Advertisements (RA - Router Advertisement), это соглашение между роутером и клиентом, чтобы назначить, какие адреса и какие типы (default gateway) ⇒ это же самое что и DHCPv6. Если у вас есть роутер ICMP RA, то он сам определяет ICMP RS (Router Solicitation) = есть ли кто-нибудь? Рассыпает Router RA; RS - это spec ff02::1 - "all routers" (multicast), а RA - это spec ff02::1 - "all hosts" (multicast)



### Как назначается Interface ID?

Нельзя 64 битов, потому что пакеты от пакета RA

1) стационарная конфигурация EUI-64 (на основе MAC-адр.) EUI = Extended Unique Identifier

2) адреса с динамической конфигурацией (на основе MAC-адр.) EUI = Extended Unique Identifier

3) адреса с временным назначением: генерируются случайным образом

При получении IP по SLAAC, не нужно звать на ручную настройку: автоматически назначаются по умолчанию

DHCPv6 и аналогичные, но в отличие от DHCPv6 (ничего не меняется)

но есть ограничение DNS серверов

3) адреса с временным назначением EUI-64: назначение 64 битов - случайно

4) адреса со статикой

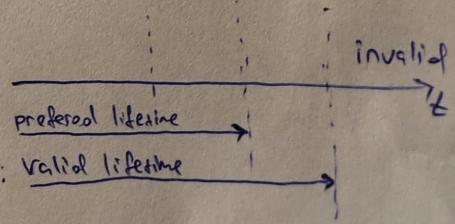
### Время жизни SLAAC адреса IPv6

Agree: времена, но SLAAC + неизменяющийся Interface ID неизменяется до конца; Valid lifetime

- Preferred lifetime: время жизни + неизменение адреса. Стандарт

(но изменениями, которые не влияют на время жизни, то есть неизменяет хеш!

(в IPv6 не ограничено M.S. неизменяющиеся IPv6 адреса, то есть различные состояния).



- Valid lifetime  
Некоторые параметры - в котором Preferred Lifetime: более высокий, - на самом деле, а это уже подразумевается.
- invalid - другие могут быть недействительными (например, если значение слишком большое).
- Preferrable/Valid lifetime ~ время жизни

## NDP (Neighbour Discovery Protocol)

Использует ICMPv6 для обнаружения соседей. NDP: блокирует MAC при работе с IP  
 Режим работы - RS и RA, поиск соседей: NS (Neighbour Solicitation (ARP-request))  
 Анонсирование: NA (Neighbour Advertisement (ARP-answer))  
 В отличие от IPv4 не имеет зон блокировки ICMP, зону можно использовать ICMPv6 - чтобы ICMPv6 могли обнаруживаться

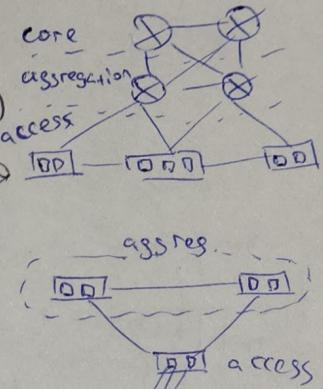
## STP (Spanning Tree Protocol) - (блуждающие гигиены)

Принцип работы на 2м уровне

3 способа в архитектуре: STP (L3 маршрутизатор), мультиплексор (L2 мост), группировка (L2 мост).  
 Важно - это fullmesh или partial mesh, STP несет все данные

На 2м уровне группировка лучше всего, но лучше всего, но худшее значение  $\infty$   
 В IP-сетях лучше всего TTL, а в Ethernet - MST

Принцип: 2 switch на одном aggregation и один порт switch уровня группировки  
 Ethernet в такой топологии работает как мост, есть механизмы, позволяющие не зацикливаться



## VLAN (Virtual Local Area Network) - (виртуальные локальные сети в группе хостов)

С основным назначением, включая общий доступ к общим подключениям и ограничение broadcast-групп.

Комиссия: назначение коммутации - по управлению в конкретной сети. Т.е. на 2-ом уровне - это коммутация коммутаторов

На 2-ом уровне назначение коммутации - это управление группами. ВМесто этого коммутаторы 1-ого уровня коммутаторы.

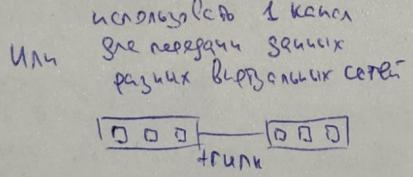
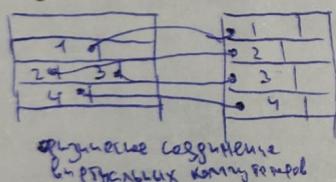
Если все порты назначения, получат 1-ую порту L2-группы. Чем больше групп, тем выше производительность

(физическое коммутаторов не много производительных)

+ 1) более тесные взаимодействия между коммутаторами (если не ограничиваются портами общего коммутатора)

2) при необходимости пересекающихся групп все уединено

Требует создания 2-ух новых switchов:



trunk - соединение, по которому могут передаваться данные нескольких VLAN-групп

Каналы, открытый в trunk, наз. мультиплексор.  
 IEEE 802.1Q - мультиплексор,  
 поддерживает VLANы на уровне  
 802.1Q стандарта.

Ethernet - мультиплексор

## QoS - Quality of Service

CFI - Canonical Format Identifier - какой канальный формат

VID - VLAN Identifier - номер VLANа - маркировка определенных

Type - тип коннективности (какой протокол)

Port-based VLAN: назначение в коммутаторах одинаковое для всех VLANов. Но на разных коммутаторах может быть разное назначение (разные коммутаторы на разных коммутаторах)

tag-based VLAN: назначение в коммутаторах одинаковое для всех VLANов. Но на разных коммутаторах может быть разное назначение (разные коммутаторы на разных коммутаторах)

↑ QoS | Идентификатор = CFI | VID | Type |  $\sum$  4 бита

38. 18. 125. 165

CFI = 0 - Ethernet

CFI = 1 - IEEE 802.3 (Token Ring)

VID 3=125  $\Rightarrow$  4096 VLANов

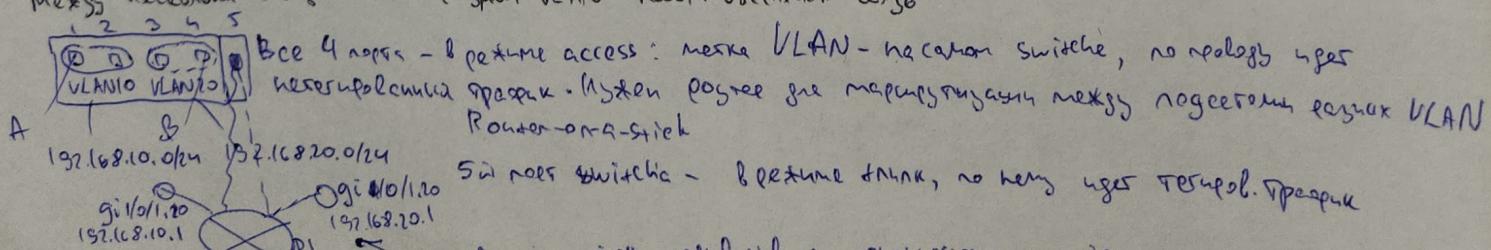
на коммутаторах

2 разных коммутаторах

5 разных VLANов

## Маркированный мульти VLAN (Router-on-a-stick)

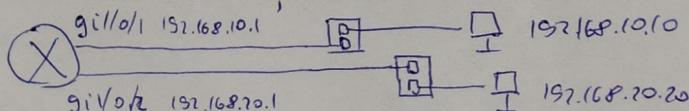
Маркированный мульти VLAN - в党总支 VLAN нестыковка данных



На каждом Sub-intf. может настроить IP address: ~~настройка Sub-interface~~  $gi1/0/1.10 \rightarrow gi1/0/1.20$   
привязки к физическому intf.  $gi1/0/1$ . На хосте за  $gi1/0$  default =  $192.168.10.1$ .  
также от A к B:

- 1) хост A: frame NOT tag I receive access на VLAN10, т.е. A знает, что B не в его сети
- 2) определяет A, что B не на switch, т.е. MAC-адр.  $gi1/0/10$  и адрес на R1
- 3) R1 получает это и возвращает ответ  $gi1/0/20$
- 4) switch отправляет на порт VLAN20 IP-адреса - на порт  $gi1/0/20$ , считает  $TG=20$  и возвращает B

т.е. фактическая схема = логическая схема



### QinQ - технология, применение 2 раза: вложенная и вынужденная

Хотим чтобы пользователь подключился к нам 2 раза L2 клиентом  
Он зайдет на наш client switch за клиентом: там вложенное  
интерфейсы - Port based VLAN. Вынужденный client, или он просто  
использует VLAN trunk. т.е. где оно клиент откуда VLAN.

Но! Тогда клиент уходит A, B - логика VLAN, A2B2 - в группу и т.д.: Терминалы - группы  
использования VLANов

Решение 1: багет ему VLANы с 120 до 129. Но клиент может терять 1000+ в сети оператора M.S. и не более  
4096 VLANов, а если какими-то образом 1000 VLANов, то физически этого нет у клиента

Решение 2: QinQ - 2 уровня: 2 метки протокола 802.11q:  
т.е. оно же клиент - VLAN в своем операторе, а дальше VLANы клиентов  
посылаются клиенту. Терминалы могут использовать только VLAN оператора

VLAN-Translation: 2 клиента хотят VLANы с 120 до 129 - реального. Там у них есть 120-129 и 140-149,  
но заявлены новые VLANы на switchах

PBB (Provider Backbone Bridging) - определение уникальных групп работников > групп & назначение коммутации  
Provider Backbone Bridging

### Классический STP = IEEE 802.1D

STP - позволяет подключать сети между собой L2 группы.

Правило: проприетарные сети не могут находиться на одном SW.  
Порядок работы STP:  
 1) определен Root switch - который будет называться Root SW1  
 2) все остальные Switchы называют Root Port (RP) - интерфейс, через который они будут подключены к Root switch  
 3) для всех остальных switchов называется Designated Port (DP) - назначенный порт: чтобы не было...  
 Задача метки SW2 и SW3 2 различных порта - between them, остальные заблокированы  
 a) блокировка все остальные ( кроме RP и DP ) - (выделено)  $\Rightarrow$  называем затяжкой  
 Если пакет приходит из RP, то STP не распространяется дальше.

Когда switch становится Root Bridge и может всем BPDU-сообщениям присвоить BID  
RID - Root Identifier, BID = priority + MAC = 800000 + проприетарный MAC-адрес

У обычных L2-switches, нет MAC, но есть, что такое в STP, 3 MAC.

Раскрытие по BID: чем меньше, тем предпочтительнее (наибольший имеет приоритет. Root = самый old)  
Получивший первое сообщение, становится Root (его MAC < MAC2 < MAC3 и т.д. - root).  
Т.е., это же root, BPDU-сообщения не идет, а когда BPDU от root'a и распространяется

Root уходит всем BPDU, в котором:

1) BID (наиболее старые switch-ы читают старые BID)  
2) RPC = Root Path Cost - стоимость пути до корневого switch

Root port - min. bridge (the more number min)  $\rightarrow$  1st bridge in!  
designated - min. message (the more number min)  $\rightarrow$  1st bridge in!

## Проблемы classic STP

- Тайм-аут: BPDUs - в зоне, если нет-то сократить до 200 мс и пересыпать дальше - только через 20 с (не раньше 10 BPDUs)  $\Rightarrow$  20 с. Проблемы могут возникнуть.

Соединение интерфейсов: blocking  $\rightarrow$  listening (ненадежные соединения не передаются, 15с = forwarding delay),  
 $\rightarrow$  learning (такие 15с: заслонение соседней станции, то же нет надежных взаимодействий)  $\rightarrow$  forwarding  
 (это же неудобно для них) listening  $\rightarrow$  learning - недостаточно времени.

Короткое & длинное - бессмыслица для протокола.

## Быстро меняющееся STP

RSTP (Rapid Spanning Tree Protocol - быстрый STP).

Минимизация временных склонностей: в случае отказа обоговариваются временные ограничения для передачи BPDUs.

RSTP не наследует все таймауты, но они все равно есть: RSTP  $\neq$  STP временные = 0!

В RSTP есть алгоритм Proposal / Agreement - упрощает склонности

1) выбирает корневого фабрика: root switch - как в STP

2) Root switch становится беем интерфейса и несет имя proposer чтобы BPDUs Proposal

3) Все остальные switch'ы становятся беком и несут имя proposal

и не несут имя отвечающих фабрик - Agreement

4) Root переносится беком, несет имя agreement

5) Остальные switch'ы не называют корневым switchом и несут proposal, не неся имя agreement - и т.д.

Итог: идет "борьба" - все интерфейсы должны работать в режиме full duplex  
 Коэффициент Proposal / Agreement зависит от количества интерфейсов, необходимого для изоляции - это никаких таймаутов

В RSTP blocking  $\rightarrow$  discarding: только неиспользованные BPDUs

3) Spanning tree  $\rightarrow$  alternative spanning tree  $\rightarrow$  backup spanning tree назначение

- PVST+ (Per VLAN Spanning Tree Protocol) [Cisco]  $\rightarrow$  PC

PC  $\rightarrow$  802.1Q. Для каждого VLAN - отдельные фабрики

- Rapid PVST+ [Cisco] = RSTP + PVST+

Запускает сразу для каждого VLAN один RSTP

## Оптимизация STP

1) Root Guard - предотвращение наличия нескольких корневых фабрик.

настройка на 1 root для switch'ов, но наличие к корню. Выбирает root-a

2) Loop Guard - избегает образования петель (  $\leftrightarrow \Rightarrow \rightarrow$  )

3) Port Fast: blocking  $\rightarrow$  forwarding: раньше listening и learning

4) BPDUs forward: выходят от получивших BPDUs, временно блокируя разобщение

5) Backbone Fast: более быстрое время.

6) Bridge Assurance

## Технологии агрегации каналов

2 приложений к которым, но не может хранить информацию:

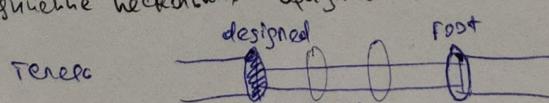
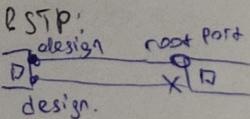
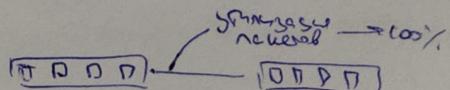
Большой временной:

1) более высокопроизводительные соединения - головной!

2) несколько каналов: работы лучше 1! производительность выше (генератор BPDUs)

$\rightarrow$  технология LAN = Link Aggregation Group (Ether Channel / Port Channel,)

оединение нескольких физических Ethernet-каналов в 1 логический



Объединение портов одного коммутатора для передачи данных. Т.к.

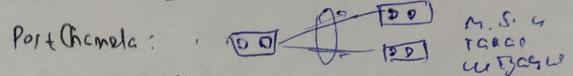
- switch
- port trunk
- native VLAN (для VLAN, данные которого передаются без ферма)
- VLAN tagging (все управл. и доступ, разделяются на trunk)
- trunk (разделение порта / trunk)

Число ограничено, но не - STP!

Максимум LAG между L2-L3 устройствами

### Способы построения LAG

- Статическое объединение каналов - устанавливается вручную ELAG (ручно) — не требуется VLAN на соединение (настройка не требуется)
  - Динамическое объединение каналов с помощью LACP/PAgP но! не все знают, что это такое
  - LACP = Link Aggregation Control Protocol
- Также существует коммутация VLAN-ов, т.к. они обмениваются не данными  
В принципе ID используется — настройка MAC-адреса LACPDB с публичным MAC-адресом  $\Rightarrow$  канал не потерян  
LACP: go to 16 универсальных PortChannel, которых обнаруживаются и не разбиваются



### Балансировка трафика в LAG

Как поделить трафик при соединении? Хеш-функцией

Распределение M.S. MAC, IP (+ порты, + протокол: TCP/UDP)  $\Rightarrow$  Hash = f(MAC<sub>A</sub>, MAC<sub>B</sub>, IP<sub>A</sub>, IP<sub>B</sub>, Protocol, Port<sub>A</sub>, Port<sub>B</sub>)  
И смотрим основной алгоритм этого хеша на хешах пакетов IP и TCP/UDP

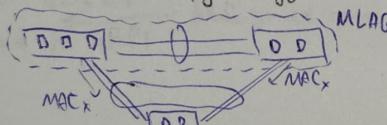
### Оптимизация LAG

В классическом PortChannel используется один и тот же MAC-адрес

$\rightarrow$  vPC (Virtual PortChannel) / MLAG (Multi-chassis LAG) / VSS (Virtual Switch System)

Traffic 2 switches передается в vPC / MLAG — reply и request ссылаются на один и тот же MAC-адрес (отдельные MAC-адреса)

Также можно использовать скрепки



## OSPF

OSPF = Open Shortest Path First

Информация о минимальных маршрутизациях (как в RIP), но этот — не DVA, а топология: он считает в себе  
области и Link State Database.

Zones (area) — собирательство интерфейсов маршрутизаторов. На интерфейсах и связях между зонами сопоставляются

Переходные OSPF-зоны называются промежуточными и переходными

Разные типы зон:

1) area 0 (backbone) — основная зона, все остальные подчинены ей

2 зоны (не 0) не могут соединяться без подключения к 0-му

2) area k (standart)  $k > 1$

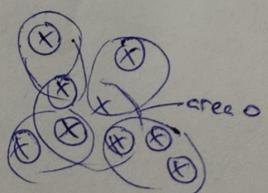
Типы переходов:

Internal (изнутри зоны) — маршрутизатор, все интерф. которого в одной зоне (подразд.)

ABR — Area Border Router (переходной) — есть его интерф. — в 0-й зоне, но есть в других

ASBR — Autonomous System Boundary Router (переходной между системами) — стоит на границе OSPF зонам  
и границам зон — RIP, BGP... ~

Пограничные ABR и ASBR называются



3) Stub (stubzone) — где не могут находиться другие OSPF-зоны переходные. ASBR — поставлен в зоне не M.S.

4) totally stubby (non-backbone transition zone) — где не могут находиться другие OSPF-зоны переходные

Изоляция, чтобы избежать ошибок. Рекомендации практика

## ▪ Типы маршрузов между зонами:

- Intra-zone (внутризональные)
- Inter-zone (межзональные) - OSPF-зоны, но в разных зонах
- External (Внешние) - различные протоколы (RIP/OSPF...)

В рамках одной зоны OSPF - link state информации: посыпки генерируются LSA = Link State Advertisement:

На этой основе они составляют топологию сети: A routers report ее на основе LSA от соседей.

При передаче между зонами OSPF = DNA: передается только метрика до приватной из зоны 3061.

Идентификатор зоны - для упрощения назначения топологии. Топология передается через зону.

ABR имеет топологию двух зон, к которым он подключен.

Если изменение в one zone, то router все пересчитает, если в другой - он от ABR будет забирать информацию

## Типы LSA

- LSA1 (Router LSA) - описывает router и его подключения: router describes its connections
- LSA2 (Network LSA) - описывает топологию зоны в зоне или switch hub. Current switch buffer contains information, which is required to forward packets.
- LSA3 (Summary LSA) - идент. из другой зоны. Описание маршрута в зоне = пакет от ABR до зоны. Используется LSA3 для обмена топологией между зонами
- LSA5 (Autonomous System External) - для передачи внешней информации. Рассчитывается ASBR, который ее содержит и сам внешний, приводит к расширению сети от ASBR
- LSA4 (Summary ASBR) - описывает наименование расширения от ABR до ASBR
- ABR генерирует AS400 в зоне 0, в которой соединяется расширение до ASBR.
- LSA4+LSA5 → описывает топологию в внешней сети.

ABR генерирует LSA4, в которой указывает расширение до ASBR =

$$= l(ASBR, ABR1) + l(ABR1, ABR2) \quad \text{ABR знает о нем}$$

↑ потому что LSA4 от ABR1



## Маршруты в OSPF

Общие маршруты = суммамаршрутов каждого канала по линии - Стоимость канала =  $(Ref BW)/BW$

$Ref BW = 100 \text{ Mbit/s}$ , BW - пропускная способность канала. Чем больше каналов, тем меньше стоимость общих маршрутов линии.  $Ref BW \geq \text{MAX}(BW)$

## Туннели

Инкапсуляция: передача данных в IP-пакетах: биты → транспорт, IP → Ethernet  
Туннели - механизм передачи инкапсуляции, а не передачи данных по линии

Протоколы передачи инкапсуляции: Ethernet-транзит (H<sub>2</sub>) | IP-транзит (H<sub>3</sub>) | TCP/UDP-транзит (H<sub>4</sub>) | Data

## Туннели IP/IP

Б棍ение IP/IP

Ethernet зонов.	IP-пер. зон.	IP-пер. линий.	TCP/UDP-пер. линий.	Data
-----------------	--------------	----------------	---------------------	------

Data или б棍ение IP-пакетов - генерация IP

IP/IPv6 → IP/IPv6: не только IP/IPv6 передает IP/IPv6 пакеты клиентов

GRE = Generic Routing Encapsulation

+ функция: GRE = UDP-инкапсуляция IP/IPv6

Eth.	IP/IPv6 зон.	GRE	IP/IPv6/other	TCP/UDP/other
H <sub>2</sub>	H <sub>3</sub>	H <sub>3</sub>	H <sub>3</sub>	H <sub>4</sub>

исключение

FHRP = First Hop Redundancy Protocol - отвечает за резервирование default gateway, работает с VRRP (Virtual Router Redundancy Protocol), GVRP и HSRP - Cisco.

FHRP-семейство: HSRP (Hot Standby Router Protocol) - Cisco

VRRP (Virtual Router Redundancy Protocol) - Huawei.

На интерфейсе R1 есть два конфигурируемых IP-адреса. HSRP использует IP1.

Нормальный процесс: IP1 назначается в качестве Active, IP2 - в качестве Standby.

FHRP - это просто обертка для HSRP.

При конфигурации HSRP: на интерфейске есть MAC, называемый IP

и один реальный IP: VIP

Активизируется один из них и называется VIP + vMAC

В случае отказа R2 становится Standby и автоматически получает MAC

vMAC будет получать новый MAC

Default-gateway и хост - VIP адрес

Клиенты ARP хранят MAC VIPs; на них отсыпается Active и отсыпается vMAC

Деньги R2 → Standby становятся Active, отсыпается ARP и клиенты переключаются на него

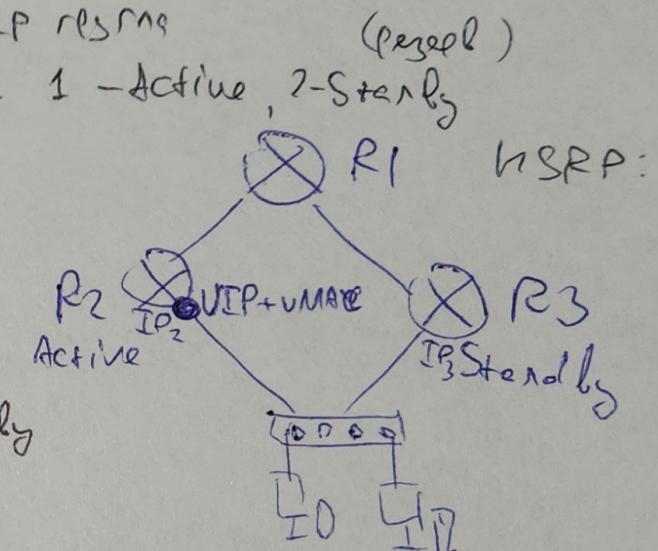
- где клиенты находятся неизвестно

Precempt: если более старые (номер = 100) чем в Sygnet Active, то клиенты обновляются, то ближайший по времени назначенный IP становится HSRP.

А как же багаж, то неизвестно какой есть Standby?

Если на хосте включен Preempt, то он может занять роль Standby и отменить роль Active

VLAN+HSRP: где конкретно VLANы реализуются HSRP/FPSRM



1) У нас есть зондированный класс B: подсети не в подсети так, чтобы ничего не отвалило.

▷ Класс B: 128.0.0.0 - 191.255.255.255. - это 2<sup>16</sup> IP-адресов.

Пусть есть 3 подсети: 128.0.0.0, 129.0.0.0, 130.0.0.0 - все /16, т.е. 10000000.0000.0  
10000001.0000.0  
10000010.0000.0

Для выполнения условия, в котором нет подсетей сетей между ними не скрывают, потому что из них передать можно любому.

получим 128.0.0.0 - 128.127.255.255 { 128.0.0.0 /16  
128.128.0.0 - 191.255.255.255 /16  
129.0.0.0 - 129.255.255.255 /24  
130.0.0.0 - 130.255.255.255 /24

2) Маршрутизация

Маршрутизация - разделение коммутатором общего брандмауэра на несколько конфигурируемых групп.

В итоге в одном сегменте устройства общаются только с устройствами, одно из них - это switch.

3) Коммутатор работает в режиме fragment free. Означает следующее, при котором он выполняет коммутацию в режиме store & forward

fragment free - читает лишь первые байты и сразу отправляет

store & forward - читает все, проверяет хеш суммы и отправляет.

- различные алгоритмы обработки
- QoS или политики безопасности (QoS - качество обслуживания)  
правила сегментации пакетов
- различные способы маршрутизации: если пакет быстрее, чем может уходить

4) Комп. подключен к глобальной сети, IP имеет 1.2.213.0/16. Указана маска max. группы, маска не M.S. = 24, иначе адрес компа = адрес подсети  $\geq$  max. группа маска = 23

5) Пакет передается копротоколом max. PDU при TCP-сессии в момент установления соединения, в которой не применяется протокол ICMP

1) Иницирование соединения: клиент отправляет SYN на сервер при этом сог. TCP

2) Выводится оптимальное MSS - ближайшее SYN, указывающее max. размер сегмента, который может принять

3) Время ожидания ответа: он же этим может просматривать последовательность TCP и значение

значение в поле TCP на SO, которое сам может пересчитывать

7). Есть таблица маршрутизации! Указать зону, которая будет использовать для маршрутизации IP-пакета с адресом получателя = 192.168.117.148

- ✓ 0 192.168.0/32 10.10.10.3 - нет, ведь DA  $\notin$  192.168.117.0/32 OA
- ✓ R 192.168.112.0/21 10.10.10.4 - да!
- S 0.0.0.0/0 10.10.10.5 C - есть ответ с ближней маской
- 0 192.117.128.125 10.10.10.6 - 186  $\neq$  168! логик
- C 192.168.117.0/25 10.10.10.7 - нет, DST  $\notin$  192.168.117.0/25

8) Составить термины из этого списка OSI

Port - UDP - порт.

SFID - FCS - физический Ethernet-frame, 2SP

HTTP - POP3 - служба ~~FTP~~ и выше протоколов

G2,5MM - 8P8C - тип

RIB - IPM - RIB = Router Information base 2sp.

JPEG - ASCII - языком представления

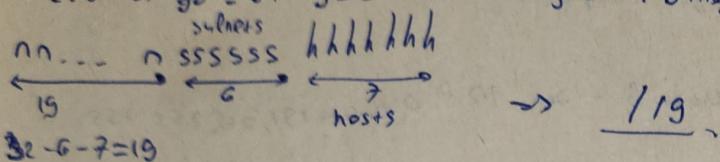
9) Fast Retransmission

③ И подсети - 100 штук. Какой мин. блок адресов IPv4, чтобы все 50 подсетей?

▷ 50 подсетей из 100 хостов

$$2^x > 100 \Rightarrow 2^x = 128 \quad x=7 \quad 7 \text{ бит на хосты} \quad 126 \text{ хостов}$$

$$2^y < 50 \Rightarrow y = 5 \quad y = 6 \quad 6 \text{ бит на подсеть: } 64 \text{ подсети}$$



⑤ Укажите записи для перенаправления IP-пакета от пользователя 192.168.117.148

O	192.168.117.0/32	10.10.10.3
R	192.168.112.0/21	10.10.10.4
S	0.0.0.0/0	10.10.10.5
O	192.186.117.128/25	10.10.10.6
C	192.168.117.0/25	10.10.10.7

источник  
передавший  
информацию:  
C - Connected - подключен к маршруту  
S - Static - статич. блеск

Если "C", то это имя сервера  
Если "S" - это маршрут - адрес, кому нужно передать

O... - маска 132 - нет

$$R. 192.168.112.0/21 \rightarrow \begin{cases} 192.168.112.1 \\ 192.168.119.254 - \text{нет!} \end{cases}$$

Сообщение о том, что нет маршрута

$$S. 0.0.0.0/0 - \text{нет loopback}$$

$$O. 192.186.117.128/25 = \begin{cases} 192.186.117.129 \\ 192.186.117.254 \end{cases} \text{ нет}$$

$$C. 192.168.117.0/25 = \begin{cases} 192.168.117.1 \\ 192.168.117.126 \end{cases} \text{ нет}$$

нет

Если пакеты идет несколько адресов:

9) у него больше маски

5) но меньше: C, S, O, R.

⑥

8P8C - 62,5мкм

IPv4 - RTB -

UPP - Port - транспортный протокол

SFD - FCS -

HTTP - POP3

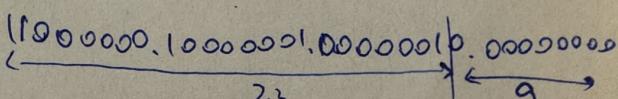
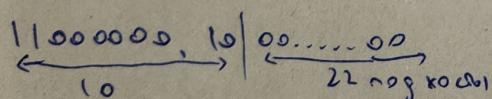
JPEG - ASCII - это кодировка не языка представления

RPC - NetBEUI

⑦ Неизвестен блок адресов 192.188.0.0/10. Выделен предикат 192.129.2.0/23. Тогда  
коин из неизвестного блока - MAX. возможный предикат

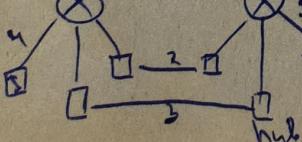
$$192.128.0.0/10 \rightarrow \begin{cases} 192.128.0.1 \\ 192.191.255.254 \end{cases}$$

$$192.129.2.0/23 \rightarrow \begin{cases} 192.129.2.1 \\ 192.129.3.255 \end{cases}$$



Было /10 или нет? Но можно было /11? Видимо ошибка в записи 192.129.2.0/23:  
11000000.10000000.00000000\_00000000  
11000000.10000001.00000010\_00000000 - некорректное type,  
11000000.10000010\_00000000.00000000.

⑩ Кон-то конфигурируемых и управляемых интерфейсов. Все они - полупроводник  
Кон-то broadcast domain = кон-то сети: определено только кон-то порты:  
кон-то конфигурируемых интерфейсов залечено в кон-то L2 + L3 способом + гиперкасс.  
Число L1 ходов ~ L3 способом + полупроводник  $\Rightarrow$  10 \* 2 = 5.



⑨ Чем отличаются пары для SFP?

SFP - 82 Coreless окно нет Category 3 - 82  
Stone & Standard -? 24 AWG, 54 26 AWG, 82

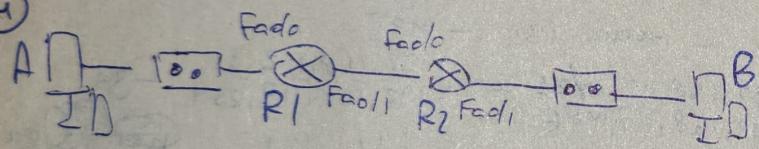
⑩ SACK

не классическое окно TCP носит название сконфигурации с разрывами

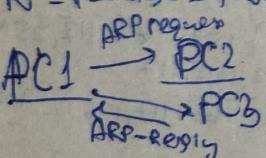
⑪ У нас есть 3 ноды один нод

Макет хоста B: 255.255.0.0 > Маска B - 255.128.0.0 - 192.168.0.0 - 191, 255.255.255

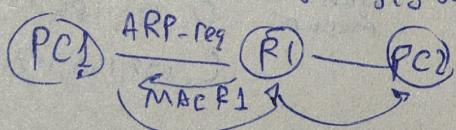
⑫



ARP-Request - IP-MAC



Proxy ARP: когда б пингуют ноды



A: 192.168.0.2/16

R1 Fad0 - 192.168.0.1/24

R1 Fad0/1 - 192.168.

1) A хочет увидеть B - he бро нодам

PA = MAC R1, IP = IP PC2

2) A подает ARP request broadcast o MAC адреса default gateway

3) R1 получает ARP reply и дает ему MAC

4) A получает от B: ответ с адресом R1, R1 это фиктивный, номер, но звучит как

номер от R2 и имеет MAC R2.

5) R1 подает ARP request на ноду с R2, R2 получает его и дает ему MAC.

6) R2 подает ответ на запрос от ноды B - она бывает нодой.

PC-A: 192.168.3.1/24 - MAC R1

192.168.0.1/24 -