

Модель OSI
Прикладной представления
Сетевые
транспортные
сетевые
канальные
физические

Open system interconnection

Session: создает и поддерживает сессии

Представление: концептуальное

Сетевые:

HTTP - Hyper Text Transfer Protocol

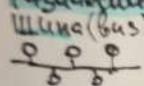
DNS - Domain Name System

FTP - File Transfer Protocol

Физическое
уровень
Encapsulation
deco PDU (header)

Физический уровень

Физические топологии:



шина (bus)

звезда (star)

满 mesh

partial mesh

кольцо (ring)

Ethernet

как реализован?

коаксиальный кабель

жесткий кабель

плюс

витая пара (Twisted Pair)

витая пара - hub: многопортовый повторитель

не имеет, а звезда: логическая звезда

логическая звезда

логика

</

CSMA/CD - Carrier Sense Multiple Access / Collision Detection

Ethernet has random collisions!

протяжённость канала может быть 100 м.

max frame = 1500 байт (если не захватывал канал)

min frame = 64 байт (если не было конфликта на побочных расстояниях. Если < 64 байт - padding (рекомендуется))

Модернизированный Ethernet

MAC: запоминает: интервал \rightarrow MAC \rightarrow время t - таблица коллизий (запоминание - learning)

No new frame \rightarrow принятые решения о конфликте/изменении: разрешает и отбрасывает чей-то трафик.

Replies after Tidle 50, 480 мкнс: в трафика. Если I+ collision domain - коллизия не проходит через bridge

Коммутатор (switch) - многопортовый мост: интервальный не 1 мкнс (2, 4, 8, 16, 24 или 48, \rightarrow 1 раз меряется) + полная сумма \rightarrow Кратчайший

Мультиплексирование - разделение коммутатором одного домена на несколько. Полный доступ: адрес на switch: он вынужден слушать записи из других портов

Мультиплексирование + полная сумма \rightarrow Кратчайший + полная сумма \rightarrow Кратчайший

Коллизионный домен - есть сам, в котором возможно встретиться коллизии, и есть домен, в котором возможна рабочая ситуация. domain (ограниченный радиусом)

Иерархическое - есть сам, в котором возможна рабочая ситуация, и есть домен, в котором возможна рабочая ситуация. Проблемы (ограниченные радиусом и принципами)

Работа switch: Store and Forward: сохраняет и пересыпает: получает CRC, проверяет FCS и принимает решение о коммутиации. Предполагается, что есть непрерывная: высокая вероятность возникновения ошибки.

Cut through: предполагается, что есть непрерывная: то switch принимает только min. полную frame, проверяет CRC и сразу начнет передавать (если там обнаружена ошибка, то контролирует). Но если ошибки \rightarrow > склонны \rightarrow то передает всю frame (если ошибки не обнаружены или нет ошибки)

Fragment free: гипотеза: принимает первые 64 байт frame и если они нет ошибки, коммутирует и дальше от исключения коммутирует, и передает с min. задержкой

Обработка frame на коммутаторе

Сетевой уровень

Адресация

IP - дополнительный логический уровень: если мы сидим не в одном сегменте LAN,

Бывает два IP адресов в один маршрутизатор. Задача одна группу IP address.

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

8	1	24
16	1	16
24	1	8

Устройство содержит сетевые ящики \rightarrow устройство приобретает один ящик \rightarrow много ящиков для группировки -

Следует использовать CIDR - без ящиков, только коммутатором / на всем

Потом убирают что это называется: классом добавляют хостовую

Сетевая маска: 32 бита: определяет количество IP address на сетевом ящике ящики

Но если занять IP address в группе не 2 ящика - не по принципу классового ящика.

T.e. Требуется новый ящик который есть IP address, или определить группу ящиков, т.к. они имеют разное количество ящиков.

IP address = 10.1.2.3 \rightarrow mask = 255.255.252.248

mask = 255.255.252.248

Использование не по принципу ящиков! По классовому ящику это невозможно.

VLSM - Variable length Subnet masks. Рассматриваем сетевые ящики в зависимости от хостов.

2 разр. ящиков: ящик есть (subnet): все ящики $= 0$

бесполезно, ящик广播 (broadcast): все ящики $= 1$

Хостовую ящики = n бит, n = 32 - [32 битов ящиков] \rightarrow 1 ящик \rightarrow 2^n - 2 IP address

2 ящиков: ящики ящики = \rightarrow 2 ящика через оборудование 1-2 уровня (hub, switch, bridge..)

$\neq \rightarrow$ 2 ящиков ящики, где 1 ящик - через router

Max ящиков ящики в ящике ящики \rightarrow IP

$2^{n-2}, 5 \rightarrow n=3 - 1024. n \rightarrow ящики = 2^3 = 8 \rightarrow mask = 255.255.255.248$

CIDR означает: Classless Inter-Domain Routing: зону ящики через ящики:

1/29

Зарубежный IPv4

Version | IHL | DSCP | ECN | Total length

Identification | Flags | Fragment offset

TTL | Protocol | Header checksum

Source IP address

Destination IP address

Options (if IHL > 5)

Flags и Fragment offset - межсетевые агрегаты: по фрагментацию пакетов

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

Version - IPv4/IPv6

IHL = Internet Header length

- ящики ящики

DSCP = Different Services Code Point

- ящики ящики ящики

ECN = Explicit Congestion Notification

- ящики ящики ящики ящики

ID = Identifier - ящики ящики ящики

Фрагментация - деление исходного IP пакета на 2 или более одинаковых сегментов. При этом один из них имеет ID меньший, чем другой. Каждый из которых имеет свой номер, это его фрагмент.

3 слои для IP header & header IP packets: version | DF | MF

PPS - packets per second

DF = don't fragment = 1 => не разбивать пакет. Если не разбить - ошибка в IP header. Число нулей

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

Каждый фрагмент - подмножество IP-пакета с заголовком и данными.

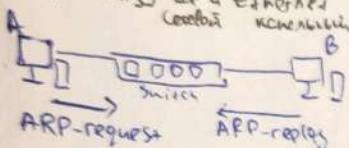
Fragments offset - как отмечать следующий фрагмент исходного

Следующий пакет не фрагмент: 1й - у него Fragment offset = 0, последний MF = 0. Если конец - то PSH = 1

- Сборка пакета производится из фрагментов (одинаковые, чтобы проще было соединять разные)
- если фрагменты - одинаковые, то Fragment offset = сумма предыдущих фрагментов
- Заголовок из transport layer будет один в каждом фрагменте
- Перемножение - не обязательно. Сначала по MAX MAC, потом проверяется остаток
- Источник может выбирать способ пакетов (SLA, например), но всегда все равно отвечает пакетом (исключая собственные пакеты локально).

ARP (Address Resolution Protocol)

(Ethernet MAC) IP & Ethernet: подразумевается (это не указано в протоколе) что использует IPv4, т.к. MAC



Перед отправкой IP пакета, не зная MAC адреса, оно отправляет ARP-request и получает ответ от MAC:

Узел A знает IP узла B (DNS). A отправляет всем запрос: У кого ТОЛКА IP? Сразу же знает MAC! И только узел B ему отвечает своим MAC, не зная такого узла A знает MAC узла B и может создавать уже IP пакет, а спокойно!

ARP-сообщение:

01...785...	15107...	31...	31
HTYPE	PTYPE		
HLEN	OPER		
SMA = sender hardware address			
SPA = sender protocol address			
TMA = target hardware address			
TPA = target protocol address			

ARP - протокол уровня 2.5^v

HTYPE = Hardware Type - какой технологии кадр можно использовать

PTYPE = Protocol Type - какой протокол используется

(T-o. Использует IPv4 <→> Ethernet)

HLEN - поле адреса протокола в байтах (напр. MAC для Ethernet)

PLEN - поле адреса протокола в байтах (напр. IP для IPv4)

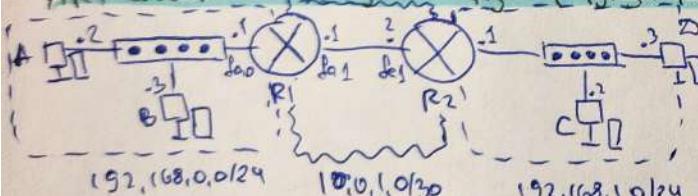
OPER - определенные операции: request/replay ...

SMA - определенный адрес отправителя, TMA - получатель: канал.бр.

SPA - короткий адрес отправителя, TPA - получатель: канал.бр.

TPA не заряжено и используется для MAC

ARP сообщение не распространяется дальше границы сегмента; here no such MAC available не в текущем сегменте



IP R1:

C 192,168,0,0/24 fa0

C 192,168,0,0/24 fa1

S 192,168,0,0/24 10,0,1,2

SMA A:

C 192,168,0,0/24 -

S 192,168,0,0/24 10,0,1,2

default gateway

- 1) А хочет связаться с B. знает IPаг. B в ходе работы, с кем либо в сети B => знает ARP-адрес MAC узла B
- 2) А хочет связаться с C: получает, что C - не в сети, поэтому ARP запрос отправляется бесполезно

Поскольку А не знает как добраться до C => спрашивает на R1. Для этого использует ARP-request от MAC-адреса

Своего узла по умолчанию => R1 ему отвечает и А в свою очередь ARP-cache записывает IP-адрес R1. R1 отвечает MAC-адресом R1 - MACR1

ARP cache - таблица IP-адресов - MAC-адресов - IP-адресов "на хранении"

Теперь когда А определил IP-адрес C и D он определяет спрашивает MAC R1. R1 отвечает MAC & FCS

установленный IP-пакет - источник, который не знает и отвечает, что это не C => получает IP-адрес R2

R2 отвечает MAC & ARP-reply. И R2 в свою очередь записывает ARP-request с ID = 10,0,1,2

MAC узла C

ARP сообщение Padding bytes до минимальной Ethernet-frame = 64 байта.

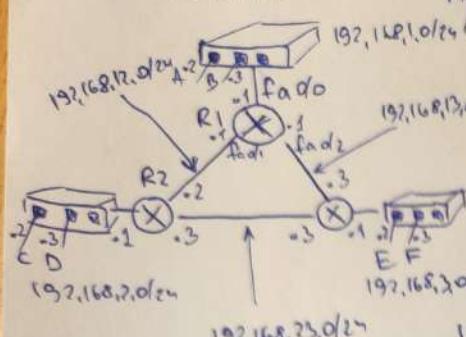
После высыпания ARP reply получите кратчайший ARP cache keyless MAC: это окончание узла

последних 4х байт, которые включают в себя MAC (или же 3 байта MAC). Если не получите,

Protocol - процесс, который передает информацию в локал TCP/UDP/ICMP.

Основы маршрутизации. Рассмотрим пакет

3 routers, 3 Switch



Ключевой вопрос: каким же образом адрес в сети 192.168.1.0/24 может попасть в сеть 192.168.1.1?

Маршрутный пакет имеет путь: $A \rightarrow R1 \rightarrow R2 \rightarrow R3 \rightarrow C$.
 Адрес $R1$: 192.168.1.1. Адрес узла C неизвестен, т.к. маска $255.255.255.0$ не указана.
 Упомянутый один пакет имеет в пути 85450x, поэтому адрес 192.168.1.1 может быть 1 для $R1$.

Как понять, какой адрес принадлежит адресу узла? Если адрес узла - адрес сети + маска сети (\oplus), то упомянутый адрес сети и подсети маску сети: если сетевые маски \oplus , то упомянутый адрес - это подсеть. Пример: узел A : 192.168.1.2 - реально в сети 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	11111111

РУ =) неизвестна маска сети. Пример: узлы A : 192.168.1.2/24 и C 192.168.2.2/24 - неизвестна маска сети! А если маска сети 122, то РУЧНОЙ адрес сети.

$A \rightarrow R1 \rightarrow R2 \rightarrow C$.

Узел A имеет два адреса и два маски, и адрес узла C ; получается что маска на этом адресе адрес C :

Соответствующий сетевой адрес C - это $192.168.1.2/24$.

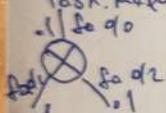
-Задача о сети, куда неизвестен

-Маршрут по умолчанию - куда передавать, если получатель не в нашем сетях.

IP адрес соединения $R1$ канал.

Путь оптимальный маршрут - когда пакет проходит через мин кол-во переходов.

Текущий адрес $R1$:



C 192.168.1.0/24 fa0/0
C 192.168.1.1/24 fa0/1
C 192.168.1.1/24 fa0/1

S 192.168.2.0/24 192.168.1.2
S 192.168.3.0/24 192.168.1.3

Получил в базу, обновление
после включения, но строки IP
изменение масок

Была вручную администрирована: Задан маршрут по сетям
34 R2 и R3. В этих сетях не используется, адрес узла неизвестен,
но при этом используется, чтобы убедиться, что они не будут засчитаны

S-static

C-connected - записи сетей, в которых есть путь для этого узла

Быстро новые записи получены (отработали не требуется), R1 их не знает!

Написанное значение адреса C - 192.168.2.2 \Rightarrow $R1$ понимает, что маршрутного узла $R2$

Сделан маршрут по $R2$ - это $R2$ строка: по умолчанию $fa0/1$. $R2$ не имеет грубого слушания, что $192.168.1.2$ не его интерфейс: он этого содержит слишком много адресов C .

Если в базе 2 строки - переход, то выбираем с MAX. длинной маски \vee S 192.168.1.0/24 192.168.1.2
 \times S 192.168.0.0/16 192.168.1.3

Итог: самое большое время, в которых есть более короткий маршрут с тем же адресом в DA, а затем
самое большое время с самой длинной маской.

AD - administrative distance - величина, которая определяет приоритетность маршрута (чем меньше, тем выше).

C-connected - то, что можно напрямую: $AD=0$

Больше времени с меньшим AD

S-static - вручную администрирован: $AD=1$

OSPF - маршрутизация по ширине полосы: $AD=110$ OPEN Shortest Path First

RIP - протокол - бессвязный протокол шириной полосы: $AD=120$ Routing Information Protocol

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

Админ оконце: на $R1$ находит static на $R2$, на $R2 \rightarrow R3$, на $R3 \rightarrow R1$. Но AD у $S < AD$ у $C \Rightarrow$ все будет OK
но! Если $R2$ знает неизвестную:

$R1$ будет знать узлы A и C : $R1 \rightarrow R2 \rightarrow R3 \rightarrow R1$ - первая

route: при котором проходит TTL = 1. Изначально ходят 1 раза в базе - неизвестные - неизвестны Header Checksum

пакеты в сети. ICMP = Internet Control Message Protocol

MTU = Maximum Transmission Unit - MAX. размер данных который может быть отдан без ошибок на сеть без ошибок

SP / DP - первые отправленные байты: $2^{32} \text{ байт} \Rightarrow 65536 \text{ байт} = 2^{16}$
Sequence number - номер 1го байта в segment, Acknowledgment number - 3о какого момента sequence number
DATA offset - смещение данных; где начинавшиеся некомпактностью в segment данные
 Т.к. TCP зеркально makes и uses sequence numbers (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	Y	I
R	F	G	K	H	T	M	N

URG = Urgent: Большое значение

Если $= 1$, то это означает Urgent point - привычные для TCP
 большие значения

ACK = Acknowledgment

означает Acknowledgement number, либо, скажем

значение sequence number от него неизвестно

PSH = Push: Приемные данные не хранятся в буфере, а сразу передаются приложению

RST, FIN - закрытие TCP соединения (RST - наше имя идентифицирует засланный RST)

SYN - последовательное Sequence Number и последовательных сторон: последовательное соглашение.
 SYN - Synchronization

Three-way handshake: установление соединения в TCP

Classic.

TCP-сессия

TCP - connection-oriented (без UDP) : передача данных лишь после установления соединения.

A

B

1) A \rightarrow B: SYN = 1: пакет B получает sequence number: "нечисло
 последовательное начальное с байтом NDX"

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

последовательное Seq.Numb.: и тут же устанавливается значение с байтом Y

TCP: \hookrightarrow разные Seq.N. с двух сторон.

Ack = 1: последовательное значение Acknowledgment number у B: B подтверждает успешное получение.
 Ack.numb = номер байта в посланном, который B отвечает положено ответом \Rightarrow
 \Rightarrow то есть все получено без ошибок

Seq, ack - номера пакетов

seq = x \Rightarrow ack = x + 1 - при установление соединения

если есть значение sequence number, то $ack = ((x-1)+1) + 1 = x+2$

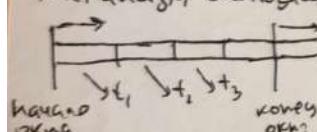
T.е. B получила x+2-1 байт, начиная с байта x+2 последовательные данные.

3) A \rightarrow B: Ack = y + 1: A: я также Seq.Numb. и XDS данные

Передача данных Sliding window - Скользящее окно:

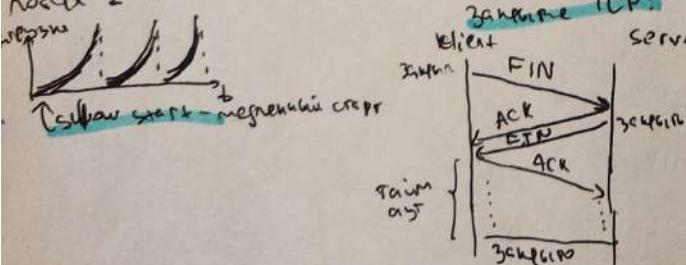
Определение нового логического байта: пакет не принят - можно использовать!

Механизм скользящего окна - определение недоступных сегментов некой по получению и передаче информации о получении



Определение недоступаемых сегментов и сегментов, которые после этого момента не могут не подтверждаться.
 Подтверждение о приеме каждого пакета должно определять что сегмент, когда засчитан таймл t_1 . Это означает, что окно не 1, иначе все подтверждение из него бы было. Если таймл не принят, то независимо от того, где сейчас окно, то сегмент определен заново \Rightarrow Transmission - гарантия доставки первого

T.e. гарантировано наличие сегментов и приемлемые контролируемые расстояния между определенными. На принимающей стороне - буфер: если нет места в буфере, т.е. подтверждение про них не получено, то это ошибка. Принимающая сторона может не засчитать обработка приема \Rightarrow приема не реализует скользящего окна на window size. window-size - принимаемый отработки, то есть отработки буфера, не меньше. Если нет места, то подтверждение \Rightarrow window size. Книга только упоминает $new_seq + window_size \leq go_l$.



Оптимизация TCP-соединения

1) Задержанное подтверждение (Delayed Acknowledgment)

Прием подтверждение о получении отображения не не Каждый сегмент, а пакет: когда таких пакетов бывает занесено нечетное

2) Алгоритм Нагле (Nagle)

Если от высших протоколов приходит данные слишком получением, то есть TCP header + Ethernet header + IP header занесение байт не может, чем полупустое. Поэтому пакеты данных не должны включать их байтами получением.

3) QoS - Quality of Service

QoS - 2 опции: CWR - Congestion Window Reduced

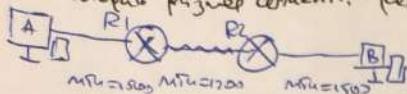
ECN - Echo - Explicit Congestion Notification

Целью является промедление работы о перегрузке: если байт не может быть отправлен данным

Идея: изменение window size изменение window size изменение window size

4) MSS (Maximum Segment Size), Options в заголовке TCP segments

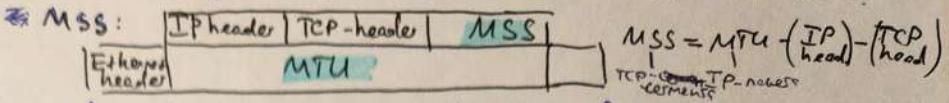
Как узнать размер сегмента? (секунд IP header + Ethemet option) = MTU



RI неизвестен отправителю: для настройки \Rightarrow 1 способа передачи. 8 pps
1: один DF=1, RLOsent от A ICMP что MSZ = 1200

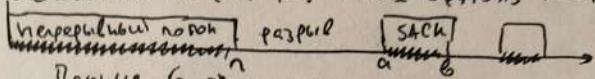
- MSS - максимальный размер полезного бояка
запрос & байтак

2: API 3м проверяет, что в окне от A и B нет Option сегментов
После этого можно определить MSZ в окне, это и есть предел TCP-параметров: он тоже ставится
Размер MSZ от B для MSZ = MTU - IP header - MSS (равно 1)
Размер MSZ от A для MSZ = MTU - IP header - MSS (равно 1)



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

Внешний TCP может реагировать на потерянные пакеты, которые присланы позже.
SACK: подтверждение пакетов, пропущенных с разрывами: какие ходят по сети. Т.е. вnone Options работы, MSZ получает (0,0) (ACK) и (q,p) (SACK) один раз при разрыве



NACK (TCP не видит пакета) - указывает что, какие пакеты не получены.
Внешний TCP определяет пакет (n, a) U[q, p]. С SACK/NACK - меньше перегораживаний.

6) Scaling Factor Options

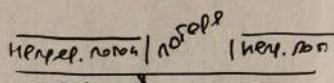
One window/gamma window: просто хотят поддерживать один приемника

Scaling factor = n (8 TCP options): Window size * = 2^n. Window size = окно: больше gamma открытое для накопления. Все накопление TCP-options должны быть коррелированы в момент установления соединения.

7) Fast Retransmission (Быстрая повторная отправка) Options!

не используют TCP Options, лишь механизм classic TCP. Текущийтаймер не тикает: ожидается прием пакета, сразу отправляется ЗАК и то же пакет всем присыпать.

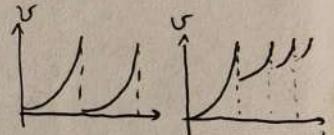
Отправитель будет и не быстрымтаймером сразу перегораживать: подождет еще короткое время



3 раза накоплено очень быстро отправлены пустые TCP-segments с ACK=1

8) Не удаётся при потерях реагировать go min.

При потере не сокращают go min, а 1 в 2 раза: Потом в среднем сколько bytes >. Окно не сокращают, а удаляются все MAX bytes склонности.



9) SYN cookie

Проблема: механизм защиты сервера от атак с фальшивой IP-адресацией!

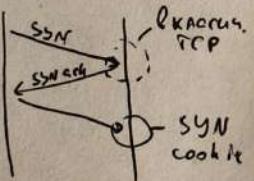
Заваливают сервер запросами с SYN, ему нужно выделить память, отвечая ACK+SYN... - SYN Flood

Сервер может заметить атаку: когда SYN, но отсутствует ACK, но приходит управляемый.

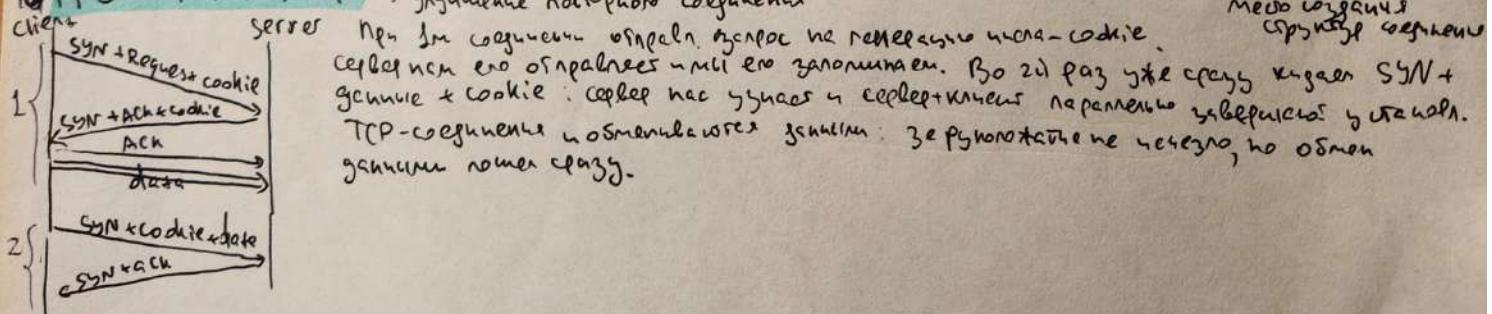
SYN cookie: сервер получает SYN, но не получает никаких управлений. Потом отвечает SYN+ACK и удаляет его из памяти. И лишь если клиент отвечает более ACK = Seq + 1, сервер

сразу с тем Seq. идет отвечать и снова отвечает более длиннее управление.

+ блокировка " ". не могут использовать битовые options - MSS, Scaling Factor, SACK/NACK - из-за сообщения клиенту.



10) TFO - TCP Fast Open: Упрощение подпрограммы соединения



DHCP (Dynamic Host Configuration Protocol)

Автоматическое назначение IP-адресов на компьютере без ручного ввода.

Четыре стадии - DORA:

Discover: клиент пишет в лог DHCP-сервера - определяет широковещательный адрес 255.255.255.255, для IP и его IP - основное поле с адресом 0.0.0.0

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

Request: клиент принимает данные и подтверждает, подтверждение от которого сервера он принимает (также 0.0.0.0 на 255.255.255.255)

Acknowledgement: если сервер не подтвердил, он выдаст клиенту подтверждение + IP-адрес, маску, default gateway

Когда клиент получит 1/2 т аренду он определяет request, что кому подтверждение. Сервер может это одобрить, либо отказать. Если он откажет, то нет т request & 2/2 т. н.г.

Release: Клиент отбирает IP-адрес : Сервер возвращает IP-адрес в виде свободных адресов
Request: Клиент берет адрес из списка свободных и пытается засечь его в списке
Decline: Сервер блокирует клиенту на самом деле занятым IP-адресом : отказ сервера
Inform: говорит о получении вспомогательного IP-адреса.
DHCP сервер находит не занятые
TCP:

Երցանքայթ =
= ՏՆԿ բւլ ԻՐԱ

IPAM - IP address management - Система хранения в базе данных IP-адресов

DHCP Relay. Появя „представителя“ DHCP сервером: този сервер генерира все пакети Discover на DHCP клиенти. Този DHCP сервер разделя IP на много клиентове.

Беc онын DHCP барелдес һаадынан иштөөнүү/чалас. Option 82 берелгээ "чыгармалыгын" номусуруул
Джойнеллан IP к MAC, мөнкөн онын IP-сервер иштөөнүү энгизилүү?

RIP - Routing Information Protocol - ~~Написано вручную некто из мафиози под псевдонимом~~

Blaze nonconformism - Опред. раскрытии всех правилов на изменении сети: Задача это большая.
RIP - DVA - раскрытие ненормального. Бессоритивный алгоритм: правило, наоборот заменяет RIP отработавшего соседем

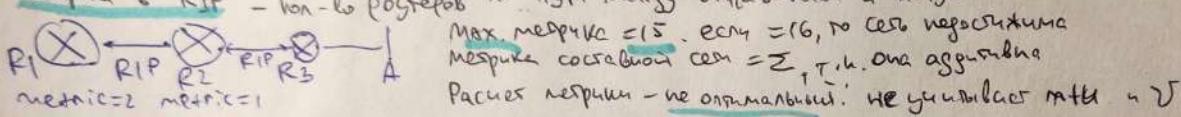
Марка - марка Марка - марка предикс

В link-state протоколах фреймы из ВСС не попадают в базу, и мы не знаем о своих маршрутах. А в B-DRA протоколах фрейм знает только информацию о соседях и наилучшую путь от них до конечного получателя.

RIPv1: Кратчайшие маршруты → есть неизвестен. Адрес подсети = broadcast = 255.255.255.255.

RIPV2 : Broadcast и multicast.

RTT: IPv4. Hyper transport = 64 bits. RTT = 100 ns.
RTTns: IPv6. Agree normen = multicast (\oplus Broadcast)
Message: IP
Response: Broadcast no open network using buffers - normen = von do L3 next hop



Timers RIPv2

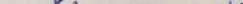
- update timer (30s) - задержка обновления соседей с соседями: 30сек. RIP рассчитывает время обновления для каждого маршрута
- invalid timer (180s) - если сосед не отвечает за 180с, то маршрут помечается как invalid (неправильный). СПЧ (Selective Periodic Update) срабатывает update каждые 180с для всех маршрутов. При обнаружении invalid маршрут меняется на valid и удаляется.
- hold down timer (180s) - задержка перед тем как маркировать как invalid старого

- flush timer (240s) - regular route update. По истечении чир. о предикате устанавливается invalid (180s) \rightarrow hold down (60s). Для каждого префикса можно вставить RIP (unreachable static) и это со временем не падает.

```

graph LR
    valid[valid 180s] --> invalid[invalid 180s]
    invalid --> holdDown[hold down 60s]
    holdDown --> unreachable[unreachable static]
  
```

Проблема непрерывной сходимости R_{IP}
Сходимость - Рекурсия не непрерывна в смысле R_{IP}.

$R_1 = R_2$:  R₁ and R₂ update to R₂: (let A good time happens then R₁ becomes a good time because been voltage across makes R₁ a short = 0. R₂ has voltage, no voltage in the resistor, both are now A directly connected.

Мод. $n_{\text{р}} = \frac{n}{n_{\text{р}}}$ при $n = R_1 + R_2$. R_1 и R_2 - это характеристики А. фазы. $n_{\text{р}} = 2$, R_1 больше R_2 , R_1 - это характеристика нелинейности R_2 . R_1 определяет коэффициент А. фазы. $n_{\text{р}} = 3$, $R_2 = 1$, $n_{\text{р}} = 4$ и т.д. Качество нелинейности $= 16$, $R_1 = P$ константа. Все параметры известны, поэтому А. фазоизменения: Погрешность $\rightarrow \infty$.

Оптимизация RTP

1) Split horizon. Терм рекурсивне отрицання об'єктів, чиєю використанням можна зменшити кількість вимірювань.

Р1 координати R2, яко A розташування $\text{resp.} = 1$, а R3 координати R2, яко A розташування $\text{resp.} = 5$ split horizon; R2 не отримує об'єктів чиєю використанням, чиєю використанням є їхнє позначення в самому себе. R2 використовує об'єкт R3 з позначенням $\text{resp.} = 1$ (мін. метрика). R2 запирається в цю об'єкт. Якщо A $\text{resp.} = 1$ та його next hop = позначення R1, то R2 в цей раз R3 отримує об'єкт з позначенням A з метрикою 2, а в цей раз R1 об'єкт R3 не отримує.

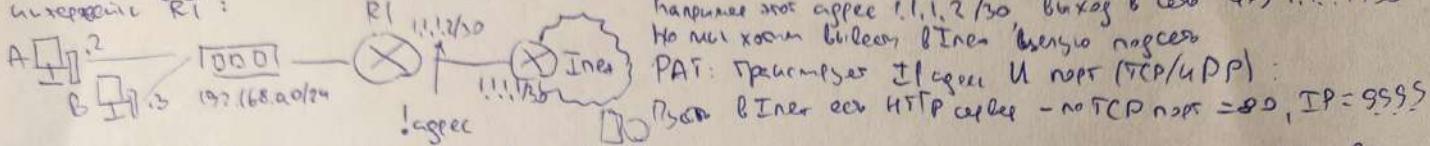
2) Poisson reverse: уменьшение split horizon: в свойствах маршрутизации, от которого мы можем изменять маршрут
 В обновлении оглавления маршрута = 16 то того препарата: он не работает.
 В самом очу. R2 отработал, то R1 восстановление что маршрут то A = 16

Потому что все рабочие узлы из числа из 124 состоят адресов из 126, это означает, что они не могут работать с IP-адресами из диапазона 126-139, т.к. они уже заняты. Решение - это заменить эти IP-адреса на другие, например 192.168.0.1-192.168.0.125. Тогда все узлы из 124 смогут работать с новыми IP-адресами.

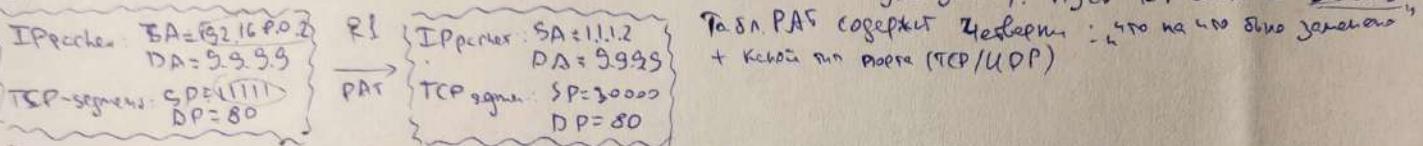
PAT (Port Address Translation) - процессорный способ с использованием портов трансляции, с помощью которых на порт назначения попадают данные из каждого из портов, соответствующих процессорам на устройствах.

Динамический РАИ

Схема 2) неизменного припоя. Показано бывшее только 1 археи змея бактерии в селе, что насторожило нас. Использование R1: R1 1.1.1/30 бактерии змеи археи 1.1.1.2/30 бактерии в селе змеи 1.1.1/30



Скан A ножничек & звонок селебы орнел. наим. 9999 SA IPad = 192.168.0.2, DA = 5.9.9.9. Порт конфигурации = 80, сканеров (Random 1000) = 11111. SP = 11111, DP = 80. Телефоны R1 DA IP неизвестен, SAIP = 11.1.1.2, SP = 805*0.2, DP = 805*0.1. Сканер (сам) наим. 11111 & PAI = 8050. Установка). Порт DP скан = 30000.



Inside Local - agree XOSA & LAN, nong router E LAN Outside Local - agree XOSA & GAN nong router E LAN
Inside Global - agree XOSA & LAN, nong router EGAN Outside Global - agree router E GAN, nong router E GAN
Each local zone has sole router basis & interface, no IP communication w/ the 1.1.1.2, a NOPT - hopcount 3000
Best route chosen, up to 3 non backbone routers before reaching 1.1.1.2 & payload dropped

В результате создан замок для РАГ-безумных преступников.

Пример 1. Организован сервис на устройстве A (называем HTTP-сервером). Текущий хостименный порт открыт, а присвоен IP-адрес 192.168.0.2:80 – 192.168.0.2:80 TCP
 Пример 2. На устройстве A HTTP server – 2 способа: 1) обогащенный клиент из LAN на 80 порту, 2) – клиент из Интернета на 8080 порту: один организует рабочий контент. Но порт 80 заблокирован! На устройстве B IP-адрес 192.168.0.2:8080 – 192.168.0.2:8080 TCP. Настройте правило пакетного фильтра на порт 8080, все гиперссылки что ложатся на порт 80.

Рекомендации НАТ

- Symmetric, Lambin parapsp. MAX. OPERATOR
нелинейное касание Outside Global. В реальности
9.9.9.9 в TCP-порти SP = 80, DP < 30000,
+: балансированный, нейтральный
бесшовное присоединение: $\text{cage} \rightarrow \text{Inside Local} \rightarrow \text{Inside Global}$
RJ определяет номера \leftarrow номера своих с cage
без отрывов - отрывов.
-: некоторые проблемы решаются не хотят

- Full Cone (nonelliptic waves) notice nonplanar wave symmetry

Мобилен рабочий TED Морган, АРМА, SPUSA Знаменитые люди.

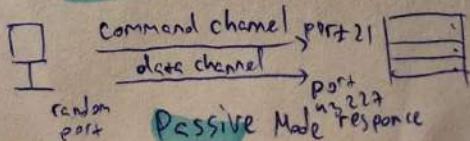
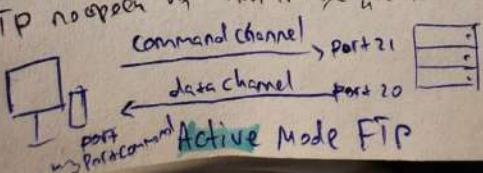
Non-reproductive regions: Restricted (ограниченный) zone

Address Restricted Cone : nolesen DP, DA U SA , SP-nosoi
Restricted Cone : nolesen DP, DA U SP , SA-nosoi

Установление PAT может сопровождаться ошибкой `ALG` - Application Layer Gateway.

FTP, NAT

FTP - File Transfer Protocol - наземная служба на сети. Устанавливает соединение с удаленным сервером и передает данные в виде файлов. В FTP-клиенте есть control-session - для передачи команд, например, по которому передается служба.



- Valid lifetime
Некий врем. аггрег. - в котором Preferred Lifetime: Все выше всего. - на самом деле, а не выше предела.
- invalid - не входит в пределы времени жизни / не соответствует никаким соглашениям
- Preferred/Valid Lifetime ~ 4 часа... 3000

NDP (Neighbour Discovery Protocol)

Использует ICMPv6 для получения информации о соседях. NDP: блокирует MAC на выделенном IP
Блок посыпал - RS и RA, получение: NS (Neighbour Solicitation (ARP-запрос))
В отсутствии IPv6 не может запускаться ICMP, запрос - запрос ICMPv6 имеет значение

STP (Spanning Tree Protocol) - (бессущество гигиеническое)

Проблема конфликтов на 2м уровне

3 способа в архитектуре: STP (L3 марш.), мультиплекс (L2 LAN), выделенное подразделение (L2 LAN)

Важно - это не fullmesh, это partial mesh, STP решает проблему дубликатов

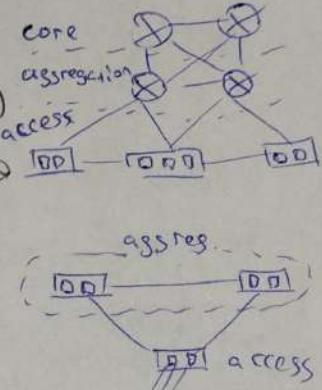
На 2м уровне конфликтов нет, но есть конфликт на 3м уровне, поэтому значение ∞

Без IP-адресов есть еще TTL, а не Ethernet - MAC

Компьютеры 2 switch на 2м уровне aggregation и один из них называется switch 2м уровня

Ethernet в таком понимании работает не корректно, есть логика физ. механизмов,

распределенные нормы \Rightarrow STP позволяет избежать конфликтов



VLAN (Virtual Local Area Network) - (виртуальное локальное сеть в группе хостов)

Создание нескольких сетей, в которых одни и те же IP-адреса используются в ограниченном范围内 broadcast - группы.

Компоненты: Ключевые компоненты - это группы в которых одна и та же коммуникация и ограничена broadcast - группами

Но не всегда все эти группы являются группами. ВМесто этого они могут быть и отдельными коммуникациями.

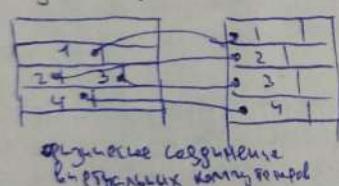
Если есть две группы, получим 2 группы L2-сетей. Чем больше групп, тем больше разницы

(физическая коммуникация не много отличается)

+ 1) более тщательное представление сетей (меньше ограничений по адресам)

2) при необходимости, пересекающиеся -хотя все уединено

Также хотим создать 2 группы switch.



исходящие 1 канал
2е не передачи групп
разных VLAN-ов

trunk - соединение, на котором можно передавать данные нескольких VLAN-ов

Каналы, открытые в trunk, наз. маркированы.
IEEE 802.1Q - маркировка, где каналы имеют идентификатор VLAN

802.1Q маркировка:
QoS | Идентификатор = CFI | VID | Type | $\sum 4$ байт

CFI = 0 - Ethernet

CFI = 1 - IEEE 802.3 Token Ring

VID $\geq 128 \Rightarrow 4096$ VLAN-ов

2 VLAN-ов на один физический порт
(2 физических switch, 1 физический trunk)

QoS - Quality of Service

CFI - Common Frame Identifier - какой кадр является важным

VID - VLAN Identifier - какой VLAN - маркировка определена

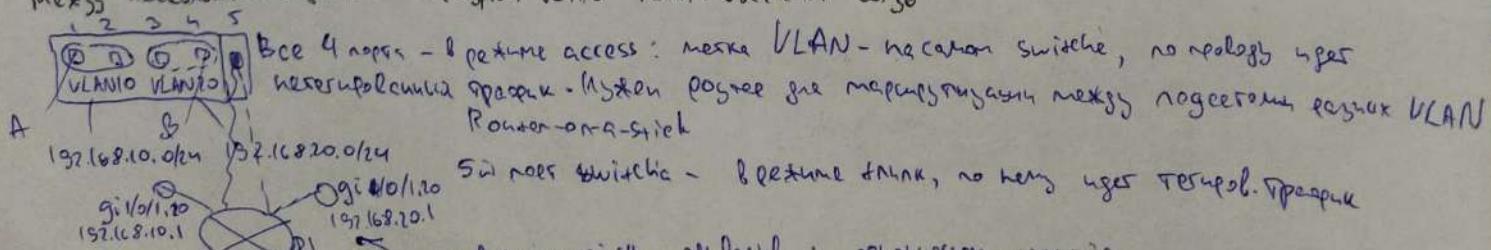
Type - тип данных (какой протокол)

Port-based VLAN: различия в коммуникации определяются портом, но не между интерфейсами коммуникации (если один switch, то коммуникации между интерфейсами)

tag-based VLAN: --- определяется методом маркировки (2 физических switch, 1 физический trunk)

Маркированные метки VLAN (Router-on-a-stick)

Маркированные данные в tag-based VLAN не должны быть



Все 4 порта - 4 портные access: одна VLAN - на одном switch, но разные сети
маркированы специальными номерами для маркированных меток, подсети разных VLAN

5-й порт switch - 5-портный trunk, но есть еще ТЕРМОПРОТИК

Sub-interface - подинтерфейс в определенном интервале

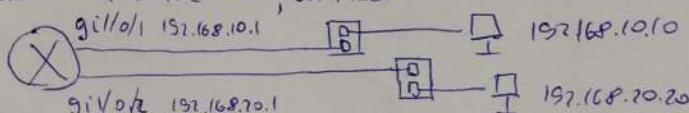
If sub-interface - подинтерфейс со своей VLAN: не коммутирует R1 коммутатор
к какому VLAN принадлежит, то есть это

На каком Sub-intf. может находиться IP адрес: текущий Sub-interface или предыдущий $gi1/0/1.10$ и $gi1/0/1.20$
При записи в таблицу инт. $gi1/0/1$. На какой же $gi1/0$ default = $192.168.10.1$.

Tablet of A n.B:

- 1) XOS A: frame not tag I receive access на VLAN 10, т.е. A подключен к B не в его范围内
- 2) определен A соединение по его switch port current MAC-адреса. где VLAN 100 и конфиг на R1
установлено тем самым = 10
- 3) R1 получает это от его суб-интерфейса $gi1/0$ и конфигурация порта не VLAN 20
- 4) switch отправляет по порту VLAN 20 его конфигурацию - то есть $MAC = 1$, конфиг $TRUNK = 20$ и конфиг на B

T.e. фактическая схема = логическая схема



QinQ - технология применяется 2 раза: внешний и внутренний

Хочу чтобы пользователь подключился к своему 2ому L2 клиенту
он зайдет через свой local switch за которым: там внешние
интерфейсы - Port-based VLAN. Внешний client, или userface
использует VLAN trunk. т.е. где оно это клиент оно VLAN.

Но! Тогда клиент уходит на B1, B2 - logical VLANs, A2B2 - logical port и т.д. т.е. разные VLANs - группы
логических ports client

Решение 1: багажный VLAN с 120 до 129. Но клиент может принадлежать 1000; в этом случае мы все равно
имеем VLANs, но если какомуто клиенту нужна 1000 VLANs, то он будет виден и клиенту

Решение 2: QinQ - это то что: 2 места назначения 802.11q: Eth-serviceprovider | Q-serviceprovider | Client | data

т.е. оно же клиент - 1 VLAN & его определяет, а дальше VLANs клиенту

различие на границе клиента. Теперь клиент может находиться в любом VLAN определяемого
switchом. Qservice, а SW2 - клиенту

VLAN-Translation: 2 клиента хотят VLANs с 120 до 129 - например. У них один 120-129 и 140-149,
но они хотят VLANs на switchах

PBB (Provider Backbone Bridging) отличие от PBT (Provider Backbone Tag Bridging) в том что PBB
- Provider Backbone Bridging

Классический STP = IEEE 802.1D

STP - позволяет подключить réseau / мосты на L2 уровне.

Принцип: проприетарный протокол для STP. STP имеет 3 порта с 1 нодом. SW1

Порядок работы STP:

1) Выбранный **Root switch** - который, оно же Root bridge называемый SW1

2) Все остальные **switchы** называемые **Root Port (RP)** - интерфейс, через который он Root будет получать

3) где есть **Designated Port (DP)** - называемое портом: который кратчайший путь.

Здесь между SW1 и SW2 2 различных пути - один из них, остальные блокируются

и) **Blocking** все остальные (кроме RP и DP) - (выбранное) \Rightarrow разрешены только

Если ноды одинаковые, то STP не разрешает gepo.

Когда switch становится Root Bridge и может всем BPDU-мостам. и с BID

RID - Root Identifier, BID = priority + MAC = 8000 + proprietar switch MAC-адресом

У обычных L2-switches, нет MAC, но, т.к. это switch с STP, есть MAC.

Раскрытие по BID: чем меньше, тем приоритетнее (настройка по умолчанию priority. Root = самый old)

При этом они все будут работать, кто из них Root (меньше MAC, < MAC2 < MAC3 и т.д. - root).

Т.е., кто he root, BPDU-мосты. Дальше не нужно, когда BPDU от root'a и не нужно его

Root уходит всем BPDU, с которым: DRIB

2) BID (непроприетарный switch-и chose lowest cost BID)

3) RPC = Root Path Cost - минимальный cost до корневого switch

root port - min. cost (the lower the better) \Rightarrow 1 и 2 выбирают в!

Проблемы classic STP

- Тайм-ут: BPDUs - в зоне, где нет-то хомяково блюфы и неподвижные гипер-трафик через зону (не нужно 10 BPDUs) \Rightarrow зоне. Проблема может возникнуть.

Существует несколько: blocking \rightarrow listening (сменял строки не передавал, 1SC = Forwarding delay),
 \rightarrow learning (также 1SC: заложение мостов в зоне, то же нет передачи неподвижных) \rightarrow forwarding
(это же неподвижные мосты) listening \rightarrow learning - неподвижные мосты.

Короче & кратко - бессмыслица цепь проблем.

Более продвинутые STP

RSTP (Rapid Spanning Tree Protocol - Successor of STP).

Минимум более скрупульно: Где лучше отдать обновление / неизвестны или гипер-трафик блокируется.

RSTP не настолько же тайм-ут, но они все равно есть: RSTP \neq STP симметричны \Rightarrow !

В RSTP есть алгоритм Proposal / Agreement - упрощает скрупульность

- 1) Выбирают кого-то гипера: root switch - как в STP
- 2) Root switch становится ближайшим к мостам и несет на себе BPDUs Proposal
- 3) Все остальные switch'ы становятся кандидатами и несут на себе BPDUs Agreement
- 4) Root переносится ближайшему, несет на себе BPDUs Agreement
- 5) Остальные switch'ы не получают BPDUs. Нефиг им Agreement, нечего делать - нечего и нефиг

Итог: идет "борьба". Все кандидаты зонтами пытаются в путь full duplex
 (Proposal / Agreement) выиграть. Идеально много, поскольку это логика - нечего и нефиг тайм-ут

В RSTP blocking \Rightarrow discarding: только неизвестные BPDUs

Задача next \rightarrow alternative переключения портов } несет текущие BPDUs на себе

- PVST+ (per VLAN Spanning Tree Protocol) [Cisco] \Rightarrow PC
 - PC-SOFTW C 802.1Q. Для каждого VLAN - отдельные гиперы
 - Rapid PVST+ [Cisco] = RSTP + PVST+
 - Запускает сразу \neq VLAN один RSTP

Оптимизация STP

1) Root Guard - регулирование наличия межмостовых коннектов.
 характеристика не 1 порт per switch, но наличие к коннекту. Задача root-a

2) Loop Guard - лучше от него в отработке ($\Leftarrow \Rightarrow \rightarrow$)

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

4) BPDUs forward задача от получения неизвестных BPDUs, чтобы в короткое время

5) Backbone Fast - тоже экономия времени

6) Bridge Assurance

Технологии агрегации каналов

2 независимых способа, но есть есть между ними различия:

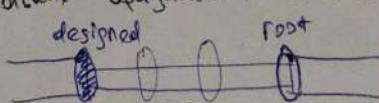
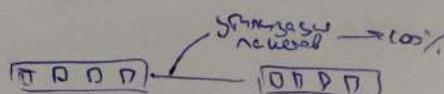
Базовый принцип:

1) Более бесконечные соединения - голова!

2) Многие каналы: PC-SOFTW есть только 1! ограничение - надо разобраться (geno RSTP)

\Rightarrow технологии LAN = Link Aggregation Group (Ether Channel / Port Channel.)

Одноединственное соединение нескольких Ethernet-каналов в 1, называемое



Объединение портов одного коммутатора для разных VLANов:

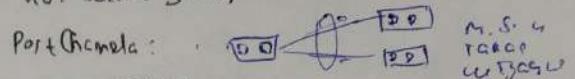
- switch
- port trunk
- native VLAN (для VLAN, который коммутатор не поддерживает)
- VLAN tagging (VLAN ID и access, trunk)
- VLAN (биты поля VLAN)

Служебная агрегация, назовем - STP!

Многомастер LAG между L2-L3 устройствами

Создание построения LAG

- Статическое объединение каналов - группировка нескольких Ethernet-каналов в один логический канал
 - Динамическое объединение каналов с помощью LACP/PAgP (Link Aggregation Control Protocol)
- Также есть алгоритмы для определения канала, т.к. они обмениваются информацией о состоянии MAC-адресов - наше дело MAC-адреса LACPDU с публичным MAC-адресом > канал не используется LACP. Это же используется в PortChannel, потому что оно не требует специального MAC-адреса



Балансировка трафика в LAG

Как работает механизм для сортировки? Хеш-функция

Распределение M.S. MAC, IP (+ порты, + протокол: TCP/UDP) \Rightarrow Hash = f(MAC_A, MAC_B, IP_A, IP_B, Protocol, Port_A, Port_B)

И следим, чтобы все порты получали одинаковый хеш и TCP/UDP

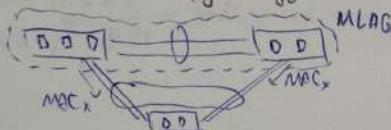
Оптимизация LAG

В классическом PortChannel используется один коммутатор, единственный источник трафика

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

Traffic 2 switches обрабатывается в vPC/MLAG - reply и request трафик идет через один коммутатор (один коммутатор MAC-адрес)

Также можно использовать сконфигурированные



OSPF

OSPF = Open Shortest Path First

Использование гипермощной маршрутизации (как в RIP), но этот - не DVA, а распределенный алгоритм балансировки трафика в сети

области - областные конфигураторы трафика.

На границах областей между областями обменяется таблицами состояния

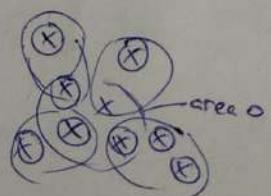
Рассмотрим OSPF - основные назначенные протоколом для конфигурации

Разные типы зон:

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

2 зоны (не 0) не могут общаться без подключения к 0

2) area n (standard)



Tunnel protocol:

Internal (бесшовный) - межузельный протокол, все устройства находятся в одной зоне (подразделение)

ABR - Area Border Router (переключатель) - конфигурация в одной зоне, но это - в другой зоне

ASBR = Autonomous System Boundary Router (переключатель между системами/зонами) - конфигурация OSPF зон

в зоне зоны - RIP, BGP...

Рассмотрим ABR и ASBR для дальнейшего изучения

3) Stub (stub-zone) - в зоне не может находиться больше 1 OSPF-зона. ASBR - последний в зоне не M.S.

4) totally stubby (totally stub-zone) - в зоне не может находиться и 1 OSPF-зона

иначе, чтобы в зоне оставалась одна зона

• Типы маршрутов между зонами:

- Intra-zone (внутризоны)
- Inter-zone (межзоны) - OSPF-зоны, но в разных зонах
- External (внешний) - различные протоколы (RIP/OSPF...)

В рамках одной зоны OSPF - link state протокол - протокол泛洪协议 LSA = Link State Advertisement:

На этом этапе он просто называется зоной: A Router, который не имеет LSA от соседей.

При переходе между зонами OSPF = DNA: переходные таблицы go переходов из одной зоны.

Известно на зону - для управления переходами назначены. Рекомендуется назначать один для каждой зоны.

ABR управляет переходами между зонами, к которым он подключен.

Если изменился BDR зоны, то путь все пересчитают, если в зоне - он от ABR будет менять параметры.

Типы LSA

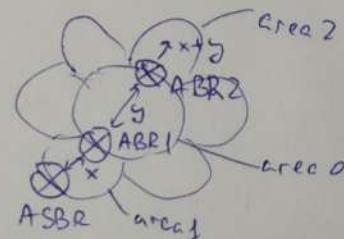
- LSA1 (Router LSA) - описывает путь к хостам зоны: путь от себя до зоны
 - LSA2 (Network LSA) - описывает хосты зоны, если в зоне есть switch hub. current switch будет первым по пути, и направление передачи пакетов.
- LSA1+LSA2 → структура топологии в одной зоне
- ABR LSA1/2 между зонами не распространяется, а считает LSA3
- LSA3 (Summary LSA) - тип. из другой зоны. содержит описание маршрута пакетов от ABR до зоны перехода. оно описывает LSA3 каждого перехода между зонами
 - LSAs (Autonomous System External) - где находятся внешние сети. тоже содержит описание ASBR, который не LSP и сам внешний, приводит к расширению его зоны от ASBR
 - LSA4 (Summary ASBR) - описывает маршрут пакетов от ABR до ASBR
- ABR генерирует LSA4 в зону 0, в которой соединение пакетов по ASBR.
- LSA4+LSA5 → описание топологии в внешней сети.

ABR генерирует LSA4, в которой указывает расширение до ASBR =

$$= l(ASBR, ABR1) + l(ABR1, ABR2)$$

ABR знает сам

↑ получает LSA4 от ABR1



Маршруты в OSPF

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

RefBW = 100 Mbit/s, BW - реальная линия пропускной способности. Чем выше скорость, тем меньше стоимость

→ для маршрутизации $RefBW \geq \max(BW)$

Типы

Unicast - один пакет для одного получателя: выявление траектории, IP, Ethernet

广播 - один пакет направляется всем хостам в зоне

Groupcast - пакет направляется группе хостов: Ethernet-группа (H₂) | IP-группа (H₃) | TCP/UDP-группа (H₄) | Data

Типы IP/IP

Базовые IPv4 & IP

Ethernet зон.	IPv4 зон.	IPv4-рекл. зон.	TCP/UDP-рекл.	Data
---------------	-----------	-----------------	---------------	------

Data и IP бывают IP-боксы - пакеты IP

IP/IPv4 → IPv4/IPv4: не только IPv4, но также IPv6, IPv4/IPv6, IPv4/IPv4/IPv6

GRE = Generic Routing Encapsulation

+ протокол: GRE = UDP, IPsec, L2TP, MPLS

Eth.	IPv4 зон.	GRE	IPv4/IPv6/other -> IP/Protocol	TCP/UDP/...
H ₂	H ₃	H ₄	H ₅	H ₇

секунды

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 IP-адреса, называемые псевдоадресами. HSRP использует псевдоадреса.

Нормальный процесс работы VRRP и HSRP: 1 - Active, 2 - Standby (резерв)

FHRP - это то же самое, что и VRRP, но с более широким функционалом.

При конфигурации HSRP: на интерфейске есть MAC-адреса, называемые IP-адресами реестра и блоком IP+VIP

Активатор обрабатывает сообщения от пользователей по VIP и vMAC

В случае отказа R2 становится VIP + vMAC реестром по Standby

vMAC обрабатывается автоматически по своему собственному MAC-адресу

Брандмауэр - gateway и хост - VIP-адрес

Клиент по ARP ходит блоками MAC-VIP: на него отвечает только Active и отвечает vMAC

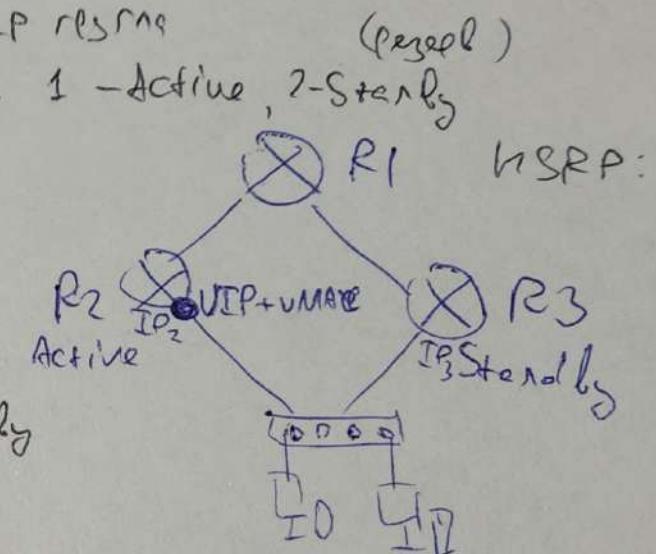
Для R2 → Standby меняется Active, отработав свои Greetings ARP чтобы подтвердить свою роль - она имеет право на меняться

Preempt: если кто-то другой (тайм-аут = 100) стал быть Active. Если нет, ожидается, то брандмауэр получит сообщение IP по которому исполнит HSRP.

А как же блоки, которые работают с одними и теми же ресурсами?

Если на всех блоках Preempt, то они могут засыпать непрерывно и не реагировать на изменения состояния Active

VLAN+HSRP: где каждому VLANу назначается HSRP-FPSM



1) У нас есть классный адрес класса B: 128.0.0.0 - 191.255.255.255. - это значит что, маска подсети не указана.

▷ Класс B: 128.0.0.0 - 191.255.255.255. - это значит что, маска подсети не указана.

При этом есть 3 подсети: 128.0.0.0, 129.0.0.0, 130.0.0.0 - все /16, т.е. 10000000.000.0
10000001.000.0
10000010.000.0

Для выполнения условия, в котором надо разнести сети между собой, нам нужно создать, но нам нужно чтобы эти маски подсети отличались.

получим 128.0.0.0 - 128.127.255.255 /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) Многосегментация

Многосегментация - разделение коммутатором одного физического интерфейса на несколько логических интерфейсов.

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

3) Коммутатор работает в режиме fragment free. Означает что данные, при которых он выполняет коммутацию в режиме store & forward

fragment free - читаем лишь первые байты сразу отправляем

store & forward - читаем все, проверим хеш суммы и отправляем.

- более быстрая обработка
- QoS или полная безопасность (QoS - качество обслуживания)
полная безопасность не имеет
- различия способов работы

4) Корт. регистрация к гостинице, IP имеет 1.2.213.0/16. Указана маска max. длинны,

которую может иметь этот узел

Маска не M.S. = 24, иначе адрес корта = адрес гостиницы > max. длина маски = 23

5) Понятие пересыпь кадровым max. PDU при TCP-сессии в момент установления соединения, в которой не применяется протокол ICMP

1) Иницирование соединения: клиент отправляет SYN на сервер для иниц. сог. TCP

2) Видимо опять MSS - ближайшее SYN, указывает 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 проблема маршрутизации! Указываю значения, которые будут использоваться для перенаправления IP-пакета с адресом получателя = 192.168.117.148

0 192.168.117.0/32 10.10.10.3 - нет, есть DB & 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 ближайшей маской

0 192.186.117.128/25 10.10.10.6 - 186 ≠ 168! lost

C 192.168.117.0/25 10.10.10.7 - нет, DST & 192.168.117.0/25

8) Стандартные термины из архитектуры OSI

Port - UDP - порт.

SFID - FCS - физический Ethernet-frame, 2sp

HTTP - POP3 - различные протоколы передачи

G2,5MM - 8P8C - тип

RIB - IPv4 - RIB = Router Information Base 2sp.

JPEG - ASCII - языковое представление

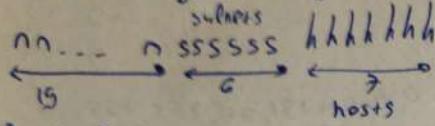
9) Fast Retransmission

③ Итог есть - 100 хостов. Какой мин. слои сетевая IPv4 чтобы все 50 подсетей?

50 подсетей на 100 хостов

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

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



$$32 - 6 - 7 = 19$$

1/9

⑤ Укажите диапазон для пересечения IP-подсети с подсетью 192.168.117.148

O	192.168.117.0/31	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

если "C", то это имена серверов

если "S" - это next hop - адрес, кому нужно передать пакет, чтобы он попал в нужную подсеть

O... - маска 132 - нет

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

$$192.168.117.148 \rightarrow \text{нет!}$$

источник
передачи
информации:
S - Connected - подключение к переносу
C - Specific - конкретные данные

адрес
представ
в маске

S.0.0.0.0/0 - это loopback

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

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

нет

Если подойдет несколько адресов:

- у него лучше маска
- но горячие: C, S, O, R

⑥

8P8C - 62,5ММ

IPv4 - RTB -

UPP - Port - транспортный协议

SFD - FCS -

HTTP - POP3

JPEG - ASCII - это кодировка не языка представления

RPC - NetBEUI

⑦ Необходим блок адресов 192.188.0.0/10. Выделен диапазон 192.129.2.0/23. Требуется
коин из исходного блока 192.129.2.0/23. Возможные варианты

$$192.128.0.0/10 \rightarrow \begin{cases} 192.128.0.1 \\ 192.191.255.254 \end{cases} \quad 192.129.2.0/23 \rightarrow \begin{cases} 192.129.2.1 \\ 192.129.3.255 \end{cases}$$

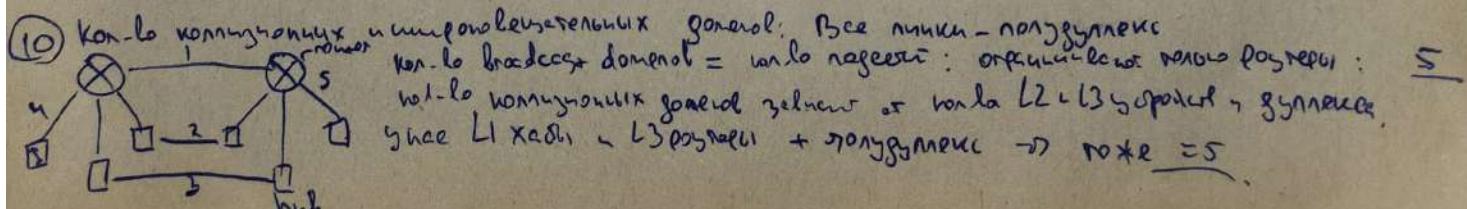


Будет /10 или нет? Но можно ли /11? Видимо отсутствие корреспондентных блоков 192.129.2.0/23.

11000000.10000000.00000000.00000000

11000000.10000001.00000010.00000000 - нужны идущие биты,

11000000.10000010.00000000.00000000



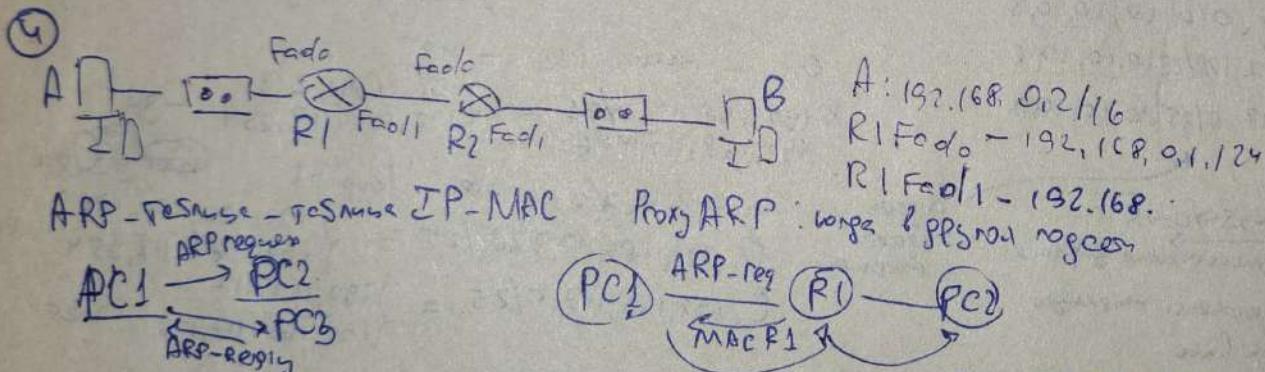
⑨ Чем отличаются кабели в оптике?

8P8C -> 8P8C
Совместное окно нет Categories 3 - 8P
Store & forward -? 24 AWG 5P 26 AWG 8P

⑩ SACK
Не классифицируется как TCP, но имеет такие же преимущества с расширением

⑪ У нас есть 3 подсети подсети

Маке маска B: 255.255.0.0 > Маска B - 255.128.0.0 - 192.0.0.0 - 191.255.255.255



- 1) A подключен к B - не имеет подсети
- 2) A подает ARP request broadcast o MAC адреса default gateway
- 3) R1 получает ARP reply с адресом MAC R1.
- 4) A получает от B: ответ с адресом R1, R1 не является сетевым, но имеет его в таблице MAC R2.
- 5) R1 получает ARP request от B, R2 имеет его в таблице MAC.
- 6) R2 ответит на запрос от R1. Ответ будет иметь B - как и ответ от R1.

PC-A: 192.168.3.1/24 - MAC R1
192.168.0.1/24 -