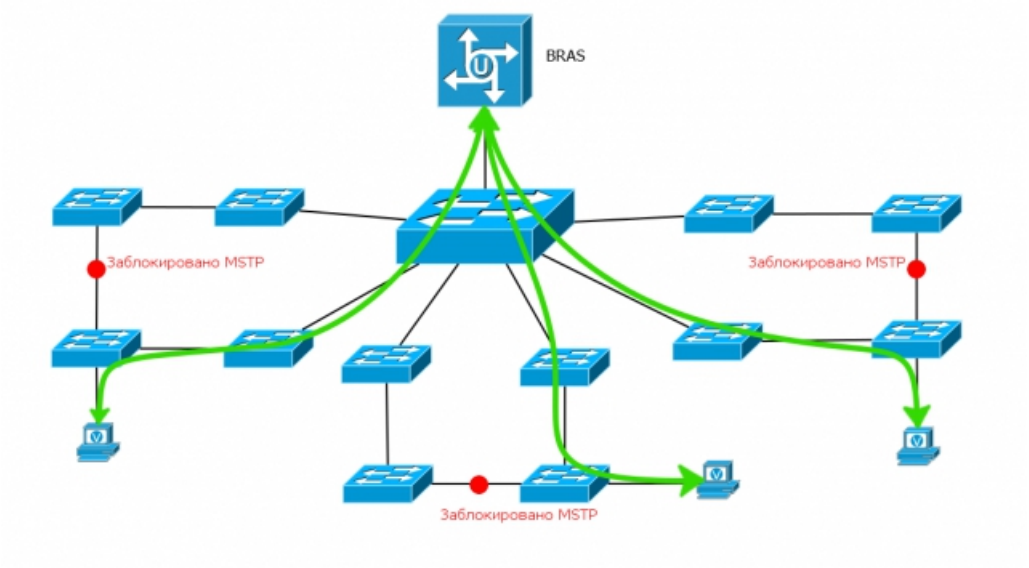
Такие разные проблемы

Чем крупнее сеть, тем более разношёрстнее оборудование там стоит, тем сильнее различается уровень компетенции обслуживающего персонала, тем больше проблем у них возникает.

Хочу представить вам подборку разноплановых аварий на сетях, доставивших немало часов увлекательной отладки.

По следам забытых RFC

Имеем вот такую сеть.



На доступе кольца коммутаторов младшей линейки, защищённые MSTP. Кольца агрегируются на вышестоящем коммутаторе, который напрямую подключен к BRAS'у. Кольца разорваны посередине, чтобы уравновесить нагрузку на разные порты агрегаторов.

Помимо базового доступа в Интернет предоставляется услуга IPTV.

Симптомы проблемы: Периодически наблюдается повышенная загрузка CPU сразу огромного количества коммутаторов по всему L2-домену, и в такие моменты они становятся недоступными для управления на несколько секунд. Сервисы при этом не страдают.

Анализ и решение

Раз сразу несколько коммутаторов страдают в одно время, надо начинать поиски с внешних факторов. Начинаем, как обычно, с проверки логов.

В них встречаются такие сообщения:

> 2013-09-16 20:46:37 The CPU is overloaded, and the tasks with top three CPU occupancy are frag_add(16%), SNPG(15%), FTS(13%). (CpuUsage=95%, Threshold=95%)

Frag_add отвечает за изучение MAC-адресов и заполнение ARP-таблицы, *FTS* - за отправку и получение пакетов, а *SNPG* - за обработку мультикаста.

Учитывая некую периодичность появления таких сообщений, предполагаем, что есть какое-то внешнее событие, которое запускает механизмы коммутатора. А обновление таблицы MAC'ов и ARP обычно происходит сразу после получения TC-пакета процессом STP. *Это делается для того, чтобы сразу отреагировать на изменения топологии.*

Не пришлось долго искать такие сообщения в логах:

> 2013-09-16 20:09:14 MSTP received BPDU with TC, MSTP process 0 instance 0, port name is GigabitEthernet0/0/1.

То же самое показывает и статистка полученных TC:

```
<Quidway>display stp tc-bp statistic
------------------------- STP TC/TCN information -------------------------
MSTID Port                TC(Send/Receive)    TCN(Send/Receive)
0   GigabitEthernet0/0/1    8/60            0/0
0   GigabitEthernet0/0/2    88/6            0/0
```

Этим объясняется процесс *frag-add*. Частично этим объясняется и активность *FTS* (получение и отправка пакетов) - первое время каждый принятый кадр нужно обработать на CPU и загнать MAC-адреса в таблицу. Но это событие не из ряда вон выходящее, и если каждое отключение порта приводило бы к повышению CPU, жить такому вендору было бы сложно.

Есть ещё процесс *SNPG*, отвечающий за мультикаст. Что не так с ним?

Дело в том, что в конфигурации присутствует команда **igmp-snooping send-query enable**. Она отвечает за отправку IGMP General Query во все свои порты при получении STP TC. Сделано это для того, чтобы быстро перестроить таблицу мультикастовой коммутации, когда изменилась топология. Таким образом, коммутатор быстро узнаёт, где появились клиенты, которые прежде получали трафик по другому пути. По умолчанию GQ отправляется на адрес 192.168.0.1. Естественно, он уходит и в тот порт, который выполнял роль Router Port, потому что при перестроении кольца он мог поменяться, и там могли появиться клиенты. В общем, не хилая такая работа для коммутатора младшей линейки - обработать весь поток Report'ов такого большого L2-домена.

Далее опытным путём подтвердили предположение: после удаления этой команды загрузка CPU оставалась в пределах нормы во время перестроения кольца. Кроме того, обратили внимание, что об обрыве в одном кольце уведомляются коммутаторы во всех других (рассылаются TC BPDU). Это следствие того, что на весь немаленький L2-домен натянут один MSTP процесс. По этой причине страдало так много коммутаторов. Вот казалось бы и распутали клубок - удалить команду и желательно разнести разные кольца по разным MSTP Instance или даже процессам, чтобы не было такого острого взаимовлияния. Но вот оказия: как только удалили эту команду, после изменения топологии прекращает работать мультикаст. А через несколько минут восстанавливается сам по себе. Повторяемость 100%. Найти ответ в лоб не удаётся. Что происходит, совершенно непонятно.

Остаётся только дампить трафик.

Вот результаты:

```
91 15:25:15.788948   0.0.0.0   224.0.0.1   IGMPv2   60 Membership Query, general
92 15:25:15.844759   0.0.0.0   224.0.0.1   IGMPv2   60 Membership Query, general
93 15:25:15.846257   0.0.0.0   224.0.0.1   IGMPv2   60 Membership Query, general
```

```
Jan 14 2008 02:32:13 26 %%01IFNET/4/IF_STATE(l)[41]:Interface GigabitEthernet0/0/2 has turned into DOWN state.
Jan 14 2008 02:32:13 26 IFNET/2/IF_PVCDOWN:OID 1.3.6.1.6.3.1.1.5.3 Interface 32 turned into DOWN state.
Jan 14 2008 02:32:17 26 %%01IFNET/4/IF_STATE(l)[42]:Interface GigabitEthernet0/0/2 has turned into UP state.
Jan 14 2008 02:32:17 26 IFNET/2/IF_PVCUP:OID 1.3.6.1.6.3.1.1.5.4 Interface 32 turned into UP state.
Jan 14 2008 02:32:17 26 MSTP/1/TOPOC:OID 1.3.6.1.2.1.17.0.2 Bridge topology change.
Jan 14 2008 02:32:17 26 MSTP/4/PFWD:OID 1.3.6.1.4.1.2011.5.25.42.4.2.1 The port has been set to forwarding state. (InstanceID=0, PortInstanceID=0, PortID=25, ifIndex=32, PortName=GigabitEthernet0/0/2)
<Quidway>dis igmp-snooping statistics vlan 4
IGMP Snooping Packets Counter
 Statistics for VLAN 4
  Recv V1 Report      0
  Recv V2 Report      0
  Recv V3 Report      0
  Recv V1 Query       0
  Recv V2 Query       11
  Recv V3 Query       0
  Recv Leave          0
  Recv Pim Hello      0
  Send Query(S=0)     3
  Send Query(S!=0)    0
  Proxy Send General Query          0
  Proxy Send Group-Specific Query   0
  Proxy Send Group-Source-Specific Query 0
```

То есть дёрнули порт и непонятно почему, коммутатор отправил три пакета IGMP General Query с адресом отправителя 0.0.0.0. Что это такое, откуда, зачем? На вопрос отвечает ~~Максим Поташёв~~ RFC4541.

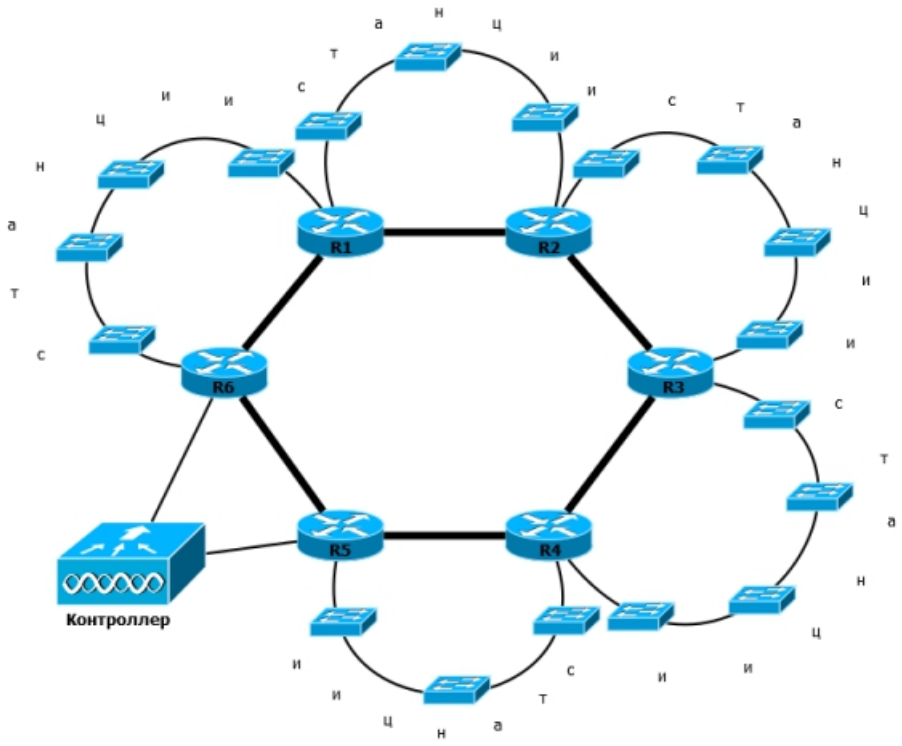
> The 0.0.0.0 address represents a special case where the switch is proxying IGMP Queries for faster network convergence, but is not itself the Querier. The switch does not use its own IP address (even if it has one), because this would cause the Queries to be seen as coming from a newly elected Querier. The 0.0.0.0 address is used to indicate that the Query packets are NOT from a multicast router.

Иными словами, это специальный Query, который позволяет ускорить сходимость. Адрес 0.0.0.0 означает, что запрос не от мультикастового маршрутизатора, и не нужно производить перевыборы Querier. И тут картина начинает складываться. **BRAS просто складывал с себя обязанности Querier после получения 0.0.0.0 (он же меньше текущего)**. Заблокировали отправку IGMP Querier с агрегирующего коммутатора на BRAS - никаких проблем больше нет. Потрачено много времени на отладку, а всё потому, что кто-то не придерживается RFC. Кстати, вышла в свет новая статья из цикла Сети Для Самых Маленьких. Она посвящена кокрастыке мультикасту и поможет вам разобраться в деталях.

Loop, да Loop around

Список продолжает проблема с широковещательными штормами. Пожалуй, это настоящий бич сетей MetroEthernet. Огромная часть запросов первого уровня открывается именно по штормам: петли физические, петли логические, петли в MPLS L2VPN. И в то время, как все стремятся уменьшить радиус широковещательного домена до локальной сети, некоторые расширяют его до масштабов города.

Имеем вот такую сеть

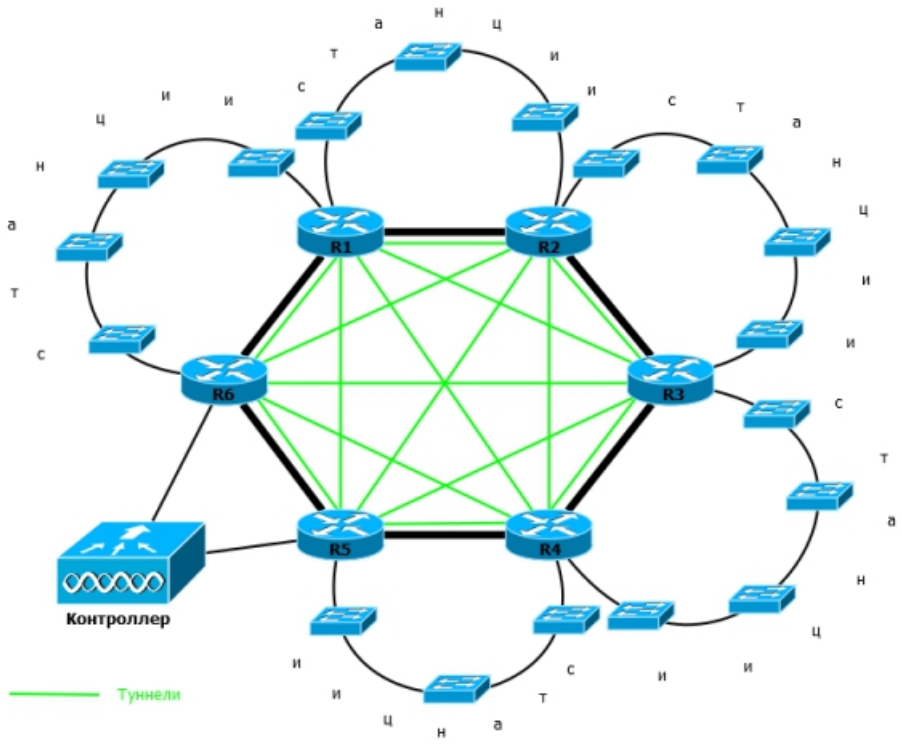


Есть один центральный узел-контроллер. К сети MetroEthernet подключаются станции, которые хотят взаимодействовать с контроллером, причём не просто взаимодействовать, а именно так, словно они подключены непосредственно друг к другу безо всяких маршрутизаторов.

Для этого на сети MetroEthernet поднята услуга MPLS L2VPN. Все станции являются членами одного широковещательного домена. В кольцах доступа существует чистый IP/Ethernet, а внутри агрегации уже MPLS.

Кольцо защищено RRPP - проприетарным протоколом Huawei для кольцевых топологий. Пакеты RRPP - это BPDU, чтобы они проходили через сеть MetroEthernet, создан специальный VSI исключительно для RRPP. То есть пакеты RRPP Hello прозрачно проходят через маршрутизатор благодаря L2VPN.

Внутри сети между маршрутизаторами Full-Mesh с помощью MPLS TE туннелей для передачи данных L2VPN. В качестве IGP выступает ISIS (*хочу напомнить, что он активируется на интерфейсах, в отличие от OSPFv2*).

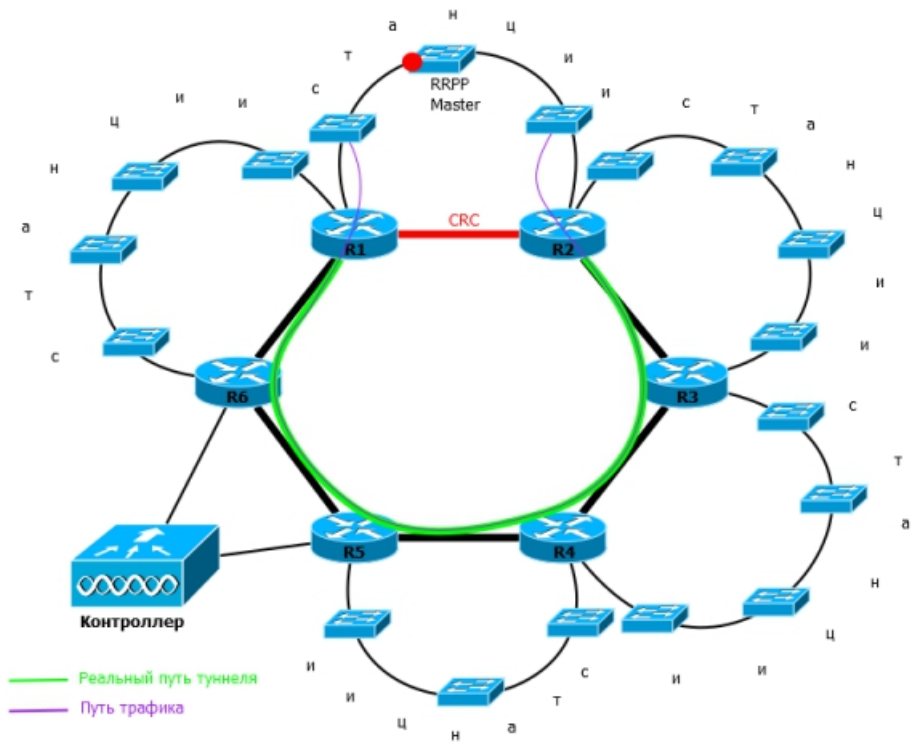


В какой-то момент на одном из магистральных интерфейсов сети MetroEthernet начал расти счётчик CRC-ошибок. Естественно, это сразу сказалось на сервисах. Интерфейс выключили, ошибки кончились, но кольцо MetroEthernet оказалось разомкнутым, что как бы страшно.

Но, разумеется, всё продолжает работать, туннель между двумя маршрутизаторами перестроился на резервный путь по кольцу.

RRPP тоже работает, потому что Hello-пакеты проходят успешно.

То есть сейчас, если два устройства в одном кольце захотят пообщаться друг с другом, они пойдут по длинному пути.



Но долго без резервирования жить нельзя - сеть не Украина - отделить кусочек нельзя. Инженеры решили провести тест линии. Для этого они выбрали время наименьшей нагрузки, взяли с собой запасные SFP и платы и поехали на узел.

Чтобы не проводить операции на живой сети, решили изменить настройки IP на интерфейсе - взять адреса из какого-нибудь неиспользуемого в сети диапазона - они же ведь в этом случае ни с кем не пересекутся.

Поменяли IP, включили интерфейсы, запустили пинг. Наблюдают ошибки на портах, потери в пинге - проблема на лицо. Наблюдают какое-то время ещё перед тем, как менять линию, не обращая пока внимания, что терминал начал подтормаживать. Через несколько минут звонок от центра мониторинга - "недоступна большая часть станций".

Глядь, а там загрузка интерфейсов под 100%, CPU тоже перегружен - как есть шторм.

Разорвали проблемный линк, подождали минут 5 - ничего не изменилось - шторм прогрессирует. Начали рвать все имеющиеся кольца - помогло.

Два вопроса: почему случился шторм, ведь предприняли, казалось бы, меры и почему отключение проблемной линии не помогло?

Легче оказалось ответить на второй вопрос. В силу особенностей дизайна сети широковещательный шторм распространился очень быстро. И к моменту, когда инженеры узнали о шторме, он уже вовсю царствовал на коммутаторах других колец. Процессоры были заняты обработкой всяческих ARP'ов и других широковещательных пакетов и начали отбрасывать пакеты RRPP Hello. RRPP Master, не получая Hello, разблокировали свои резервные порты, считая, что кольцо разорвано. А на самом-то деле кольцо было целым и продолжало передавать трафик.

В итоге даже после разрыва линка между R1 и R2 петли продолжали плодить шторм и в других кольцах.

Но что же пошло не так с самого начала?

Несложно догадаться, что корень проблемы именно в CRC-ошибках. Результаты работы чёрного ящика показали, что в буфере чипа линейной платы произошёл сбой, который, грубо говоря, инвертировал бит, записанный в него, тем самым повреждая данные. Ирония заключается в том, что мутировали именно пакеты RRPP Hello, другие данные были затронуты в меньшей степени. RRPP Master не понимал, чего от него хотят, не обрабатывал Hello и снимал блокировку со своего резервного порта, смыкая тем самым петлю, в то время, как все другие данные ходили вполне нормально, и сразу после этого начали размножаться.

"*Но мы же... поменяли IP-адрес - не должно было ничего подняться*", "*таблица маршрутизации не знает*" - говорили они, "*MPLS TE-туннели должны идти по нормальному пути*" - говорили они.

Развязка

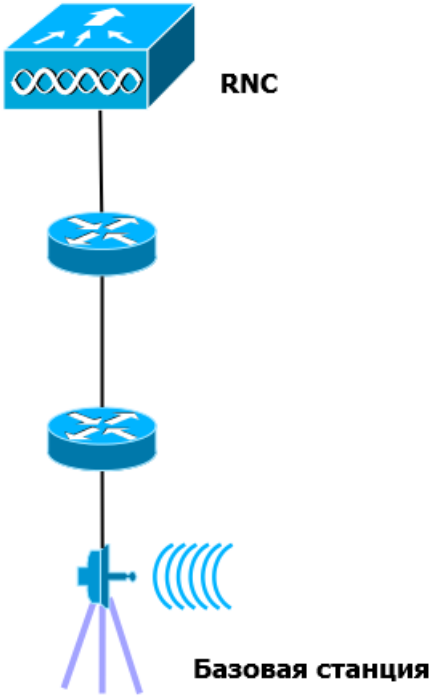
Концами TE-туннелей являются адреса Loopback-интерфейсов. RSVP строит LSP по кратчайшему пути. Да, пока линк был в дауне, путь лежал через всё кольцо MetroEthernet, но потом его подняли. И... на интерфейсе не был выключен ISIS, который не посмотрел, что IP-адреса какие-то левые, а просто изучил, что через них доступны Loopback'и других маршрутизаторов, туннели легли через этот путь, данные начали корёжиться.

Штормы это всё-таки один из самых сложных типов проблем - устранить сравнительно легко и сделать это надо очень быстро, а разбираться потом, когда уже всё кончилось, где там, что коротнуло.

Если стандарты пишут, значит это кому-нибудь нужно

Интереснейшая проблема, которую, слава Лейбницу, решал не я. Копий было сломано много, война была затяжной.

Очень простая сеть



Всё работает, базовые станции успешно общаются с RNC, трафик ходит. Далее происходит плановое обновление ПО на маршрутизаторах, и после него на RNC вываливается авария о том, что потеряна синхронизация с БС - сервисы разваливаются. Откат - всё работает. Ещё один тест для подтверждения - ситуация повторяется.

Важно понять, как происходит синхронизация. Существует не так уж много известных механизмов для этого: NTP, TOP, SynchroEthernet, 1588v2 (он же PTP).

Заказчик с полной уверенностью утверждает, что для синхронизации используется NTP. Хоть мы такого и не встречали никогда, но ладно - радиооборудование другого вендора, мало ли что он там накрутил. Ответственный инженер и целая команда разработчиков, проглядела все глаза, пытаясь найти хоть какие-то различия в NTP-пакетах до и после обновления. R&D поклялись Китайской Великой Стеной, что не вносили никаких изменений в алгоритм обработки NTP-сообщений. Сравнивали до байта данные в пакете.

И тут закралось первое подозрение, а что если не по NTP? Заказчик продолжал спорить, вендор радиооборудования молчал, как рыбой об лёд, потому что с ним не было контракта на техподдержку.

Мы же практически с завязанными глазами должны были угадать, что идёт не так в многомегабитном потоке данных. Пытались найти следы других механизмов синхронизации - ноль по вдоль.

В итоге чёрный ящик крутился, крутился и через несколько дней выплюнул - VSS Monitoring (кажется, так Wireshark определял эти пакеты). Пакеты этого типа ходили между RNC и BS. Мы до сих пор не уверены в том, что именно в них ходила синхра (возможно, внимательный читатель прояснит эту ситуацию), но именно в них обнаружили кардинальное отличие между различными версиями.

Это оказались пакеты сравнительно маленького размера - несколько байтов полезной нагрузки. Но, как известно, Ethernet не приемлет радикального минимализма - ему подавай как минимум 64 байта. Понятное дело, что даже заголовки MPLS+L2VPN+Ethernet+VLAN (4+4+18+4=32 байта) не всегда добьют до нужного, поэтому есть специальный механизм - **Padding**. Он дополняет размер Ethernet-кадра до минимального - до 64 байтов.

И как обычно, дьявол в мелочах. Заполняться, согласно стандарту, должно нулями, мол проще и легче определить, где кончаются полезные данные, где начинается padding. Но этого стандарта почему-то не принято строго придерживаться. Часто оказывается эффективнее заполнить тем, что уже есть в буфере, а отделение зёрен от плевел переложить на вышестоящий протокол.

Так вот оказалось, что в старой версии всё работало согласно стандарту, а вот в новой было решено оптимизировать механизм Padding, и кадр дополнялся не нулями, а куском из этого же самого кадра. Учитывая, что это не исключительное нарушение стандарта, подавляющее большинство протоколов с этим как-то справляется, а кто-то не справился, скорее всего, приняв всё заполнение за полезную нагрузку. Тут остаётся только гадать.

В кратчайшие сроки был выпущен временный патч, возвращающий назад режим работы - проблема

решилась. Это пример очень сложного запроса и показатель хардкорности инженеров техподдержки. Даже со связанными руками, закрытыми глазами, кляпом во рту и работающим рядом отбойным молотком эти ребята способны решать сложнейшие запросы.

Как выстрелить себе в ногу и ничего не понять

Проблемы первого уровня называются также критическими, но довольно таки часто на деле они являются скорее истерическими. Это ситуации когда заказчик хочет открыть запрос первого уровня, но в действительности ничего страшного не случилось, либо случилось, но сервисы уже восстановлены.

Вот вам пьеса в трёх действиях:

Саботажник

У инженерного состава паника - уже несколько раз они наблюдали падение линка между ядром сети и агрегацией. Пять минут лежит, потом само поднимается. Падает сервис всего города. Во внутренней системе заявок идут ожесточённые дискуссии на эту тему, считают, сколько пользователей страдает, смотрят логи со всех узлов, статистику. Я в этом участия не принимаю и могу с чувством, с толком, с расстановкой изучать логи.

Кстати, надо тут заметить, что в Hi-End маршрутизаторах Huawei два хранилища для логов - просто буфер, который выводится по команде **display logbuffer**, и файл подробнейших логов на карте памяти (*cfcard2:/log/log.log*).

В последний сохраняются все события и все действия инженеров на коммутаторе. Вот его я и начал смотреть. Догадаетесь, что я там увидел?

```
Mar 8 2014 10:15:03 %%01SHELL/5/CMDRECORD(l): Record command information. (Task=vt0, ip=X.X.X.X, User=newbie, Command="interface Eth-Trunk 3")
Mar 8 2014 10:20:13 %%01SHELL/5/CMDRECORD(l): Record command information. (Task=vt0, ip=X.X.X.X, User=newbie, Command="shutdown")
```

Торопыга

А вот случай совсем недавно. Инженер жалуется, что после команды **undo mpls** на интерфейсе упали все LSP, LDP-сессии, туннели на устройстве - восстановилось после ребута. Полный предвкушения, что будет что-то интересное, я полез в логи. Печенька не заставила себя долго ждать:

```
2014-1-1 00:00:00.000.1 LoI-gw1 %%01SHELL/5/CMDRECORD(l)[112789]:Record command information. (Task=VT0, ip=10.20.30.40, User=newbie, Command="system-view")

2014-1-1 00:02:21.000.1 interface GigabitEthernet1/1/1 %%01SHELL/5/CMDRECORD(l)[112791]:Record command information. (Task=VT0, ip=10.20.30.40, User= newbie, Command="sysname interface GigabitEthernet1/1/1")

2014-1-1 00:02:09.000.1 interface GigabitEthernet1/1/1 %%01SHELL/5/CMDRECORD(l)[112792]:Record command information. (Task=VT0, ip=10.20.30.40, User= newbie, Command="undo mpls")
2014-1-1 00:02:28.000.1 interface GigabitEthernet1/1/1 %%01SHELL/5/CMDRECORD(l)[112793]:Record command information. (Task=VT0, ip=10.20.30.40, User= newbie, Command="undo mpls ldpsi")
```

System-view - режим конфигурации
sysname - команда для смены имени устройства.

Мазохист

Проблема: не работает команда по добавлению статического Router port в IGMP - после настройки клиенты перестают получать мультикастовый трафик.

На канальном уровне для мультикаста есть много плюшек для управления. Одна из них - статическая настройка порта в сторону маршрутизатора.

Вообще Router port изучается коммутатором благодаря IGMP-Snooping после получения IGMP General Query от маршрутизатора. Когда на коммутатор приходят Report от клиентов, он отправляет их именно в Router port. Бывают ситуации, когда маршрутизатор по какой-то причине не посылает General Query, но посылать Report'ы на него по-прежнему нужно. Либо вы хотите чётко определить, что отправлять Report нужно именно в этот интерфейс.

Делается это в режиме настройки интерфейса командой igmp-snooping static-router-port vlan X.

Есть дополнительная команда для выключения изучения того, где находится маршрутизатор вообще: undo igmp-snooping router-learning. В этом случае он просто начинает игнорировать Query с этого порта.

Вместе эти две настройки позволяют жёстко зафиксировать порт, куда следует отправлять Report, и соответственно откуда получать трафик.

Как же можно здесь навредить себе? А всё довольно просто - достаточно включить IGMP-Snooping Proxy.

Ведь на оборудовании Хуавэй у него три базовых правила работы:

1. Если коммутатор получил Query от вышестоящего маршрутизатора, он отправляет ему ответный Report, не опрашивая клиентов - просто сообщает, какие группы ему нужны.
2. Если коммутатор получил самый первый Report для данной группы, он отправляет его наверх. Все другие Report он учитывает, но не отправляет
3. Если коммутатор получил самый последний Leave (от последнего клиента группы), он отправляет его наверх. Все другие Leave учитывает, но не отправляет.

И что же это получается, если настроить все команды разом? Первый клиент подключается, посылает Report. Это первый Report для группы, поэтому коммутатор отправляет его в Router port. Трафик начинает литься и клиент его получает.

Далее маршрутизатор тщетно отправляет GQ на в интерфейс, но коммутатор их отбрасывает, потому что настроено undo igmp-snooping router-learning.

Сам по себе клиент, как правило, не отправляет Report периодически.

По прошествии трёх минут маршрутизатор прекращает свои потуги, не получая Report, и удаляет интерфейс из списка OIL. Все следующие клиенты уже были не первыми и их Report'ы погибали на коммутаторе.

Что объединяет все описанные ситуации? Человеческий фактор, расхлябанное отношение, невнимательность.

Решение сетевых проблем часто сравнивают с работой врача. Пациент жалуется на боль, а мы с помощью наводящих вопросов и анализов находим болезнь. Разница лишь в том, что цена нашей ошибки - деньги и репутация, а не человеческие жизни. У нас есть право на ошибку, но не нужно им злоупотреблять.

Пожалуй, из ряда выбивается только проблема с паддингом - всё-таки разработчики действовали из лучших побуждений - они хотели оптимизировать работу и в общем-то не их вина, что стандартом не достаточно чётко это прописано. И не вина другого вендора, что он не предусмотрел такую возможность и не адаптировал к ней свои механизмы.

Ещё более глубокая проблема - широковещательные штормы. Понятны причины, по которым на L2 существует принцип широковещания, можно понять и причины, по которым нет поля TTL в кадрах Ethernet, понятно даже то, почему изначально разработчики не задумывались о таких проблемах - в конце концов, эта технология задумывалась для LAN, а не для сети провайдера или уж тем более MAN. Однако эта недальновидность дорого теперь обходится операторам. Но это уже совсем другая история.