Бесчеловечные сети

Любой инженер стремится автоматизировать повторяющиеся действия, стремится свести к минимуму количество работы, которую нужно сделать вручную.

Несложно представить сколько тоски и боли причиняют нам проблемы, которые можно было бы решить без нашего участия.

Ведь как было бы здорово, если бы оборудование могло само найти ошибку в настройках и поправить её, если бы оно в читаемом удобном виде предоставляло информацию о ситуации в сети и потенциальных проблемах, само анализировало события и выстраивало зависимости. А ещё лучше, чтобы после его установки на сайтах оно само настраивалось и следило за своим здоровьем, просто оповещая нас о том, что нужно заменить неисправную плату или пора расширить пропускную способность.

В этой статье я хочу начать рассмотрение концептуального вопроса о "бесчеловечных" сетях - сетях с минимальным участием человека в настройке и обслуживании.

Чтобы понять, как выполнить автоматическую настройку оборудования, мне кажется, в первую очередь нужно научить программу находить ошибки в конфигурации. Ну как, например, можно заставить программу выбрать IP-адреса для линковой подсети, если она не может определить, что адреса на одной стороне и на другой из разных подсетей? Или как заставить прокинуть VLAN между двумя коммутаторами, если программа не сможет определить, что настройки транка на разных сторонах не совпадают?

Поэтому будем отталкиваться от траблшутинга и для начала введём классификацию проблем на оборудовании:

1. Критические ситуации реального времени;
2. Проблемы, которые имеются в данный момент, но не затрагивают сервисы;
3. Аппаратные проблемы оборудования;
4. Потенциальные проблемы ПО;
5. Некорректная конфигурация.

Разберём каждый из этих типов и подумаем, как можно автоматизировать поиск причины и устранить проблему.

1. Критические ситуации реального времени

Первый тип - это то, над чем бьются все умы телекома с момента его зарождения:стабильность работы и минимальные потери сервисов при работе.

К таким проблемам относится то, что затрагивает трафик прямо сейчас - отключение интерфейсов, выход из строя платы или узла, пожар, отключение электричества, сбой ПО или протоколов.

В любом случае, всё это сводится к тому, что внезапно становится недоступным какой-то из путей и нужно в срочном порядке найти новый - перевести на него трафик.

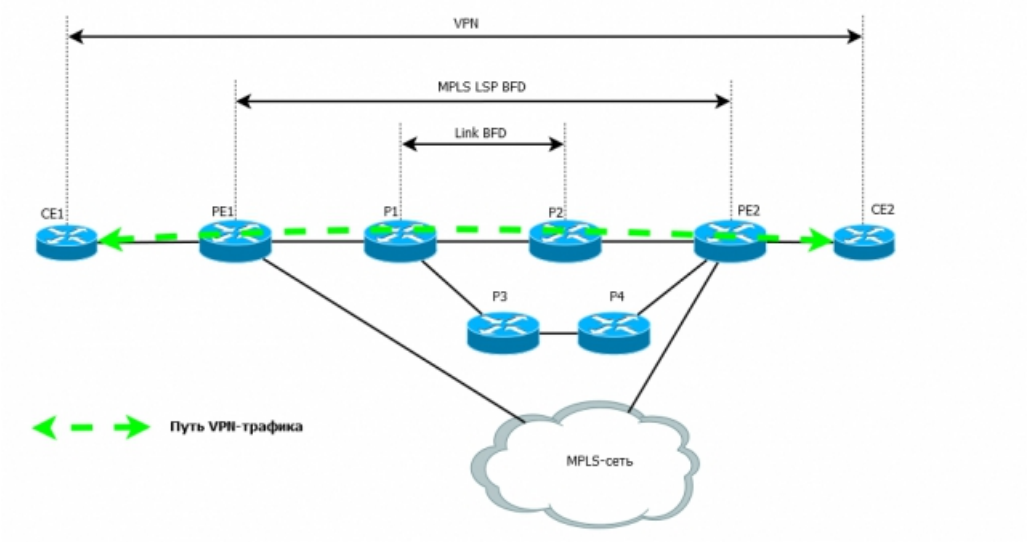
На этой ниве разработано огромное количество протоколов и механизмов: от STP с его бесконечными модификациями до MPLS TE.

Учитывая современную тенденцию к All IP, в сетях с коммутацией пакетов всё более актуально встают вопросы времени переключения трафика. Сейчас все стремятся к цифре в 50 мс возле места возникновения проблемы и 200 мс End-to-End. Такие потери не успеет заметить конечный пользователь.

Механизмы BFD, OAM, MPLS TE FRR и прочие, вроде бы, чудесно справляются с этим, **при условии их корректной настройки**. Но всё это чуждые друг другу по своей сути вещи.

Дело в том, что ни у IP, ни у Ethernet нет встроенных средств мгновенного обнаружения проблем, в отличие, например, от SDH. И приходится мудрить разнообразные схемы взаимодействия.

Вот пример реального дизайна.

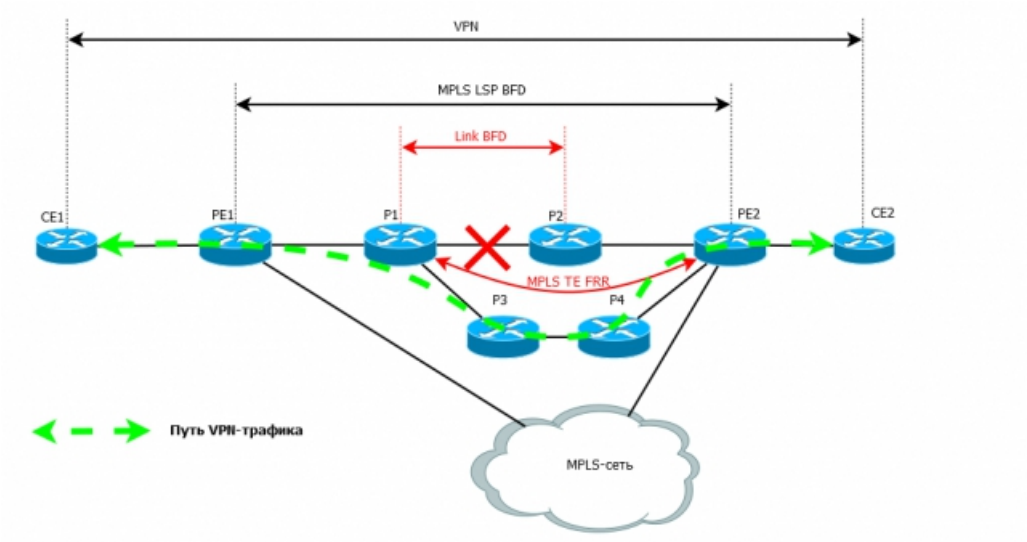


1. Организован MPLS TE Tunnel между двумя PE-устройствами. В него зарулены VPN, в которых передаётся трафик 2G и 3G базовых станций.

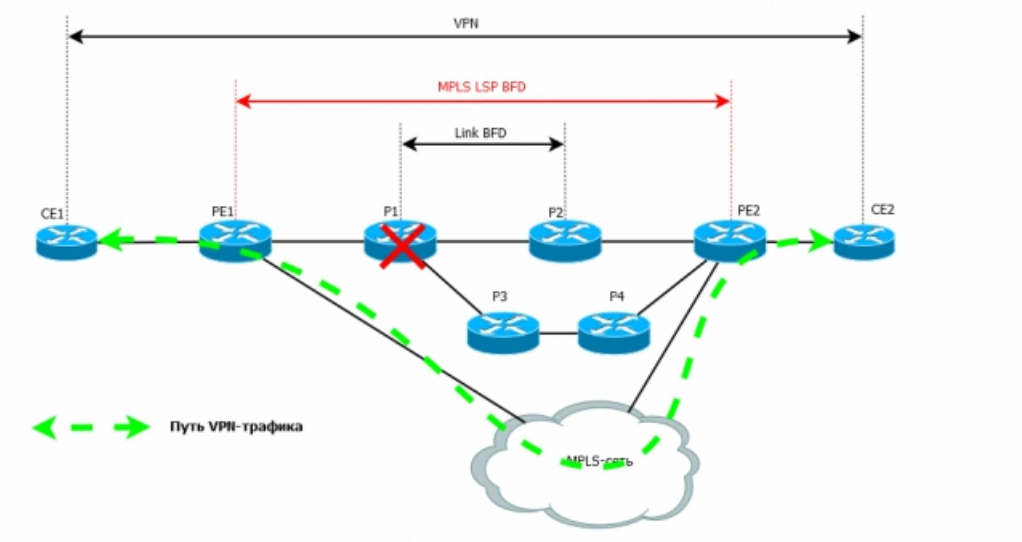
2. BFD запущен между узлами P1 и P2 для отслеживания статуса линии (на самом деле он может быть активирован на всех линковых интерфейсах).

3. На сети активирован функционал Fast Reroute.

В случае обрыва линии между P1 и P2 мгновенно отрабатывает BFD и переводит интерфейс в состояние Down. Это замечают все протоколы и начинают активные действия. В первую очередь срабатывает FRR и находит временный обходной путь. Это случается в пределах 50 мс. Никто и глазом не успевает моргнуть. Даже BFD на TE Tunnel, потому что он быстрее перестроился благодаря FRR.



Если выйдет из строя узел P1, то уже сработает MPLS TE Tunnel BFD. Если есть заранее настроенный резервный канал, всё переключится на него. Опять же незаметно.



Много тонкостей, масса настроек. Все эти разные по своей природе протоколы нужно сращивать друг с другом. Ювелирная работа. Однако это то, что мы имеем, и оно работает.

Естественно, тут едва ли кто-то будет придумывать новый стандарт вместо Ethernet, IP или TCP, сочетающий в себе плюсы всех предыдущих, потому что, как известно, в результате появится ещё один стандарт со своими недостатками. Поэтому нужно очеловечить существующую ситуацию.

Конечно, здорово было бы, чтобы всё настроилось само правильно и красиво, но, как я сказал выше, путь к этому лежит через автоматический траблшутинг, который мы и должны сейчас обсудить.

А вот другой момент - поиск первоначальной причины - что вызвало такой кипиш на сети? Сейчас это выглядит, как перебор тонны логов со всех устройств сети и их анализ. Не всегда это тривиально, не всегда можно однозначно понять, где причина, где следствие. А сделав неправильные выводы, можно выработать неверные превентивные шаги и ухудшить положение.

Однако если вдуматься, можно разработать универсальный стандарт логов, аварий, которого должны придерживаться все производители. В этом формате оборудование будет отправлять сообщения на Интеллектуальную Систему Контроля Работы. Получив кучу таких сообщений, Система принимается за их анализ. Далее она локализует проблему одним-двумя устройствами. Если необходимо, то потом по специальному протоколу Система Контроля обращается к узлам-подозреваемым и снимает данные.

В итоге она подготовит логичное сообщение, которое будет понятно без многочасовых анализов. Хронология аварий, затронутые сервисы, причина, решение, рекомендации.

Причём, в принципе, это можно делать в реальном времени.

Пример 1

Вот, например, был у меня такой запрос:

Внезапно упали все базовые станции, подключенные к магистральному коммутатору. Произошло примерно в 10:20, 03.03.2013. Всё, что мне нужно было сделать - получить доступ на этот коммутатор и проверить подробные логи, в которых я нашёл следующее:

> Mar 3 2013 10:15:03 %%01SHELL/5/CMDRECORD(l): Record command information. (Task=vt0, Ip=X.X.X.X, User=xyz, Command="**interface Eth-Trunk 3**") Mar 3 2013 10:20:13 %%01SHELL/5/CMDRECORD(l): Record command information. (Task=vt0, Ip=X.X.X.X, User=xyz, Command="**shutdown**")

Что было нужно для достижения результата:

1. Инженер звонит на горячую линию ТП и регистрирует запрос.
2. Сотрудники горячей линии передают запрос на группу.
3. Менеджер назначает ответственного инженера.
4. Инженер связывается с заказчиком и уточняет проблему, запрашивает логи, удалённый доступ.
5. В ходе анализа находит причину.

Как бы это могло выглядеть при наличии Интеллектуальной Системы Контроля Работы?

Система Контроля обнаружила недоступность целого сегмента за этим магистральным коммутатором. Тут же на неё стали приходить логи с других устройств, в том числе и с самого

коммутатора. Она проанализировала это и локализовала проблему именно на данном устройстве. Причём одно из сообщений - отключение интерфейса, другое, предваряющее его - выполнение команды **interface Eth-Trunk 3** пользователем xyz. И тут всё сложилось. Причём это всё в реальном времени. В панели аварий высветилось сообщение:

"*Пользователь Иван Иванов (ссылка на страницу с данными о пользователе, подтянутыми с RADIUS-сервера и кнопки связи с ним: телефон, корпоративный чат и тд.) авторизовался на устройстве (ссылка на параметры устройства с кнопкой для удалённого подключений к нему), в 10:20:13 03.03.2013 и выполнил команду по отключению интерфейса (ссылку на страницу с логами, где ключевые моменты выделены красным), после чего стали недоступны следующие узлы (ссылка на список)*".

Специалисты отдела мониторинга узнают о проблемах и её причине раньше, чем очухается первый пользователь.

Ну, разве это не прелестно?

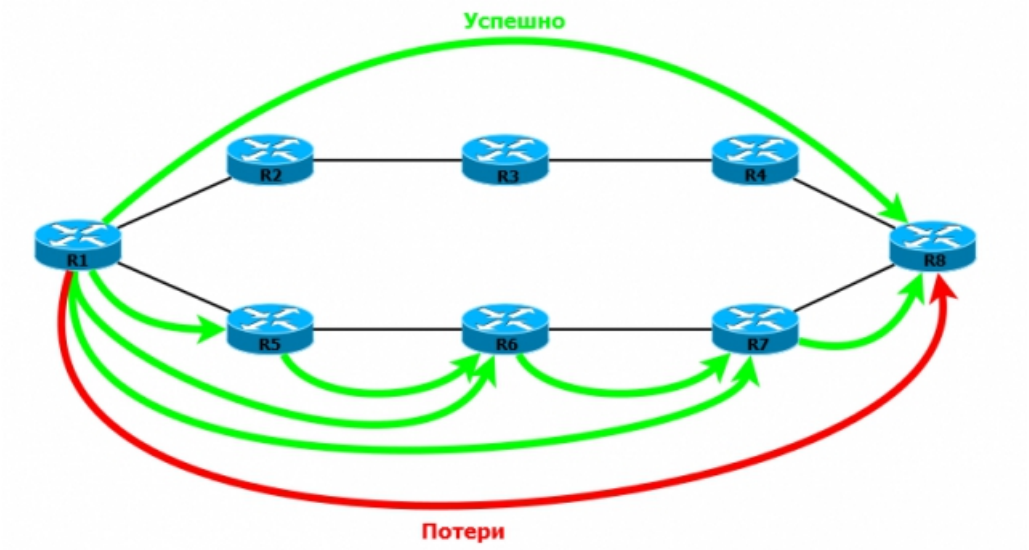
Разумеется, это очень простой пример, но даже он потребует огромной работы по стандартизации логов и разработку интеллекта Системы Контроля.

Есть здесь и контрпример - то, что видится сложным для машинного траблшутинга.

Пример 2

Например, такой запрос:

В такой сети теряются пакеты при пинге с R1 на R8 до другого. При этом все непосредственные линки работают без каких-либо ошибок или потерь.



Расскажу кратко ход решения. Подробно он был рассмотрен в моём блоге.

Во-первых, проверил, естественно, все линки point-to-point - всё отлично, во-вторых, проверил маршрутизацию - всё везде безо всяких проблем - FIB тоже в полном порядке.

В-третьих, тестировал большие/маленькие пакеты, пинги с разных направлений. Выяснилось, что чем больше размер пакетов, тем больше процент потерь. Проблема наблюдалась только с некоторых направлений. Причём никакой зависимости установить не удалось.

Тогда я начал собирать статистику со всех устройств со всех транзитных интерфейсов - сколько пакетов вошло, сколько вышло. Это помогло установить, что пакеты терялись на узле R6 - вошло 5 вышло 2. Пошла проверка ACL, политик, настроек QoS, статистику ошибок. Всё в порядке. Тут я сдался и эскалировал запрос более компетентным инженерам. В ходе суровой отладки в скрытых режимах выяснилось, что повреждено взаимодействие между линейной платой в 9-м слоту и коммутационной фабрикой в 1-м. Все пакеты, которые направлялись сетевым процессором на эту плату, просто отбрасывались. Достали плату - несколько контактов замято при неаккуратной установке.

То есть обнаружить неисправность платы в принципе можно, но как до этого дойти, исходя из наличия потерь на сети?

С другой стороны, правильный подход заключается в самопроверке оборудования. Не Система

Контроля должна инициировать проверку плат, оборудование само должно сообщать Системе о неисправности тех или иных своих элементов. Но об этом дальше.

2. Проблемы, которые имеются в данный момент, но не затрагивают сервисы

Переходим на тёмную сторону луны, которую не сканируют тысячи телескопов BFD. В логи часто сыпятся сообщения, которые непонятны даже после упорного гугления и чтения посвящённой оборудованию документации. И мне до сих пор непонятно почему. Почему нельзя здесь сделать адекватное описание, понятную формулировку? Кажется, что это, как у докторов - простому смертному не дано понять. Признаюсь вам, что и инженеру ТП они тоже иногда бывают непонятны.

Но даже если понятен смысл, часто непонятны причины. В примерах это будет видно.

Сюда же относятся проблемы, которые могут не появляться в логах: ошибки на интерфейсах, которые пока под слабой нагрузкой, и они не дают о себе знать, флаппинг интерфейсов или маршрутов на резервных линках, слишком лёгкие пароли, отсутствие ACL на VTY или на внешнем интерфейсе, большое количество широковещательных сообщений, поведение, похожее на атаки (обилие ARP, DHCP запросов), высокая утилизация ЦПУ каким-либо процессом, отсутствие чернодырых маршрутов (blackhole) при настроенной агрегации маршрутов и многое другое.

Причём если долго игнорировать такие предвестники, рано или поздно, они могут перерасти в серьёзные неприятности.

Однако я думаю, что оборудование самостоятельно или с помощью Интеллектуальной Системы Контроля Работы может отследить причины таких сообщений. Возможно, даже двум или более устройствам придётся "поговорить" друг с другом, чтобы найти корень болезни.

Ну, давайте несколько примеров, чтобы не быть голословными.

Пример 3

В логи посыпались сообщения в большом количестве:

> Feb 1 2013 05:40:26 SuperRouter %%01FIBSPT/3/ERROR(l):-Slot=7; There is error entry in TCAM. (TableType=[1], VrfIndex=[0], IpAddr=[x.x.x.x], VlanId=[0], ErrCode=[1])

Ну да, "Там у вас проблема в TCAM", и знание, что такое TCAM никак не поможет мне понять, что это значит. Разработчики подключились, что-то сделали и выдали ответ:

"*Данные сообщения были вызваны тем, что при изучении маршрутов не полностью обновилась таблица FIB. После этого возникла проблема с поиском префиксов в ней. Это информационные сообщения - никаких действий не требуется*".

Простите, если действий не требуется, на кой мне эти сообщения видеть? Неужели оборудование само не может продиагностировать эту проблему? Ну, то есть, как только появилась такая авария, запусти алгоритм полной проверки - найди причину. Это ведь всё можно формализовать. Нашёл причину - отошли на Систему Контроля человеческое сообщение информационного характера, чтобы инженеры были в курсе, что было такое дело: так, мол, и так: "Был сбой записи в память, в результате чего была сгенерирована авария. Проблема уже исправлена повторным обновлением FIB".

Пример 4

Внезапно начали появляться сообщения "*GigabitEthernet1/1/2 Received Pause Frames Exceeded Threshold*" на оборудовании. Также растут счётчики Pause Frames на интерфейсах.

Казалось бы, PAUSE FRAME - это известный механизм Flow Control в Ethernet, который говорит о перегрузках на интерфейсах - противоположное устройство сообщает этому, что оно шлёт слишком много трафика. Но статистика на интерфейсах говорит об утилизации в несколько процентов, пиковые значений даже близко не дотягивают до 50%. Более того, на обратной стороне не растёт счётчик Pause Frames.

Начинаем разбираться, и оказывается, что проблема вызвана переполнением какого-то внутреннего буфера на плате и есть специальный алгоритм, как проверить, что именно в этом дело. На передачу трафика это никак не влияет.

И тут снова возникает вопрос - а разве оборудование само не может провести эту проверку? Если подтвердится, то либо выслать информационное сообщение на Систему Контроля, либо вообще скрыть этот факт, если он действительно не влечёт за собой никаких последствий - только сделать запись в логи.

Разумеется, тут больше работы вендора - предусмотреть все такие ситуации и заставить своё оборудование заниматься самодиагностикой. Но разве не должен вендор стремиться к минимизации издержек на техподдержку, чтобы ей не приходилось заниматься такими случаями?

Как решать проблемы с ошибками на интерфейсах, флапингом и прочими вещами, когда вовлечено более чем одно устройство? Для этого нужен либо протокол взаимодействия, чтобы устройства могли определить, что не так с противоположной стороны, либо таким анализом должна заниматься Система Контроля.

Например, у ошибок на интерфейсах чётко определённый список причин. Ну, или во всяком случае его можно определить. Ну, вот на вскидку:

- Проблема с платой/интерфейсом (тут к предыдущим примерам - самоанализ состояния);
- Разные режимы скорости/дуплекса - это можно легко обнаружить и даже исправить;
- Плохое качество линии - находится методом отсекания. Система Контроля анализирует переданные/принятые данные, режимы работы и тд. на обеих сторонах. Она же, кстати, может переключить, например, трафик на другой пути/отключить проблемный интерфейс.

Вопросы, вроде флапинга маршрутов или MAC-адресов, тоже можно отследить при правильном подходе к стандартизации. Интеллектуальная Система Контроля Работы знает всю сеть, знает все MAC-адреса, таблицы коммутации и маршрутизации, проверяя одно устройство за другим, оно находит источник бед и выясняет причину.

Причём это явно значительно быстрее, чем такой же анализ, проводимый человеком, и это возможно делать в автоматическом режиме даже во время наличия проблемы. А ведь часто бывает, что инженер пытается разобраться в ситуации постфактум - уже всё случилось, сервисы лежали 15 минут, а теперь сеть в порядке и нужно на основе скудных логов восстановить хронологию событий и найти виновника, чтобы предотвратить рецидив. А компьютеру 15 минут за глаза, чтобы сделать это всё прямо в тот момент, когда на сети творится беспорядок.

Пример 5

Опять же приведём пример:

На сети периодически наблюдаются потери трафика.

Я приступил к анализу и увидел, что статистика OSPF показывает огромное количество полученных и удалённых маршрутов - счётчики перманентно растут - флаппинг? Да, флаппинг. Включаем отладку обновлений OSPF и выясняем, откуда валится спам. Раскручивая цепочку, находим, что в сети два маршрутизатора с одинаковыми Router ID. С учётом получения удалённого доступа решение у меня заняло около двух часов.

Сколько бы потребовалось Системе Контроля? Нисколько - она бы не допустила такую конфигурацию, отследив два одинаковых Router ID в одном домене OSPF ещё при настройке.

Если бы такие веерные обновления были вызваны флаппингом интерфейса, то Система Контроля явно узнала бы сначала об этом и сумела предвидеть/предотвратить проблемы с OSPF.

Здесь проблемы с поиском неисправности могут возникнуть в тот момент, когда мы упираемся в физическую невозможность проверки чего-либо. Например, состояния медной линии, как в случае выше, или радиоканала.

3. Аппаратные проблемы оборудования

Частично мы их уже обсудили выше. Речь о багах и выходящем из строя оборудовании.

Я бы разделил их на три типа:

- Внезапные
- Накопительные
- Потенциальные

Внезапные

Вышли из строя блоки питания, пробило плату, выгорел порт, повреждена шина, потерян сигнал синхронизации между слотами, внезапная перезагрузка итд. Иными словами всё то, что нельзя спрогнозировать.

Для таких случаев должна быть возможность контроля, что именно случилось. То есть на плате должен быть ещё какой-то независимый контур контроля, который даст знать управляшкам, что с

ней не так.

В случае перезагрузки платы/устройства оборудование в силах само установить причину. Даже в случае самого наивнезапнейшего сбоя управляющие платы должны иметь возможность знать, что спровоцировало перезагрузку. Возможно, это будет даже не управляющая плата, а какой-то внесистемный модуль, как сейчас на серверах бывает.

То же самое касается ситуации, когда плата, например, не может загрузиться. Внесистемные модули мониторинга на проблемной плате и на шасси/управляющей плате должны суметь друг другу сообщить причину.

Что здесь может быть сложно отследить - это если текущее ПО не поддерживает данный тип плат, к примеру. Но на этот случай существует Интеллектуальная Система Контроля Работы. Она получит сообщение о неопознанной плате с устройства, запросит у управляющих модулей тип платы, версию ПО она уже знает, далее обращается на сайт производителя и проверяет список поддерживаемых плат. Видит что да, все критерии бьют и выводит об этом сообщение. И это вместо регистрации трабл-тикетов и долгих поисков вручную. После этого инженер уже может смело наезжать на менеджеров, зачем они купили платы, не поддерживаемые программным обеспечением.

Накопительные

Тут могут быть разные возможные варианты - накапливающиеся ошибки на чипах, утечки памяти, количество потерянных пакетов синхронизации и т.д.

Ни у кого не будет сомнений, что такие вещи либо отслеживаются, либо их можно отследить. Да, есть вероятность, что их не получится предотвратить, но Система Контроля или само оборудование может отслеживать тенденцию и опасные значения ключевых элементов системы.

Пример 6

Пример из реальной практики:

Внезапная перезагрузка маршрутизатора.

Устройство хранило три таблицы Full-view. Было использовано около 80% от 1 ГБ оперативной памяти. Далее в какой-то момент начался флаппинг BGP-маршрутов, что потребовало дополнительного пространства оперативной памяти. После определённого порога в качестве защитного действия выполнилась автоматическая перезагрузка маршрутизатора. Нам пришлось путём сложных команд и логов на различном оборудовании восстанавливать историю аварии, чем была заполнена память.

Система Контроля в реальном времени могла бы отследить приближение к критическому значению размера использованной оперативки. Да и более того, она бы гораздо раньше информировала инженеров о проблемах с BGP.

Потенциальные

Это тёмные попутчики - никак себя не проявляют, просто ждут своего часа.

Все вы знаете, что у вендоров есть список известных проблем. Он всегда довольно подробный: тип оборудования, условия возникновения, влияние, способ решения.

Система Контроля может загружать в свою БД эти данные (при условии их открытости вендором, разумеется) и предупреждать такие проблемы - либо просто информируя пользователя об этом, либо проявляя какие-то активные действия, в виде, например, обновления ПО или применения скрипта.

Возможно, также создание некой открытой базы системных проблем, например, уязвимость каких-либо протоколов, которые будут отслеживаться Системой Контроля.

Например, совсем недавно была обнаружена уязвимость OSPF (представлен на конференции Blackhat 2013), некоторые вендоры её быстро закрыли патчами, но кто из операторов на это среагировал? А кто из крупного интерпрайза (банки, предприятия)? А кто из миллионов мелких?

4. Потенциальные проблемы ПО

Тут всё просто. Либо у вендора есть хорошая база ПО, патчей и их описания со списком доступного функционала и решённых проблем, либо нет. Естественно, если нет, нам надо, чтобы да.

Система Контроля просто следит за всеми обновлениями и при необходимости загружает их и устанавливает.

5. Некорректная конфигурация

Думаю, это самая сложная часть. Формализовать правила конфигурации, означает создать универсальный язык взаимодействия между Системой Контроля и оборудованием. Ну, нельзя же пытаться разгрести на одном сервере разрозненные данные от Juniper, Cisco, ZTE и Dlink. Нельзя создавать парсер, который будет подстраиваться под данные с разных устройств.

То есть будет необходима стандартизация как минимум хранения конфигурации и передачи её на Систему Контроля.

Как это видится мне: должен быть блок описания возможностей системы: что за тип (коммутатор, маршрутизатор, файрвол итд.), функционал (OSPF, MPLS, BGP). Дальше должны быть секции собственно конфигурации. Такая структура должна поддерживаться любым оборудованием от коммутатора доступа до VoIP шлюза в IMS-ядре.

Тогда с лёгкостью можно находить разнообразные неточности: неконсистентные настройки параметров на оппозитных устройствах (например, дискриминаторы BFD, уровни сетей IS-IS, соседи BGP, IP-адреса), совпадающие Router-ID, неактивированный PIM между двумя мультикастовыми маршрутизаторами итд.

Но, положа руку на сердце, - уже это вещи нетривиальные и реализуемые только должной стандартизации топологий или формализованного LLD (Low Level Design).

Но есть также пласт задач/проблем, которые видятся мне неразрешимыми на данный момент без создания какого-то примитивного ИИ.

Пример 6

Чтобы не ходить далеко за примером, возьмём уже описанную мной проблему.

Кратко: в MPLS TE туннеле задан обязательный Next-Hop в primary Explicit path. Резервного нет. В случае если Next-hop доступен, LSP обязан строиться через него, даже если ему для этого придётся идти через всю сеть по периметру. При этом трафик может передаваться нормально. Как тут понять, что что-то не так с конфигурацией?

Или настройка стыка двух провайдеров (даже простая IP-связность без CSC или Option X). Система Контроля по моему видению ограничена только нашей AS, максимум сетью компании, включающей несколько AS. То есть отследить здесь неработоспособность чего-то вообще не представляется возможным. *Только если Системы Контроля не смогут договориться между собой*.

Интеллектуальная Система Контроля Работы

Данной теме будет посвящена отдельная статья. Пока вкратце хочу рассказать о том, какими возможностями она должна обладать.

Мне довольно часто задают вопрос о том, как консолидировать всё в одном месте: схемы сетей, коммутации, мониторинг, бэкапы. Как поддерживать актуальное состояние L1/L2/L3 схем одновременно? И я не знаю, что на это ответить. Нет, что-то, конечно, в этой области реализовано. Но во всех существующих решениях есть какие-то недостатки или ограничения.

В итоге схемы в Visio (одна для физического соединения, другая для логической L2 сети, третья описывает взаимодействие на третьем уровне), для наблюдений за линией и оповещения - Nagios, для контроля утилизации интерфейсов - Cacti, SysLog для отслеживания аварий и ещё скрипты для резервного копирования конфигураций. Да, в коммерческих продуктах некоторые функции бывают реализованы одновременно. Но, согласитесь, далеко не все.

Да, мы все к этому привыкли. Мы знаем, на каком IP-адресе, какой сервер, и за что он отвечает. Мы мастерски выравниваем по линейке в Visio маршрутизаторы на новых узлах, чтобы всё было красиво и знаем, в какую из ста экселевских форм надо вписать номера интерфейсов интерконнекта.

Но, чем крупнее сеть, тем более яркой и динамичной жизнью она живёт и тем больше сил нужно на актуализацию документации и мониторинг.

Однако что здесь есть принципиально нереализуемого? Допустим, протоколы маршрутизации (вроде OSPF или IS-IS) знают L3-топологию, LLDP или некий его заменитель могут помочь составить L1/L2 топологию. Причём там уже есть и номера интерфейсов и список VLAN'ов и информация о программно-аппаратной начинке. Задача инженера потом - только расставить их на карте в удобном виде.

На любое устройство можно зайти, посмотреть аварии, логи, статусы интерфейсов, сервисов,

историю проблем, замены плат. На схеме отражены абсолютно все линии, вплоть до абонентских терминалов, подписаны все интерфейсы, сабинтерфейсы.

Из Системы Контроля можно легко перейти в консоль конфигурации или запросить последние бэкапы файлов с карты памяти устройства.

Она же следит за всеми обновлениями, готовит формы для замены неисправных элементов, отчёты любых типов и т.д.

Но самая важная и самая сложная в реализации функция Интеллектуальной Системы Контроля Работы - анализ проблем на сети. Система Контроля - это мозг всей сети, получающий импульсы от нервных окончаний и анализирующий их. Как мозг контролирует все процессы организма, начиная от сердцебиения и заканчивая решением дифференциальных уравнений, так и Система Контроля знает абсолютно всё, что происходит на сети: аппаратные неисправности, потери пакетов на сегменте, высокие задержки и джиттеры, ведущие к плохому качеству сервиса, отключение интерфейсов, проблемы протоколов. В более глобальном смысле она может отслеживать тенденции и предлагать варианты оптимизации, расширения сети.

Я не готов пока сказать, как это может быть реализовано, но сам факт принципиальной возможности отрицать нельзя.

В заключение я бы хотел предупредить возможные крайности в комментариях.

Во-первых, я предлагаю не какое-то приложение или их комплекс, которое будет универсальным парсером и анализатором и станет "фишкой" какой-то конкретной компании. Я говорю о системном подходе к автоматизации процессов в сетях связи: создание стандартов для конфигурации, аварий, логов, разработка новых протоколов взаимодействия Устройство-Устройство, Устройство-Система Контроля.

То есть это некий базис, опираясь на который, многие компании смогут делать универсальные системы гораздо лучшего качества и функционала, чем существуют сейчас.

Во-вторых, да я знаю, что сейчас есть, так или иначе, комплексы, похожие на те, что я описал, но либо они моновендорные, либо обладают минимальным набором функций.

В-третьих, это концепция хоть и напоминает SDN, однако подход принципиально отличается. В "бесчеловечных" сетях каждое устройство остаётся индивидуальностью, самостоятельной единицей, сеть вполне может обойтись и без Системы Контроля и продолжать функционировать в том виде, в котором оно сейчас. Система предоставляет лишь интеллектуальное дополнение с огромным функционалом. Таким образом, здесь не встаёт вопрос узкого места, как в SDN, когда имеет значение стабильность и ширина каналов до самого "мозга" и, собственно, его наличие.

Коллеги, я не претендую на полноту раскрытия проблематики. Наверняка, я не усмотрел ещё множество узких мест. Но это лишь мои размышления.

Данная статья остро нуждается в критике и предложениях. Впереди будет ещё серия публикаций, посвящённых различным аспектам таких "бесчеловечных" сетей: собственно Интеллектуальная Система Контроля Работы, новые протоколы взаимодействия, подходы к решению проблем и автоматической настройке.