Управляется программно: идеальный маршрутизатор

В реальном мире не бывает ничего идеального. Никому и никогда еще не удалось создать идеального зеркала, которое отражает 100% фотонов. В природе не встречается ни абсолютно упругого, ни абсолютно черного тела. Но очень хотелось бы. Хотя бы с целью определения модели идеальных сетей передачи данных, которые, конечно же, имеют идеальные операторы связи.

И в данной статье совершенно нет цели назвать все маршрутизаторы (кроме марки "XXX") неидеальными. Но есть желание разобраться с вероятностными параметрами сетей будущего, какими они будут, как их строить и как этими сетями управлять.

А по моему глубокому (авторскому) убеждению, сети передачи данных стоят на пороге глобальных изменений, связанных с экономическими и технологическими сдвигами в обществе. Причем, тренды эти глобальны и никак не зависят от субъективизма автора и/или читателя. Тренды просто есть и они объективны, ибо мир меняется (ох, как часто я эту фразу пишу - простите) заметно, неуклонно и необратимо. Даже если кто-то этого и не хочет.

Сдвиг связан с двумя причинами, и вряд ли кто-то сможет поспорить с этими утверждениями. Назовем их "два И".

Первая "И" - инновации. Вторая "И" - инвестиции.

Инновации заключаются в том, что постоянно придумывается что-то новое и неизвестное ранее. Причем, не с целью "придумать новое", а как раз с целью как можно больше продать и заработать. Это нормально. Человечество последние пять тысяч лет только так и живет. Придумываются новые бизнес-схемы, технологии, разрабатываются новые микросхемы и алгоритмы. И именно для того, чтобы удовлетворить вторую "И" - инвестиции.

Технологии перманентно получают свою долю мирового производства благ. С целью сделать благ еще больше. И дешевле. Или, по меньшей мере, больше благ за те же деньги.

Самым каноничным примером, который можно привести - микропроцессоры и "Закон Мура". Это даже и не закон вовсе, в научном его понимании. Это просто эмпирическое наблюдение, которое многократно пытались опровергнуть или "перезагрузить" с предсказаниями "достигли физического предела". Но всегда появляется что-то новое и все равно каждые два года вычислительные мощности (или количество логических элементов на чипе) удваивается.

Интернет-отрасль, которая надо отметить также находится под сильнейшим влиянием "закона Мура", подвержена "двум И". Здесь есть постоянно растущие и инновации и инвестиции. Инновации обеспечивают инвестиции, которые ведут к следующим инновациям, обесценивая предыдущие инвестиции.

Так и живем.

Вот давайте проведем небольшой экскурс исторический:

Это первый в Мире маршрутизатор - BBN Interface Message Processor (IMP). Разрабатывался для ARPANet. Работал в КалТехе, начиная с 1969 года. Внутри был ~~процессор~~ микрокомпьютер Honeywell 516 с памятью на "6,000 слов" - тогда память измеряли "словами" - что соответствует примерно 12 килобайт. Стоил этот шкафчик $82,200.

Вот так с открытой дверцей:

Музейный экспонат.

Или вот, например, один из первых аппаратов Cisco ASM/2-32EM:

Это через 20 лет - 1987 год.

Технических характеристик и цен не нашел, но именно такой прибор обслуживал суперкомпьютерный центр в CERN.

И вот теперь, в 2016-ом - прибор, который хотелось бы назвать "идеальным":

Это OnHub от Google. Бытовой маршрутизатор с кучей сетевых функций, двумя радиодиапазонами доступа и прицелом на "Интернет вещей".

И понятно, что это устройства разного класса, но функционал "вычислительного устройства, пересылающего пакеты данных из разных сегментов сети" у них одинаков.

Но вернемся к тенденциям и "двум И".

А в сетестроительстве происходит следующее: интернет становится повсеместным, непрерывным и бесшовным. Это значит, что производительность сетей растет (ну, не было пока случаев, чтоб трафик не рос) и, что логично, количество подключаемых устройств растет так же. Значит, нужно больше маршрутизаторов. Еще больше! Маршрутизаторы становятся дешевле, массово продающимися, повсеместными.

И тут возникает вопрос по управлению этой массой довольно сложных телекоммуникационных приборов. И даже не управлению, а, как сейчас модно говорить - оркестрацией.

Давайте пофантазируем (хотя, какая там фантазия - это фактически уже обыденность) относительно "сетей будущего" и "всеобщей интернетизации людей и вещей". У вас есть ряд устройств, которые подключаются к Сети каким-либо интерфейсом - радио (Wi-Fi), ethernet-патчкордом или 3G/LTE. Впрочем, патчкорд, наверное, нужно исключить из обзора - только радио сейчас интересует продвинутых консьюмеров. Возиться с проводами всем лениво и дорого - с оговоркой, что если это не собственно операторы связи.

Пользователю неважно, как IP-связность придет к нему на смартфон, планшет или Amazon Dash Button. Важно только цена и ценность этой связности.

Было бы совсем прекрасно (и это вовсе не фантастика давно), чтобы операторские сети просто подключали устройства к Сети, учитывали эти соединения, и в конце расчетного периода выставляли соответствующий счет. Ubiquitous Network. Вы просто в Сети.

Но для этого необходимо обеспечить:

1. Узнавание устройства пользователя, или этот самый AAA (Authentication, Authorization, Accounting).
2. Взаимодействие всех операторских устройств (собственно, маршрутизаторов), хотя бы для получения данных для пункта 1.
3. Адекватный, в смысле цены-качества, инструментарий наблюдения-мониторинга за устройствами, средства диагностики и желательно дистанционной починки.
4. И все это великолепие должно стоить как можно дешевле. Потому что а) таких устройств очень много, и б) устройства находятся повсеместно.

Мировое телекоммуникационное сообщество уже занимается созданием таких сетей. Например, китайские телекомщики уверенно и целенаправленно пилят ~~гири~~ концепцию C-RAN - Cloud Radio Access Network.

Американская концепция называется DAS - Distributed Antenna Systems. Почти то же самое, что и китайский C-RAN, только чуть менее универсальное.

Опять же Facebook пилит свои устройства в прожекте OpenCellular с целью дать доступ к самому себе как можно большему числу народа. Вот такие красивые устройства делают:

Гугловский OnHub мы уже упоминали…

Все эти тренды связывает одно: общее централизованное управление и мониторинг.

Остается только понять, как это будет устроено...

Для этого вернемся на грешную землю и реальность данную нам в ощущениях, и вспомним базовые экономические принципы строительства и эксплуатации сетей передачи данных.

Где экономика выражается вот таким уравнением:

S = CAPEX + N * OPEX

, где:

S - общая сумма затрат,

CAPEX - капитальные затраты, связанные с закупкой оборудования и его монтажом,

OPEX - операционные затраты, связанные с обслуживанием оборудования,

N - количество временных периодов. Например, месяцев или лет - кому как удобнее.

Поскольку формула очень простая и понятная, не углубляясь в особенности национального бухгалтерского учета, можно сразу понять, что чем меньше затрат как капитальных, так и операционных, тем дешевле собственно сеть. И тем выше прибыль, которую мы получим, если из всех доходов вычтем "S".

То есть, нужно сокращать расходы путем более дешевого оборудования. И его обслуживание.

И это вполне достижимая задача, если оборудование у нас стандартизировано и тиражно (основной закон массового производства), а процессы OSS отлажены и автоматизированы.

На пути у нас, разумеется, будут разложены специальные грабли для набития шишек, но в этом-то и заключается поступательное развитие прогресса - на ошибках учатся.

Например, для целей автоматизации процессов эксплуатации давным-давно была изобретена Operations Support Systems (На русском это звучит немного проще - "система эксплуатационной поддержки") и множество протоколов. Да вот хотя бы Simple Network Management Protocol - хорошая же штука. Уже более 15 лет на всех устройствах мира.

Или развитие оного - TR-069. Отличная вещь, которую более десяти лет внедряют крупнейшие операторы, да никак внедрить не могут.

Суть протоколов заключается в создании некой инструментальной среды для управления действительно большим количеством сетевых устройств и вообще.

А вообще, конечно, обслуживанием оборудования занимаются люди. А люди ~~грешны~~ делают ошибки. И это ведет к потерям финансовым и временным. Например:

7 смертных грехов сетевого администратора

Люди делают ошибки, которые приводят к последствиям. Иногда ошибки имеют катастрофические результаты. Иногда просто несколько часов простоя. Чаще их просто не замечают - подумаешь, отвалился сегмент на несколько сотен абонентов. Остальные-то триста тысяч - работают!

Тем не менее, сетевые регламенты подобны боевым уставам. Регламенты написаны кровавыми слезами сейлзов, у которых ушли клиенты и потом техподдержки третьего уровня.

Очень коротко перечислим греховные действия/бездействия ответственных лиц. Есть мнение, что список неполный и может быть продолжен.

Отсутствие документации и нежелание делиться информацией

Обычно не хватает времени и, главное, желания на написание многобукв. И так же все понятно. И вообще - пусть пишет Донцова и ее литературные негры:

Понимание масштаба бедствия придет только после смены/болезни/смерти администратора, который ушел, и унес тайну конфигураций с собой. Зато администратор ощущает себя незаменимым и трудноувольняемым персонажем.

Особенно наличие отсутствия (обожаю этот оборот) документации доставляет удовольствий, если в системе происходят постоянные недокументированные изменения, которых потом никто не помнит, включая авторов изменений.

Отсутствие резервных копий

Во-первых, резервные копии занимают место на дисках... Зачем тратить на создание резерва ценное время - и так все работает. Ровно до того момента, как все пойдет не так. А момент настанет всенепременно.

Наплевательское отношение к авторским правам и лицензионной политике

Никто не любит платить "лишнее". Зачем платить, если можно скачать и установить. До момента, когда производитель софта найдет критическую ошибку, которую не обязан исправлять на контрафактных копиях.

Отсутствие внятной политики безопасности

Политика - вообще штука тонкая. Особенно, когда дело касается дилемм класса "ограниченные ресурсы против гипотетических угроз". Тем более, если для этого опять нужно что-то писать...

Отсутствие средств организации своего труда или их переизбыток

Крайности "и так сойдет" против "не стану делать это руками". Существует огромное количество историй про то, как откладывается запуск новых сервисов по причине "вот тут нужно написать еще один скрипт, потому что ... (следует минимум шестнадцать причин автоматизации процедуры, которая гипотетически может когда-то случиться). Или обратная ситуация, когда админ приезжает в выходные на рабочее место, чтобы перезагрузить сервер, потому что скриптам WatchDog "он не доверяет".

Мораль одна - все хорошо в меру.

Отсутствие выработанной стратегии развития

Любая система, которая не развивается, в конечном итоге скатывается к деградации. Система просто ломается или теряет преимущества.

С другой стороны, если ты не знаешь, куда идешь, то нечему удивляться, что ты оказался именно здесь.

Впрочем, стратегические стратегии прямо связаны с пунктом про документацию. Потому что писать эксплуатационную документацию без целей верхнего (стратегического) уровня занятие не менее бессмысленное, чем книжки Донцовой.

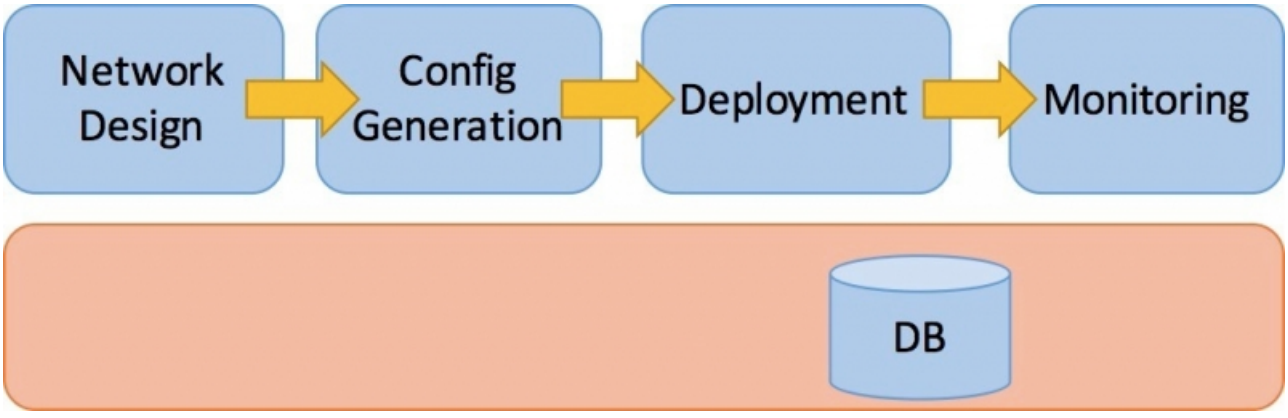
Забыть пароль от консоли

Думаю, тут очевидно.

Новейшая история дизайна сетей

Есть ли ~~жизнь на Марсе~~ отклик от этих букв в сердцах настоящих и будущих системных администраторов - я не знаю. Перечисленные выше банальности были всего лишь прелюдией к тому, что я сейчас попытаюсь донести до читателей. И наверняка тоже банальность. Но, как вы помните, есть концепция "двух И", где все новое ставит перед собой только одну цель: "инновации для инвестиций". Сократить издержки путем вложения денег.

И вот перед вами более другая концепция дизайна, которая выглядит как-то так:



То есть схема, когда дизайн, конфигурация и эксплуатация сетей сами по себе стратегия развития, документация, фиксация изменений и управление версиями ПО. Мечта сетевого инженера-администратора.

Суть концепции в одно предложение: "Все записывается в базу данных". А раз это база данных, следовательно, мы можем легко и непринужденно сделать выборку по параметрам и визуализировать нашу сеть.

А поскольку каждое изменение в Сети - это тоже запись в БД, то отследить историю изменений также несложно.

Подробнее по этапам.

Проектирование сети: первый, я бы даже сказал нулевой, этап жизненного цикла сей концепции управления сетями - Network Design. Это очень просто: пишется та самая стратегическая стратегия с целями и задачами. Ответ на главный вопрос: "Зачем мы вообще это делаем". И потом потихоньку описываем ее детально: где и какие у нас узлы связи, топология сети, физика каналов, количество стоек и оборудование в них. Объекты, интерфейсы и IP-адресация. Обычная работа проектировщиков.

Создание конфигураций: более детальное описание проекта сети. Точнее активных сетевых элементов в виде их конкретной описательной модели: что как и куда пересылается. Разумеется, что в спроектированной сети есть очень стандартные элементы, которые отличаются только ID и/или конкретными адресами src/dst. И, разумеется, есть не очень стандартные элементы. Но теперь все конфигурации телекоммуникационных приборов у вас не только документированы, но и описаны в практическом смысле - накатил конфиг, да и шабаш!

Прелесть заключается в том, что все изменения также логируются в БД. Например, всегда можно сделать Ctrl+Z. И ко всему процессы создания конфигураций довольно неплохо автоматизируются и масштабируются.

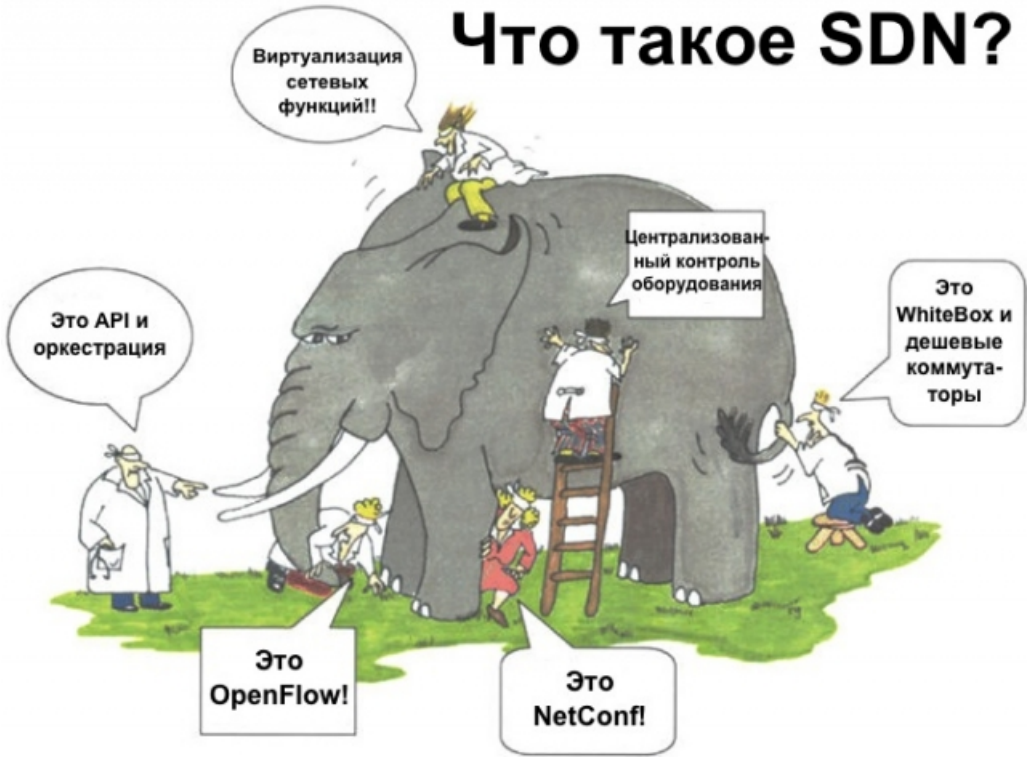
Развертывание сети: после создания настроек-конфигураций (которые, напомню, сами по себе документация) следует команда Deploy. И все работает!

На самом деле, конечно, не так все просто - должны существовать некоторые процедуры тестирования и верификации конфигураций. Отладка. Но зато у вас есть функция "отката", и это многого стоит. Процедура развертывания должна и обязана быть последовательной и иметь продуманные фазы для уменьшения риска фатальных сбоев.

Мониторинг и эксплуатация: Последний штрих и вишенка на торте: каждый сетевой компонент, находящийся в эксплуатации, должен перманентно отчитываться о своем состоянии. Все ли устройства онлайн? Правильно ли ходят пакеты? Нет ли перегрузки некоторых элементов по десятку технологических параметров (загрузка ЦП, например, или наличие свободной памяти). Тот, кто не измеряет объект управления - не управляет им. А если измеряет, то имеет инструмент по исправлению ошибок, включая на переделку всего, вплоть до сетевого дизайна логического, и если потребуется - физического.

Идея кажется прикольной. Осталось понять, как же нам реализовать такую замечательную систему, желательно за невоенные деньги и без шаманизма command line interface. И без колдунов.

Напомню вот такую картинку:



Она замечательна тем, что очень наглядно объясняет, где будут ~~теряться~~ тратиться инвестиции провайдеров в ближайшем будущем. ~~В поисках слона.~~

Одна из строчек в составе SDN пока не очень понятная аббревиатура NETCONF.

Сейчас попробую наглядно объяснить что это.

RFC-3535

История началась довольно давно. Аж в 2001 году, когда сразу на нескольких мероприятиях (конкретно: NANOG-22, RIPE-40, LISA-XV и IETF-52) операторы поставили вопрос ребром разработчикам протоколов, оборудования и, разумеется, сетевого ПО. Боль эксплуатационщиков заключалась в том, что хотя все кажется хорошо и прекрасно (год, напомню, 2001-ый), но что-то вот с управлением сетевыми устройствами не очень. В том смысле, что количество сетевых элементов растет и управлять ими все сложнее и сложнее.

В результате обсуждений было принято "Рабочее предложение", оно же RFC за номером 3535: "Обзор семинара IAB 2002 года по управлению сетями".

В данном документе были зафиксированы результаты диалога между операторами сетей и разработчиками протоколов "об углубленной работе IETF над управлением сетями в будущем". Всего было выявлено 14 требований операторов.

Главное требование - простота использования любой новой системы управления сетью. Простота использования подразумевает возможность управлять сетью, а не только устройством в сети, а также требует четкого разграничения между данными о конфигурации, операциях и статистике. Откуда уже вышли и прочие требование — поэтапная конфигурация, проверка конфигурации перед ее применением и откат к предыдущей конфигурации в случае отказа.

Документ этот, по сути, описывает, что существующие системы управления никуда не годятся и надо делать что-то более лучшее.

Вот, например, пункты о том, что SNMP не так уж хорош, как кажется производителям оборудования:

1. Низкая производительность (напомню, что в протоколе SNMP за одно обращение можно выполнить одну операцию над устройством).
2. Концепция MIB (Management Information Base- база управляющей информации) не выдерживает критики, поскольку производители сотворили там полнейший бардак.
3. Из MIB вообще нельзя понять, что там происходит с устройством. Ибо на низком уровне и ~~трезвым~~ человеком не читается.
4. При этом собственно реализация MIB сильно избыточна.
5. И практически непредсказуема в использовании.
6. Программирование операций SNMP слишком трудоемкая задача.

...

И еще много претензий. Особенно, связанных с использованием UDP-транспорта, который не очень-то и подходит для бизнес-критичных целей управления оборудованием.

В общем, SNMP не так хорош, как о нем рассказывают.

Кроме того, в документе есть размышления о других способах управления, и, само собой, об их достоинствах и недостатках. Например, о Common Information Model (CIM) и Common Open Policy Service (COPS) [RFC2748]. Но в конечном итоге участники воркшопа склонились, что на текущей фазе развития науки и сетевого оборудования наиболее удобна схема управления посредством CLI (TELNET/SSH).

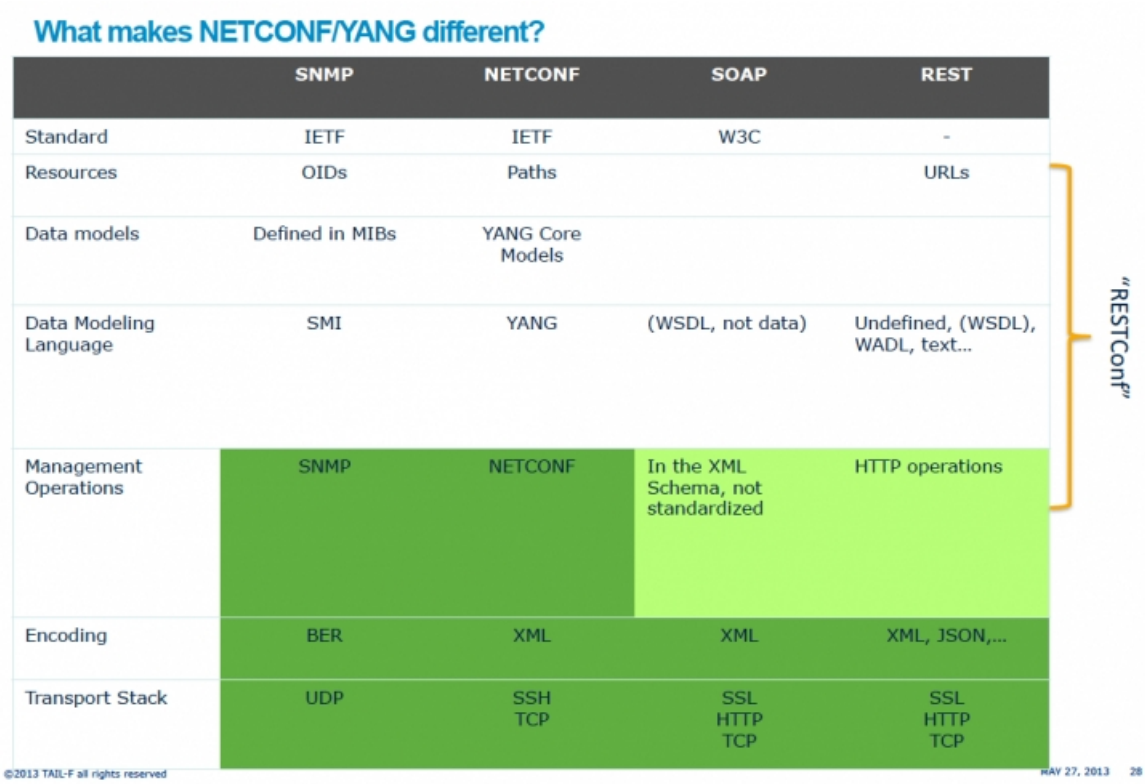
Что, в общем и целом, происходит и по сию пору. Есть исследования, что до 70% рынка управления телеком-системами базируется на технологии CLI scripting.

И это при наличии уже третьей версии SNMP и присутствии на рынке систем с прочими способами управления устройствами.

NETCONF

И только к 2011 году сообщество пришло к созданию более лучшей модели, которую достаточно полно описали в RFC 6241. Новый протокол называли Network Configuration Protocol (NETCONF) - "Сетевой протокол конфигурации".

Отличия от SNMP (плюс еще пара рассматриваемых способов управления) приведены в такой вот табличке:



Если очень коротко (подробнее, конечно же, надо читать первоисточник), то суть NETCONF заключается в том, что:

1. Транспортом для этого прикладного протокола стал TCP. Точнее SSH, как способ установления коннекта вообще.
2. Процедуры более высокого уровня описываются на базе "Удаленного вызова процедур" - RPC.
3. Собственно базовые операции NETCONF (их не очень много - типа <get-config> или <edit-config>).
4. Структура данных описывается на XML-подобном человеко-ориентированном языке, разработанного специально для этих целей. Все представления текущего состояния устройства, конфигурации, оповещения и операции описываются на языке YANG (подмножество YAML) RFC 6020/

Вот такая схема:

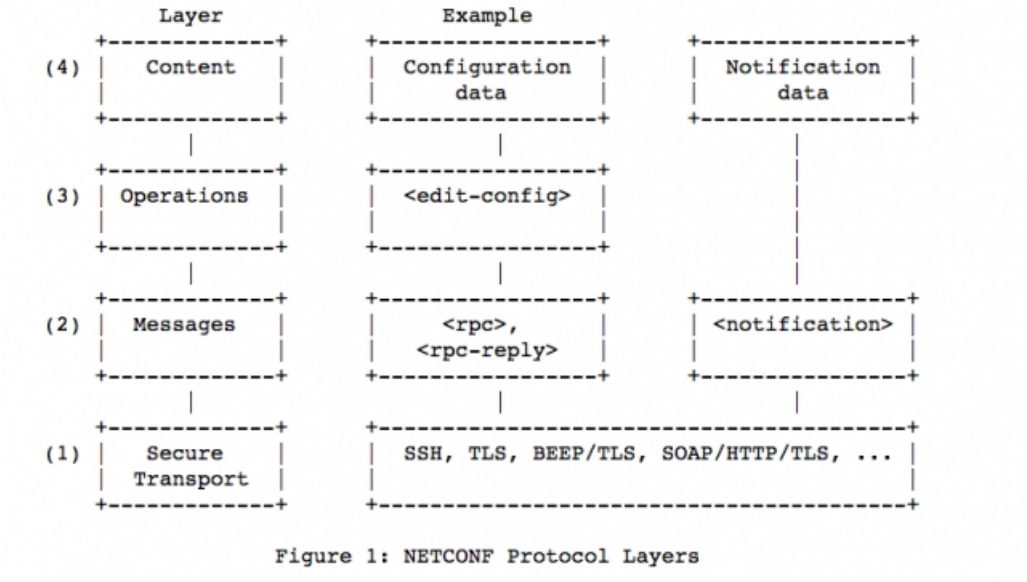


Figure 1: NETCONF Protocol Layers

Она предполагает довольно простой и понятный способ "общения с прибором":

1. Сначала устанавливается секурное соединение по SSH.
2. Формируется RPC-процедура сообщения.
3. Указывается одна или несколько операций из исчерпывающего списка:
   - <get-config> - прочесть текущую конфигурацию
   - <edit-config> - отредактировать конфигурацию
   - <copy-config> - скопировать конфигурацию
   - <delete-config> - удалить конфигурацию
   - <lock>/<unlock> - заблокировать/разблокировать устройство (например, на исполнение функций, пока мы правим конфиг)
   - <get> - взять произвольный параметр, например, для целей мониторинга
   - <close-session>/<kill-session> - штатное или принудительное завершение NETCONF-сессии.
4. Отправляются/принимаются данные в виде, который может прочесть человек, или обработан машиной.

Пожалуй, стоит еще упомянуть о столбце REST в табличке. Это еще не утвержденный протокол RESTCONF, который является развитием NETCONF. По сути, это тоже самое, только вместо транспорта в связке TCP+SSH используются обычные механизмы запросов HTTP. То есть, разработка управления устройствами будет напоминать разработку web-приложений. Да, вот как сайты сейчас разрабатываются - аналогично будут разрабатываться и системы управления сетями. По той же технологии из ~~желудей и спичек~~ ссылок и веб-интерфейсов.

Идеальный маршрутизатор

Ох, продравшись через информационный ворох, наконец, приходим к тому, ради чего писалась эта статья. К идеальному маршрутизатору. И очевидно, что идеальным он станет, если будет управляться программно и удаленно. Напомню классификацию про "тяжелую маршруторизационную систему" и оговорюсь, что программно должны управляться все сетевые элементы без исключения и на уровне ядра, и на уровне агрегации и особенно на уровне доступа.

Все активные сетевые элементы должны быть (в идеале! не завтра после обеда! - так и быть, дам время на модернизацию и получение кредитов) включены в систему управления, мониторинга и конфигурации оператора. Чем лучше эта система работает, тем качественнее сеть и тем больше ее будут любить абоненты.

Но к идеальному ~~сферическому в вакууме~~ маршрутизатору на уровне доступа у меня есть особые требования:

1. Очень дешевый. Настолько дешевый, чтобы оператор мог позвоить себе его дарить абонентам и монтажить во всех локациях, где оператор в принципе может найти абонента.
2. Достаточно мощный для выполнения поставленных прибору задач - пересылке пакетов по маршруту src/dst для достаточного количества абонентов. Не больше, но и не меньше.
3. Уметь использовать NETCONF и/или RESTCONF. Обязательное требование, иначе прибор неуправляем.
4. Иметь популярные интерфейсы соединения со стороны uplink - например, медный ethernet и/или SFP-гнездышко. Возможно радиоинтерфейс, который можно использовать как главный или

резервный - 3G/LTE или вообще 5G в том виде, в каком они появятся в коммерческой эксплуатации рано или поздно.
5. Иметь популярные интерфейсы со стороны абонента - Wi-Fi в обоих нелицензируемых диапазонах обязательно. Медные Ethernet-кабели желательно, но не обязательно. Но USB для подключения пользовательских модулей должен быть.

   Пожалуй, вот так. Но добавлю один пункт "на вырост":

6. Поскольку грядет бум IoT, то было бы неплохо иметь на борту интерфейсы для организации коннекта "с вещами". Сейчас победившего стандарта еще нет. Возможно, он появится в будущем. Вот для этого и нужен как минимум USB-интерфейс из пункта 5. Но если у маршрутизатора есть потенция принимать дополнительные модули, то он становится точно идеальным. В потенциале это следующие интерфейсы: 6LoWPAN или какая-то версия Bluetooth с низким энергопотреблением. Возможно, что-то другое. Но главное - модульная расширяемость.

Где деньги оператора

В завершении напишу кейс, где эти самые идеальные маршрутизаторы будут использоваться самыми обычными провайдерами. Обычными - потому что те провайдеры, которые не будут использовать идеальных маршрутизаторов, станут обанкротившимися провайдерами. И это не шутка.

Потому что Рей Курцвейл уже к 2019 году вангует наличие компьютерно-коммуникационных чипов буквально везде, от специальных приборов и ювелирных изделий, до бытовых приборов, вплоть до мусорного ведра. То есть три года, буквально, осталось. И почти все коммуникации будут проходить без проводов - по радио в основном. Хотя, может быть и другие физические среды - свет, звук. Или другие волны. Но таки более реалистично, что радио. Впрочем, неважно.

Важно, как будет использоваться инфраструктура, то есть пользовательский опыт.

Очень возможно, что это будет выглядеть так:

Пользователь покупает "доступ в интернет" для некоторого количества устройств с заданными профилями. Не думаю, что исчезнет "обычный доступ" для смартфонов, планшетов, ноутбуков - это конечно все останется. Просто доступ нужно будет покупать для всех устройств отдельно (кстати, в России этому посодействует "закон Яровой" или его производная - всех нужно будет идентифицировать по паспорту). Но появятся профили, аналогов которым пока нет - подключение IoT-устройств. Например, видеокамер с обратной связью, каких-то датчиков и кнопок заказа стирального порошка типа Dash Button. Разумеется, отдельный профиль для "интернет-телевизора" с VoD и "виртуальной реальностью", где будут особые требования по качеству коннекта.

Заказ "доступа" будет осуществляться со специального веб-сайта провайдера - просто добавляете устройство по определенной процедуре подтверждения легитимности коннекта именно вами, именно вот такого устройства.

Все. Теперь подключенные устройства будут он-лайн, где бы вы не находились. Буквально. 85% поверхности планеты уже покрыто радиодоступом, как утверждает Курцвейл. С надлежащим качеством и неважно к какому оператору будет подключение, потому что уберизация.

Доступ осуществляется посредством подключения абонента к множеству (действительно множества) устройств доступа, например, с Wi-Fi или еще каким 5G, установленных операторами, где только возможно - в квартирах, подъездах, столбах освещения, фасадах зданий или даже на пальмах. То есть ёлках.

Но для того, чтобы любые операторские устройства доступа к Сети узнавали абонентские устройства, нужно как-то передать данные серверам AAA. И еще более сложные процессы управления и мониторинга действительно множеством устройств доступа. Без этого никак не получится обеспечить приемлемое качество. Следовательно, нужен NETCONF и прочие составляющие SDN.

Впрочем, согласен. Этот кейс очень фантастичен.

Попробуем другой.

Вот, например, простая, казалось бы, схема подключения к Wi-Fi сети амазоновской кнопки "Dash Button".

По логике процесса - это очень простая вещь: кнопка почти никогда вообще не работает и просто выключена. Но при нажатии на "заказать", она должна "проснуться", подключиться к домашней Wi-Fi сети и выйти в интернет по заранее "зашитому" URL.

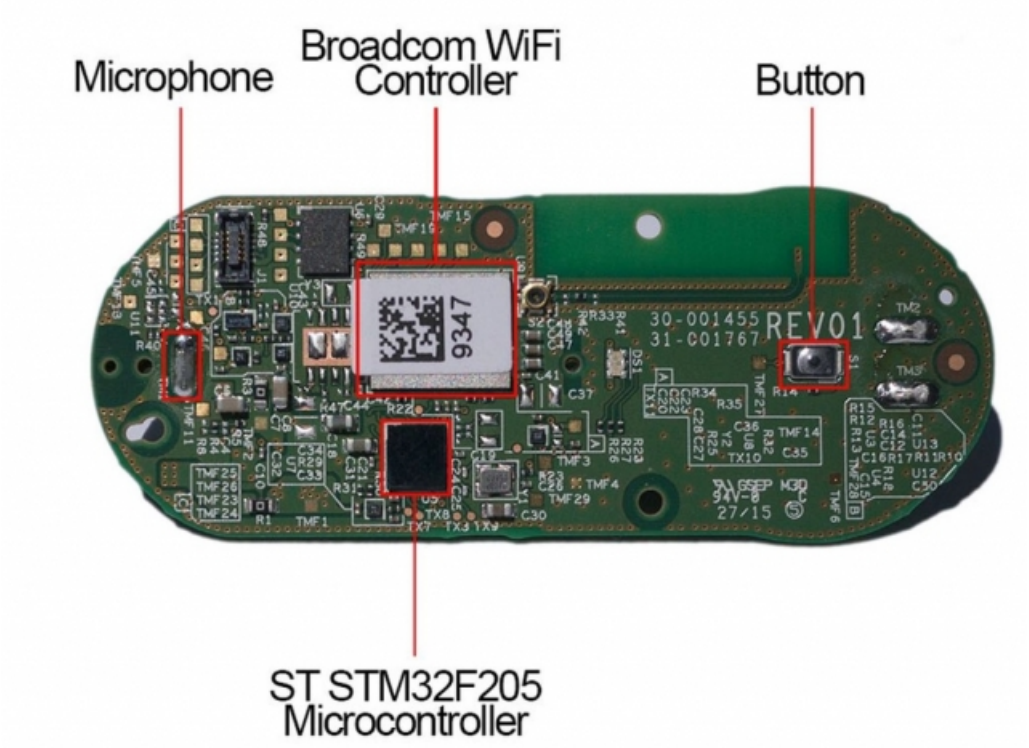
После этого считается, что заказ сформирован и к вам уже выехала бригада курьеров.

Очень просто и эффективно.

Если не задумываться о том, что вот к этой самой сети кнопка должна подключиться, для чего ей нужно знать минимум две вещи - название SSID сети и пароль доступа.

А как передать эти сведения кнопке? Это же просто кнопка и у нее нет никакого интерфейса взаимодействия с человеком - ни монитора, чтобы отразить правильность набора символов, ни собственно клавиатуры, которой эти символы можно было бы внести.

Amazon решил проблему довольно изящно процедурой подключения. Инициации, если хотите. Дело в том, что при первом подключении в качестве устройства ввода-вывода для кнопки используется смартфон с экраном и клавиатурой. И обязательно со специальным приложением. Собственно приложение и "расскажет" Dash-кнопке параметры доступа. Причем, не очень очевидным физическим каналом связи. По ультразвуку!



Дело в том, что на плате Dash Button имеется микрофон, который в момент инициации собственно "слушает". А приложение излучает несколько байт данных - SSID и пароль доступа к вашей Wi-Fi сети. Поскольку связь односторонняя, процесс может занять несколько минут, до тех пор, пока кнопка не подключится к амазоновскому серверу, который, в свою очередь, не сообщит приложению, что "все ок", подключение свершилось.

Схема гениальна. Но не идеальна. Во-первых, лишний элемент на кнопке. Во-вторых - долго. В-третьих, если вам придет в голову изменить пароль к сети, то придется проходить процедуру заново, что в случае десятков кнопок в эксплуатации наведет на мысль, что лучше ничего не трогать и пусть пароль остается прежним. Фиг с ней, с безопасностью.

Но, возможно, идеальная схема выглядела бы так:

Пользователь заходит на специальную страницу "добавить IoT-устройство" в кабинете оператора. Далее, контроллер (с NETCONF, конечно) шлет маршрутизатору команду "создать setup SSID" с заранее известной парой SSID-pass (автор знает, что такое Wi-Fi Protected Setup, но это же учебный пример - с дыркой). Дальше зажимаем кнопку "чистого" IoT-устройства, которое соединяется с сервером и путем обмена ровно двумя RESTConf-ссылками получает уже параметры доступа к "боевой сети". Пользователю остается только нажать "Подтвердить привязку", сравнив коды на корпусе устройства и появившиеся в веб-интерфейсе, и вуаля! Все работает. Безопасность соединения при этом обеспечивается минимум ~~двумя~~ тремя факторами: mac-адрес устройства жестко прописан в access-листе конфигурации, собственно транспорт защищен WPA-соединением от Wi-Fi сети, а данные от кнопки передаются на сервер в криптованном HTTPS.

А еще такая схема позволяет оператору производить учет и контроль таких подключений и, следовательно, выставлять счета за "услуги IoT".

Что и требовалось.

В завершение сего лонгрида - традиционная гифка, которая олицетворяет отношение сетевого администратора, идейного приверженца к CLI, к неизбежно подкрадывающимся изменениям в виде "программно-конфигурируемых сетей".

[Управляется программно: идеальный маршрутизатор]