Управляется программно

Идеальная жена поразительно умна и чутка. Она понимает мужа с полуслова и даже вовсе без слов, и знает все обо всем. Но она не кичится своими знаниями, нет! Они обнаруживаются только тогда, когда потребуются мужу. И в любом случае она всегда безоговорочно и даже с радостью признает, что ее самец умнее ее — иначе и быть не может, ведь он же муж! Хотя умнее других самцов она может быть, если это будет приятно ее избраннику.

[Управляется программно]

Я так понимаю, дорогой читатель, что вступление тебе понравилось. Идеальная жена - это то, что ищет каждый мужчина, но, увы, находит редко. Хотя бы потому, что "идеальная жена" может иметь маму… или подруг…

Но к делу.

Сегодня я вступаю на тонкий лед писания о вещах сугубо технологических, в которых не сказать чтобы я был полным профаном, но и узким специалистом тоже никогда себя не считал. Речь пойдет о Software-defined Networking, она же SDN, она же "программно-конфигурируемая сеть", она же ПКС. Я напишу про это как понял, а если не так понял, то технически подкованные читатели тут же поправят меня в комментариях. Инда именно этим славится Наг - всегда найдется некто, кто тему знает лучше автора и никогда не оставит ошибку безнаказанной.

Что ж, я очень надеюсь, что данная тема будет не менее полемичной, чем RIP PSTN, которая на самом деле ясна и понятна. В отличие от SDN, где маркетингового шума гораздо больше, а реальных внедрений сильно меньше.

Откуда пошла сеть программная

Есть одна небольшая теория, выведенная досужими философами, что с ростом объема знаний, накопленных человечеством, повышается эффективность жизнедеятельности. Кажется, что это заявление "Капитана Очевидность", но на самом деле, в теории скрывается методология прогнозирования и связь с базовыми законами мироздания.

Вот например, "Второе начало термодинамики", гласит, что "энтропия в замкнутой системе не убывает". Казалось бы, а при чем тут Лужков логическая цепочка “объем знаний -> эффективность”?

А при том, что кажется, что субъекты применения знаний не могут увеличивать энтропию, ибо развитие в сторону усложнения невозможно (энтропия - это еще и мера хаоса). Однако подобное применение физического закона не является корректным, так как энтропия не убывает только в замкнутых системах, в то время как человечество оперирует исключительно с системами открытыми.

В процессе познания и манипуляций с различными физическими объектами, люди превращают энергию одного вида (электромагнитную, химическую) в энергию другого вида (тепловую, кинетическую), тем самым ускоряя суммарное увеличение энтропии Вселенной. Несмотря на "локальное" уменьшение энтропии путем "упорядоченных" процессов, происходит суммарное увеличение энтропии вселенной, а человечество являются в некотором роде катализатором этого процесса. И для того, чтобы исполнять роль катализатора, мы просто обязаны эффективность повышать, тем самым приближая "тепловую смерть Вселенной". Причем, эффективность должна повышаться по логарифмическому закону и прирост роста (ну, да - прирост роста) не имеет права падать.

В качестве примера такового, могу привести общеизвестный "Закон Мура", который с некоторыми уточнениями исполняется с настырностью подростка, лезущего в колхозный сад за яблоками, уже более сорока лет.

Однако, многие знания - многие печали. Объем знаний, накопленный за тысячи лет исторического развития, превысил физический предел для понимания и запоминания 95% населения Планеты. Оставшиеся 5% пытаются осилить только некоторые, иногда очень узкие разделы наук. Опять же для примера - в 1800-м году "ученый" мог заниматься просто "наукой". В 1900-м - "химия" и "физика" уже были достаточно разными науками. В 2000-м - существуют десятки разных "химий" и "физик", и одинаково хорошо разбираться в нескольких из них - уже невозможно. Любая профессия или специальность отделяется от других при прогрессе в данной области и накоплении знаний.

Аналогичная ситуация постигла и науку о телекоммуникациях. Причем, наука сия развивается столь быстро, что знания, полученные каких-то десять лет назад, в современных условиях оказывается совершенно бесполезными при практическом применении. Нет-нет, я не утверждаю, что базовые знания по теории общей электротехники повредят современному сетевому инженеру. Но и не помогут, если речь пойдет про оптические сети - я сам, например, реально плаваю в данном разделе.

А ведь когда-то, в начале девяностых, с помощью паяльника, пинцета и оловоотсоса лихо ремонтировал сетевые адаптеры УК-УНЦ, строил сетевые интерфейсы между "Корветами" и IBM PC и подключал БК-0010 к бытовым телевизорам в школьных классах целого района. Н-да…

Чтобы не скучно было читать дальше, вот такой коуб есть для вас:

На сегодняшнем этапе сложности сетевой техники, нереально знать ВСЁ. Не хватит мощности человеческого ОЗУ/ПЗУ. Есть, конечно, уникумы, утверждающие, что знают многое, но не до каждой проприетарной реализации любого случайно выбранного RFC же! В итоге огромное количество времени даже суперспец тратит на вдумчивое курение мануалов, итогом которого становится изменение пары символов в конфигах. И еще не факт, что "полетит".

В качестве иллюстрации, приведу забавный график по количеству публикуемых RFC во времени:



Читатель может самостоятельно убедиться в том, что сегодня всевозможных RFC насчитывается ровно 6691 штука. Это много. Реально много. Причем, я даже почти уверен, что никто из присутствующих не сможет вспомнить (без подсказки Гугла), не то, что номера самых ходовых документов "рабочих предложений", но и вполне расхожих аббревиатур, типа HTCPCP (RFC 2324).

Впрочем, оно и не нужно в 99%.

Так вот, про SDN.

Этот подход (а это скорее именно "концептуальный подход") к построению сетей позволяет разделить две глобальные сущности телекоммуникационных систем - "железную" и "программную". Причем, с возможностью существенного упрощения как первой, так второй сущностей.

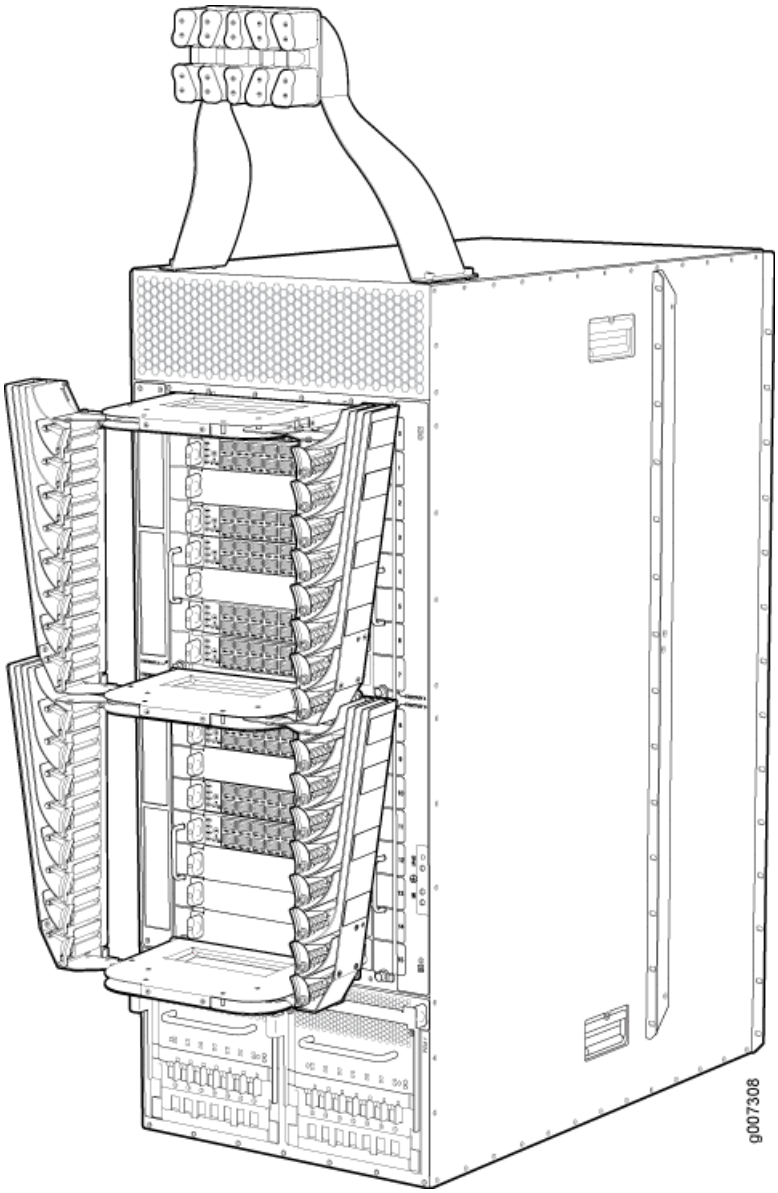
Например, вот такой зоопарк разных реализаций сетевых хардверных устройств:



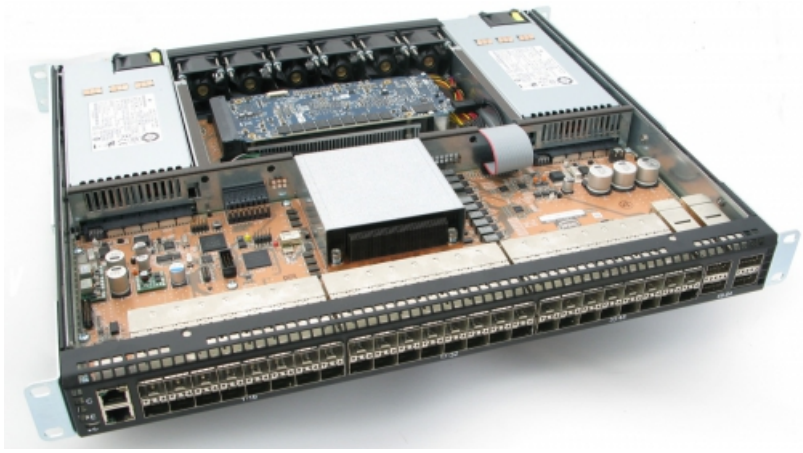
…свести к нескольким более простым сущностям. В идеале - к двум:

1. Программно-конфигурируемым коммутаторам, представляющим собой сервер с достаточно

мощным (и) микропроцессором (ами), сетевыми интерфейсами и специализированной операционной системой:



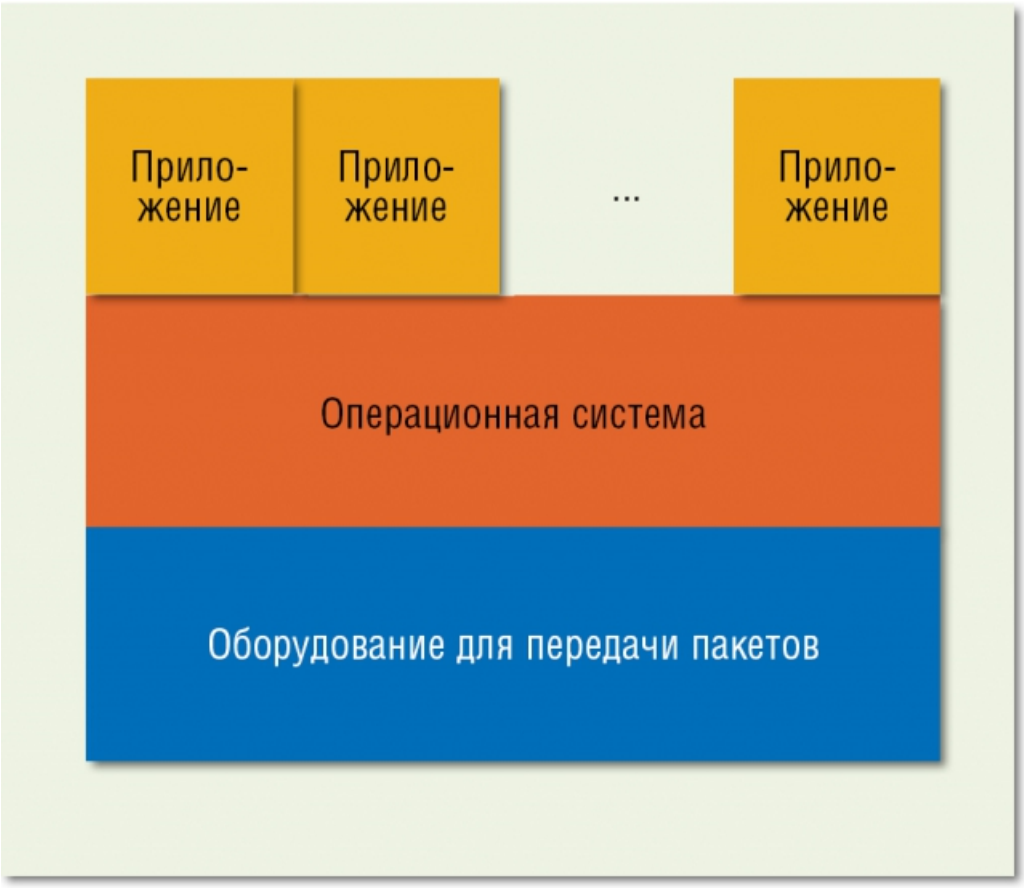
2. Контроллером, управляющим ПК-коммутатором(ами):



…представляющим собой сервер с достаточно мощным(и) микропроцессором(ами), сетевыми интерфейсами и специализированной операционной системой.

Есть мнение, что ПК-коммутаторы все же должны различаться по мощности процессоров и объему памяти на борту - типа дешевые и хиленькие на доступ, и мощно-дорогие на ядро, но это уже детали. По сути это допиленный по необходимости компьютер со специальной операционной системой и ничего больше.

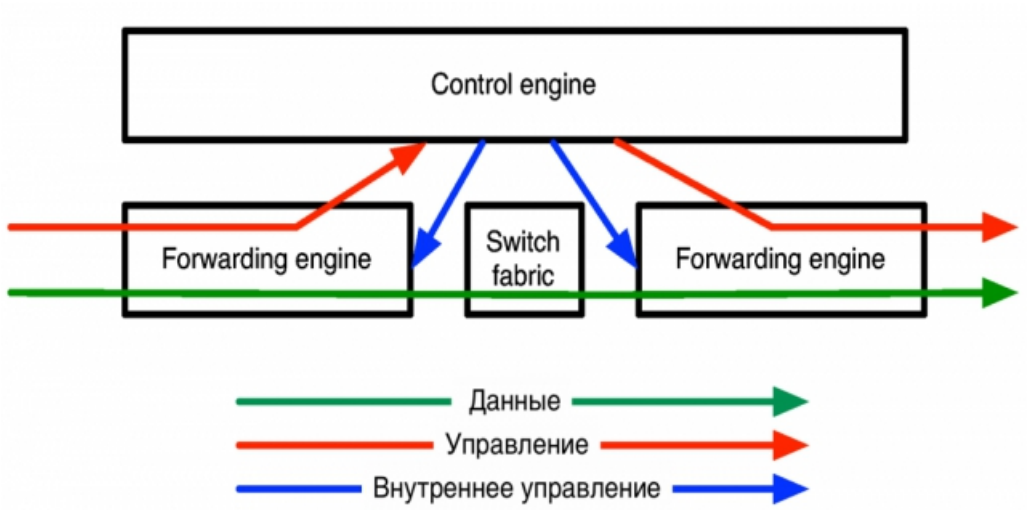
Но очень желательно, чтобы устройства были вообще одинаковыми, и функции контроллера мог выполнять какой-нибудь один стандартный роутер.



Такой подход приведет к тому, что многостраничные спецификации будут просто не нужны. Зачем? У нас есть проектируемая сеть с N активными элементами. Следовательно, нам нужно купить N+1 универсальных SDN-роутеров. +1 будем использовать в качестве контроллера.

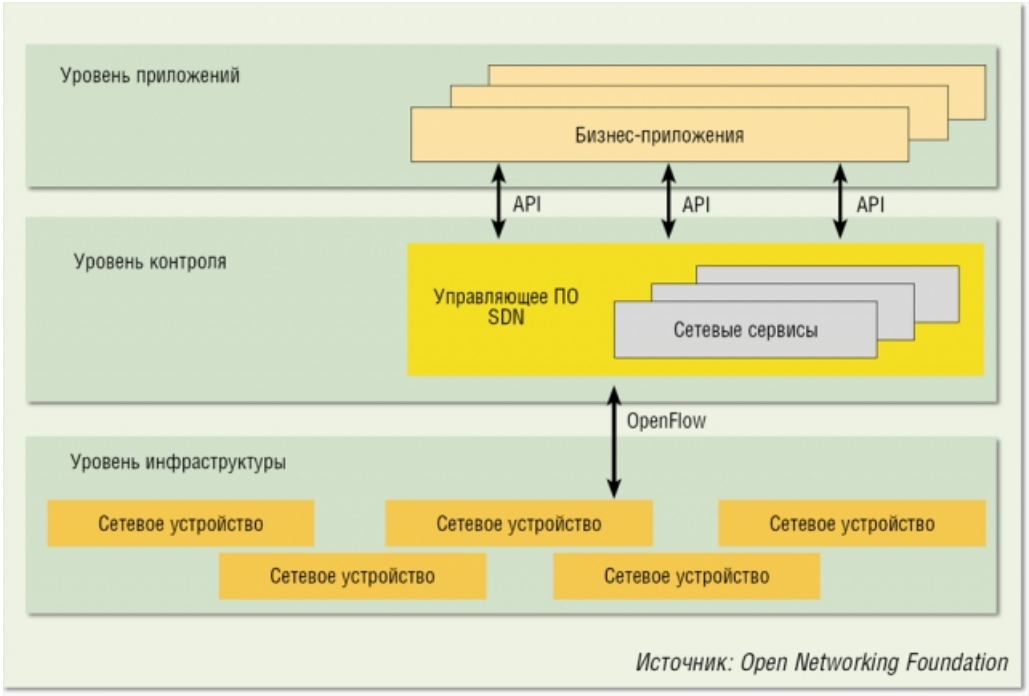
Коммодитизация (превращение в ликвидный товар) коммутаторов будет так же на высоте - сняли с демонтируемых объектов SDN-устройства, продали. Перенесли в другое место, воткнули патчкорды - работает.

Разумеется, нужен еще соответствующий софт, который будет рулить процессом. В обще-абстрактной схеме ПКС будет выглядеть так:



Это всего лишь абстракция "идеального коммутатора", который по команде "управляющего движка" переправляет данные в нужном направлении.

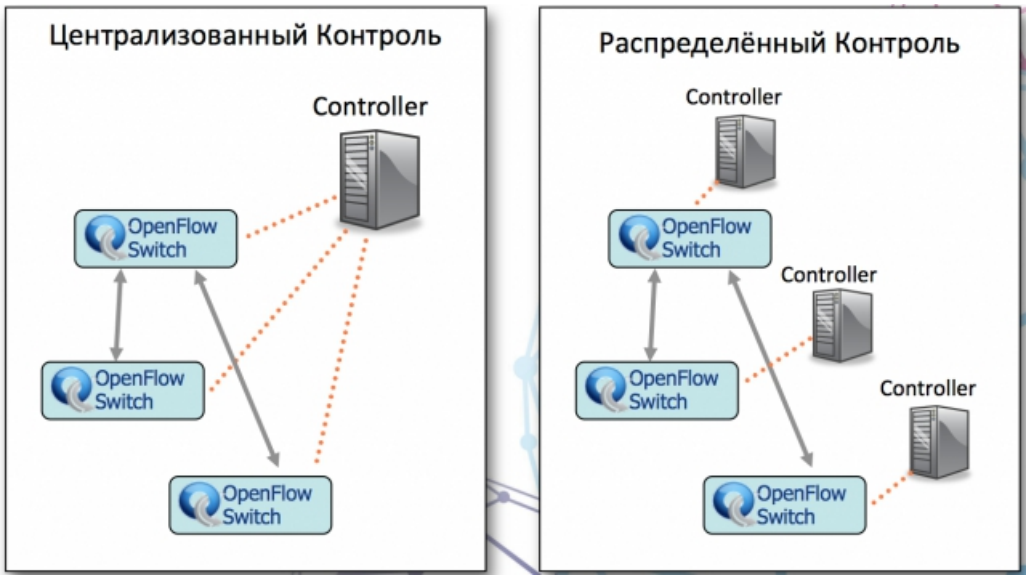
Вот более высокий уровень абстракции:



Есть физический инфраструктурный уровень, на котором располагаются все эти стандартные коммутаторы/сетевые устройства. Посредством специального протокола OpenFlow (специальный протокол, на котором базируется SDN) контроллер SDN может управлять всеми устройствами, назначая им соответствующее данному элементу поведение, таблицы маршрутизации и прочие тонкости, задуманные администратором сети, позволяющие устройствам адекватно реагировать на события.

Еще выше уровнем находятся "бизнес-приложения". Да хотя бы старая добрая телефония, видеостриминговые сервисы, управление доступом и прочая-прочая-прочая.

Причем, поскольку контроллеры так же могут общаться между собой и управлять друг другом, можно строить сети различной степени вложенности иерархий:



… а контроллеры могут быть и централизованными, и распределенными.

Думаю, основную идею на уровне детского сада мне удалось передать, а тонкости архитектуры и протоколов, которые, должен огорчить читателя-неспециалиста, находятся еще в весьма экспериментально-зачаточном состоянии.

Упомянутый уже OpenFlow пока ругают на проф-ресурсах за общую сырость и ненадежность. Кроме того, различные особо умные вендоры пытаются придумать свой Луна-парк, и имеются "конкурирующие" SDN подходы в виде, например, Network functions Virtualization (NFV) или Network Virtualization (N.V.).

Разница между этими концепциями не так уж и велика, но есть:

● SDN - пытается разделить две плоскости: управление сетью (мозги) и транспорт (мышцы). В итоге пытается обеспечить централизацию управления распределенной сети с целью более эффективного использования ресурсов и автоматизации управления сетевыми сервисами.

● NFV - сосредоточена на оптимизации сетевых сервисов внутри сети за счет разделения сетевых функций (например, DNS, кэширование etc), от собственно реализации аппаратного обеспечения. Таковое разделение позволяет универсализировать программное обеспечение, ускорить внедрение новых функций сети и служб.

● N.V. - это тотальная виртуализация всех элементов сети, что позволяет разворачивать устройства специального назначения на любой (поддерживающей концепт, разумеется) аппаратуре. В том числе и использовать одно устройство для нескольких функций путем использования виртуальных машин с разным функционалом. Ключевое слово - multi-tenancy.

Практический пример с гипотетическими объектами

Снова обратимся к понятиям философическим:

...как целое понимается из отдельного, но и отдельное может быть понято только из целого, имеет такую важность для данного искусства и столь неоспоримо, что уже первые же операции невозможно проделать без применения его, да и огромное число герменевтических правил в большей или в меньшей степени основывается на нем...

...говаривал немецкий философ конца XVIII века Фридрих Даниэль Эрнст Шлейермахер. Если перевести на язык уральских подворотен предельной ясности, то понимание какого-то тонкого и абстрактно-сложного явления достигается с двух сторон - если вы можете от общего объяснить частное и одновременно от частного перейти к общему. Это называется "герменевтическим кругом" и применяется некоторыми достаточно умными педагогами в процессе обучения.

Поскольку написание сего опуса как раз в большей степени связан с наукой научения (больше, надо сказать, себя самого), то попробуем идею SDN изложить с другой стороны. А именно, со стороны рядового маркетолога от телекома специалиста-техника низшего звена.

И вот что получается. В процессе строительства, главный архитектор будучи в изрядном подпитии спроектировал сеть небольшого районного провайдера наиболее экономичным в данном временном отрезке способом. Запроектировано все по классической трех-уровневой схеме и все прекрасно работало, пока не появилась необходимость сеть эту модернизировать. Цель модернизации оставим за скобками, ибо не очень-то оно и важно, но пусть это будет переход на IPv6. Или еще какая напасть, типа смены способа авторизации или внедрение IP-TV (ха-ха).

Модернизация на самом деле, не такая уж и сложная - нужно просто взять, и перенастроить примерно 500 коммутаторов уровня доступа, 50 коммутаторов уровня агрегации, и пусть пяток маршрутизаторов. Прописать новые правила, обновить firmware, где нужно, поменять пароли за одно.

Сделать это можно собственными силами, и даже почти не обновляя зоопарк этих самых коммутаторов/маршрутизаторов. Но есть пара проблем: делать это нужно а) на рабочей сети, желательно без перерыва оказания услуг, и б) почти одновременно, ибо получится, что некоторые сегменты сети не только не получат новых услуг, но и старые-то загнутся с большой вероятностью.

Действия техслужбы вполне предсказуемы. Нужно:

1. Планируется собственно изменения, что и как нужно сделать, чтобы сеть начала работать "по-новому";
2. Спланировать работы, для чего сеть разбивается на сектора, на каждый из которых назначается ответственный;
3. В час "X" начинается беготня и толкотня плановая работа.

Это все очень утрировано конечно, но как правило, результат такой работы немного предсказуем.

Нужно заметить, что проведение такого действа с массовыми отказами вообще возможно только если на коммутаторах есть возможность удаленного конфигурирования (напомню, что при стройке использовался метод "подешевле") без подключения к технологическим портам, есть довольно не малый склад "подменки" на случай "если что-то пойдет не так", есть актуальная и рабочая база

данных о текущей конфигурации устройств (CMDB) и довольно прямых рук персонала. И почти всегда при таких процессах используется двойная работающая система, ибо только очень смелый человек рискнет очень быстро выключить боевую конфигурацию, а затем очень быстро включить другую "потенциально боевую", но не оттестированную. Обычно же включают рядом новую систему, тестируют, а затем последовательно переводят сектора, что в итоге приводит к двум одновременно рабочим системам со всеми вытекающими экономическими последствиями.

Еще раз оговорюсь, что ситуация гипотетическая, но вполне вероятная где-то в параллельных вселенных с такими "великими техническими специалистами", как я.

А вот бы, фантазирует технический директор, иметь такую штуку, чтобы и CMDB вела сама, и чтобы в любой момент можно было firmware обновить, и мониторила в процессе (с красной лампочкой алярм!), и изменения на сети происходили не так плачевно. Собственно, фантазия эта и есть SDN.

Смотрите, что происходит: каждое устройство с определенной периодичностью "постукивает" по определенному сетевому адресу на предмет каких-то изменений. На собственно "определённом сетевом адресе" и располагается тот самый контроллер, который получает команды от администратора, например, через банальный веб-интерфейс.

Если изменения таки есть, "унифицированный коммутатор" просто скачивает себе новый конфиг, новую прошивку или новое приложение и, если нужно, перезапускается. Красота же!

Догадливый администратор тут же сообразит, что изменения контроллеру можно отдавать как индивидуально для каждого "унифицированного ПКС-коммутатора", так и создавать групповые политики. В итоге получается не просто "CMDB с посмертным учетом", а именно что действующая, боевая и постоянно актуализируемая база данных работающей сети. Мониторинг, кстати, заложен архитектурно - если устройство не обратилось к контроллеру в заданное время, значит, с ним что-то не так: "алярм! алярм!"

Разумеется, у такой сети должны быть политики обеспечения живучести - дублирование и резервирование. Политики безопасности - всякие пароли доступа, шифрование, криптоключи. И политики угождения администратору, как та самая “идеальная жена” из начала статьи. Ибо какая же еще жена может быть у вот такого красавца-админа?

[Управляется программно]