**MPLS L2VPN**

L3VPN покрывает собой огромное количество сценариев, необходимых большинству заказчиков, но не все. Он позволяет осуществлять связь только на сетевом уровне и только для одного протокола — IP. Как быть с данными телеметрии, например, или трафиком от базовых станций, работающих через интерфейс E1? Существуют также сервисы, которые используют Ethernet, но тоже требуют связи на канальном уровне.

Какие вообще виды L2VPN существуют.

**VLAN/QinQ** (**QinQ IEEE 802.1QinQ** — расширение к стандарту IEEE 802.1Q, описывающее как тегированный трафик может передаваться внутри уже тегированного по 802.1Q трафика. Эта технология имеет большое значение для построения Metro Ethernet-сетей. Для использования QinQ-инкапсуляции требуется поддержка со стороны коммутатора.) — их можно сюда отнести, поскольку основные требования VPN выполняются — организуется виртуальная L2 сеть между несколькими точками, данные в которой изолированы от других. По сути VLAN per-user организует Hub-n-Spoke VPN.

**L2TPv2/PPTP**/**L2TPv3**/**GRE** — протокол туннелирования второго уровня, имеют проблемы с масштабированием.

**VXLAN, EVPN** — варианты для ЦОД'ов. Очень интересно, но DCI не входит в планы этого выпуска. Зато о них был отдельный подкаст (слушайте запись 25-го ноября)

**MPLS L2VPN** — это набор различных технологий, транспортом для которых выступает MPLS LSP. Именно он сейчас получил наиболее широкое распространение в сетях провайдеров.

Возможность инкапсулировать трафик любого канального уровня в MPLS называется **AToM — Any Transport over MPLS**.

* ATM Adaptation Layer Type-5 (AAL5) over MPLS
* ATM Cell Relay over MPLS
* Ethernet over MPLS
* Frame Relay over MPLS
* PPP over MPLS
* High-Level Data Link Control (HDLC) over MPLS

Для построения любого L2VPN существуют два концептуально разных подхода.

* **Point-to-Point**. Применим к любым типам протоколов канального уровня и в принципе, в полной мере исчерпывает все сценарии применения L2VPN. Поддерживает все мыслимые и немыслимые протоколы. Причём некоторые из них ещё и по-разному может реализовывать. В основе лежит концепция PW — PseudoWire — псевдопровод. Общее название услуги: **VPWS — Virtual Private Wire Service**.
* **Point-to-Multipoint**. Этот режим только для Ethernet, поскольку только в нём фактически такая необходимость есть. В этом случае у клиента может быть три-пять-десять-сто точек подключения/филиалов, и все они должны передавать данные друг другу, причём, как одному конкретному филиалу, так и всем сразу. Название технологии: **VPLS — Virtual Private LAN Service**.

**Терминология**

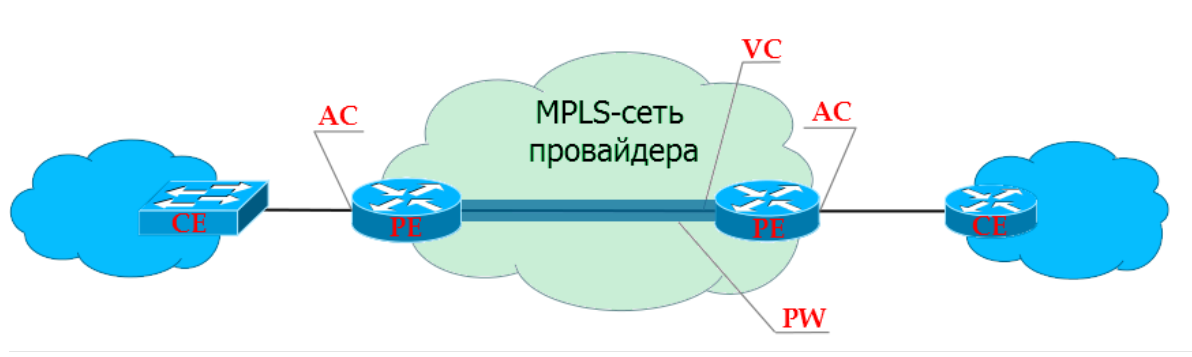
**PE — Provider Edge** — граничные маршрутизаторы MPLS-сети провайдера, к которым подключаются клиентские устройства (CE).

**CE — Customer Edge** — оборудование клиента, непосредственно подключающееся к маршрутизаторам провайдера (PE).

**AC — Attached Circuit** — интерфейс на PE для подключения клиента.

**VC — Virtual Circuit** — виртуальное однонаправленное соединение через общую сеть, имитирующее оригинальную среду для клиента. Соединяет между собой AC-интерфейсы разных PE. Вместе они составляют цельный канал: AC→VC→AC.

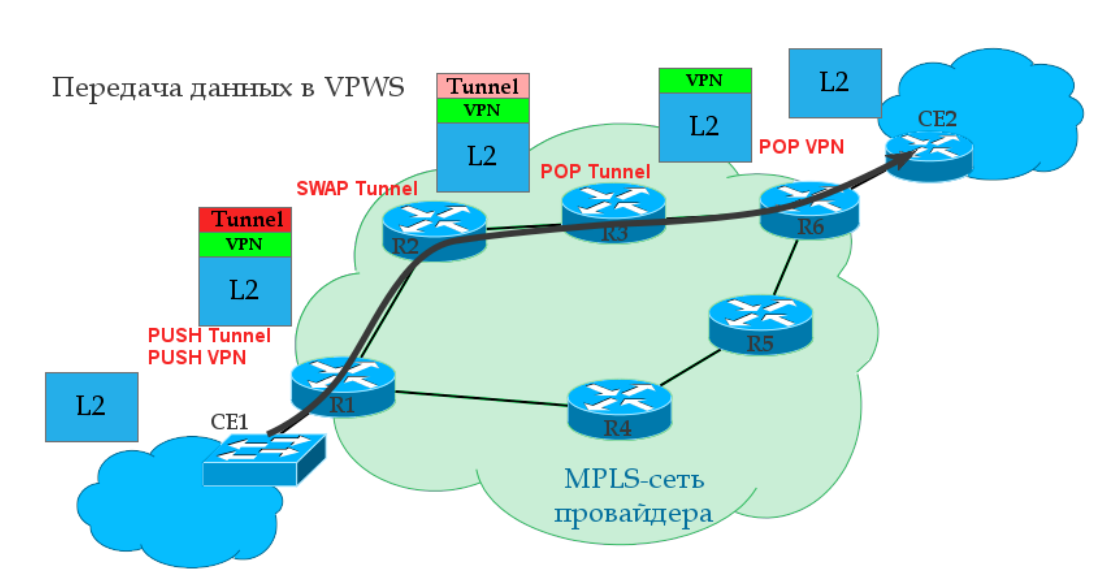
**PW — PseudoWire** — виртуальный двунаправленный канал передачи данных между двумя PE — состоит из пары однонаправленных VC. В этом и есть отличие PW от VC.



**VPWS — Virtual Private Wire Service**

В основе любого решения MPLS L2VPN лежит идея PW — PseudoWire — виртуальный кабель, прокинутый из одного конца сети в другой. Но для VPWS сам этот PW и является уже сервисом.

**VPWS Data Plane или передача пользовательского трафика**



0. Между R1 и R6 уже построен транспортный LSP с помощью протокола LDP или RSVP TE. То есть R1 известна транспортная метка и выходной интерфейс к R6.

1. R1 получает от клиента CE1 некий L2 кадр на AC интерфейс (то может оказаться Ethernet, TDM, ATM итд. — не имеет значения).

2. Этот интерфейс привязан к определённому идентификатору клиента — VC ID — в некотором смысле аналогу VRF в L3VPN. R1 даёт кадру сервисную метку, которая сохранится до конца пути неизменной. VPN-метка является внутренней в стеке.

3. R1 знает точку назначения — IP-адрес удалённого PE-маршрутизатора — R6, выясняет транспортную метку и вставляет её в стек меток MPLS. Это будет внешняя — транспортная метка.

4. Пакет MPLS путешествует по сети оператора через P-маршрутизаторы. Транспортная метка меняется на новую на каждом узле, сервисная остаётся без изменений.

5. На предпоследнем маршрутизаторе снимается транспортная метка — происходит PHP. На R6 пакет приходит с одной сервисной VPN-меткой.

6. PE2, получив пакет, анализирует сервисную метку и определяет, в какой интерфейс нужно передать распакованный кадр.

**VPWS Control Plane или работа служебных протоколов**

Транспортная метка может назначаться как LDP, так и RSVP-TE. За распространение сервисных меток отвечает тот же LDP — Targeted LDP. Теперь он может устанавливать соединение с удалёнными маршрутизаторами и обмениваться с ними метками. Обычный LDP — это link-local протокол и ищет соседей среди непосредственно подключенных маршрутизаторов, то есть TTL его пакетов — 1. Однако tLDP достаточна IP-связность.

**Виды VPWS**

**PW** — это виртуальный L2-канал, который лежит в основе любой технологии L2VPN и служит туннелем для передачи данных.

**VLL (Virtual Leased Line)** — это уже технология, которая позволяет инкапсулировать кадры различных протоколов канального уровня в MPLS и передавать их через сеть провайдера.

**PWE3 — Pseudo Wire Emulation Edge to Edge**. Строго говоря, область применения этой технология шире, чем только MPLS. Однако в современном мире в 100% случаев они работают в связке.

**VPLS. Point-to-Multipoint**

VPLS — Virtual Private LAN Service. По своей сути — это коммутатор. Провайдер подключает несколько точек заказчика к своей сети в разных её концах и обеспечивает L2 связность. И теперь это задача транспортной сети провайдера — заботиться о правильной коммутации кадров, то есть изучении MAC-адресов.

**Терминология**

**VPLS-домен** — изолированная виртуальная L2-сеть, то есть, грубо говоря, один отдельный L2VPN. Два разных клиента — два разных VPLS-домена.

**VSI — Virtual Switching Instance**. Виртуальный коммутатор в пределах одного узла. Для каждого клиента (или сервиса) он свой. То есть трафик разных VSI не может кочевать из одного в другой.

В общих чертах передача пользовательского трафика выглядит, как и для случая VPWS, но добавляется шаг изучения MAC'ов и проверки MAC-таблицы при передаче трафика.

1) На AC-порт PE-маршрутизатора пришёл пользовательский кадр. PE-маршрутизатор смотрит в заголовок Ethernet и проверяет MAC-адрес отправителя.

А) Если он уже есть в таблице MAC-ов данного VSI, PE сразу переходит к шагу 3.

Б) Если этого адреса ещё нет — он записывает соответствие MAC-порт в таблицу и тоже переходит к шагу 3.

3) PE-маршрутизатор смотрит в заголовок Ethernet и проверяет MAC-адрес получателя.

А) если таковой присутствует в таблице MAC-адресов данного VSI:

* PE ищет выходной интерфейс для кадра с данным MAC'ом. Это может быть физический интерфейс или PW.
* Если порт назначения — физический интерфейс — просто отправляет в этот порт. Если это PW, то добавляет соответствующую метку — сервисную. Эта метка будет неизменна до конца пути.
* PW — это всегда канал между двумя IP-узлами, поэтому зная IP-адрес удалённого PE, локальный PE из таблицы меток извлекает транспортную и ставит её сверху стека — она будет меняться на каждом P-маршрутизаторе.

Б) Если же MAC-адрес неизвестен, то как порядочный коммутатор наш PE должен выполнить широковещательную рассылку кадра по всем PE данного VSI.

* Локальный PE составляет список всех удалённых PE этого VSI, и, создав копии этого кадра, вставляет в них сервисные метки — каждому своя.
* Далее на каждую копию кадра навешивается ещё и транспортная метка (тоже своя для каждого PE).
* Весь этот ворох кадров рассылается по сети провайдера.
* Также копии широковещательного кадра отправляются в AC-интерфейсы, если такие есть, без заголовков MPLS.

4) Удалённый PE после получения кадра и снятия меток (то есть когда уже определил VSI) тоже действует как обычный коммутатор:

А) Если MAC-адрес источника пока не известен, вносит его в таблицу. В качестве входного интерфейса будет указан PW к Ingress PE.

Б) Если MAC-адрес назначения ему известен, отсылает кадр в тот порт, за которым он изучен. Отсылается уже чистый Ethernet-кадр, без каких-либо заголовков MPLS.

В) Если этот MAC не удалось найти в таблице? Широковещательная рассылка по всем AC-портам этого VSI. Заметьте, PE не будет рассылать этот кадр в PW данного VSI, потому что все другие PE уже получили копию этого кадра от входного PE. Это всё то же правило расщепления горизонта (Split Horizon), и так достигается отсутствие петель коммутации и широковещательных штормов.

Как и в обычном коммутаторе записи в MAC-таблице VSI периодически протухают и удаляются.

В вопросе изучения MAC-адресов в VPLS есть один нюанс, который резко отличает его от L3VPN. PE должен не просто знать физический порт, откуда пришёл кадр — важно определить соседа или, точнее PW как виртуальный интерфейс. Дело в том, что клиентский кадр нужно отправить не просто в какой-то физический порт, а именно в правильный PW, иными словами, правильному соседу.

Для этой цели каждому соседу выдаётся личная метка, с которой тот будет отправлять кадр этому PE в данном VPLS-домене. И в дальнейшем по VPN-метке, заглянув в LFIB, PE узнает, от какого соседа пришёл кадр.

Напомню, что L3VPN без разницы, откуда пришёл IP-пакет, поэтому для префикса в VRF всем соседям сообщается одна и та же метка.

**VPLS Control Plane**

Из работы Data Plane уже видно, что для VPLS требуется полносвязная топология PE для каждого VSI. Причём не все PE MPLS-сети провайдера будут соседями, а только те, где есть этот же VSI.

Поэтому один из главных вопросов в VPLS — обнаружение всех PE, куда подключены клиенты данного VSI. Существует здесь два подхода: ручная настройка и автоматическое обнаружение. Первый путь изначально предложила Cisco (драфт Мартини), отцом второго является Juniper (драфт Компелла).

Независимо от того, что мы вытворяем с VPN-метками, транспортные LSP строятся как обычно LDP или RSVP-TE. Точно также, независимо от используемого режима, VPLS при более детальном рассмотрении разваливается на point-to-point PW. То есть мы имеем не некое централизованное облако коммутации, а просто набор виртуальных линий между всем соседями. Решение о передаче кадра принимает Ingress PE или, проще говоря, выбирает нужный PW.

**VPLS Martini Mode (LDP)**

Люка Мартини — бывший сотрудник Cisco — предоставил решение на основе LDP.

Работать оно будет поверх MPLS-сети. Для сигнализации меток будет использоваться LDP, который уже является частью MPLS. VPLS Martini Mode описан в стандарте RFC4762. Именно это лаконичное решение и стало стандартом де факто в большинстве сетей по всему миру.

LDP используется для распределения сервисных меток. Удалённые сессии с каждым соседом в VPLS-домене настраиваются вручную. Поскольку заранее известны все участники этого VPLS, каждому из них LDP выделяет индивидуальную метку в сообщении LDP Label Mapping Message.

Если добавляется новый PE в VPLS-домен, необходимо настроить LDP-соседство со всеми существующими PE этого VPLS. После чего с каждым из них новый PE обменятся метками.

В течение всего времени, LDP проверяет доступность своих соседей. Если какой-то из соседей выходит из группы или становится недоступным, сессия разрывается и все изученные MAC'и в PW к этому соседу очищаются. Если состояние какого-либо из AC-портов VPLS-домена переходит в состояние Down, либо происходит другое событие, заставляющее очистить MAC-адреса с AC-порта, то PE сообщает об этом всем своим соседям, настаивая на удалении MAC-адресов в сообщении LDP MAC Withdraw

**VPLS Kompella Mode (BGP)**

Проблему поиска соседей решил Кирити Компелла — сотрудник Juniper. Он отталкивался от тех же критериев, но решил, что MBGP, опробованный на L3VPN, подойдёт лучше на роль протокола распределения меток.

Другое название VPLS Kompella-mode — VPLS Auto-Discovery, потому что именно это является его качественным отличием от Martini. Также вы можете услышать VPLS BGP Signaling.

Control Plane выполняет здесь две основные функции:

— Обнаружение соседей

— Передача маршрутной информации и обмен метками.

Обнаружение соседей или Auto-Discovery

Ничего нового выдумывать тут не пришлось. Схема обнаружения соседей уже применённая в L3VPN прекрасно работает и здесь.

Route Target, который является Extended Community — главный признак принадлежности к определённому VSI. Грубо говоря — если RT совпадают — значит в одном VSI.

Строго говоря: Если RT полученного анонса совпадает с настроенным в VSI, значит этот VSI хочет знать информацию из анонса.

Как и в L3VPN можно гибко организовать взаимодействие между различными VSI. Но так крайне редко кто делает.

Передача префиксов

В целом префикс L2VPN — вещь довольно искусственная — PE своим BGP Update, скорее, сообщает сам факт своего участия в VPLS-домене и метку этого факта. Но это не играет большой роли.

Какого-то адреса, тем более MAC'а, в поле NLRI сообщения BGP Update VPLS не передаёт. Помните, что изучение MAC-адресов — это полностью функция Data Plane.

**Matini vs Kompella**

В результате, какие преимущества даёт использование BGP Kompella mode перед Martini Mode?

* Auto-Discovery. Нет необходимости на каждом PE настраивать удалённые LDP-сессии со всем участниками этого VPLS-домена. При использовании RR вопрос полной связности совсем отпадает.
* Вместе с Auto-Discovery BGP поддерживает и обмен префиксами/метками. Это обычно ставится в плюс методу, хотя это необходимая функция. Здесь, вероятно, лучше заметить, что если у оператора уже есть инфраструктура BGP, то L2VPN сигнализацией через BGP впишется в неё органично. А вот если этот протокол до сих пор не использовался, то выглядеть он может, как пятая нога.
* У BGP всё схвачено на предмет Inter-AS VPN. Option C позволит организовать бесшовный (seamless) MPLS между различными AS и протянуть через него тысячи PW.