UNIVERSITY OF AMSTERDAM

MSC SYSTEMS & NETWORK ENGINEERING

Architecture of dynamic VPNs in OpenFlow

By:
Michiel APPELMAN
michiel.appelman@os3.nl

Supervisor:
Rudolf STRIJKERS
rudolf.strijkers@tno.nl

June 24, 2013

Summary

Hello world.

Contents

List of Figures

List of Tables

1 Introduction

Network operators today use Network Management Systems (NMSs) to get control over their devices and services that they deploy. These systems have been customized to their needs and in general perform their functionalities adequately. However, operators run into obstacles when trying to expand their business portfolio by adding new services. This will require *a*) new Application Programming Interface (API) calls to be implemented between their B/OSS and NMS, *b*) their NMS to be able to cope with potentially new protocols, and *c*) added expertise by engineers to define the requirements and restrictions of these protocols. When these obstacles are eventually overcome the setup that will result from this implementation will be relatively static, since any change to it will require the whole process to be repeated.

Until recently this limitation didn't distress operators as their networks were in fact primarily static. But with increasing demand for services requiring for example mobility and short-term virtual networks, these limitations start to become a tangible problem for operators. By solving the complexity of implementing new services or features for them, they will be able shorten their time to market, save on networking expertise and be more adaptive to changes in these services.

A potential candidate to solve this complexity is OpenFlow [1] and Software Defined Networkinging (SDN). SDN is a relatively new architecture to allow for the programmability of networks. The architecture has recently been standardized in the OpenDaylight project [2] which also includes OpenFlow, a lower level and increasingly supported API towards networking devices. Implementing the SDN architecture promises *a*) CAPEX savings due to hardware being more generic and flexible, *b*) OPEX savings because of the integration of NMSs and the control interface of the devices, thereby increasing automation, and *c*) increased network agility by using the open interfaces to program network devices directly [3].

1.1 Research Question

It is unclear however if a real-world OpenFlow and SDN implementation will actually provide any simplicity, additional flexibility or cost savings when compared to contemporary technologies [4]. Indeed, the technologies in use today have served operators well up until this point and their practicality has been proven over the past years. This research will seek to identify where exactly operators can benefit from implementing this use-case using SDN compared to the architecture in use today.

This research offers that – given the use case as defined in Section 1.2 – OpenFlow will reduce the complexity in the architecture of the management systems and the network as a whole. To prove this, we will need to answer the question: *"How much can operators benefit from using OpenFlow when implementing Dynamic VPNs?"*

1.2 Scope

Dynamic VPNs (DVPNs) are private networks over which end-users can communicate, deployed by their common Service Provider (SP). They differ from normal Virtual Private Networks (VPNs) in the sense that they are relatively short-lived. Using DVPNs, SPs can react more swiftly to customer requests to configure, adjust or tear down their VPNs. This research will prove if such a service can be implemented using contemporary technologies. And, if so, what such a network will look like with regards to the protocols needed.

More importantly, we will compare the characteristics of implementing such an environment using both available technologies and an SDN solution. The focus will primarily be on deploying Provider-provisioned VPNs (PPVPNs) at Layer 2 of the OSI-model between end-users. We haven chosen to do so because these Ethernet VPNs are characterized by their transparency to the end-user, who will be placed in a single broadcast domain with its peers and can thus communicate directly without configuring any sort of routing.

Previous research in [5] has proposed a very specific implementation for programmable networks to deploy on-demand VPNs but it predates the OpenFlow specification, and also omits a comparison with how this would look using contemporary technologies.

1.3 Approach

In the Section 2 we will define the conceptual design of DVPNs. This will result in a list of required features for the technologies to provide such a service. Section 3 will list the technologies available and will additionally determine their usability for implementing DVPNs when taking into account the requirements set forth in Section 2. In Section 4 we will distill the advantages and limitations of the different implementations and substantiate how the compare to each other. Finally, Section 5 summarizes the results and provides a discussion and future work on this subject.

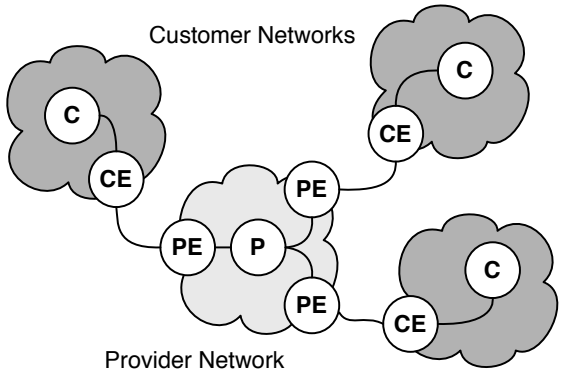
2 Dynamic VPNs

In Section 1.2 we already gave a short description of DVPNs. In this section, we will further look at the actual concept. Starting with defining what it actually provides, how it's carried over the core, the information needed to implement a VPN, from where that information is available and finally working towards a list of technical requirements that the network will need to provide.

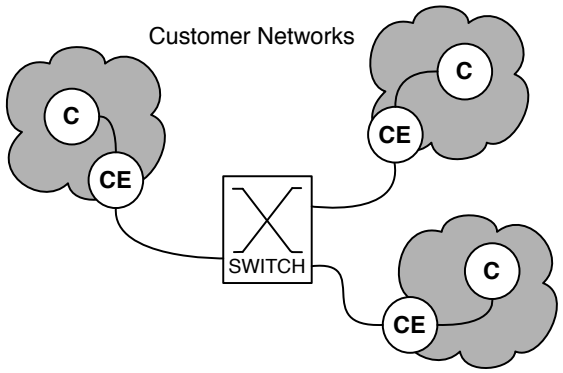
2.1 Service

To define the DVPN service, we first take a look at the concepts of non-dynamic, or static VPNs. They can be classified depending on the OSI layer which it virtualizes, the protocol that is being used and the visibility to the customer. In an IPSec VPN for example, the customer needs to setup his Customer Edge devices (CEs) at each site to actually establish the Layer 3 IP VPN. As we have already established in Section 1.2, we limit the use-case to an multi-point Ethernet Layer 2 VPN which is provisioned by the provider (PPVPN) and thus requires no action on the CE. Throughout this paper the definition of PPVPN related terms will be used as described in RFC 4026 [6] and an overview is given in Figure 1a.

What a Layer 2 PPVPN provides to the CE is a transparent connection to one or more other CEs using a single Ethernet broadcast domain. Another term to describe such a VPN service is a Virtual Private LAN Service (VPLS). It enables the interconnect of several LAN segments over a seemingly invisible carrier network. To do so, the Provider Edge device (PE) needs to keep the Customer MACs (C-MACs) ahead of the frame intact and also support the forwarding of broadcast and multicast traffic. All PEs (and of course Provider devices (Ps)) will not be visible to the CE, who will regard the other CEs as part of the VPLS as direct neighbors on the network as illustrated in Figure 1b.



(a) Terminology used to describe PPVPN devices.



(b) Appearance of VPLS from customer point-of-view.

Figure 1: Visualization of used terminology.

All these functionalities apply to VPNs as well as DVPNs. DVPNs however, also are flexible in nature. They can be configured, adapted and deconfigured within relatively short timespans. Current Layer 2 VPNs are mostly configured statically and changes in their configurations will require manual labor from the engineers. To convert them to DVPNs new tools are needed to automate this provisioning process which we will get back to in Section 2.3.

2.2 Transport

Transporting a Layer 2 frame between two CEs starts at the PE. The ingress PE learns the Source Address (SA) of the frame behind the port connected to the CE, then it needs to forward the frame to the PE where the Destination Address (DA) is present. It will need to do so while separating the traffic from other DVPNs,

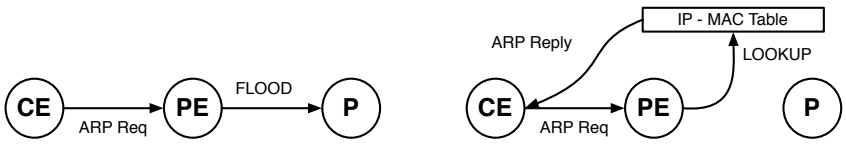
it has to make the traffic unique and identifiable from the rest of the VPNs transported over the network. This is done by giving the frame some sort of 'color' or 'tag' specific to the customer VPN. Additionally it should presume the that P devices are not aware of the DVPN and do not learn the C-MAC addresses. This is because the network will have to scale to thousands of DVPNs and possibly millions of C-MACs divided over those DVPNs. To provide this so called MAC Scalability, only PEs should learn the C-MACs.

Forwarding from ingress PE to egress PE happens over a path of several Ps. Every PE connected to a CE member of a particular DVPN, should have one or more paths available to each and every other PE with members of that DVPN. The determination of the routes of these paths takes place through a form topology discovery. This mechanism should dynamically find all available PEs and Ps with all the connections between them and allow for the creation of paths which are not susceptible to infinite loops.

The links comprising the paths have a certain capacity which will need to be used as efficiently as possible. This means that the links comprising a path will need to have enough resources available, but that other links need not be left vacant. Also, if the required bandwidth for a DVPN exceeds the maximum capacity of one or more of the links in a single path, a second path should be installed to share the load towards the egress PE.

Continuing with the processing of the ingress customer frame, when it arrives at the ingress PE with a DA unknown to the PE, the frame will be flooded to all participating PEs. Upon arrival there, the egress PE stores the mapping of the frames SA to the ingress PE and if it knows the DA will forward out the appropriate port. Because this is a virtual broadcast domain, all Broadcast, Unknown unicast and Multicast (BUM) traffic will need to be flooded to the participating PEs. To limit the amount of BUM traffic in a single DVPN rate limits or filters will need to be in place to prevent the DVPN from being flooded with it.

Another addition to rate limiting unknown unicast traffic is by pre-populating the MAC tables of the PEs. This requires that, besides the ingress PE only learning the SA from the CE, it will also actively distribute the SA to all other PEs with members of the same DVPN. Then, instead of flooding unknown unicast frames to the PEs, the ingress PE drops the frame, knowing that the other PEs will not recognize it either. This can also be extended to limit broadcast Address Resolution Protocol (ARP) traffic if the PEs also exchange the Internet Protocol (IP) address belonging to each C-MAC. When the ingress PE receives an ARP request for a certain IP address, it can look it up in its table and without flooding the frame, reply to the CE with the correct Media Access Control (MAC).



(a) Normal ARP Request. (b) ARP proxy in PE.

Figure 2: Processing of ARP requests at the PE.

With multiple DVPNs present on the network it can happen that one DVPN affects the available bandwidth of others. Therefor rate limits will need to be in place for the overall traffic coming in to the CE-connected ports. Policing rates of different DVPNs in the core is nearly impossible, the hardware cannot police traffic of separate DVPNs. And, because it burdens the core with another responsibility while it should only be concerned with fast forwarding, is also undesirable. However, by assigning a minimum and maximum bandwidth rate to each DVPN instance, it is possible to preprovision the paths over the network according to the required bandwidth. By also monitoring the utilization of individual links, DVPN paths can be moved away from over-provisioned links while they are in use. However, the impact on traffic when performing such a switch must be minimized and should ideally last no longer than 50 ms.

To monitor and troubleshoot large carrier networks Operations, Administration and Management (OAM) functionalities need to be supported by the network. Monitoring end-to-end activity needs to available through automatic continuity check messages, but also by supporting 'traceroutes' through the network manually. This enables the network react proactively to network failures by using a similar method as

presented above when switching DVPNs to a different path, also known as 'fast failover.'

2.3 Provisioning

Before implementing a DVPN in the network, the network first has to become converged. Meaning that the topology of the complete network is known and that paths can be created over this topology.

A DVPN instance consists of multiple member ports, which are identified by their PE device and the port on that PE. The instance also contains values for its minimum and maximum available bandwidth which can be used to determine the paths that the DVPN will get assigned. When member ports reside on different PEs, a bidirectional path will need to be created through the network. The route of the path will depend on *a*) the liveness and administative availability of the links, *b*) the administrative costs of the links, and *c*) the resources available on the links towards the PE. The exact algorithm used to choose the paths lies outside of the scope of this document. Paths are defined by the physical ports that they traverse through the network, with the PEs as the first or last in the list.

When the path between two PEs has been setup for the DVPN, it can be put in the DVPN description. More paths may be added over different routes and paths may be adjusted during the lifetime of the DVPN. This may for example be necessary when a certain link in the path fails, or when it nears its peak capacity and has to be rerouted. A simplified but complete information base diagram has been given in Figure 3. Individual port utilization will be monitored and when a certain link shows high utilization, the corresponding paths and DVPNs using those paths can be looked up using the information base. Also other monitoring and troubleshooting processes will profit from this information.

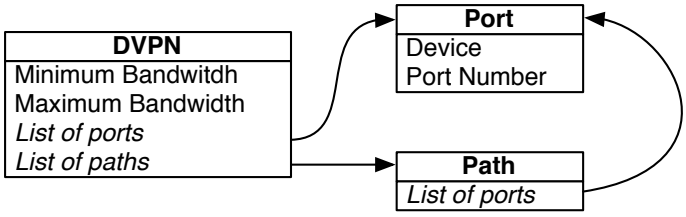


Figure 3: Information base of a DVPN.

After the complete paths between PEs with DVPN members have been setup the traffic can start flowing. However, as has been mentioned before, the rate limiting feature will need to be applied to the ingress ports to prevent the DVPN from using up all the networks resources.

2.4 Requirements

To summarize, to be qualified to deploy DVPNs, networking technologies will need to be able to provide the following features:

1. identify traffic from separate DVPNs by tagging,
2. scalable up to thousands of DVPNs and C-MACs,
3. topology discovery,
4. provision paths over the network,
5. efficient use of, and control over all network resources (Traffic Engineering (TE)),
6. share the load of traffic over multiple paths,

7. rate limiting or filtering of BUM traffic,
8. rate limiting of total DVPN traffic per port,
9. fast failover times (<50ms) to provide continuity to critical applications,
10. provide Operations, Administration and Management features to monitor and troubleshoot the network.

Besides the forwarding specific requirements the provisioning system will also need to be able to:

1. take input as certain ports to be placed in a DVPN,
2. determine routes that can be used for the paths,
3. monitor links and reroute paths on failure or peak capacity,
4. set the rate limits on ingress PE ports.

3 Implementation

Using the requirements set forth in Section 2 we can compile a list of contemporary technologies that can meet them and provide DVPN. The protocols considered had to meet 2 criteria: *1*) can provide Ethernet PPVPNs between multiple sites, and *2*) protocol stack must be supported in hardware at time of writing.

3.1 SPB

Shortest Path Bridging (SPB) is an evolution of the original IEEE 802.1Q Virtual LAN (VLAN) standard. VLAN tags have been in use in the networking world for a long time and provide decent separation in campus networks. However, when VLAN-tagging was done at the customer network, the carrier couldn't separate the traffic from different customers anymore. This resulted in 802.1Qad or Q-in-Q which added an S-VLAN tag to separate the client VLANs from the SP VLANs in the backbone. This was usable for the Metro Ethernet networks for awhile but when SPs started providing this services to more and more customers, their backbone switches could not keep up with the clients MAC addresses.

To provide the required MAC scalability problem Provider Backbone Bridging (PBB) (802.1Qay or MAC-in-MAC) was introduced. It encapsulates the whole Ethernet frame on the edge of the carrier network and forwards the frame based on the Backbone-MAC, Backbone-VLAN and the I-SID. The I-SID is a Service Instance Identifier, which with 24 bits is able to supply the carrier with 16 million separate networks. The downside of PBB remained one that is common to all Layer 2 forwarding protocols: the possibility of loops. Preventing them requires Spanning Tree Protocol (STP) which will disable links to get a loop-free network. Disadvantages of STP include the relatively long convergence time and inefficient use of resources due to the disabled links. This final problem was solved by using IS-IS as a routing protocol to distributed the topology and creating Shortest Path Trees (SPTs) originating from each edge device. This is called SPB or 802.1aq.

3.2 MPLS

Multi Protocol Label Switching (MPLS) is known for its scalability and extensibility. Over the past decade additions have been made to the original specification to overcome a plethora of issues within carrier networks. This initially started with trying to implement fast forwarding in legacy switches using labels (or tags) at the start of the frame [7]. When this issue became surmountable using new hardware, MPLS had already proven to be capable of transporting a wide arrange of protocols on the carrier backbone network, all the while also providing scalability, TE and Quality of Service (QoS) features to the operators.

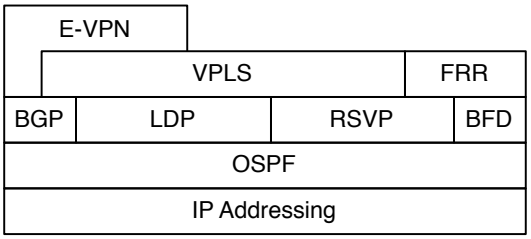


Figure 4: Dependency stack of MPLS-related technologies.

Table of protocols

MPLS - tag of traffic VPLS - encapsulates Ethernet frames RSVP - distributes labels downstream / ecmp / te LDP - distributes labels upstream / ecmp OSPF - learns topology of network BFD - provide connectivity checks

FRR depends on RSVP RSVP depends on OSPF

	SPB	MPLS	OpenFlow / SDN
Tagging of VPN Traffic	PBB	VPLS	PBB / MPLS
MAC Scalability	yes	yes	yes
Topology Discovery	IS-IS	OSPF	application
Path Provisioning	SPT	RSVP / LDP	application
Traffic Engineering	limited	RSVP	application
Equal Cost Multi Path (ECMP)	limited	yes	yes, using Groups
BUM limiting	dependent on HW	dependent on HW	yes, using Metering
Exchange C-MACs	no	E-VPN (draft)	application
Traffic Rate Limiting	dependent on HW	dependent on HW	yes, using Metering
Fast Failover	no	FRR	yes, using Groups
OAM	802.1ag	LSP Ping / BFD	application
Forwarding Decision	PBB tags	MPLS labels	flow entry
BUM traffic handling	flood	flood	sent to controller

Table 1: Required features and corresponding available technologies.

LDP depends on VPLS VPLS depends on RSVP and MPLS RSVP depends on OSPF and MPLS

BGP depends on E-VPN E-VPN depends VPLS VPLS depends on RSVP RSVP depends on OSPF and MPLS

depends all on MPLS forwarding plane

PBB - tag traffic IS-IS - learns topology of network SPB - ecmp / te

rate limiting, vendor specific

what can provide what function for DVPNs?

3.3 Contemporary Technologies

3.4 OpenFlow

4 Results

4.1 Contemporary Implementations

SPB benefits from the maturity of the Ethernet protocol by reusing protocols for OAM and Performance Measurement (PM) and the fact that only the edges of the network need to be SPB capable – the core switches just need to be able to forward 802.1Qad frames. However, due to its Ethernet-based nature it lacks TE functionalities. The paths that the VPN traffic takes are not easily manageable or customizable and provide limited scalability due to limited amounts of available paths (or trees in this case) that can be configured at this time. This also applies to the ECMP functionalities that are limited by the available paths. However, using extensible Equal Cost Trees (ECT) algorithms future, additional algorithms with multiple paths maybe introduced [8].

The failover of paths that have failures present is not optimized for speed. Although the use of hardware multicast floods allows for claimed convergence of below 100ms, reconvergence times are below par [9].

MPLS itself is more a way of forwarding frames through the network, without facilitating any topology discovery, route determination, resource management, etc. These functions are left to a stack of other protocols. To discover the topology, MPLS relies on an Interior Gateway Protocol (IGP). The distribution of labels is done using Label Distribution Protocol (LDP) and/or Resource Reservation Protocol (RSVP), of which the latter also provides granular TE and QoS functionalities.

VPNs are also provided by additional protocols. Layer 3 VPNs make use of Border Gateway Protocol (BGP) to distribute client prefixes to the edges of the carrier network. The core is only concerned with the forwarding of labels and has now knowledge of these IP prefixes. Layer 2 VPNs make use of LDP and VPLS, a service which encapsulates the entire Ethernet frame and pushes a label to it to map it to a certain separated network. Again, the core is only concerned with the labels and only the edges need to know the clients MAC addresses.

Because of its extensibility the MPLS technology and the added protocols and tools, it is commonly used in Carrier Ethernet Networks (CENs) as an alternative to legacy ATM and SDH networks. With added features such as ECMP, Fast Reroute (FRR) and explicit routing it has proven to be a technology fit for carriers to transport critical application traffic over large networks

4.2 OpenFlow

How did it do?

What are the differences?

Also: access layer intelligence.

5 Conclusion

6 Future Work

Other use cases:

- multi-domain
- mobility
- smart metering

A Acronyms

API Application Programming Interface

ARP Address Resolution Protocol

ATM Asynchronous Transport Method

BFD Bidirectional Forward Detection

BGP Border Gateway Protocol

BUM Broadcast, Unknown unicast and Multicast

CE Customer Edge device

CEN Carrier Ethernet Network

C-MAC Customer MAC

DA Destination Address

DVPN Dynamic VPN

ECMP Equal Cost Multi Path

ECT Equal Cost Trees

FRR Fast Reroute

HW Hardware

IEEE Institute of Electrical and Electronics Engineers

IGP Interior Gateway Protocol

IP Internet Protocol

IS-IS Intermediate System-Intermediate System

LAN Local Area Network

LDP Label Distribution Protocol

LSP Label-switched Path

MAC Media Access Control

MPLS Multi Protocol Label Switching

NMS Network Management System

OAM Operations, Administration and Management

OSI Open System Interconnect

OSPF Open Shortest Path First

PBB Provider Backbone Bridging

P Provider device

PE Provider Edge device

PM Performance Measurement

PPVPN Provider-provisioned VPN

QoS Quality of Service

RSVP Resource Reservation Protocol

SA Source Address

SDH Synchronous Digital Hierarchy

SDN Software Defined Networkinging

SPB Shortest Path Bridging

SPT Shortest Path Tree

SP Service Provider

STP Spanning Tree Protocol

TE Traffic Engineering

VLAN Virtual LAN

VPLS Virtual Private LAN Service

VPN Virtual Private Network

B Bibliography

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[2] "OpenDaylight Project." http://www.opendaylight.org/.

[3] S. Das, G. Parulkar, N. McKeown, P. Singh, D. Getachew, and L. Ong, "Packet and circuit network convergence with openflow," in *Optical Fiber Communication (OFC), collocated National Fiber Optic Engineers Conference, 2010 Conference on (OFC/NFOEC)*, pp. 1–3, IEEE, 2010.

[4] J. Van der Merwe and C. Kalmanek, "Network programmability is the answer," in *Workshop on Programmable Routers for the Extensible Services of Tomorrow (PRESTO 2007), Princeton, NJ*, 2007.

[5] B. Yousef, D. B. Hoang, and G. Rogers, "Network programmability for vpn overlay construction and bandwidth management," in *Active Networks*, pp. 114–125, Springer, 2007.

[6] "RFC 4026: Provider Provisioned Virtual Private Network (VPN) Terminology." http://tools.ietf.org/html/rfc4026.

[7] Y. Rekhter, B. Davie, E. Rosen, G. Swallow, D. Farinacci, and D. Katz, "Tag switching architecture overview," *Proceedings of the IEEE*, vol. 85, no. 12, pp. 1973–1983, 1997.

[8] "RFC 6329: IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging." http://tools.ietf.org/html/rfc6329.

[9] P. Ashwood-Smith, "Shortest Path Bridging IEEE 802.1aq – NANOG 49," 2010. https://www.nanog.org/meeting-archives/nanog49/presentations/Tuesday/Ashwood-SPB.pdf.

Acknowledgements

Thanks to Rudolf Strijkers for his supervision during this project.