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 28, 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. Which will potentially require *a*) new Application Programming Interface (API) calls to be implemented between their OSS and NMS, *b*) their NMS to be able to cope with potentially new protocols, and *c*) added expertise by engineers to define the possible feature interactions and restrictions of these protocols [1]. 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.

To manage resources efficiently in a carrier network operators have been using Virtual Private Networks (VPNs) between customers. By differentiating traffic between VPNs they can control their traffic flow at a granular level. However, the set of interactions between different protocols and management interfaces to them are complex. The provisioning of VPNs requires expertise and a significant amount of changes to those protocols. Until recently operators were not concerned by this rigidness as their networks were in fact primarily static. However, the demand for application specific networks is growing and operators are looking for a more flexible approach in the form of Dynamic VPNs (DVPNs). DVPNs are private networks over which end-users can communicate, deployed by their common Service Provider (SP). They differ from normal 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. However, due to the aforementioned complexity DVPNs services have not been implemented on a large scale.

A potential candidate to solve the complexity of implementing DVPNs is OpenFlow [2] and Software Defined Networkinging (SDN). SDN is a relatively new architecture to allow for the programmability of networks. The architecture is not standardized but a generalized structure has been given in the OpenDaylight project [3] which also includes OpenFlow as can been seen in Figure 1. OpenFlow is a lower level and increasingly supported API protocol 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 [4].

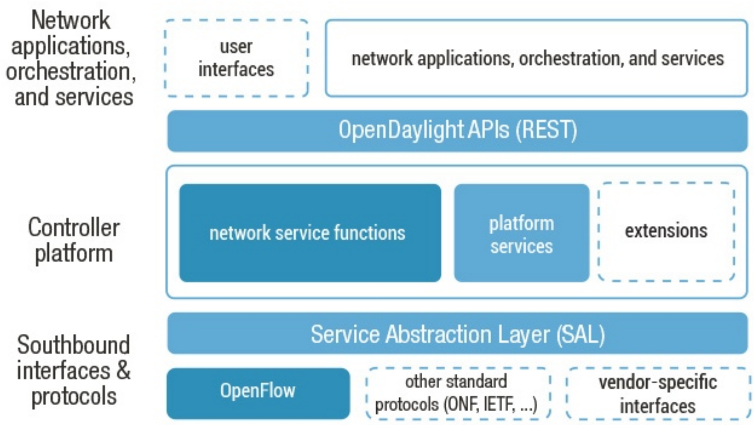


Figure 1: The proposed OpenDaylight architecture.

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 [1]. And so the question arises: *"How much can operators benefit from using OpenFlow when implementing Dynamic VPNs?"*

This requires research into whether DVPNs can be provisioned by current technologies and how this can be implemented in the network. Additionally, we will research the possibility of implementing DVPNs using OpenFlow. And finally we will attempt to prove the hypothesis that OpenFlow will reduce the complexity in the architecture of the management systems and the network as a whole.

1.2 Scope

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.

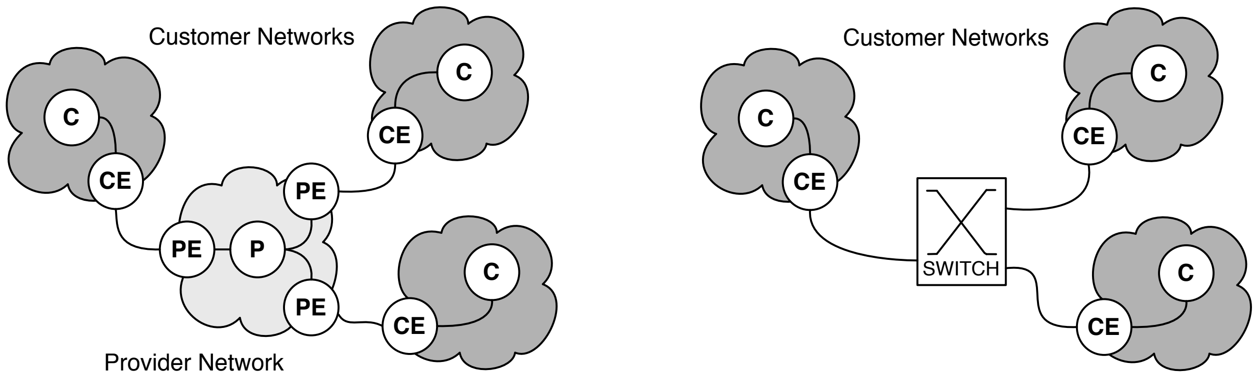
2 Dynamic VPNs

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 2a.

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 2b.



(a) Terminology used to describe PPVPN devices.

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

Figure 2: 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.

To summarize, from a service level perspective a DVPN needs to provide a network to the customer which:

1. provides a Layer 2 broadcast domain,
2. does not require configuration on the CE,
3. is transparent to CEs.

2.2 Transport

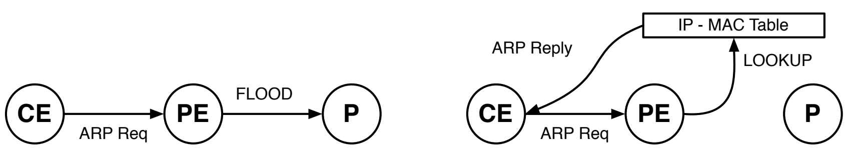
Transporting a Layer 2 frame between two CEs starts at the PE. The ingress PE learns the Source Address (SA) of the frame behind the port connected to the CE, then it needs to forward the frame to the PE where the Destination Address (DA) is present. It will need to do so while separating the traffic 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 3: 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.'

To sum up the requirements discussed in this section, the network needs to provide the following functionalities:

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.

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 4. 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.

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.

To provision DVPNs in the network, the NMS in short should be able to:

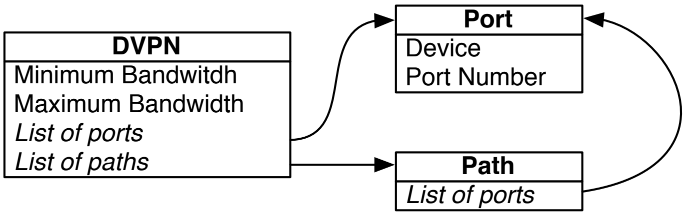


Figure 4: Information base of a DVPN.

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 DVPNs. 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.

what can provide what function for DVPNs?

3.1 Contemporary Technologies

3.1.1 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.

SPB benefits from the maturity of the Ethernet protocol by reusing protocols for OAM and Performance Measurement (PM). This allows for fast error detection and extensive troubleshooting tools by using the Institute of Electrical and Electronics Engineers (IEEE) 802.1ag and International Telecommunication Union - Technology (ITU-T) Y.1731 standards respectively. The Intermediate System-Intermediate System (IS-IS) implementation has also been adapted to rapidly detect errors however, no fast recovery function has been defined, besides complete IS-IS reconvergence. This would result in a traffic impact of several hundreds of milliseconds in large networks [7].

However, due to its Ethernet STP forwarding-based nature it lacks TE features. The paths that the VPN traffic takes are not explicitly configureable and provide limited scalability due to limited amounts of available paths (or trees in this case). This also negatively affects its Equal Cost Multi Path (ECMP) functionalities, which are limited by the available paths as well. However, using extensible Equal Cost Trees (ECT) algorithms, future, additional algorithms with multiple paths maybe introduced [8].

Provided that IS-IS has been configured on all provider devices, Figure 5 illustrates the interfaces needed to provision a DVPN in SPB. First, the I-SID has to be defined on all PEs, which in turn will setup SPTs towards the other participating PEs. The member CE ports then need to be added to the I-SID after which the VPNs is established. The rate limiting of the CE ports is a vendor-specific feature, however and may vary per hardware platform.

3.1.2 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

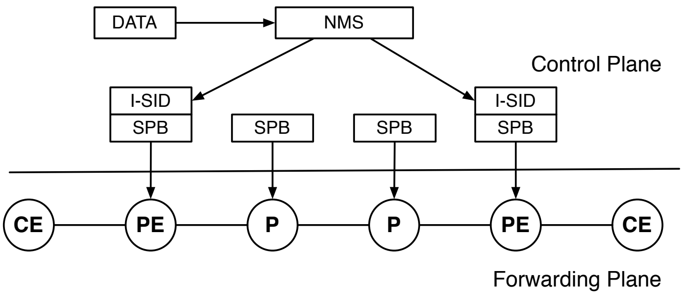


Figure 5: Provisioning a DVPN using SPB.

tags) at the start of the frame [9]. 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.

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. Without IP reachability throughout the network these protocols cannot exchange traffic and so, as a prerequisite, MPLS relies on an Interior Gateway Protocol (IGP) like Open Shortest Path First (OSPF) to discovery the topology.

The distribution of labels has to be facilitated as well, which is done using Label Distribution Protocol (LDP) and/or Resource Reservation Protocol (RSVP). These protocols run between each device in the path between two PEs and exchange the labels that the will assign to a certain path, thereby setting up a Label-switched Path (LSP). LDP does this by distributing its labels from the egress PE up towards the ingress PE based on IGP costs. RSVP, on the other hand, signals its paths from the ingress PE towards the downstream PE based on constraints, potential explicit hops or as a last resort using the IGP next hop. Label distribution is still determined from egress to ingress, but the actual path is determined at the head-end. To determine the best path to take, RSVP uses the Constrained Shortest Path First (CSPF) algorithm which can take into account link characteristics like bandwidth or Fast Reroute (FRR) support. This allows RSVP LSPs to take more well informed paths through the network and together with support for defining explicit paths, allows for granular TE features which LDP lacks. Both LDP and RSVP also allow for the use of multiple paths over the network to share traffic load towards a PE.

The aforementioned FRR feature is unique to RSVP and provides the network with fast failure recovery. It does so by preprovisioning a so-called backup LSP next to the primary LSP. When a failure is detected on the primary LSP, traffic is immediately shifted towards the standby path, yielding a sub-50ms failover. Obviously, this value also depends on the time it takes for the failure to be detected. Therefore it is important to have some sort of detection mechanism in place. One that is commonly used and integrates with OSPF is Bidirectional Forward Detection (BFD). This protocol sets up sessions between devices and triggers an alarm when the session does not behave as expected. At which point FRR kicks in.

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 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. When setting up a VPLS instance (a VPNs), LDP sessions are setup between all PEs part of the same VPLS instance. Consecutively, the PEs will exchange their chosen labels for that instance between each other.

The C-MACs in a VPLS instance are normally learned through the data plane. That is, when a frame comes in from a CE, the PE learns the SA behind the corresponding port. If it doesn't know the DA, it will flood the frame to other PEs with member ports in that instance. These PEs in turn learn the SA as well behind the ingress PE. When a large number of C-MACs are present within a VPLS instance this can cause a lot of broadcast traffic, specifically ARP traffic. To solve this, the Ethernet VPN (E-VPN) standard has been proposed [10]. This technique provides MAC learning in the control plane by exchanging learned C-MACs between PEs using Multi-Protocol BGP. Additionally it may also learn the IP address associated with the C-MAC and distribute that as well. Thereby being able to act as an ARP proxy, as earlier illustrated in Figure 3b.

The different protocols all depend on each other, as illustrated in Figure 6. Each PE device runs this stack, while P devices run a subset which is shaded.

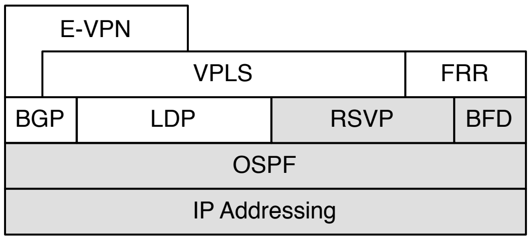


Figure 6: Dependency stack of MPLS-related technologies.

To configure a DVPN using MPLS first the participating PEs need to be configured with the new VPLS instance to which the member CE ports will be added. Next, constraints are defined by the NMS, which can be in the form of an explicit route to make a static route or by defining loose constraints based on bandwidth limits which can be used the CSPF algorithm. Using these constraints, paths are installed at each PE towards every other participating PE. These paths are then added to the VPLS instance, allowing LDP sessions to be setup between the PEs. Next, for FRR, backup LSPs need to defined similarly to the primary LSP but over a different path, which can again be done using constraints to exclude the other links. Utilization of the links in the network has to be monitored as well and when a path has a link which is nearing capacity, new LSPs have to be provisioned and some VPLS paths move to those LSPs. And finally the ingress traffic on the CE ports need to be rate limited. This procedure again, is not standardized and is dependent on support of the hardware.

The procedure above implies that the backbone network has been setup with the following protocols and features already enabled: IP addressing, OSPF routing, MPLS forwarding, RSVP with FRR and BFD. After initial setup of the backbone network the NMS is only concerned with the PEs, as can also be seen in Figure 7.

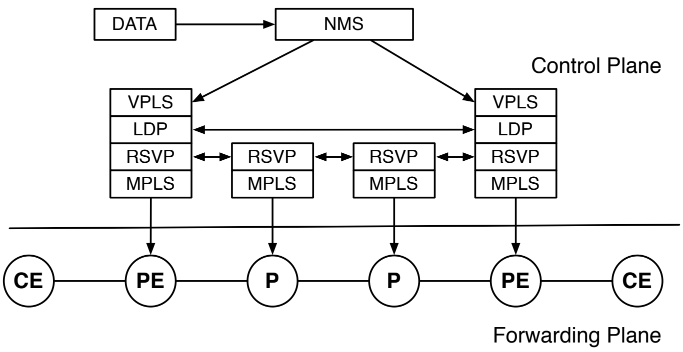


Figure 7: Provisioning a DVPN using MPLS.

3.2 OpenFlow

Section 1 gave a short introduction into what SDN and OpenFlow entail and what it promises in terms of cost savings and agility. Software Defined Networkinging is the general principle of designing flexible networks using open interfaces towards the hardware. The complete architecture is being standardized by the networking industry within the OpenDaylight project [3]. Looking at OpenFlow itself from a more technical point of view, it boils down to a protocol used to program the forwarding plane in networking devices from a centralized server called the 'controller'.

The momentum that SDN is getting might be explained by a general need for change in the networking industry. Operators primarily want to get more control over their networks, something which using the current stack of protocols is relatively complicated to get. The original OSI reference model [11] touches on the "Management Aspects" of each layer in the model, a way for management entities in the highest layer to control the behavior of lower layers. Unfortunately in the swift evolution of Transmission Control Protocol (TCP)/IP, these management interfaces are often limited or absent all together.

The OpenFlow controller provides operators with an alternative to these interfaces, namely a programmable forwarding plane. This gives them TE and QoS control by using custom applications to direct traffic through the network.

The OpenFlow specification has been through a few revision since it was first proposed. In the first version tagging of traffic was only support using a single VLAN tag. Version 1.1 added matches for MPLS and Q-in-Q tags, and version 1.3 could also match PBB tags. These features are essential for implementing VPNs because the traffic flows will need to be separated upon entering the carrier network. Also, with scalability in mind we do not want to learn the client MACs in the core of the network.

Also not included in version 1.0 were mechanisms to allow for fast failover and ECMP. These were added in version 1.1 by introducing logical groups of ports to which a flow entry can forward the frame. However, to support fast failover a *liveness monitoring* technique will need to be implemented supported by the switch. This could simply be the physical link state or more intricate tools, e.g. BFD.

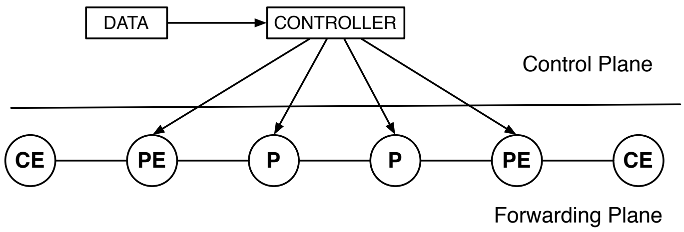


Figure 8: Provisioning a DVPN using OpenFlow.

The features and limitations of the three discussed technologies are given in Table 1. They are compared to the list of requirements as discussed in Section 2. Section 4 will continue with an evaluation of the three architectures.

	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	no	RSVP	application
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.

4 Results

4.1 Contemporary Implementations

Simplicity of setup comes at a cost:

- no TE
- limited ECMP
- no fast failover

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, 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?

	1.0	1.1	1.2	1.3
VLAN Tags	x	x	x	x
Q-in-Q Tags		x	x	x
MPLS Tags		x	x	x
PBB Tags				x
Groups		x	x	x
Metering				x

Table 2: Comparison of OpenFlow version regarding key features for DVPNs.

MPLS VPNs in OpenFlow: [12]

MPLS control plane in OpenFlow: [13]

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

CSPF Constrained Shortest Path First

DA Destination Address

DVPN Dynamic VPN

ECMP Equal Cost Multi Path

ECT Equal Cost Trees

E-VPN Ethernet VPN

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

ITU-T International Telecommunication Union - Technology

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

TCP Transmission Control Protocol

TE Traffic Engineering

VLAN Virtual LAN

VPLS Virtual Private LAN Service

VPN Virtual Private Network

B Bibliography

[1] 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.

[2] 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.

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

[4] 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.

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

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

[9] 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.

[10] A. Sajassi, R. Aggarwal, N. Bitar, W. Henderickx, S. Boutros, F. Balus, K. Patel, S. Salam, A. Isaac, J. Drake, R. Shekhar, and J. Uttaro, "BGP MPLS Based Ethernet VPN," 2013. http://tools.ietf.org/html/draft-ietf-l2vpn-evpn-03.

[11] H. Zimmermann, "OSI reference model–The ISO model of architecture for open systems interconnection," *Communications, IEEE Transactions on*, vol. 28, no. 4, pp. 425–432, 1980.

[12] A. R. Sharafat, S. Das, G. Parulkar, and N. McKeown, "Mpls-te and mpls vpns with openflow," in *ACM SIGCOMM Computer Communication Review*, vol. 41, pp. 452–453, ACM, 2011.

[13] S. Das, A. R. Sharafat, G. Parulkar, and N. McKeown, "Mpls with a simple open control plane," in *Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2011 and the National Fiber Optic Engineers Conference*, pp. 1–3, IEEE, 2011.

Acknowledgements

Thanks to Rudolf Strijkers for his supervision during this project.