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 22, 2013

Summary

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.

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) and is described in RFC 4026 [6]. 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.

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 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.

ARP PROXY?

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 hardware-wise. And, because it burdens the core with an access layer 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 of the P-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.

Related to

OAM

2.3 Provisioning

wat moet je weten uit het netwerk?

wat voor input heb je nodig?

First, we will need to define a certain input. Our design accepts a set of physical ports with an optional C-Virtual LAN (VLAN) corresponding to it. This set represents the group of users which will be placed in a single VPN and thereby connected to each other. It also requires the values for minimum and maximum bandwidth used by DVPN over the network. High bandwidth DVPNs may need load-sharing over multiple physical 1GbE links for example.

Second, the output will be defined as the VPN created throughout the network and Layer 2 connectivity between the chosen endpoints from the input.

Third, the usage of the network resources should be monitored during the lifetime of the DVPN. If certain links are nearing their capacity, DVPNs should be able to move paths to links where more resources are available.

Finally, the tearing down of the DVPN will also need to be arranged so as to free up resources for new DVPNs.

what information from the network? links link utilization

nms information bases: mac/ip address management? dvpns and ports and paths paths and links

AUTOMATION

requirements: tagging - why? ecmp - why? oam - why? te - why? rate limiting - why? (also on BUM traffic) arp proxy - why?

hoe doe je routing?

what information from the network? links link utilization

nms information bases: mac/ip address management? dvpns and ports and paths paths and links

paden: hoe bepaal je ze?

The issue of creating DVPNs within SP networks apparently comes from the inability to do so using technologies available to operators today.

To be qualified to provide network operators with DVPNs each technology will need to be able to provide the following features:

1. scalable up to thousands of (dynamic) Layer 2 VPNs and client MACs,
2. fast failover times (<50ms) to provide continuity to critical applications,
3. efficient use of, and control over all network resources,
4. provide Quality of Service features to differentiate between classes of applications, and
5. an automated way to install VPNs in the network.

3 Implementation

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

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

rate limiting, vendor specific

what can provide what function for DVPNs?

3.1 Contemporary Technologies

3.2 OpenFlow

4 Results

Also: access layer intelligence.

5 Conclusion

6 Future Work

Other use cases:

- multi-domain
- mobility
- smart metering

A Acronyms

API Application Programming Interface

BUM Broadcast, Unknown unicast and Multicast

CE Customer Edge device

C-MAC Customer MAC

DA Destination Address

DVPN Dynamic VPN

IP Internet Protocol

LAN Local Area Network

MAC Media Access Control

NMS Network Management System

OSI Open System Interconnect

P Provider device

PE Provider Edge device

PPVPN Provider-provisioned VPN

QoS Quality of Service

SA Source Address

SDN Software Defined Networkinging

SP Service Provider

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.

Acknowledgements

Thanks to Rudolf Strijkers for his supervision during this project.