UNIVERSITY OF AMSTERDAM

MSC SYSTEMS & NETWORK ENGINEERING

Architecture of a dynamic VPN in OpenFlow

By:
Michiel APPELMAN
michiel.appelman@os3.nl

Supervisor:
Rudolf STRIJKERS
rudolf.strijkers@tno.nl

June 14, 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 Networking (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].

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.

Doing so will help researchers and operators answer the question: *"How much can operators benefit from using OpenFlow when implementing dynamic VPNs?"* This research offers that – with regard to the given use case – OpenFlow will reduce the complexity in the complete architecture of the management systems and network as a whole.

Scope

Given the use-case of implementing dynamic Virtual Private Networks (VPNs) similar to gTBN [5], this research will show the advantages of implementing such an environment using both Commercial Off-The-Shelf (COTS) technologies and an SDN solution. This will include research to determine if the proposed SDN and OpenFlow architecture will actually be able to support the creation of dynamic VPNs. The focus will primarily be on deploying Provider-provisioned VPNs (PPVPNs) at Layer 2 of the OSI-model between end-users. It's one of the more commonly used virtualization models, which is also plagued by its mostly manual and static setup procedure. When configured, users will be immediately connected to each other with minimal setup.

Previous research in [6] 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.

Should there be any time left, other use-cases to fill in this void will be composed in collaboration with the supervisor. These can include for example mobility within networks, multi-domain VPNs or smart metering of network usage.

Approach

We will start by giving a taxonomy of the state of the art in the networking industry with regards to different functionalities and features of the technologies. This overview will give an indication of where OpenFlow and SDN will be able to differentiate from contemporary protocols, be it in positive or negative. Using this data we will make an architectural design of the given use-case using both contemporary technologies and the SDN architecture and compare the effort in implementing such a design.

To quantify the efforts of both designs an overview will be made of the custom APIs and management systems will be made. The procedure to extend on the designs will also be defined, listing the required amount of changes to each architectural component and the expertise needed to implement these changes. Finally this will give an overview of the initial work and required work to adapt for both designs, and result in a recommendation for either one of the deployments.

2 State of the Art

The issue of creating dynamic VPNs within Service Provider (SP) networks apparently comes from the inability to do so using technologies available to operators today. To get an understanding of where the limitations or obstacles lie, an overview of the state of the art is required.

Mention requirements for dynamic VPNs to meet for each technology?

1. scalable up to thousands of VPNs
2. fast failover times (<50ms)
3. control forwarding paths
4. apply Quality Of Service (QoS) values

2.1 MPLS

Multi Protocol Label Switching (MPLS) is known for its scalability and extensibility. Over the past decade addition 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, Traffic Engineering (TE) and QoS features to the operators.

... deep dive

Does it meet criteria? Why (not)?

2.2 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 tag its traffic anymore. This resulted in 802.1Qad or Q-in-Q and 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 Media Access Control (MAC) addresses.

To solve this 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 some sort of Spanning Tree Protocol (STP) which in turn will disable links to get a loop-free network. Disadvantages of these protocols include the relatively long convergence time and inefficient use of resources due to the disabled links. This final problem was solved by using ISIS as a routing protocol to distributed the topology and creating Shortest Path Tree (SPT) originating from each edge device. This is called SPB or 802.1aq.

... deep dive

Does it meet criteria? Why (not)?

2.3 TRILL

There has been discussion going on between SPB and Transparent Interconnection of Lots of Links (TRILL) supporters as to which is the 'better' protocol. Indeed, both try to solve the same problem to make Ethernet networks scalable to the desired scale of today, but they definitely differ in their implementation. TRILL adds a completely new header on top of the Ethernet frame with a source and destination RBridge, this allows the RBridges to actually route the frame to its destination over the TRILL backbone network.

... deep dive

Does it meet criteria? Why (not)?

2.4 SDH/SONET

What is it?

Does it meet criteria? Why (not)?

2.5 OpenFlow

What is it? Relation to SDN.

The momentum comes from a general problem with control over the network (Zimmerman OSI)

3 Design

4 Results

5 Conclusion

6 Future Work

A Acronyms

API Application Programming Interface

COTS Commercial Off-The-Shelf

IEEE Institute of Electrical and Electronics Engineers

ISIS Intermediate System-Intermediate System

LAN Local Area Network

MAC Media Access Control

MPLS Multi Protocol Label Switching

PBB Provider Backbone Bridging

PPVPN Provider-provisioned VPN

OSI Open System Interconnect

QoS Quality Of Service

SDN Software Defined Networking

SDH Synchronous Digital Hierarchy

SPB Shortest Path Bridging

SONET Synchronous Optical Network

SP Service Provider

STP Spanning Tree Protocol

SPT Shortest Path Tree

TE Traffic Engineering

TRILL Transparent Interconnection of Lots of Links

NMS Network Management System

VLAN Virtual LAN

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] M. L. Cristea, R. J. Strijkers, D. Marchal, L. Gommans, C. de Laat, and R. J. Meijer, "Supporting communities in programmable grid networks: gTBN," in *Integrated Network Management, 2009. IM'09. IFIP/IEEE International Symposium on*, pp. 406–413, IEEE, 2009.

[6] 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.

[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.

Acknowledgements

Thanks to Rudolf Strijkers for his supervision during this project.