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 21, 2013

Summary

Contents

Summary **i**

1 Introduction **1**
1.1 Research Question 1
1.2 Scope 1
1.3 Approach 2

2 Dynamic VPNs **3**

3 Implementation **4**

4 Results **5**

5 Conclusion **6**

6 Future Work **6**

Appendices

A Acronyms **7**

B Bibliography **8**

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 L2VPNs 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

wat zijn het? waarom?

wat moet je weten uit het netwerk?

wat voor input heb je nodig?

waarom moet je taggen?

hoe doe je routing?

mac/ip adres management?

paden: hoe bepaal je ze? overzicht van paden?

qos: hoe doe je rate limiting?

hoe beheer je BUM traffic?

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 VPLS RSVP LDP OSPF BFD SPB

what can provide what function for DVPNs?

4 Results

5 Conclusion

6 Future Work

Other use cases:

- multi-domain
- mobility
- smart metering

A Acronyms

API Application Programming Interface

DVPN Dynamic VPN

MAC Media Access Control

NMS Network Management System

OSI Open System Interconnect

PPVPN Provider-provisioned VPN

QoS Quality of Service

SDN Software Defined Networkinging

SP Service Provider

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.

Acknowledgements

Thanks to Rudolf Strijkers for his supervision during this project.