Architecture of a dynamic VPN in OpenFlow

Research Proposal

Michiel Appelman

June 8, 2013

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 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 facy 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 Application Programming Interface (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 VPNs similar to gTBN [5], this research will show the advantages of implementing such an environment using both 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.

Should there be any time left, other use-cases will be composed together with the supervisor to fill in this void. 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 Planning

This project will be carried out over a period of four weeks at the SNE Research Group at the University of Amsterdam. A rough timeline of the research is given in the table below:

Week	Task

1	Write proposal;
	Formulate research questions;
	Find documentation of other research/experiences within community;
	Get approval for proposal;

2 & 3	Start research; Work out use-case, hypotheses and experiment; Produce proof of concept;

4	Write report;
	Prepare presentation;
	Filter report to a scientific paper;

Products

The research will result in the following products: *a*) a report written for the System and Network Engineering master education describing the research, the process and results, and *b*) a 20 minute presentation about the research to be held on July 3rd at the University of Amsterdam. Also, if preliminary results and available time allow for it, a proof of concept will be produced to provide the community with a sample implementation to test and adapt.

References

[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.