POSTER: Performance Management in Application-controlled Software Defined Networks

Jeremias Blendin
Peer-to-Peer Systems Engineering Lab, TU Darmstadt
Email: jblendin@ps.tu-darmstadt.de

David Hausheer
Peer-to-Peer Systems Engineering Lab, TU Darmstadt
Email: hausheer@ps.tu-darmstadt.de

***Abstract*—Software Defined Networking (SDN) is an emerging new paradigm that allows for an application-driven control of network resources. Traditional network management systems (NMS) can benefit from an application-controlled SDN approach, which promises to drastically reduce OPEX by translating application requirements into network behavior without human intervention. However, delivering predictable performance for applications remains a challenge as it requires such systems to maintain and forecast the resource utilization and performance of the underlying network. Therefore, this paper proposes a new application-controlled NMS approach for SDN, considering the resource and performance characteristics of the underlying network. The goal is to model the network utilization and to forecast the usage profiles of applications to prevent resource shortages before they occur. The proposed concept focusses on OpenFlow due to its generic approach, well defined resource requirements, and prevalence in scientific research.**

I. Introduction

The recent uptake of Software Defined Networking (SDN) in research and industry opens up new possibilities in network control and management. In contrast to the traditional networking model, with SDN, the control and management planes are decoupled from the data plane and, therefore, device independent. A logically centralized SDN control plane or network operating system (NOS) acts as single point of configuration for an entire network domain. Today, network management systems (NMS), being located above the control plane in the management hierarchy, often consist of purpose-built software and manual processes [1]. This leaves room for improving the efficiency in the management plane by using NOS and standard APIs to dynamically control the network domain's behavior through software [2].

An SDN-based approach to network management is Application-controlled NMS (ANMS), which aims at translating application requirements automatically into network behavior. The advantages are reduced OPEX, easier application logic deployment, and more flexible network management. The process of short term network management functionality can be implemented in software, leaving only the setup, design, and scaling to human intervention. This paper proposes a new performance management framework that aims at enabling stable and predictable performance in ANMS.

This work has been supported in parts by the European Union (FP7/#317846, SmartenIT and FP7/#318398, eCOUSIN) and the German DFG (CRC 1053, MAKI).

Multiple applications specify their requirements, which are collected by the ANMS and merged into a single specification. Based on this, the NOS and thereby the network's behavior is configured. For example, a live video streaming application issues a requirement to be accessible by all clients in a network to the ANMS. When a client requests a specific video stream, the application issues requirements, leading to corresponding changes in the network. OpenFlow [3], as one implementation of SDN, is used in this paper due to its adoption in research and industry, as well as its generic and well defined approach to data plane programming.

The remainder of this paper is structured as follows: Section II derives requirements and discusses the related work. Following these requirements, Section III presents the proposed ANMS approach and describes first results to date. Finally, Section IV concludes the paper.

II. Requirements and Related Work

The requirements for an ANMS are (1) a generic API that is (2) agnostic of the topology and size of the network domain. Resource management is a fundamental task of network management systems in order to achieve stable operations. The understanding of the relationship between the requirements of an application and the resulting resource requirements of the corresponding network configuration enables predictable behavior of the system. Hence, further requirements are (3) to deliver predictable performance and resource consumption in the data and control plane [4] and (4) to adapt to the needs of the network operator. Resources in the control plane are memory and processing capacity; relevant performance metrics are the control message throughput and reaction speed. In the data plane, resources are memory, e.g., OpenFlow rules, as well as forwarding capacity; performance metrics are data packet and control message throughput.

The PANE approach [5] implements an ANMS for OpenFlow networks and fulfills requirement (1). PANE compiles requirements by users and applications into a policy tree from which a consistent hierarchical policy tree for the entire network domain is created. The focus of PANE is the decomposition of control and visibility as well as the resolution of resource conflicts. However, associating application requirements with their corresponding data plane resource usage is not part of the system. Therefore, it is difficult to predict the system's resource usage and performance. While the PANE

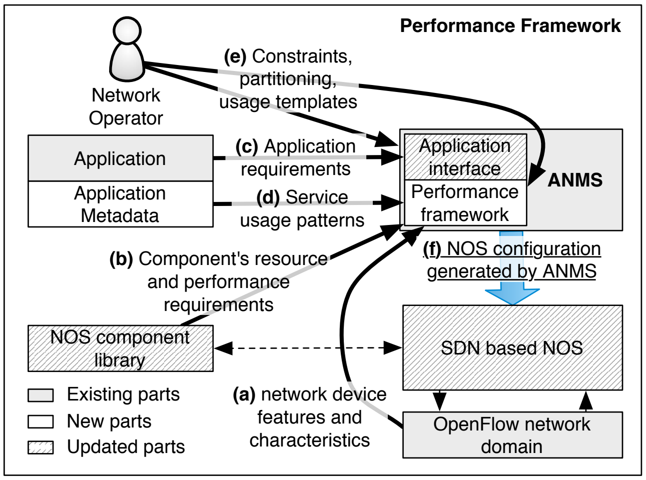

Fig. 1. Overview on the information flow in the proposed framework.

system is a promising approach to handle application resource requests the system is not resource aware. Therefore, it is difficult to adapt the system to different resource environments which range from small business networks to ISP networks. DIFANE [6] recognizes the issue of resource restrictions in the data plane and describes an approach to use OpenFlow rules efficiently. Closely related to the resource usage of networks and predictability of the performance is the notion of scalability. The necessity of exploring scalability in the context of the new opportunities of SDN is described in [4]. Furthermore, many papers, e.g. [7], [8], focus on improving the performance and scalability of control plane. While PANE meets requirement (1), none of the available approaches meets, in addition to (1), the requirements (2), (3), and (4). Therefore, an approach for a performance framework that enables ANMS to meet these requirements in addition to (1) is described in this paper.

III. An SDN Performance Management Framework

From the ANMS requirements described above features, information flow, and information sources of such a system are derived and depicted in Figure 1. Requirement (2) demands the ANMS API to be agnostic of the underlying networks' topology and size. Therefore, a model of the underlying network domain is needed, that describes the network domain's size, its resources, and its performance characteristics, labeled (a) in Figure 1. A component-based NOS is considered, similar to Corybantic [9]. The network behavior is defined through the combination and configuration of NOS components that implement network functions such as forwarding, access control lists, network address translation, and multicast. For each of these components, information on their performance and resource consumption is required (b). Delivering predictable performance should be achieved by using this information and matching them with the actual requirements which are specified by applications (c) and processed by an NMS component like PANE. Furthermore, the system should anticipate the usage of applications over time. Therefore, application meta data such as the expected usage profiles (d) are needed. This is important, considering that the operation environment can range from small company networks to data center networks or even global ISP networks. Information on the operation environment of an ANMS is used to create NOS configurations with appropriate scaling characteristics. For ensuring that the network operator is in control (4), resource constraints, and resource usage templates can be provided (e). This information needs to be considered as well by the ANMS during the generation of the NOS configuration (f), which manifests itself in the network domain's behavior.

A number of research questions result from these considerations. First, a performance and resource model and an associated specification language is required. Its purpose is to describe the characteristics of network devices, network functions and their implementation, operator-specified resource usage templates and constraints, as well as the requirements of applications. Finally, an ANMS performance component for evaluating and merging the resource requirements, invariants, and available resources into a valid network configuration is needed.

First experiences with the development of application level interfaces for a network layer multicast service in OpenFlow domains are described in [10] and demonstrated in [11]. The service named Software Defined Multicast (SDM) enables overlay networks to exploit the advantages of network layer multicast. The main results are that application interfaces are advantageous for cross-layer interaction and that the scalability properties of each OpenFlow NOS component has to be investigated in detail before deploying it, since the performance of OpenFlow software switches varies by several orders of magnitude. Therefore, in order to manage an OpenFlow domain for consistent performance, the NMS needs information on the resource and performance characteristics of the devices used. For best results, the NOS components' implementation should be adapted to the operation environment.

IV. Preliminary Conclusion and Future Work

For ANMS to deliver predictable performance, a model of the available data plane resources, the available means for implementing network behavior, and the expected application usage profiles are required. The ANMS has to offer an API for applications and for the network operator that enables them to specify their requirements. These are used by the ANMS to configure the NOS consistently, reliably, efficiently, and adapted to its usage environment. Looking further, the approach to a performance framework described in this paper should build the basis for creating general SDN scalability and performance abstractions. Next steps are the introduction of a general application level requirements language based on the SDM application API introduced in [10]. Furthermore, different multicast distribution approaches with different scalability characteristics will be developed to study the effects of using a single application requirements interface in network domains of different sizes.

References

[1] L. Velasco, A. Castro, D. King, O. Gerstel, R. Casellas, and V. López, "In-Operation Network Planning," *IEEE Communications Magazine*, Vol. 52, No. 1, pp. 53–60, 2014.

[2] H. Kim and N. Feamster, "Improving Network Management with Software Defined Networking," *IEEE Communications Magazine*, Vol. 51, No. 2, pp. 114–119, 2013.

[3] 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.

[4] S. H. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On Scalability of Software-Defined Networking," *IEEE Communications Magazine*, Vol. 51, No. 2, pp. 136–141, 2013.

[5] A. D. Ferguson, C. Liang, and R. Fonseca, "Participatory Networking: An API for Application Control of SDNs," in *ACM SIGCOMM*, 2013.

[6] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, "Scalable Flow-Based Networking with DIFANE," in *ACM SIGCOMM*, 2010.

[7] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker, "Onix : A Distributed Control Platform for Large-scale Production Networks," in *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2010.

[8] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "DevoFlow: Scaling Flow Management for High-Performance Networks," in *ACM SIGCOMM*, 2011.

[9] J. C. Mogul, A. Auyoung, S. Banerjee, L. Popa, J. Lee, J. Mudigonda, P. Sharma, and Y. Turner, "Corybantic: Towards the Modular Composition of SDN Control Programs," in *ACM Workshop on Hot Topics in Networks (HotNets)*, 2013.

[10] J. Blendin, "Cross-layer Optimization of Peer-to-Peer Video Streaming in OpenFlow-based ISP Networks," Diploma Thesis, Technische Universität Darmstadt, 2013.

[11] J. Rückert, J. Blendin, and D. Hausheer, "RASP: Using OpenFlow to Push Overlay Streams into the Underlay," in *IEEE Conference on Peer-to-Peer Computing (P2P)*, 2013.