Computer Networks 71 (2014) 1–30

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

# A roadmap for traffic engineering in SDN-OpenFlow networks

Ian F. Akyildiz [a], Ahyoung Lee [a,*], Pu Wang [b], Min Luo [c], Wu Chou [c]

[a] Broadband Wireless Networking Lab, School of Electrical & Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[b] Department of Electrical Engineering and Computer Science, Wichita State University, Wichita, KS 67260, USA
[c] Shannon Lab, Huawei Technologies Co., Ltd., Santa Clara, USA

ARTICLE INFO

*Article history:*
Available online 19 June 2014

*Keywords:*
Software-defined networking
OpenFlow
Traffic engineering
Traffic management
Traffic analysis

ABSTRACT

Software Defined Networking (SDN) is an emerging networking paradigm that separates the network control plane from the data forwarding plane with the promise to dramatically improve network resource utilization, simplify network management, reduce operating cost, and promote innovation and evolution. Although traffic engineering techniques have been widely exploited in the past and current data networks, such as ATM networks and IP/MPLS networks, to optimize the performance of communication networks by dynamically analyzing, predicting, and regulating the behavior of the transmitted data, the unique features of SDN require new traffic engineering techniques that exploit the global network view, status, and flow patterns/characteristics available for better traffic control and management. This paper surveys the state-of-the-art in traffic engineering for SDNs, and mainly focuses on four thrusts including flow management, fault tolerance, topology update, and traffic analysis/characterization. In addition, some existing and representative traffic engineering tools from both industry and academia are explained. Moreover, open research issues for the realization of SDN traffic engineering solutions are discussed in detail.



## 1. Introduction

Traffic engineering (TE) is an important mechanism to optimize the performance of a data network by dynamically analyzing, predicting, and regulating the behavior of the transmitted data. It has been widely exploited in the past and current data networks, such as ATM and IP/MPLS networks. However, these past and current networking paradigms and their corresponding TE solutions are unfavorable for the next generation networking paradigms and their network management due to two main reasons. First, today's Internet applications require the underlying network architecture to react in real time and to be scalable for a large amount of traffic. The architecture should be able to classify a variety of traffic types from different applications, and to provide a suitable and specific service for each traffic type in a very short time period (i.e., order of *ms*). Secondly, facing the rapid growth in cloud computing and thus the demand of massive-scale data centers, a fitting network management should be able to improve resource utilization for better system performance. Thus, new networking architectures and more intelligent and efficient TE tools are urgently needed.

The recently emerged Software Defined Networking (SDN) [1,2] paradigm separates the network control plane from the data forwarding plane, and provides user applications with a centralized view of the distributed network states. It includes three layers and interactions between layers as shown in Fig. 1. The details of the SDN architecture overview are explained as follows: There may be more than one SDN controller if the network is large-scale or a

* Corresponding author. Tel.: +1 404 894 6616.
*E-mail addresses:* ian@ece.gatech.edu (I.F. Akyildiz), ahyoung.lee@ece.gatech.edu (A. Lee), pu.wang@wichita.edu (P. Wang), min.ch.luo@huawei.com (M. Luo), wu.chou@huawei.com (W. Chou).

http://dx.doi.org/10.1016/j.comnet.2014.06.002
1389-1286/

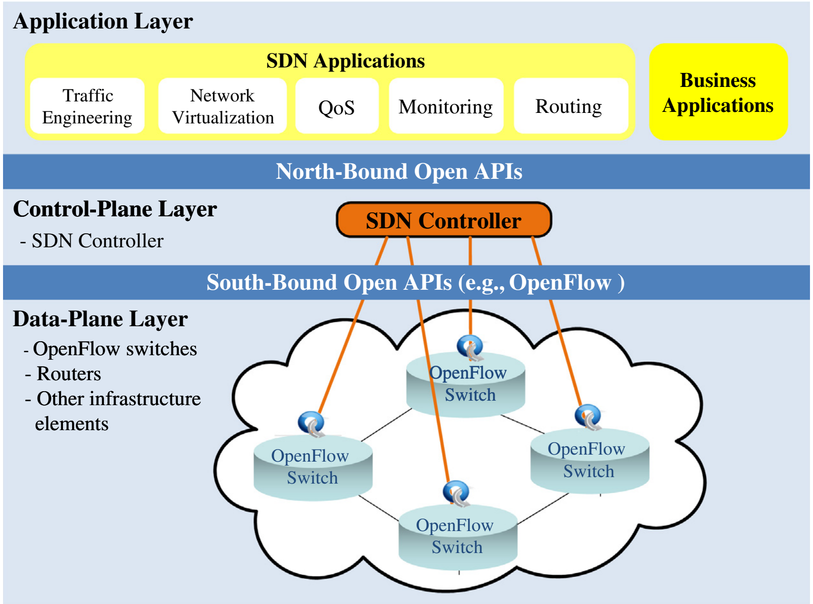


**Fig. 1.** Overview of SDN architecture.

wide-area region network. The control layer globally regulates the network states via network policies in either a centralized or distributed manner. Due to the unrestricted access to global network elements and resources, such network policies can be updated timely to react to the current flow activities. Furthermore, SDN applications exist in the application layer of the SDN architecture. A set of application programming interfaces (such as North-bound Open APIs) are supported to communicate between the application layer and the control layer in order to enable common network services, such as routing, traffic engineering, multicasting, security, access control, bandwidth management, quality of service (QoS), energy usage, and many other forms of the network management. In other words, these interfaces facilitate various business objectives in the network management. On the other hand, the data forwarding layer can employ programmable OpenFlow switches through OpenFlow controller, and the switches communicate with the controller via South-bound Open APIs (e.g., OpenFlow protocol) [1]. The OpenFlow (OF) protocol provides access to the forwarding plane of a network switch over the network and enables software programs running on OF switches to perform packet lookups and forwarding the packets among the network of switches or routers. These programmable switches follow the policies of the SDN/OF controller and forward packets accordingly in order to determine what path the packets will take through the network or switches or routers. In short, through the interactions among these layers, the SDN paradigm allows an unified and global view of complicated networks, and thus provides a powerful control platform for the network management over traffic flows. In the literature, most of the work so far is focused on developing the SDN architecture and with less effort on developing TE tools for SDN. While current TE mechanisms are extensively studied in ATM networks, IP-based and MPLS-based Internet, it is still unclear how these techniques perform under various traffic patterns, and how to obtain the enormous traffic and resource information efficiently in the entire network when the SDN is deployed. On the other hand, SDN promises to dramatically simplify the network management, reduce operating costs, and promote innovation and evolution in current and future networks. Such unique features of SDN provide great incentive for new TE techniques that exploit the global network view, status, and flow patterns/characteristics available for better traffic control and management. Therefore we first briefly discuss the classical TE mechanisms developed for ATM, IP and MPLS networks, and then survey in detail the state-of-the-art in TE for SDN from both academia and industry perspectives. Then, we examine some open issues in TE for SDN, and review some recent progresses in extending traditional TE techniques for SDN networks.

The remainder of the paper is organized as follows. Early TE issues and mechanisms based on ATM, IP and MPLS networks are given in Section 2. An overview of SDN traffic engineering solutions is provided in Section 3. From Section 4 to Section 7, the major SDN traffic engineering technologies, including flow management, fault tolerance, topology update, and traffic analysis, are presented, respectively. Existing TE tools for SDN with OF switches are further introduced in Section 8. The paper is concluded in Section 9.

## 2. Lessons learned from the past

Traffic engineering (TE) generally means that the network traffic is measured and analyzed in order to enhance