Computer Networks 127 (2017) 109–125

Contents lists available at ScienceDirect

Computer Networks

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

Achieving minimum bandwidth guarantees and work-conservation in large-scale, SDN-based datacenter networks

Daniel S. Marcon[a,b,*], Fabrício M. Mazzola[b], Marinho P. Barcellos[b]

[a] University of Vale do Rio dos Sinos (UNISINOS), RS, Brazil
[b] Federal University of Rio Grande do Sul (UFRGS), RS, Brazil

ARTICLE INFO

Article history:
Received 31 January 2017
Revised 8 July 2017
Accepted 14 August 2017
Available online 16 August 2017

Keywords:
Datacenter networks
Software-Defined Networking
Network sharing
Performance interference
Bandwidth guarantees
Work-conservation

ABSTRACT

Performance interference has been a well-known problem in datacenters and one that remains a constant topic of discussion in the literature. Software-Defined Networking (SDN) may enable the development of a robust solution for interference, as it allows dynamic control over resources through programmable interfaces and flow-based management. However, to date, the scalability of existing SDN-based approaches is limited, because of the number of entries required in flow tables and delays introduced. In this paper, we propose Predictor, a scheme to scalably address performance interference in SDN-based datacenter networks (DCNs), providing minimum bandwidth guarantees for applications and work-conservation for providers. Two novel SDN-based algorithms are proposed to address performance interference. Scalability is improved in Predictor as follows: first, it minimizes flow table size by controlling flows at *application-level*; second, it reduces flow setup time by proactively installing rules in switches. We conducted an extensive evaluation, in which we verify that Predictor provides (*i*) guaranteed and predictable network performance for applications and their tenants; (*ii*) work-conserving sharing for providers; and (*iii*) significant improvements over DevoFlow (the state-of-the-art SDN-based proposal for DCNs), reducing flow table size (up to 94%) and having similar controller load and flow setup time.

1. Introduction

Cloud providers lack practical and efficient mechanisms to offer bandwidth guarantees for applications [1–4]. The datacenter network (DCN) is typically oversubscribed and shared in a best-effort manner, relying on TCP to achieve high utilization and scalability. TCP, nonetheless, does not provide robust isolation among flows in the network [5–8]; in fact, long-lived flows with a large number of packets are privileged over small ones [9], a problem called *performance interference* [10–12]. The problem is a long-term challenge, and previous work on the field [1–3,11,13–16] has allowed important advances. In this context, we study how to address the problem in large-scale, SDN-based datacenter networks (DCNs). We aim at achieving minimum bandwidth guarantees for applications and their tenants while maintaining high utilization (i.e., providing work-conserving capabilities) in large DCNs.

Software-Defined Networking (SDN) [17] may enable the development of a robust solution to deal with performance interference, as it allows dynamic control over resources through programmable interfaces and flow-based management [18]. However, to date, the scalability of existing SDN-based approaches is limited, because of the number of entries required in flow tables and delays introduced (mostly related to flow setup time) [18–20]. The number of entries required in flow tables can be significantly higher than the amount of resources available in commodity switches used in DCNs [19,21], as such networks typically have very large flow rates (e.g., over 16 million/s [22]). Flow setup time, in turn, is associated with the transition between the data and control planes whenever a new flow arrives at a switch [1] (latency for communication between switches and the controller), and the high frequency at which flows arrive and demands change in DCNs restricts controller scalability [23]. As a result, the lack of scalability hinders the use of SDN to address interference in large DCNs.

The scalability of SDN-based datacenters could be improved by devolving the control to the data plane, such as proposed by DevoFlow [19] and Difane [24], but deployability is limited since they require switches with customized hardware. Another approach would be using a logically distributed controller, such as proposed

* Corresponding author.
E-mail addresses: daniel.stefani@inf.ufrgs.br, daniel.stefani@gmail.com (D.S. Marcon), fmmazzola@inf.ufrgs.br (F.M. Mazzola), marinho@inf.ufrgs.br (M.P. Barcellos).

[1] We use the terms “switches” and “forwarding devices” to refer to the same set of SDN-enabled network devices, that is, data plane devices that forward packets based on a set of flow rules.

http://dx.doi.org/10.1016/j.comnet.2017.08.008
1389-1286/

by Kandoo [25]. However, it does not scale for large DCNs where communications occur between virtual machines (VMs) connected by different top-of-rack (ToR) switches. This happens because the distributed set of controllers needs to maintain synchronized information (strong consistency) for the whole network. This is necessary in order to route traffic through less congested paths and to reserve resources for applications.

We rely on two key observations to address performance interference and scalability of SDN in DCNs: (*i*) providers do not need to control each flow individually, since they charge tenants based on the amount of resources consumed by applications [2]; and (*ii*) congestion control in the network is expected to be proportional to the tenant's payment (defined in their Service Level Agreements – SLAs) [12,13]. Therefore, we adopt a broader definition of flow, considering it at application-level,[3] and introduce Predictor, a scheme for large-scale datacenters. Predictor deals with the two aforementioned challenges (namely, performance interference and scalability of SDN/OpenFlow in DCNs) in the following manner.

Performance interference is addressed by employing two SDN-based algorithms (described in Section 5.3) to dynamically program the network, improving resource sharing. By doing so, both tenants and providers have benefits. Tenants achieve predictable network performance by receiving minimum bandwidth guarantees for their applications (using Algorithm 1). Providers, in turn, maintain high network utilization (due to work-conservation provided by Algorithm 2), essential to achieve economies of scale.

Scalability is improved in two ways. First, as we show through measurements (Section 3), reducing flow table size also decreases the time taken to install rules in flow tables (stored in Ternary Content-Addressable Memory – TCAM) of switches. In the proposed approach, flow table size is minimized by managing flows at application-level and by using wildcards (when possible). This setting allows providers to control traffic and gather statistics at application-level for each link and device in the network.

Second, we propose to proactively install rules for intra-application communication, guaranteeing bandwidth between VMs of the same application. By proactively installing rules at the moment applications are allocated, flow setup time is reduced (which is important especially for latency-sensitive flows). Inter-application rules, in turn, may be either proactively installed in switches (if tenants know other applications that their applications will communicate with [11] or if the provider employs some predictive technique [26,27]) or reactively installed according to demands. Proactively installing rules has both benefits and drawbacks: while flow setup time is eliminated, some flow table entries may take longer to expire (they might be removed only when their respective applications conclude and are deallocated). Our decision is motivated by the fact that intra-application traffic volume is expected to be the highest type of traffic [12].

In general, Predictor's strategy to address scalability of SDN/OpenFlow in large-scale datacenters presents a trade-off. The benefits are related to reducing flow table size and flow setup time in datacenter networks. Reducing flow table size enables (*i*) providers to use cheaper forwarding devices (i.e., with smaller flow tables); and (*ii*) forwarding devices to install rules in a shorter amount of time (as shown in Section 3.2). Reducing flow setup time greatly benefits latency-sensitive applications. The drawbacks are related to the time rules remain installed in forwarding devices and the ability to perform fine-grained load balancing. First, rules for intra-application communication (i.e., communication between VMs of the same application) are installed when applications are allocated and are removed when applications conclude their execution and are deallocated. Hence, some rules may remain installed longer than in other proposals. Second, since rules are installed at application-level, the ability to perform fine-grained load balancing in the network (e.g., for a flow or for a restricted set of flows) may be reduced. Note that Predictor can also install and manage rules at lower levels (for instance, by matching source and destination MAC and IP fields), since it uses the OpenFlow protocol. Nonetheless, given the amount of resources available in commodity switches and the number of active flows in the network, low-level rules need to be kept to a minimum.

Contributions. In comparison to our previous work [28], in this paper we present a substantially improved version of Predictor, in terms of both efficiency and resource usage. We highlight five main contributions. First, we run experiments to motivate Predictor and show that the operation of inserting rules at the TCAM takes more time and is more variable according to flow table occupancy. Thereby, the lower the number of rules in TCAMs, the better. Second, we extend Predictor to proactively provide inter-application communication guarantees (rather than only reactively providing it), which can further reduce flow setup time. Third, we develop improved versions of the allocation and work-conserving rate enforcement algorithms to provide better utilization of available resources (without adding significant complexity to the algorithms). More specifically, we improved (*i*) the allocation logic in Algorithm 1, so that resources can be better utilized without adding significant complexity; and (*ii*) rate allocation for VMs in Algorithm 2, so that all bandwidth available can be utilized if there are demands. In our previous paper [28], there could be occasions when some bandwidth would not be used even if there were demands. Fourth, we address the design of the control plane for Predictor, as it is an essential part of SDN to provide efficient and dynamic control of resources. Fifth, we conduct a more extensive evaluation, comparing Predictor with different modes of operation of DevoFlow [19] and considering several factors to analyze its benefits, overheads and technical feasibility. Predictor reduces flow table size up to 94%, offers low average flow setup time and presents low controller load, while providing minimum bandwidth guarantees for tenants and work-conserving sharing for providers.

The remainder of this paper is organized as follows. Section 2 discusses related work, and Section 3 examines the challenges of performance interference and scalability of SDN in DCNs. Section 4 provides an overview of Predictor, while Section 5 presents the details of the proposal (specifically regarding application requests, bandwidth guarantees, resource sharing and control plane design). Section 6 presents the evaluation, and Section 7 discusses generality and limitations. Finally, Section 8 concludes the paper with final remarks and perspectives for future work.

2. Related work

Researchers have proposed several schemes to address scalability in large-scale, SDN-based DCNs and performance interference among applications. Proposals related to Predictor can be divided into three classes: OpenFlow controllers (related to scalability in SDN-based DCNs), and deterministic and non-deterministic bandwidth guarantees (related to performance interference).

OpenFlow controllers. DevoFlow [19] and DIFANE [24] propose to devolve control to the data plane. The first one introduces new mechanisms to make routing decisions at forwarding devices for small flows and to detect large flows (to request controller assistance to route them), while the second keeps all packets in the data plane. These schemes, however, require more complex, customized hardware at forwarding devices. Kandoo [25] provides a logically distributed control plane for large networks. Nonetheless,

[2] Without loss of generality, we assume one application per tenant.

[3] An application is represented by a set of VMs that consume computing and network resources (see Section 5.1 for more details).