Computer Networks 112 (2017) 36–51

Contents lists available at ScienceDirect

Computer Networks

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

A high-performance Two-Phase Multipath scheme for data-center networks

Lyno Henrique G. Ferraz[a,c,1], Rafael Laufer[b], Diogo M.F. Mattos[a,c], Otto Carlos M.B. Duarte[a,*], Guy Pujolle[c]

[a] *Universidade Federal do Rio de Janeiro - GTA/POLI-COPPE/UFRJ, Rio de Janeiro, Brazil*
[b] *Bell Labs, Alcatel-Lucent, Holmdel, USA*
[c] *Laboratoire d'Informatique de Paris 6 - Sorbonne Universities, UPMC Univ Paris 06, Paris, France*

ARTICLE INFO

Article history:
Received 2 September 2015
Revised 3 May 2016
Accepted 21 September 2016
Available online 5 November 2016

Keywords:
Multipathing
Cloud
Data-center
SDN

ABSTRACT

Multipath forwarding has been recently proposed to improve utilization in data centers leveraged by its redundant network design. However, most multipath proposals require significant modifications to the tenants' network stack and therefore are only feasible in private clouds. In this paper, we propose the Two-Phase Multipath (TPM) forwarding scheme for public clouds. The proposal improves tenants' network throughput, whereas keeping unmodified network stack on tenants. Our scheme is composed of a smart offline configuration phase that discover optimal disjoint paths, and a fast online path selection phase that improves flow throughput at run time. A logically centralized manager uses a genetic algorithm to generate and install sets of paths, summarized into trees, during multipath configuration, and a local controller performs the multipath selection based on network usage. We analyze TPM for different workloads and topologies under several scenarios of usage database locations and update policies. The results show that our proposal yields up to 77% throughput gains over previously proposed approaches.

1. Introduction

In cloud computing, data centers share their infrastructure with several tenants having distinct application requirements [1]. This application diversity within data centers leads to multiple challenges for network design in terms of volume, predictability, and utilization. First, traffic between Top-of-Rack (ToR) switches is currently estimated to be 4x higher than incoming/outgoing traffic [2]. This high traffic volume requires specific network topologies for data centers in order to guarantee full bisection bandwidth and to provide fault tolerance [2–5]. Second, the random arrival and departure of virtual machines from multiple tenants result in an unpredictable traffic workload, making it hard to provide manual solutions for traffic management. Therefore, automated solutions that respond quickly to changes are required to efficiently allocate the network resources. Finally, in order to avoid forwarding loops, legacy network protocols, such as the Spanning Tree Protocol (STP) [6], are usually employed to disable certain network links. This ensures that every pair of ToR switches communicates over a single path and that the network is loop-free; however, it also restricts the switches from taking advantage of the multiple available paths in data center topologies.

Whereas volume and predictability are inherent to the traffic nature of the application, network utilization can be significantly improved by multipath forwarding. The idea is to split traffic at flow-level granularity among different paths in order to fully utilize the available capacity. Although promising, most approaches rely on heavy modifications to the network stack of end hosts, ranging from explicit congestion notification (ECN) [7,8] to multipath congestion control [9]. These modifications are not an issue on private clouds, whose sole purpose is to provide services within a single domain. However, in infrastructure-as-a-service (IaaS) clouds, in which tenants rent virtual machines and have complete control of their network stacks [10], these solutions are not feasible. Therefore, solutions that only enhance the network infrastructure while not touching the end hosts are required.

A well-known approach for deploying multipath forwarding without modifying the end host is Equal Cost MultiPath (ECMP), commonly adopted in data-centers [2,3,11–13]. Network switches supporting ECMP find multiple paths with the same cost and

[*] Corresponding author.
E-mail addresses: lyno@gta.ufrj.br (L.H.G. Ferraz), rafael.laufer@alcatel-lucent.com (R. Laufer), diogo@gta.ufrj.br (D.M.F. Mattos), otto@gta.ufrj.br (O.C.M.B. Duarte), Guy.Pujolle@lip6.fr (G. Pujolle).
[1] Grupo de Teleinformática e Automação - GTA Universidade Federal do Rio de Janeiro (UFRJ) P.O. Box: 68504 - ZIP Code 21945-972, Ilha do Fundão, Rio de Janeiro, RJ, Brasil phone: +55 21 3938–8635.

http://dx.doi.org/10.1016/j.comnet.2016.09.025
1389-1286/© 2016 Elsevier B.V. All rights reserved.

choose between them applying a hash function to fields of the packet header in order to find the next hop. ECMP is expected to evenly distribute the flows among the multiple paths and thus prevent network congestion. Nevertheless, since hash-based path selection does not keep track of path utilization, ECMP commonly causes load imbalances when long-lived flows are present on selected paths [14]. Similarly, in Valiant Load-Balancing (VLB), the flow source sends traffic to a random intermediate node which, in turn, forwards it to the destination. As ECMP, this also achieves uniform flow distribution on paths; however, due to the stateless selection of the intermediate node, VLB suffers from the same problems as ECMP. Moreover, ECMP as well as VLB choose the next hop without taking into account the utilization of the links.

In this paper, we propose a Two-Phase Multipath (TPM) forwarding scheme that calculates multiple trees on the network, and keeps track of the usage of all trees. Our proposal is based on the key properties:

- **No modifications at end hosts:** TPM is an in-network load balancing scheme that does not require modifications to the end hosts. This is required in multitenant clouds where the provider does not have any access whatsoever to the tenants' network stack.
- **No modifications in hardware:** TPM increases the performance of the data center with no hardware modifications and avoids changes to the infrastructure fabric. In addition, it also requires only a handful of configurable features to keep the implementation cost low.
- **Robustness to path and topology asymmetries:** TPM handles path asymmetry due to link failures and topology design. It can also be deployed in arbitrary topologies and covers the entire spectrum of data center topologies.
- **Incremental deployment:** TPM can be deployed in only part of the data center, and work with other segments of the data center.

The proposed TPM multipath scheme separates the forwarding functionality into two distinct phases, namely, multipath configuration and multipath selection. Multipath configuration is the offline phase that computes the best possible paths and configures switches when the network is not yet operational. It creates several VLAN trees interconnecting all ToR switches, and therefore the path selection can be performed by simply tagging packets with the proper VLAN ID at the outgoing ToR switch. To find these trees, the multipath configuration phase uses network topology information to reduce path lengths and increase link usage. In particular, we propose and formulate a genetic algorithm to find an optimal set of trees with disjoint links. Finding the optimized configuration of a datacenter network is a complex task and, for some datacenter topologies, it is an unstructured problem. Thus, it needs a back-end support for optimizing the VLAN tree configuration and to assure that all configured VLAN trees are correct [15]. The genetic algorithm formulation answers to these needs. Multipath selection is the online phase that chooses the best path for a new flow. The selection is based on path utilization in order to select the least used path.

The key idea of the proposal is to keep track of the usage of multiple disjoint paths on the network. We summarize a set of paths that shares links into a tree and, thus, the problem of calculating disjoint paths for all pairs of hosts is reduced to calculate disjoint trees that contain all hosts. We optimize the generation of trees, maximizing the diversity of links used by different trees through a genetic algorithm. The proposed genetic algorithm model achieves up to 100% of diversity between the calculated trees, i.e., the calculated trees do not share any link. Moreover, we develop a discrete-event simulator at flow-level granularity to model the data center to evaluate our proposed approaches for selecting a tree to forward a new flow. We test several scenarios inspired in realistic traffic workloads [16]. The results show that TPM always performs better than the traditional forwarding schemes with gains up to 77%.

The rest of the paper is structured as follows. Section 2 presents the architecture of TPM and our design choices. Section 3 describes the offline multipath configuration phase and Section 4 presents the online multipath selection phase. We simulate different scenarios and topologies and present the results in Section 5. We present the related work in Section 6 and conclusions in Section 7.

2. Architecture of the Two-Phase Multipath scheme

The proposed Two-Phase Multipath (TPM) scheme explores the path diversity of the network to load balance flows using an in-network approach, without requiring any modification to the tenants' protocol stack. As previously explained, this is performed in two phases: The multipath configuration phase and the multipath selection phase.

TPM requires two types of devices to manage the entire network: a global logically centralized manager responsible for the multipath configuration phase, and local controllers responsible for the multipath selection phase. In essence, the global manager collects network topology information, calculates the available paths, and sends them to the network devices to be used later during online path selection. The path computation is performed before the network becomes operational and also upon any topology change. To obtain the topology, the global manager may use either the Simple Network Management Protocol (SNMP) for topology discovery [17] or OpenFlow [18]. The VLAN for each tree can also be configured using either approach, which guarantees that most commercial off-the-shelf (COTS) devices are suitable to be used with TPM.

In order to exploit multiple paths without having to modify the network core, TPM uses VLANs (IEEE 802.1Q). Each VLAN uses a subset of aggregation/core switches to interconnect all ToR switches in a tree topology.

Instead of assigning a VLAN to each path, TPM uses a VLAN for each tree in order to aggregate multiple paths into a single VLAN ID, thus saving precious VLAN ID space (each VLAN ID has only 12 bits). Assuming a data center with n ToR switches, each tree contains $n(n-1)/2$ symmetric paths; our approach is then able to support up to $n(n-1)2^{11}$ different paths between every pair of ToR switches. This increases the path availability by a factor of at least $n(n-1)/2$ when compared to the case of using a VLAN per path. In addition to increasing path availability, VLAN trees do not require a routing protocol, since there is only a single path between any pair of ToR switches in each VLAN.

The trees of each VLAN are not entirely disjoint, and thus each link may belong to multiple trees. During the multipath configuration phase, however, the trees are selected to be as disjoint as possible in order to ensure maximum path availability between any pair of ToR switches. To find a set of trees that share the lowest number of links, we propose a genetic algorithm in Section 3.

We assume that each physical machine in a rack has a virtual switch [19,20] connected to the same local controller. During packet forwarding, the virtual switch inserts a VLAN tag into each outgoing packet and also removes the VLAN tag from each incoming packet [10]. Upon arrival of a new outgoing flow, the virtual switch contacts the controller to select an available path for it. The controller then queries a database with network usage information to determine the least congested path for the new flow, as explained later in Section 4. Once the path (and its corresponding VLAN) is selected, the local controller installs an OpenFlow rule on the virtual switch to handle future packets of this flow. Each