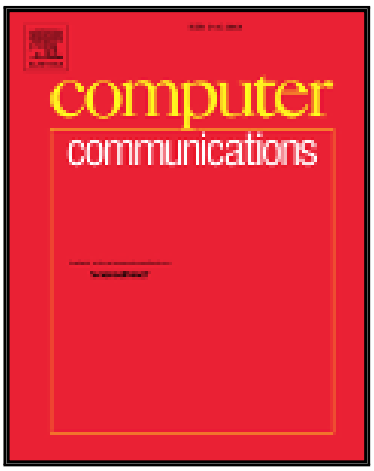

Efficient Load Balancing Over Asymmetric Datacenter Topologies

Syed Mohammad Irteza, Hafiz Mohsin Bashir, Talal Anwar,
Ihsan Ayyub Qazi, Fahad Rafique Dogar

PII: S0140-3664(17)31153-2
DOI: 10.1016/j.comcom.2018.05.010
Reference: COMCOM 5703

To appear in: *Computer Communications*

Received date: 2 November 2017
Revised date: 17 May 2018
Accepted date: 20 May 2018

Please cite this article as: Syed Mohammad Irteza, Hafiz Mohsin Bashir, Talal Anwar, Ihsan Ayyub Qazi, Fahad Rafique Dogar, Efficient Load Balancing Over Asymmetric Datacenter Topologies, *Computer Communications* (2018), doi: 10.1016/j.comcom.2018.05.010

Efficient Load Balancing Over Asymmetric Datacenter Topologies

Syed Mohammad Irteza[a,*], Hafiz Mohsin Bashir[b], Talal Anwar[a], Ihsan Ayyub Qazi[a], Fahad Rafique Dogar[b]

[a] *Department of Computer Science, SBASSE, Lahore University of Management Sciences (LUMS), DHA, Lahore, Pakistan*
[b] *Department of Computer Science, Tufts University, Medford, MA 02155, USA*

Abstract

Datacenter networks are often structured as multi-rooted trees to provide high bisection bandwidth at low cost. To utilize the available bisection bandwidth, an efficient load balancing algorithm is required. Under symmetric network conditions, *packet spraying* is known to perform well, as it enables fine-grained (packet-level) load balancing over equal cost paths. However, packet spraying performs poorly in asymmetric topologies. To make packet spraying effective under asymmetry while retaining its simplicity, we propose SAPS, "Symmetric Adaptive Packet Spraying", a Software-Defined Networking (SDN) based scheme that uses packet spraying over symmetric virtual topologies. SAPS is based on the key insight that if we provide each flow with a symmetric view of the network fabric, then packet spraying can produce near-optimal performance. Through simulations and testbed experiments, we evaluate SAPS. Over a variety of application workloads and asymmetric network scenarios, including single and multiple link failures, results indicate that SAPS performs well, e.g., under single link failure, outperforming state-of-the-art load balancing schemes by up to 61% for average flow completion times.

Keywords: Datacenter, Network Layer, Load Balancing, SDN, Packet Spraying, Asymmetry, Failures

1. Introduction

Email, search and social networks are examples of services available on the cloud today used by millions of users. One of the features of such services that make them so attractive is that they are always available, irrespective of the user's location [1]. Datacenters (DCs) today provide the base from where all these services flow, and often consist of tens of thousands of hosts, each providing both compute power and storage. Users of popular large-scale web applications like those mentioned above expect minimal response times, and a small increase in response times can lead to a tangible impact on operator revenue. Along with these user-facing services, DCs also host public cloud offerings such as Microsoft Azure, and Amazon Web Services, which enable businesses to move their applications away from expensive in-house solutions. In either case (i.e., public or private cloud) the performance of the underlying network connecting the hosts within the DC has a major impact on user experience [2].

Today, hosts within most DCs are inter-connected via commodity Ethernet switches. This network fabric, i.e., the *DC network*, is often structured as a multi-rooted tree, with multiple paths between every pair of hosts [1]. Multiple paths offer greater reliability to DC applications (e.g., F10 [3]), as well as improved throughput (e.g., full-bisection bandwidth [1]). An efficient and fine-grained *load balancing* algorithm is the key to effectively using these multiple paths.

In contrast to schemes like Equal Cost Multi-Path (ECMP) [4] that use only one path for each unique flow, the *packet spraying* (PS) [5, 6] approach sprays packets of each flow across all available paths. The design of PS is simple, and it achieves near-optimal performance over symmetric topologies [7, 8]. However, failures (at links and routers) are a common phenomenon in large-scale DC deployments [3, 9]. These failures are one of the factors that contribute towards making DC topologies *asymmetric*. Under asymmetric topologies, PS performs poorly, mainly due to (a) increased packet reordering (e.g., in case of full link failures [6]), and (b) the slowing down of TCP's control loop (e.g., under partial link failures) [8].

Although various load balancing schemes have addressed asymmetric topologies, they seldom handle the different forms of asymmetry that arise within the DC network, such as both partial link failures and full link failures. Often, their designs are also of much greater complexity relative to PS. For instance, Presto [7] is a state-of-the-art load balancing scheme, that sprays flows across paths in a weighted manner, in order to overcome PS's limitations over asymmetric topologies. Such schemes, while outperforming PS, still tend to perform sub-optimally in the case of *partial* failures—whereby a link's speed drops due to a failure or Ethernet's auto-negotiation [8]. Under such circumstances, the bandwidth-delay product of the partially failed link (i.e., the slowest link) limits the growth of a flow's congestion window. For example, if a few flowcells (i.e., a fixed number of packets) are sent over links with 10Gbps capacity, the successive ACKs received will prompt a proportional increase in the congestion window. However, if a subsequent flowcell is sent over a link with 1Gbps capacity, the enlarged congestion window (well-suited for a 10Gbps link) will result in congestion

*Corresponding author
Email addresses: syed.irteza@lums.edu.pk (Syed Mohammad Irteza), hmmohsin@cs.tufts.edu (Hafiz Mohsin Bashir), 14030013@lums.edu.pk (Talal Anwar), ihsan.qazi@lums.edu.pk (Ihsan Ayyub Qazi), fahad@cs.tufts.edu (Fahad Rafique Dogar)