Exploiting Path Diversity in Datacenters using MPTCP-aware SDN

Tomas Urban and Martin Oravsky
Faculty of Informatics and Information Technologies, Slovak Technical University

Recently, Multipath TCP (MPTCP) has been proposed as an alternative transport approach for datacenter networks. MPTCP provides the ability to split a flow into multiple paths thus providing better performance and resilience to failures. Usually, MPTCP is combined with flow-based EqualCost Multi-Path Routing (ECMP), which uses random hashing to split the MPTCP subflows over different paths. However, random hashing can be suboptimal as distinct subflows may end up using the same paths, while other available paths remain unutilized. In this paper, we explore an MPTCP-aware SDN controller that facilitates an alternative routing mechanism for the MPTCP subflows. The controller uses packet inspection to provide deterministic subflow assignment to paths. Using the controller, we show that MPTCP can deliver significantly improved performance when connections are not limited by the access links of hosts. To lessen the effect of throughput limitation due to access links, we also investigate the usage of multiple interfaces at the hosts. We demonstrate, using our modification of the MPTCP Linux Kernel, that using multiple subflows per pair of IP addresses can yield improved performance in multi-interface settings.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Animation*; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Physically based modeling*

General Terms: Experimentation, Human Factors

Additional Key Words and Phrases: Face animation, image-based modelling, iris animation, photorealism, physiologically-based modelling

ACM Reference Format:

Vitor F. Pamplona, Manuel M. Oliveira, Gladimir V. G. Baranoski, and Sean Fogarty. 2009. Photorealistic models for pupil light reflex and iridal pattern deformation. *ACM Trans. Graph.* 28, 4, Article 106 (September 2009), 10 pages.
DOI: http://dx.doi.org/10.1145/1559755.1559763

Manuel M. Oliveira acknowledges a CNPq-Brazil fellowship (305613/2007-3). Gladimir V. G. Baranoski acknowledges a NSERC-Canada grant (238337). Microsoft Brazil provided additional support. Authors' addresses: Sean Fogarty, (Current address) NASA Ames Research Center, Moffett Field, California 94035.

DOI: http://dx.doi.org/10.1145/1559755.1559763

1. INTRODUCTION

The Transmission Control Protocol (TCP) is used by the vast majority of applications to transport their data reliably across the Internet. TCP was designed in the 1970s, and neither mobile devices nor computers with many network interfaces were an immediate design priority. On the other hand, the TCP designers knew that network links could fail, and they chose to decouple the network-layer protocols (Internet Protocol) from those of the transport layer (TCP) so that the network could reroute packets around failures without affecting TCP connections. This ability to reroute packets is largely due to the use of dynamic routing protocols, and their job is made much easier because they dont need to know anything about transport-layer connections.

Todays networks can be multipath: mobile devices have multiple wireless interfaces, datacenters have many redundant paths between servers, and multihoming has become the norm for big server farms. Meanwhile, TCP is essentially a single-path protocol: when a TCP connection is established, the connection is bound to the IP addresses of the two communicating hosts. If one of these addresses changes, for whatever reason, the connection will fail. In fact, a TCP connection cannot even be load balanced across more than one path within the network, because this results in packet reordering, and TCP misinterprets this reordering as congestion and slows down.

This mismatch between todays multipath networks and TCPs single-path design creates tangible problems. For instance, if a smartphones WiFi loses signal, the TCP connections associated with it stall; there is no way to migrate them to other working interfaces, such as 3G. This makes mobility a frustrating experience for users. Modern datacenters are another example: many paths are available between two endpoints, and multipath routing randomly picks one for a particular TCP connection. This can cause collisions where multiple flows get placed on the same link, thus hurting throughput to such an extent that average throughput is halved in some scenarios.

Multipath TCP (MPTCP)[A. Ford and C. Raiciu and M. Handley and O. Bonaventure year] [Raiciu et al. 2012] is a major modification to TCP that allows multiple paths to be used simultaneously by a single transport connection. Multipath TCP circumvents the issues mentioned above and several others that affect TCP. Changing TCP to use multiple paths is not a new idea; it was originally proposed more than 15 years ago by Christian Huitema in the Internet Engineering Task Force (IETF), and there have been a half-dozen more proposals since then to similar effect. Multipath TCP draws on the experience gathered in previous work, and goes further to solve issues of fairness when competing with regular TCP and deployment issues as a result of middleboxes in todays Internet. The Multipath TCP protocol has recently been standardized by the IETF, and an implementation in the Linux kernel is available today[C. Paasch and S. Barre and J. Korkeaniemi and F. Duchene and G. Detal year].

2. ANALYSIS

2.1 Chapter introduction

This chapter is a brief analysis of Software defined networking and how it can benefit from multipath transmission control protocols, especially in datacenter-like topologies by exploiting path diversity.

In chapter 2.2, the basic concept of software defined networking will be introduced. Chapter 2.3 will briefly talk about Multipath TCP protocol which will be used later in this document. Chapter 2.4 will consider several problems regarding poor performance of SDNs due to random path selection.

2.2 Software defined networking

Todays networks mainly consist of routers, switches, hosts and many other network devices. In order for hosts to communicate, routers and switches performs as a transfer point between source and destination. These devices must provide very fast and reliable transfer of information, which, since the need for fast Internet connection is bigger and bigger every year, can be challenging.

The basic network device consists of data plane and control plane. Data plane is responsible for receiving the incoming packet and forwarding it to devices control plane, where the packet is processed. Control plane then forwards the packet again to the data plane, which then sends the packet via correct outgoing interface. This causes network device to store a lot of information and the CPU load can become quite high.

Software defined networking is an architectural approach that optimizes and simplifies network operation by more closely binding the interaction (i.e., provosioning, messaging and alarming) among applications and network services and devices, whether they be real or virtualized. [Nadeau and Gray 2013]

SDNs use centralized object known as controller, which does all the computing and routing of the communication. The controller installs various forwarding rules based on global view of network topology on forwarders. Forwarders (also known as switches) are simple devices which forward the communication based on the rules installed by controller. If packet arrives on forwarder interface and no rule is matched, forwarder sends the packet to controller using OpenFlow protocol. The controller then processes the packet (e.g. flooding ARP request) and sends the packet back to forwarders along with corresponding rule.

This enables the network to better handle the communication while using the hardware resources better.

2.3 Multipath TCP

There are several options when sending communication on multiple path simultaneously while providing better redundancy and throughput. One can use Stream control transmission protocol (SCTP) which uses the concept of streams. When there is a need for L2 redundancy, bonding has proven itself as a good alternative.

In this document we propose Multipath TCP as a great alternative for improving path diversity in multipath communication in datacenter-like topologies.

Multipath TCP utilizes one TCP connection on multiple paths, on which separate TCP subflows are created, thus providing improved throughput and redundancy. Multipath TCP eliminates single point of failure by ability to switch communication to a working path when another path goes down. Multipath TCP enabled hosts use Multipath TCP options to establish a connection or to add a new subflow to an existing connection.

2.4 Problems with Multipath TCP and SDNs

Software defined networking can greatly benefit from Multipath TCP capabilities. However, software defined networks do not handle MPTCP traffic differently than the classic single-path TCP traffic. In order to exploit path diversity better and to provide true redundancy to the network, further implementation on the controller is necessary.

A key aspect that affects MPTCP performance is the routing mechanism of the subflows. Currently, the most prominent and widely deployed routing mechanism in datacenters is a flow-based variant of Equal-Cost Multi-Path Routing (ECMP) [2]. However, ECMP uses random hashing to split the subflows over different paths. This can cause a variety of problems, from which the most important is the high probability that multiple subflows end up on the same path while other paths remain not utilized properly. This also destroys the idea of redundancy.

The number of subflows is also a crucial aspect to performance of the network. Although the idea of more subflows can mean faster networks, the more subflows MPTCP use, the bigger the overhead is. More subflows mean more rules installed in forwarders and higher CPU utilization for the controller.

As shown in previous papers, MPTCP throughput also correlates with number of used interfaces on hosts. We will also deal with this issue since one-homed devices can quickly become a bottleneck in this type of network.

The purpose of this assessment is to implement an intelligent routing protocol that makes software defined networks MPTCP-aware by exploiting path diversity.

3. DESIGN

3.1 Topology

The right choice of topology is crucial for datacenter network performance. One of the most used datacenter network topology is FatTree topology, which is shown below on picture 1. The topology consists of pods, which are connected on multiple core switches. The higher the level, the faster the bandwidth.

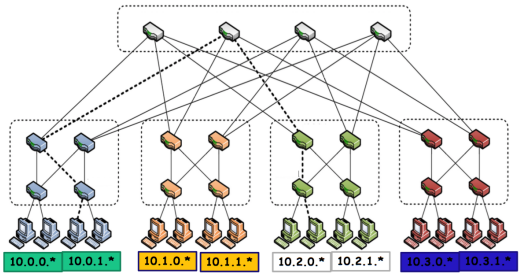

Fig. 1. Fattree topology

3.2 Controller

To make SDN MPTCP-aware, it is necessary to let controller know that multipath communication occurs in the network. This can be achieved by packet inspection on controller. When a packet arrives on a forwarder and no rule is matched, the packet is sent to SDN controller which calculates the path for the subflow by using its global view of the topology. SDN controller then installs the rules

onto every single forwarder that lies on the chosen path. The implementation will use Floodlight controller written in Java.

The shortest paths available will be computed using a graph traversal algorithm. Sets of paths will be stored on controller and these will be deterministically assigned to subflows. While using multiple redundant paths, MPTCPs performance should be maximized using a small number of subflows. Each path should be assigned to only one subflow, this can be achieved by storing IP addresses to paths.

When a new packet arrives on the controller, it will extract the needed information from the packet (IPs, ports, MPTCP options). If the option says we are dealing with a new subflow (MP_CAPABLE), it will find the shortest path between IPs and store the path into a hashtable. This path is then assigned to a subflow. If the options says we are dealing with additional subflow (M _JOIN), we extract the token from the packet and find another shortest path for the subflow. The rules are installed on all switches belonging to the chosen path.

3.3 Testbed

To properly evaluate the algorithm, a virtual testbed will be needed. The tests will be performed on a Linux virtual machine running Floodlight controller and Mininet emulator. We create two more Linux virtual machines with multiple interfaces that will be connected to ports on the switches emulated in Mininet. After running FatTree topology in Mininet, we will evaluate how many subflows are needed in order to maximize MPTCP performance. We will then compare this approach to random approach used by ECMP protocol.

4. TYPICAL REFERENCES IN NEW ACM REFERENCE FORMAT

A paginated journal article [Abril and Plant 2007], an enumerated journal article [Cohen et al. 2007], a reference to an entire issue [Cohen 1996], a monograph (whole book) [Kosiur 2001], a monograph/whole book in a series (see 2a in spec. document) [Harel 1979], a divisible-book such as an anthology or compilation [Editor 2007] followed by the same example, however we only output the series if the volume number is given [Editor 2008] (so Editor00a's series should NOT be present since it has no vol. no.), a chapter in a divisible book [Spector 1990], a chapter in a divisible book in a series [Douglass et al. 1998], a multi-volume work as book [Knuth 1997], an article in a proceedings (of a conference, symposium, workshop for example) (paginated proceedings article) [Andler 1979], a proceedings article with all possible elements [Smith 2010], an example of an enumerated proceedings article [Gundy et al. 2007], an informally published work [Harel 1978], a doctoral dissertation [Clarkson 1985], a master's thesis: [Anisi 2003], an online document / world wide web resource [Thornburg 2001], [Ablamowicz and Fauser 2007], [Poker-Edge.Com 2006], a video game (Case 1) [Obama 2008] and (Case 2) [Novak 2003] and [Lee 2005] and (Case 3) a patent [Scientist 2009], work accepted for publication [Rous 2008], 'YYYYb'-test for prolific author [Saeedi et al. 2010a] and [Saeedi et al. 2010b]. Other cites might contain 'duplicate' DOI and URLs (some SIAM articles) [Kirschmer and Voight 2010]. Boris / Barbara Beeton: multi-volume works as books [Hörmander 1985b] and [Hörmander 1985a].

APPENDIX

A. CLASSICAL MULTIDIMENSIONAL SCALING

Let D be an $n \times n$ matrix of pairwise distances. The matrix D is symmetric with a zero diagonal. We are interested in finding a $d \times n$ matrix X where each column $\boldsymbol{x}_i$ is the representation of the point i in R^d and $\mathrm{D}_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$. Denote the inner product (or Gram matrix) for this set of points by $\mathrm{K} = \mathrm{X}^\top \mathrm{X}$.

K is an $n \times n$ symmetric positive semidefinite matrix. Let us now abuse notation and use D^2 to indicate the matrix of squared pairwise distances $\mathrm{K} = -\frac{1}{2}(\mathrm{I} - 11^\top)\mathrm{D}^2(\mathrm{I} - 11^\top)$. Here, I is the $n \times n$ identity matrix and 1 is the n-vector of all ones.

ACKNOWLEDGMENTS

We are grateful to the following people for resources, discussions and suggestions: Prof. Jacobo Melamed Cattan (Ophthalmology-UFRGS), Prof. Roberto da Silva (UFRGS), Prof. Luis A. V. Carvalho (Optics-USP/SC), Prof. Anatolio Laschuk (UFRGS), Leandro Fernandes, Marcos Slomp, Leandro Lichtenfelz, Renato Silveira, Eduardo Gastal, and Denison Tavares. We also thank the volunteers who allowed us to collect pictures and videos of their irises: Alex Gimenes, Boris Starov, Christian Pagot, Claudio Menezes, Giovane Kuhn, João Paulo Gois, Leonardo Schmitz, Rodrigo Mendes, and Tiago Etiene.

REFERENCES

A. Ford and C. Raiciu and M. Handley and O. Bonaventure. current-year. TCP Extensions for Multipath Operation with Multiple Addresses. (current-year). http://tools.ietf.org/html/draft-ietf-mptcp-multiaddressed-09.

Rafal Ablamowicz and Bertfried Fauser. 2007. CLIFFORD: a Maple 11 Package for Clifford Algebra Computations, version 11. (2007). Retrieved February 28, 2008 from http://math.tntech.edu/rafal/cliff11/index.html

Patricia S. Abril and Robert Plant. 2007. The patent holder's dilemma: Buy, sell, or troll? *Commun. ACM* 50, 1 (Jan. 2007), 36–44. DOI:http://dx.doi.org/10.1145/1188913.1188915

Sten Andler. 1979. Predicate Path expressions. In *Proceedings of the 6th. ACM SIGACT-SIGPLAN symposium on Principles of Programming Languages (POPL '79)*. ACM Press, New York, NY, 226–236. DOI:http://dx.doi.org/10.1145/567752.567774

David A. Anisi. 2003. *Optimal Motion Control of a Ground Vehicle*. Master's thesis. Royal Institute of Technology (KTH), Stockholm, Sweden.

C. Paasch and S. Barre and J. Korkeaniemi and F. Duchene and G. Detal. current-year. MPTCP Linux Kernel Implementation. (current-year). http://mptcp.info.ucl.ac.be.

Kenneth L. Clarkson. 1985. *Algorithms for Closest-Point Problems (Computational Geometry)*. Ph.D. Dissertation. Stanford University, Palo Alto, CA. UMI Order Number: AAT 8506171.

Jacques Cohen (Ed.). 1996. Special Issue: Digital Libraries. *Commun. ACM* 39, 11 (Nov. 1996).

Sarah Cohen, Werner Nutt, and Yehoshua Sagic. 2007. Deciding equivalances among conjunctive aggregate queries. *J. ACM* 54, 2, Article 5 (April 2007), 50 pages. DOI:http://dx.doi.org/10.1145/1219092.1219093

Bruce P. Douglass, David Harel, and Mark B. Trakhtenbrot. 1998. Statecarts in use: structured analysis and object-orientation. In *Lectures on Embedded Systems*, Grzegorz Rozenberg and Frits W. Vaandrager (Eds.). Lecture Notes in Computer Science, Vol. 1494. Springer-Verlag, London, 368–394. DOI:http://dx.doi.org/10.1007/3-540-65193-4_29

Ian Editor (Ed.). 2007. *The title of book one* (1st. ed.). The name of the series one, Vol. 9. University of Chicago Press, Chicago. DOI:http://dx.doi.org/10.1007/3-540-09237-4

Ian Editor (Ed.). 2008. *The title of book two* (2nd. ed.). University of Chicago Press, Chicago, Chapter 100. DOI:http://dx.doi.org/10.1007/3-540-09237-4

Matthew Van Gundy, Davide Balzarotti, and Giovanni Vigna. 2007. Catch me, if you can: Evading network signatures with web-based polymorphic worms. In *Proceedings of the first USENIX workshop on Offensive Technologies (WOOT '07)*. USENIX Association, Berkley, CA, Article 7, 9 pages.

David Harel. 1978. *LOGICS of Programs: AXIOMATICS and DESCRIPTIVE POWER*. MIT Research Lab Technical Report TR-200. Massachusetts Institute of Technology, Cambridge, MA.

David Harel. 1979. *First-Order Dynamic Logic*. Lecture Notes in Computer Science, Vol. 68. Springer-Verlag, New York, NY. DOI:http://dx.doi.org/10.1007/3-540-09237-4

Lars Hörmander. 1985a. *The analysis of linear partial differential operators. III*. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Vol. 275. Springer-Verlag, Berlin, Germany. viii+525 pages. Pseudodifferential operators.

Lars Hörmander. 1985b. *The analysis of linear partial differential operators. IV*. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Vol. 275. Springer-Verlag, Berlin, Germany. vii+352 pages. Fourier integral operators.

Markus Kirschmer and John Voight. 2010. Algorithmic Enumeration of Ideal Classes for Quaternion Orders. *SIAM J. Comput.* 39, 5 (Jan. 2010), 1714–1747. DOI:http://dx.doi.org/10.1137/080734467

Donald E. Knuth. 1997. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms (3rd. ed.)*. Addison Wesley Longman Publishing Co., Inc.

David Kosiur. 2001. *Understanding Policy-Based Networking* (2nd. ed.). Wiley, New York, NY.

Newton Lee. 2005. Interview with Bill Kinder: January 13, 2005. Video, *Comput. Entertain.* 3, 1, Article 4 (Jan.-March 2005). DOI:http://dx.doi.org/10.1145/1057270.1057278

T. D. Nadeau and K. Gray. 2013. SDN: Software Defined Networks, Sebastopol. *O'Reilly Media* (2013).

Dave Novak. 2003. Solder man. Video. In *ACM SIGGRAPH 2003 Video Review on Animation theater Program: Part I - Vol. 145 (July 27–27, 2003)*. ACM Press, New York, NY, 4. DOI:http://dx.doi.org/99.9999/woot07-S422

Barack Obama. 2008. A more perfect union. Video. (5 March 2008). Retrieved March 21, 2008 from http://video.google.com/videoplay?docid=6528042696351994555

Poker-Edge.Com. 2006. Stats and Analysis. (March 2006). Retrieved June 7, 2006 from http://www.poker-edge.com/stats.php

C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley. 2012. How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP. *USENIX Symposium of Networked Systems Design and Implementation (NSDI12)* (2012).

Bernard Rous. 2008. The Enabling of Digital Libraries. *Digital Libraries* 12, 3, Article 5 (July 2008). To appear.

Mehdi Saeedi, Morteza Saheb Zamani, and Mehdi Sedighi. 2010a. A library-based synthesis methodology for reversible logic. *Microelectron. J.* 41, 4 (April 2010), 185–194.

Mehdi Saeedi, Morteza Saheb Zamani, Mehdi Sedighi, and Zahra Sasanian. 2010b. Synthesis of Reversible Circuit Using Cycle-Based Approach. *J. Emerg. Technol. Comput. Syst.* 6, 4 (Dec. 2010).

Joseph Scientist. 2009. The fountain of youth. (Aug. 2009). Patent No. 12345, Filed July 1st., 2008, Issued Aug. 9th., 2009.

Stan W. Smith. 2010. An experiment in bibliographic mark-up: Parsing metadata for XML export. In *Proceedings of the 3rd. annual workshop on Librarians and Computers (LAC '10)*, Reginald N. Smythe and Alexander Noble (Eds.), Vol. 3. Paparazzi Press, Milan Italy, 422–431. DOI:http://dx.doi.org/99.9999/woot07-S422

Asad Z. Spector. 1990. Achieving application requirements. In *Distributed Systems* (2nd. ed.), Sape Mullender (Ed.). ACM Press, New York, NY, 19–33. DOI:http://dx.doi.org/10.1145/90417.90738

Harry Thornburg. 2001. Introduction to Bayesian Statistics. (March 2001). Retrieved March 2, 2005 from http://ccrma.stanford.edu/~jos/bayes/bayes.html

Received September 2008; accepted March 2009