Evaluation of Multipath TCP's response to latency in wide-area software-defined networks

Laboratory for Software Design & Analysis
established in 2005

Chawanat Nakasan, Kohei Ichikawa, Hajimu Iida

⚓ Introduction

Multi-homed, multi-site system setup is useful for various large-scale and distributed systems, such as high-performance computing, distributed storage, cloud computing network, and disaster recovery.

Multipath TCP (MPTCP) is an extension of TCP that allows multiple paths or routes to be used at once, as opposed to the original TCP/IP stack. However, ***MPTCP does not have control over path selection***, and multiple paths may clash at the network level resulting in performance-degrading congestion.

Simple Multipath OpenFlow Controller (smoc) was developed to address this problem by taking advantages of ***OpenFlow*** software-defined network (SDN) framework to provide multipath routing functionality. However, thorough performance evaluation in different environments is necessary to ensure that *smoc* can function well in a general OpenFlow network.

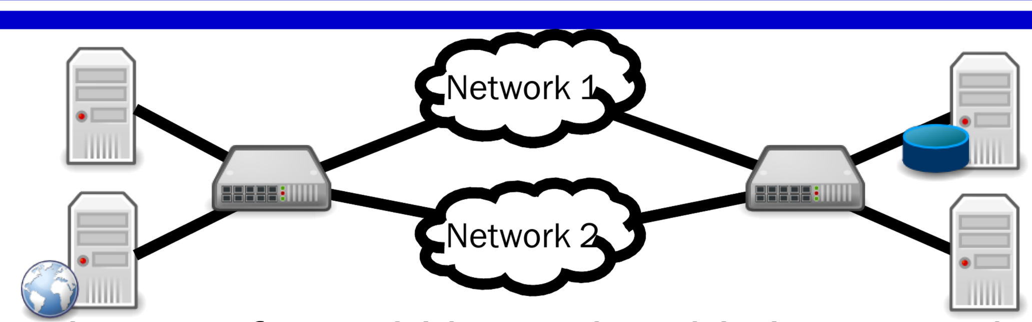
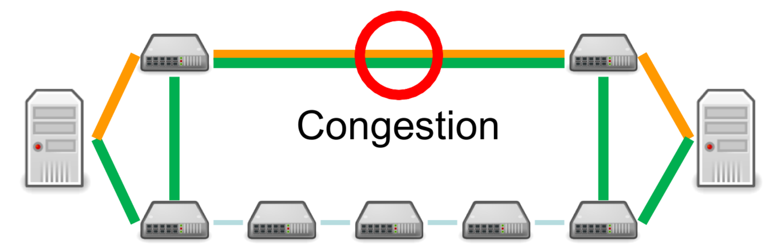
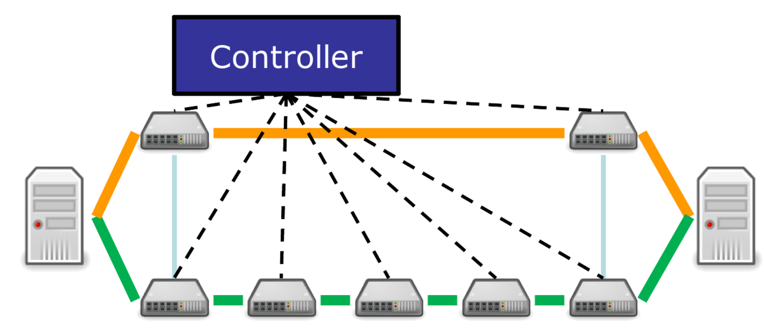


Image of a multi-homed multi-site network



Congestion in traditional hop-based IP routing

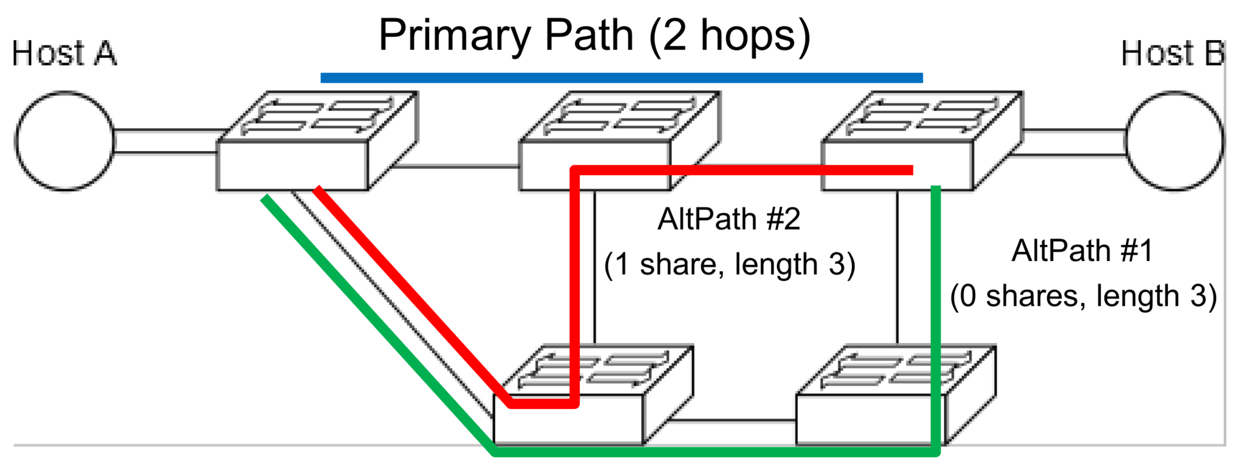


Optimized routing powered by OpenFlow centralized controller

⚓ smoc: Simple Multipath OpenFlow Controller

Path Calculation and Selection

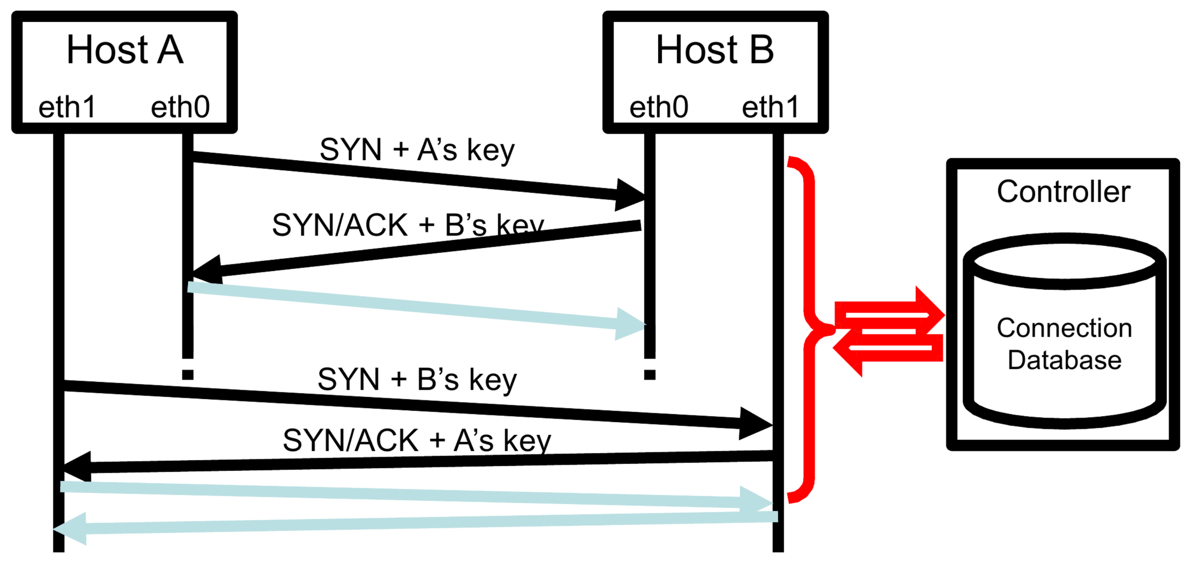
smoc runs on an original algorithm that (1) finds the primary path which is the shortest path, then (2) ranks all other simple paths by number of edges shared with primary path, then by rank. This ensures that collisions are avoided in a simple way.



Sample routing results from our algorithm

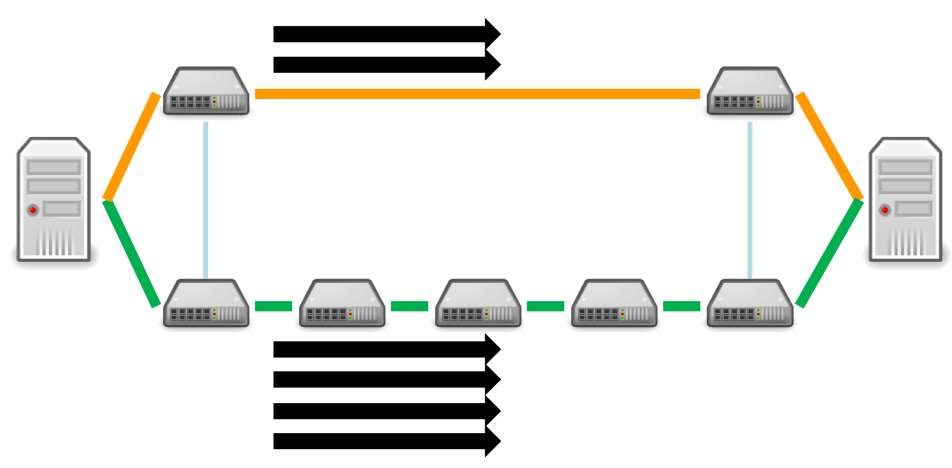
MPTCP Flow Group Detection

In order to detect and differentiate flow groups, we use information exchanged during MPTCP handshake. Relevant information is stored in the controller and queried/matched as needed.



Adjusting Number of Flows

MPTCP kernel supports adjustment of number of TCP subflows to be used per MPTCP connection. We are currently investigating how it affects throughput and how we can exploit it to our advantage in high-latency unequal networks.


Adjust the number of subflows on each path based on the network metrics of that path

⚓ Evaluation and Results

The primary objective of this work is to determine the effects of latency on smoc and MPTCP, especially when multiple paths are used at the same time and when those paths have unequal latency and bandwidth limit. We are also exploring how number of concurrent MPTCP subflows affect the measured throughput.

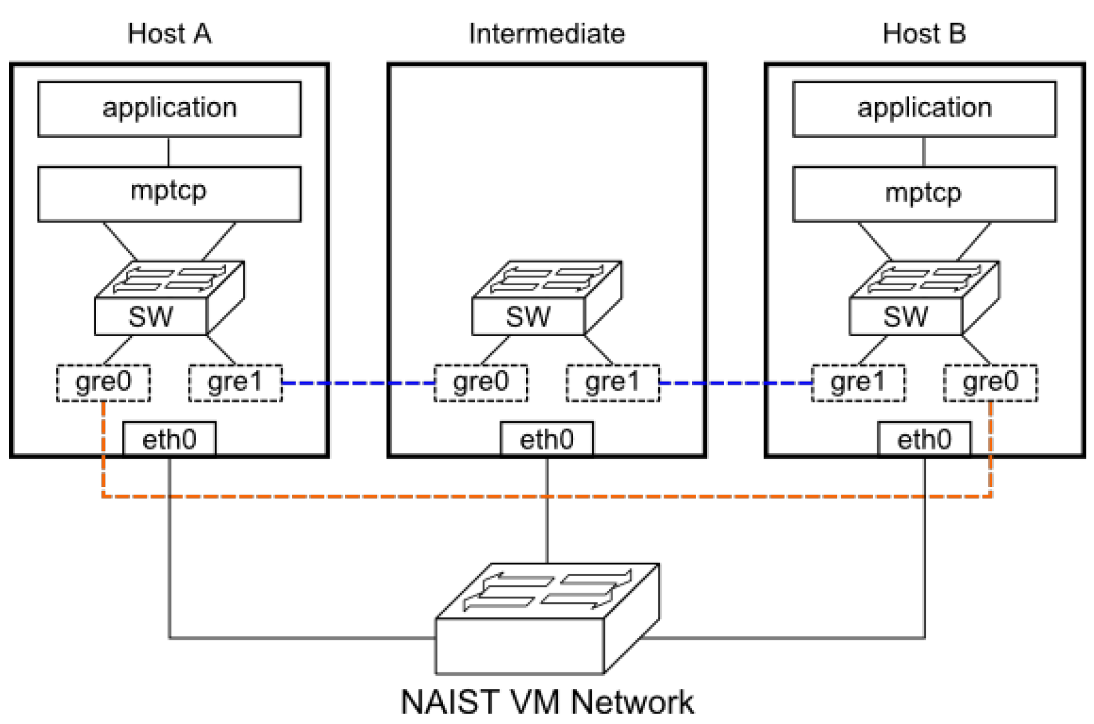
For this work, we used a local virtual machine cluster in our campus to provide the virtual infrastructure as we can explicitly control the latency in the network.

In all tests, all network paths are limited to 100 Mbps.

Results indicate that it is possible to increase the throughput of the application by using multiple TCP subflows, especially when the latency is high. **Setting the number of concurrent streams to 12 or 16 provided the best results within the scope of this work.** Future stages of this project concerns optimization of subflow distribution to improve throughput.

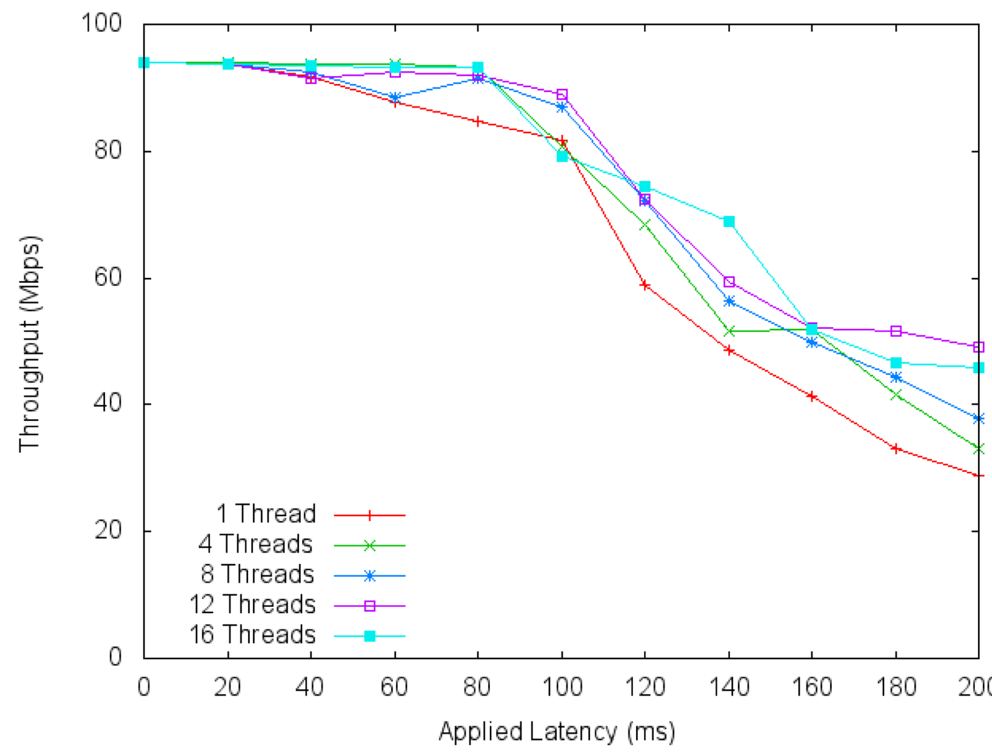
Experiment List:

No.	Paths	Latency (ms)	# TCP Streams
1	1 Path	0..20..200	1, 4, 8, 12, 16
2	2 Disjoint Paths	0..20..200	2*, 4, 8, 12, 16

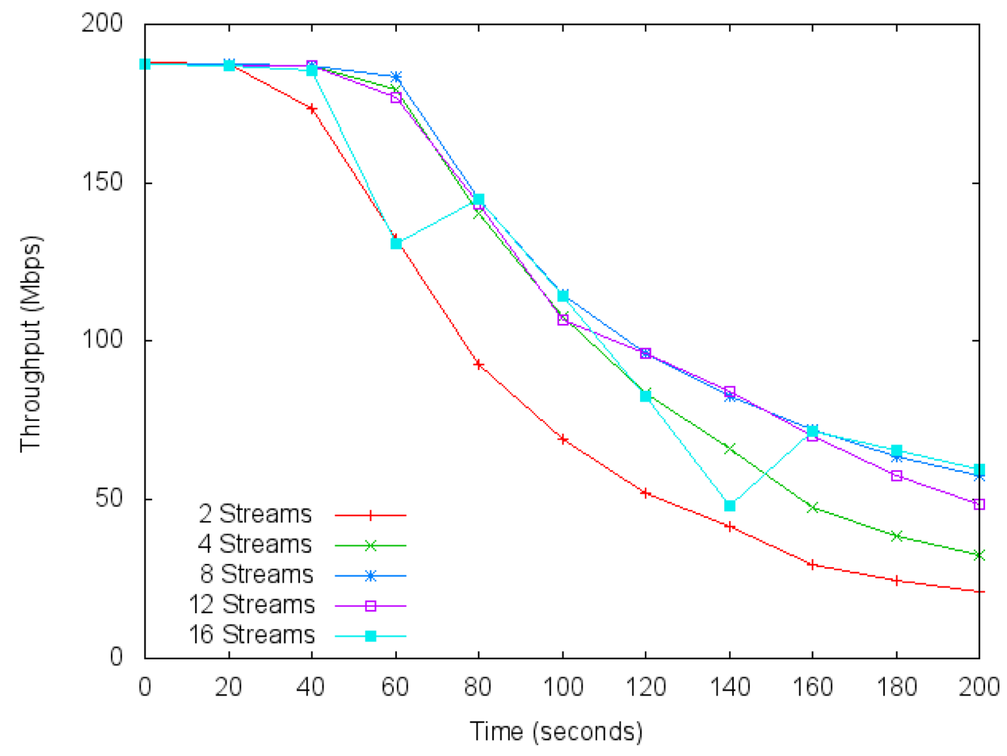


🡐The testbed was set up with 3 hosts as shown. The direct path (red) is used in Experiment 2.

*Experiment 1 starts at 1 TCP stream, while Experiment 2 starts at 2 streams. This is because with n streams configured, MPTCP will produce n identical TCP subflows. In the case of Experiment 2, testing 1 stream will be meaningless as it will be identical to using a single path.



Throughput plot measured in Experiment 1



Throughput plot measured in Experiment 2