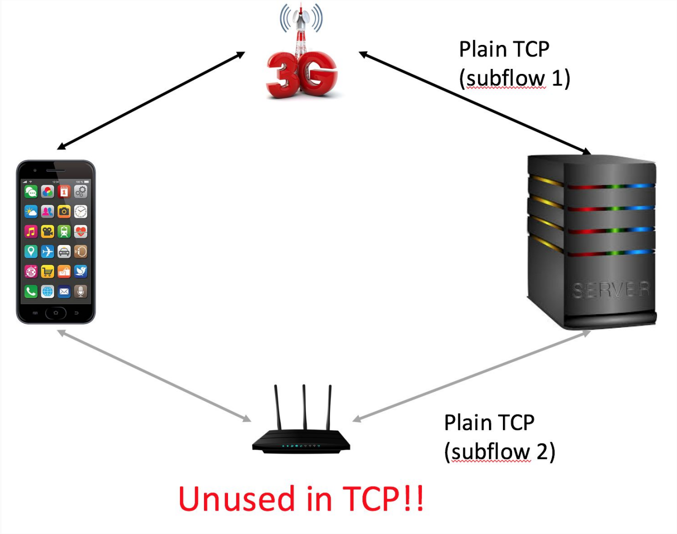
A comparative study of the Multipath TCP for the Linux kernel

Korhan Demirkaya, Shubham Saini

CSE 222a Class Project
Instructor: Dr. Alex Snoeren

TCP vs MPTCP

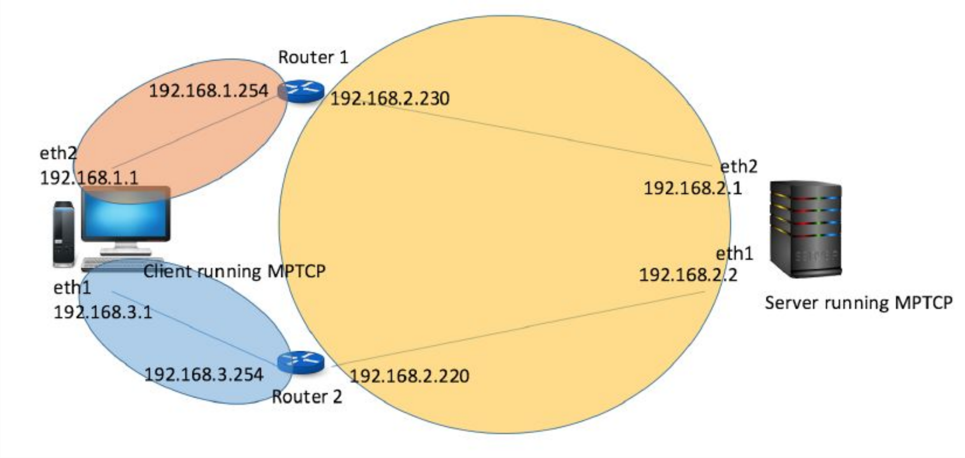
TCP	MPTCP
Connection between two interfaces	Connection between two hosts
Single path, one interface for transmission.	● Can use multiple interfaces and links including Wi-Fi and mobile networks. ● Backward compatible
No link handover capability	Link handover at the end points through abstraction in the transport layer.



Implementation - Topology

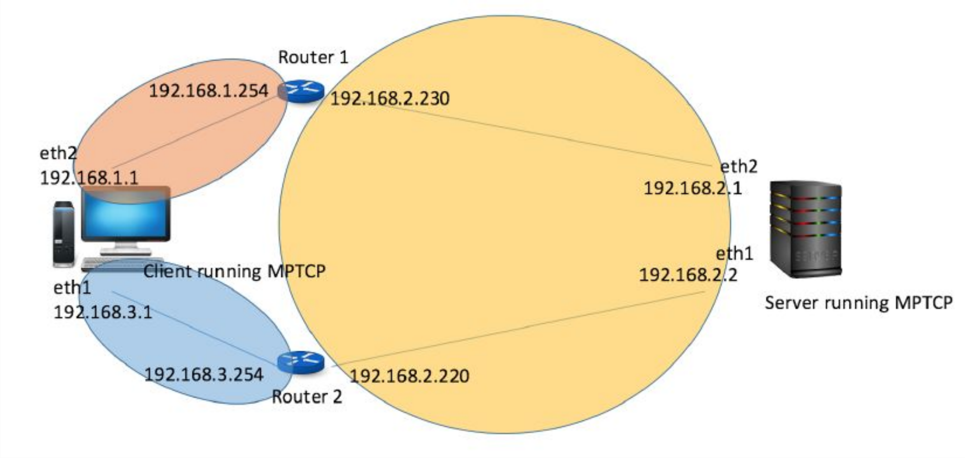
Each router connect two different subnets, allowing client and server to communicate.

Two possible paths between client and server



Implementation - Technologies

- Each network element a separate VM running over VirtualBox.
- Client and Server run Ubuntu 14.04 with MPTCP kernel.
- Routers run Ubuntu Server 14.04 with Quagga routing suite.
- Network bandwidth limited using wondershaper
- Bandwidth measurement using iperf3



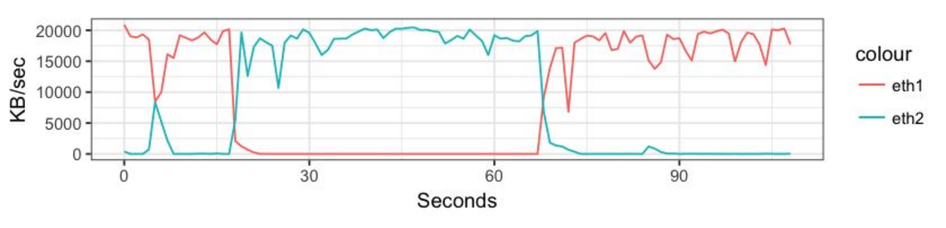
Overview of experiments

Validate the key benefits of MPTCP, and compare against TCP

1. Efficiency
2. Reliability
3. Bandwidth Aggregation
4. Fairness to TCP

Results - MPTCP Efficiency

MPTCP should send traffic on the most efficient (least congested, least RTT) path



Experiment:
1. Start with two paths with equal RTT, congestion
2. Introduce congestion on Path 1 (20 seconds)
3. Reduce congestion on Path 1 (40 seconds)
4. Increase congestion on Path 2 (75 seconds)

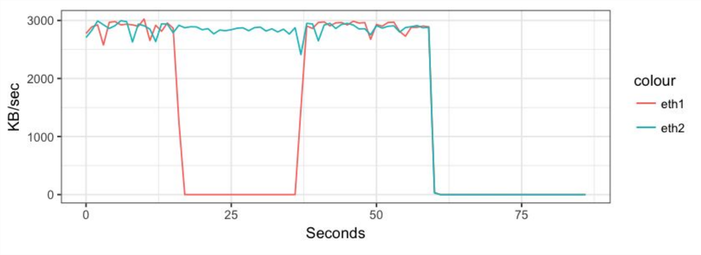
Observation:
1. MPTCP quickly adapts to congestion
2. Sends <u>ALL</u> its traffic on its least-congested paths

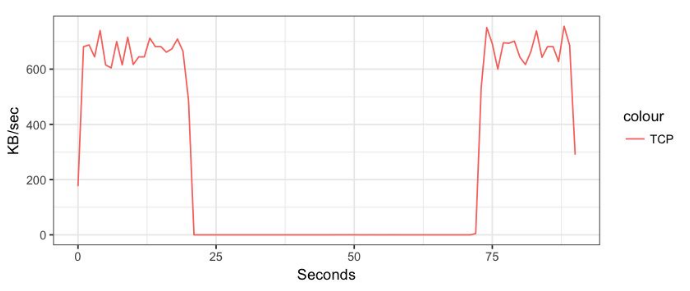
Results - MPTCP Reliability

MPTCP should provide reliable communication across the available paths.

Experiment:
1. Limit the capacity of each available path to ~2 MBPS.
2. Send data using MPTCP.
3. Disconnect one of the paths.
Repeat the same experiment for TCP.

Observation:
1. MPTCP maintains continuous data transfer through all available paths, even if some of the paths are unavailable for some duration.
2. TCP does not switch to the other available path when one a path in use is broken.



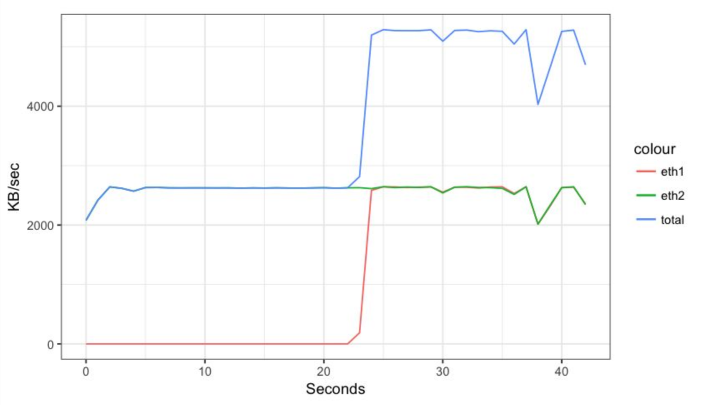


Results - MPTCP Bandwidth Aggregation

MPTCP provides bandwidth aggregation capabilities by making use of all available resources. (limited capacity)

Experiment:
1. Start data transfer on one of the paths.
2. Make a new path available to MPTCP in between the transfer.

Observation:
MPTCP starts sending data through the new path as it becomes available, thus increasing total available bandwidth.

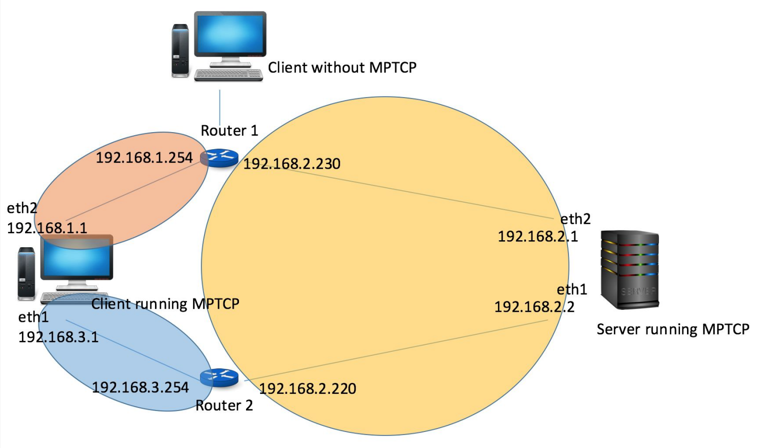


Results - Fairness to TCP

Goal: MPTCP should use as much resources as available, <u>while being fair to TCP.</u>

Experiment: Introduce a new client (client 2) without MPTCP.

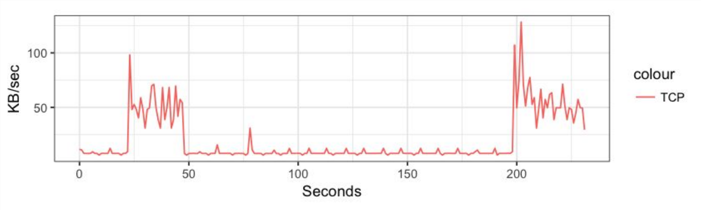
1. Start data transfer from client2 to server.
2. Start MPTCP transfer from client1 to server.



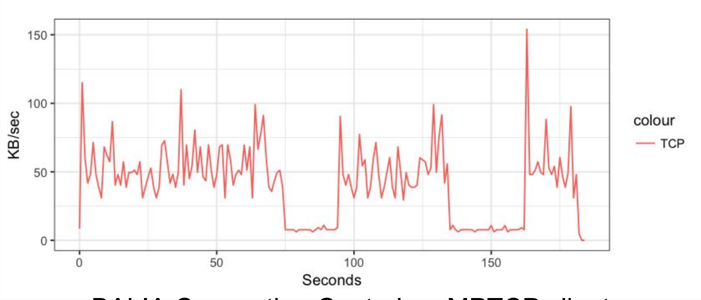
Results - Fairness to TCP

Observation:
1. Default TCP congestion control mechanism used in MPTCP.
2. When sharing bottleneck links, MPTCP gets higher bandwidth than competing TCP flows.

BALIA (Balanced Linked Adaptation Congestion Control) turns out to be more fair to TCP.
TCP still fails to get equal bandwidth share as MPTCP, and oscillates between 0% and 50% share.



Default TCP Congestion Control on MPTCP client



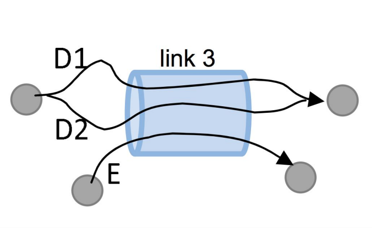
BALIA Congestion Control on MPTCP client

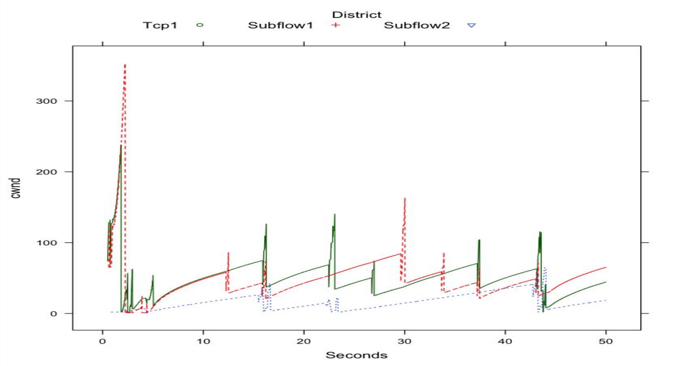
HTSIM results

Use HTSIM to verify the results obtained using our implementation.

Implemented a “bottleneck” topology with 2 multipath flows and 1 competing TCP flow.

Observation:
1. Contrary to our experiments, TCP congestion window size in HTSIM is larger than MPTCP.
2. Bandwidth output seems to be random values, not correlated to the cwnd. Needs further investigation.





Thank you!

Acknowledgements:

Dr. Alex Snoeren

Dr. Costin Raiciu

CSE 222a peers