Improving Datacenter Performance and Robustness with Multipath TCP

Costin Raiciu
Department of Computer Science
University Politehnica of Bucharest

Sebastien Barre (UCL-BE), Christopher Pluntke (UCL), Adam Greenhalgh (UCL), Damon Wischik (UCL) and Mark Handley (UCL)

Thanks to: trilogy
cisco

Presented by Gregory Kesden
Motivation

- Datacenter apps are distributed across thousands of machines
- Want any machine to play any role

To achieve this:

- Use dense parallel datacenter topologies
- Map each flow to a path

Problem:

- Naïve random allocation gives poor performance
- Improving performance adds complexity

This is the wrong place to start
Contributions

Multipath topologies need multipath transport

Multipath transport enables better topologies
To satisfy demand, modern datacenters provide many parallel paths

- Traditional Topologies are tree-based
  - Poor performance
  - Not fault tolerant

- Shift towards multipath topologies: FatTree, BCube, VL2, Cisco, EC2
Fat Tree Topology [Fares et al., 2008; Clos, 1953]
Fat Tree Topology [Fares et al., 2008; Clos, 1953]

K=4

Aggregation Switches

K Pods with K Switches each

Racks of servers
Collisions
Single-path TCP collisions reduce throughput
Collision
Not fair
Not fair
No matter how you do it, mapping each flow to a path is the wrong goal.
Instead, we should pool capacity from different
Instead, we should pool capacity from different
Instead, we should **pool** capacity from different
Instead, we should **pool** capacity from different sources.
Multipath Transport
Multipath Transport can pool datacenter networks

- Instead of using one path for each flow, use many random paths

- Don’t worry about collisions.

- Just don’t send (much) traffic on colliding paths
Multipath TCP Primer [IETF MPTCP WG]

- MPTCP is a drop in replacement for TCP
- MPTCP spreads application data over multiple subflows
Multipath TCP: Congestion Control [NSDI, 2011]
MPTCP better utilizes the FatTree network
MPTCP on EC2

- Amazon EC2: infrastructure as a service
  - We can borrow virtual machines by the hour
  - These run in Amazon data centers worldwide
  - We can boot our own kernel

- A few availability zones have multipath topologies
  - 2-8 paths available between hosts not on the same machine or in the same rack
  - Available via ECMP
Amazon EC2 Experiment

- 40 medium CPU instances running MPTCP
- For 12 hours, we sequentially ran all-to-all iperf cycling through:
  - TCP
  - MPTCP (2 and 4 subflows)
MPTCP improves performance on EC2

![Graph showing MPTCP and TCP performance](image)
What do the benefits depend on?
- How many subflows are needed?
- How does the topology affect results?
- How does the traffic matrix affect results?
At most 8 subflows are needed

Total Throughput

Throughput (% of optimal)

TCP  2  3  4  5  6  7  8

Multipath TCP
MPTCP improves fairness in VL2 topologies

Fairness is important: Jobs finish when the slowest worker finishes.
MPTCP improves throughput and fairness in BCube
Oversubscribed Topologies

- To saturate full bisectional bandwidth:
  - There must be no traffic locality
  - All hosts must send at the same time
  - Host links must not be bottlenecks

- It makes sense to under-provision the network core
  - This is what happens in practice
  - Does MPTCP still provide benefits?
Performance improvements depend on traffic matrix

![Graph showing relative MPTCP throughput vs. connections per host. The graph compares TCP and MPTCP performance, highlighting the sweet spot where MPTCP outperforms TCP. The x-axis represents connections per host, ranging from 0.001 to 10, while the y-axis shows relative MPTCP throughput from 0.9 to 1.8. The sweet spot is marked by an increase in load, where MPTCP provides better performance than TCP.]
What is an optimal datacenter topology for multipath transport?
In single homed topologies:

- Hosts links are often bottlenecks
- ToR switch failures wipe out tens of hosts for days

Multi-homing servers is the obvious way forward
Fat Tree Topology
Fat Tree Topology

Upper Pod Switch

ToR Switch

Servers
Dual Homed Fat Tree Topology
Is DHFT any better than Fat Tree?

- Not for traffic matrices that fully utilize the core
- Let’s examine random traffic patterns
  - Other TMs in the paper
DHFT provides significant improvements when core is not overloaded.
Summary

- “One flow, one path” thinking has constrained datacenter design
  - Collisions, unfairness, limited utilization

- Multipath transport enables resource pooling in datacenter networks:
  - Improves throughput
  - Improves fairness
  - Improves robustness

- “One flow, many paths” frees designers to consider topologies that offer improved performance for similar cost
Effect of MPTCP on short flows

- Flow sizes from VL2 dataset
- MPTCP enabled for long flows only (timer)
- Oversubscribed Fat Tree topology
- Results:

<table>
<thead>
<tr>
<th></th>
<th>TCP/ECMP</th>
<th>MPTCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completion time:</td>
<td>79ms</td>
<td>97ms</td>
</tr>
<tr>
<td>Core Utilization:</td>
<td>25%</td>
<td>65%</td>
</tr>
</tbody>
</table>
MPTCP vs Centralized Dynamic Scheduling

![Bar chart comparing MPTCP and Centralized Scheduling](image)

Throughput (% of max)

Centralized Scheduling

- Infinite
- 1s
- 500ms
- 100ms
- 10ms

MPTCP
Effect of Locality in the Dual Homed Fat Tree
Overloaded Fat Tree: better fairness with Multipath TCP