Experimenting with Multipath TCP

Sébastien Barré*, Olivier Bonaventure*
Costin Raiciu**, Mark Handley**

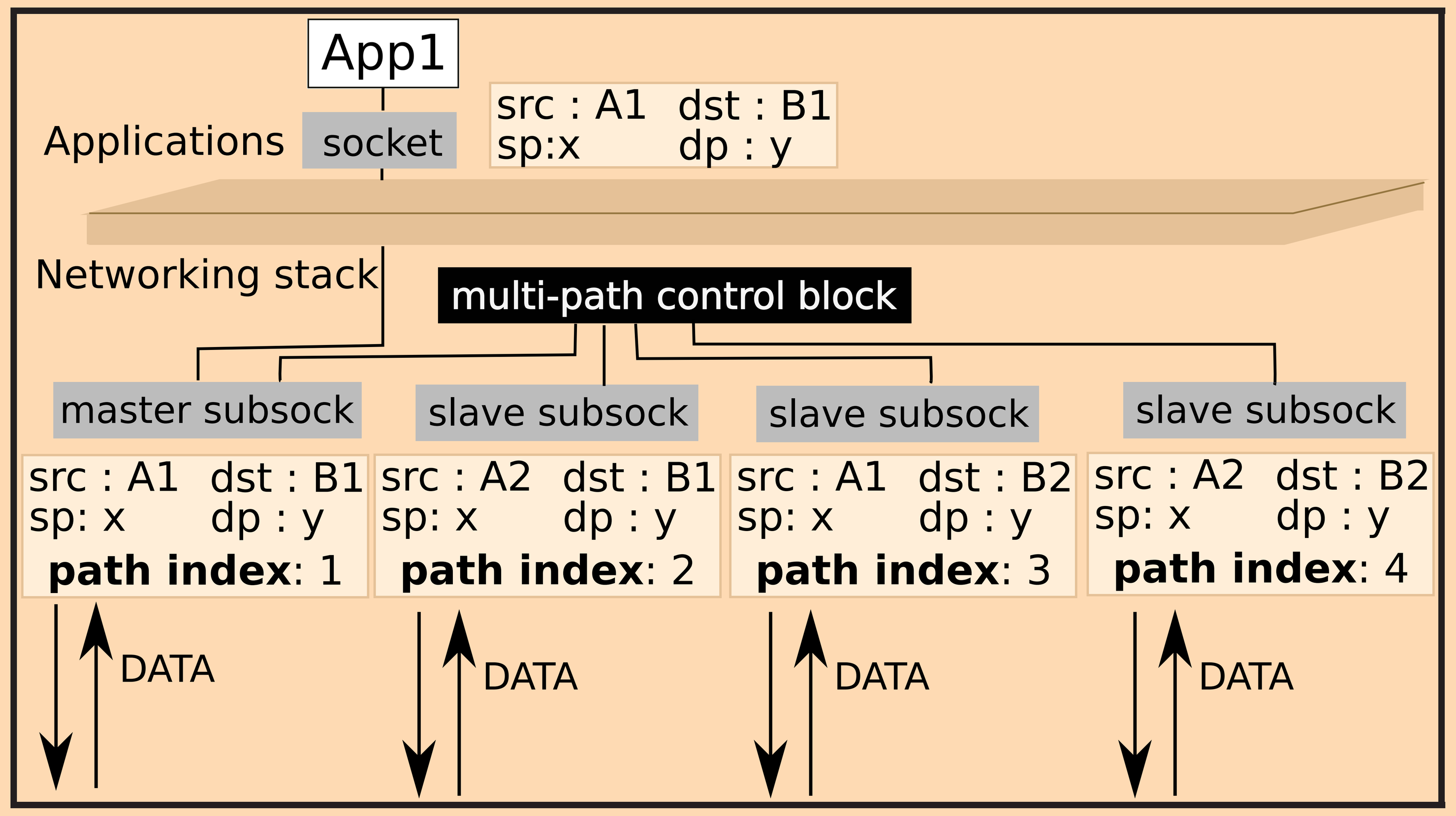
*IP Networking Lab, Université catholique de Louvain
**University College of London

Introduction

- More and more mobile devices can use multiple mediums to access the Internet (3G, wifi, . . .).
- Why not to use them **simultaneously** ?
- Advantages:
  - Better experienced throughput
  - Better tolerance to failures
  - Possibility to switch to a cheaper medium when available
- At which layer to (de)multiplex ? ⟶ The transport layer
  - Doing it lower could create reordering in TCP, causing a drop in the performance.
  - TCP handles a lot of information about path properties, and can quickly react to changes of those properties.

Architecture overview

- One socket manages the communications with the application (master subsocket)
- Upon a `connect()` or `accept()`, a new multipath control block is created.
  - Responsible for shared buffer management,
  - Holds the list of available subflows,
  - Manages connection-level reordering queues.
- When new paths are discovered (alternate local or remote addresses), slave subsockets are created.
  - Slave subsockets behave like normal TCP sockets, but are completely hidden to the application.

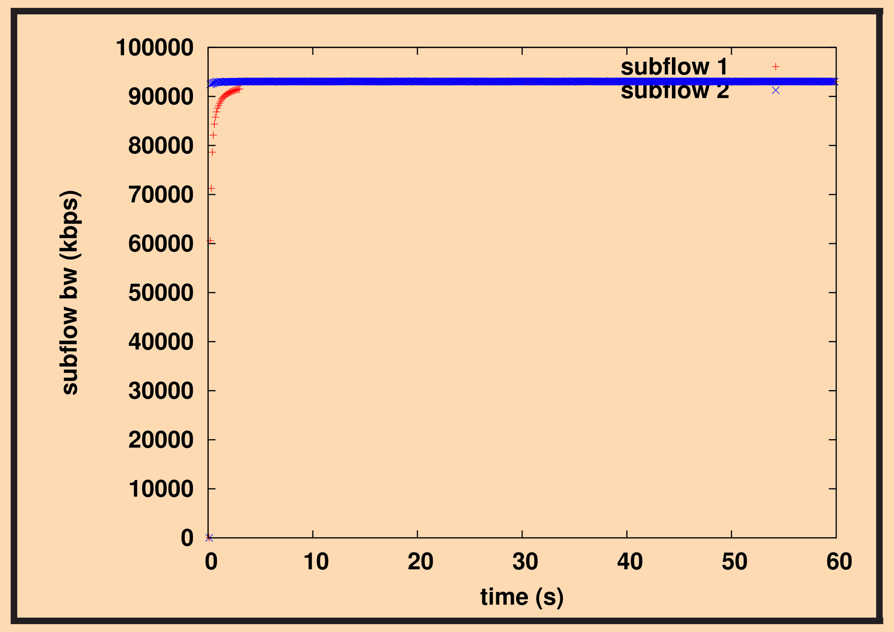


MPTCP protocol overview

- SYN exchange with "Mulipath capable" option.
- If the peer replies with the same option, it supports MPTCP.
- Optional address exchange to allow more paths to be established.
- Additional paths are established just like independent TCP flows (3-way handshake)
- A scheduler decides on which path to send every segment.
- Upon timeout on any path, data is retransmitted on another path.
  - Allows recovering from failures.

MPTCP in action

- Adding one 100mbps wire just doubles the bandwidth:



- MPTCP can use several subflows with different path properties

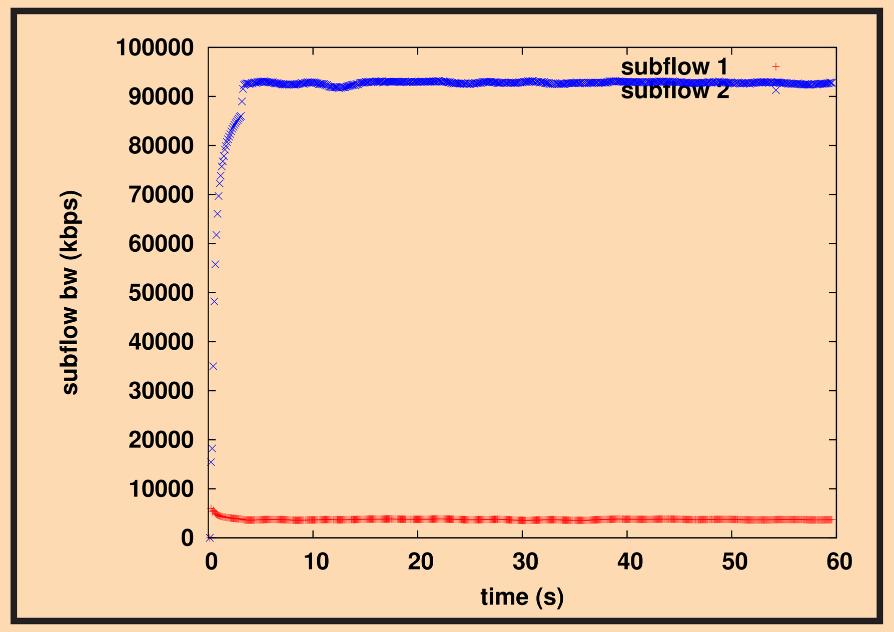
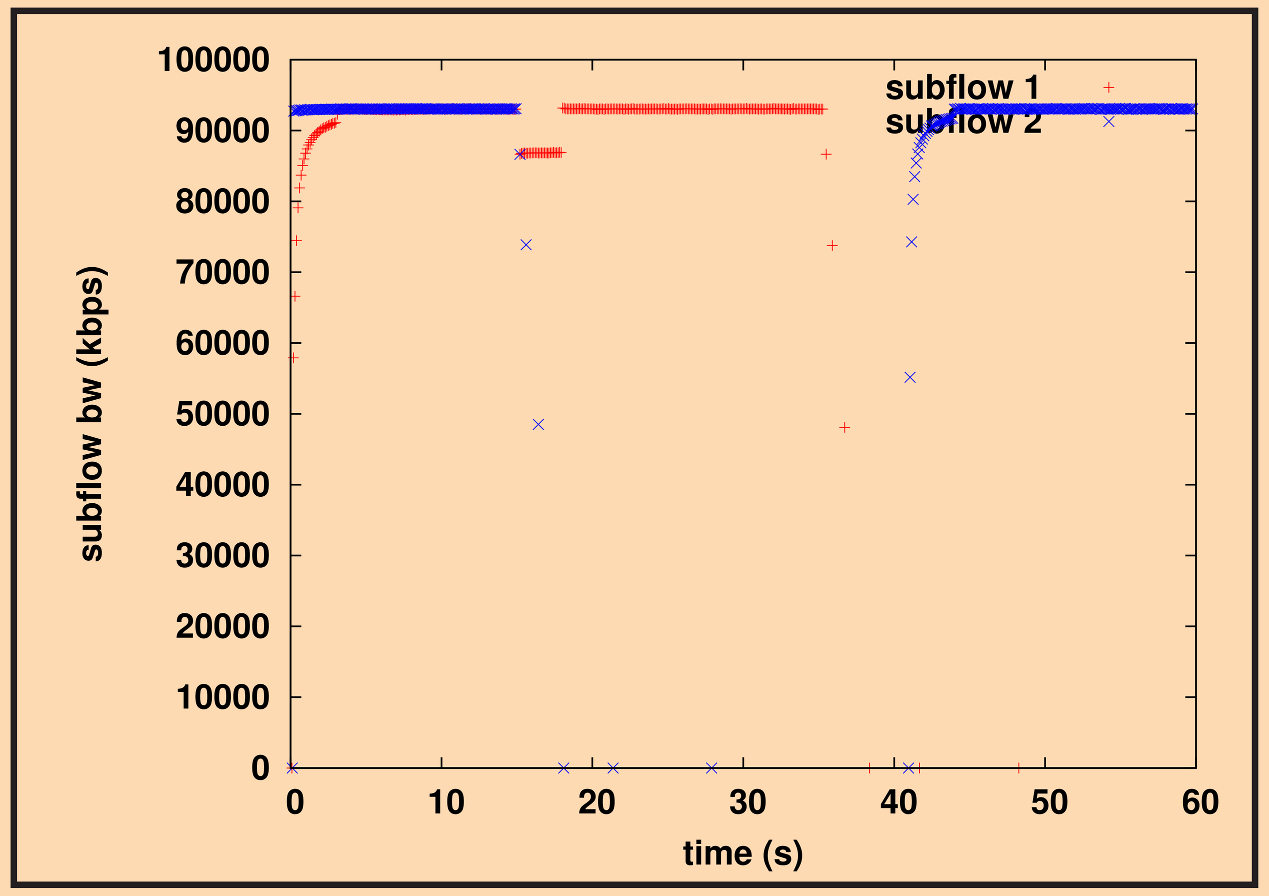


Figure: First path at 100mbps, second one at 10mbps

Failure support

- Upon failure, MPTCP moves to another working subflow:



  - Communication starts with two 100mbps paths
  - After 15 seconds, we simulate a link failure on path 2.
  - Very fast (one RTO later), path 1 resends data originally sent on path 2.
  - After 35 seconds, we repair path 2 and simulate a failure on path 1.
  - Due to exponential backoff and slow start, path 2 needs some time to recover, but then comes back to the original 100mbps transmission rate.

Conclusions

- MPTCP allows using several paths **concurrently**
- It supports transparent failover, and reacts at the scale of RTO.
- **unmodified** applications can directly benefit from multipath.
- Current work:
  - System performance impact of MPTCP
  - MPTCP applicability to datacenters
  - Path selection heuristics
- To download MPTCP for Linux:
  `http://inl.info.ucl.ac.be/mptcp`

Acknowledgements

The research results presented herein have received support from the Trilogy (http://www.trilogy-project.eu) research project partially funded by the European Community. The views expressed here are those of the authors only. The European Commission is not liable for any use that may be made of the information in this document.

sebastien.barre@uclouvain.be