

MultiPath TCP : Linux Kernel implementation



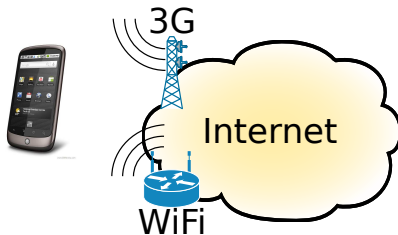
Presenter: Christoph Paasch
IP Networking Lab
Université catholique de Louvain

August 22, 2012

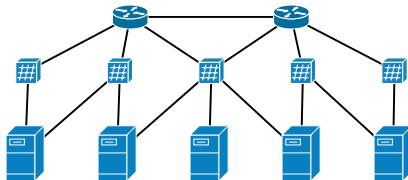
<http://mptcp.info.ucl.ac.be>

Networks are becoming Multipath

- Mobile devices can connect to the Internet via different interfaces

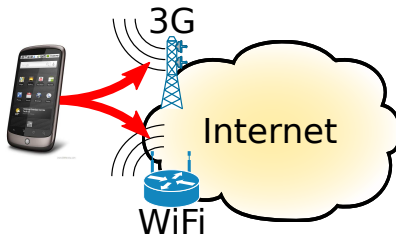


- Data-centers have a large redundant infrastructure

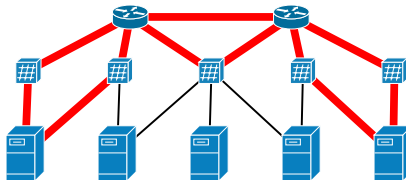


Networks are becoming Multipath

- Mobile devices can connect to the Internet via different interfaces



- Data-centers have a large redundant infrastructure

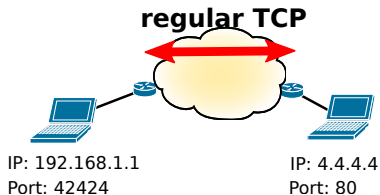


However, the protocols are single-path

- TCP is used for 95% of the Internet communications

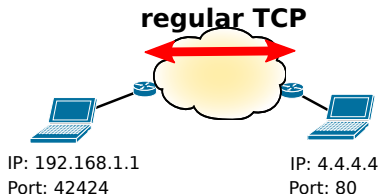
However, the protocols are single-path

- TCP is used for 95% of the Internet communications
- TCP identifies connections by the 5-tuple



However, the protocols are single-path

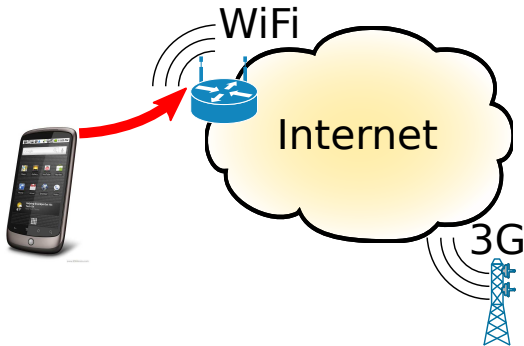
- TCP is used for 95% of the Internet communications
- TCP identifies connections by the 5-tuple



- A single TCP connection cannot be used across different interfaces.

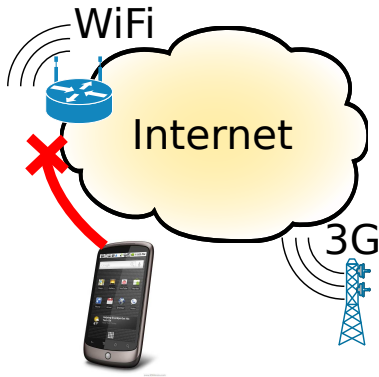
End-hosts don't use the multipath network

Smartphones have to restart their data-transfer when moving away from the WiFi access-point.



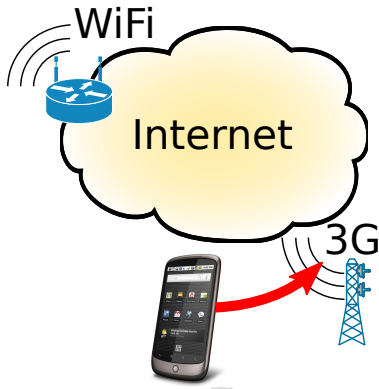
End-hosts don't use the multipath network

Smartphones have to restart their data-transfer when moving away from the WiFi access-point.



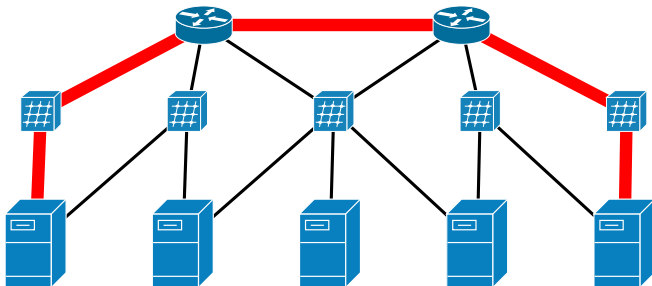
End-hosts don't use the multipath network

Smartphones have to restart their data-transfer when moving away from the WiFi access-point.



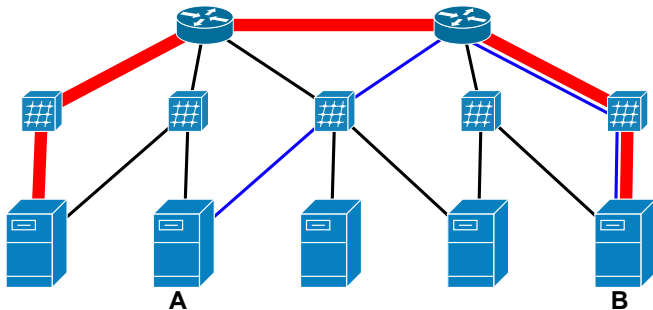
End-hosts don't use the multipath network

Collisions in data-center reduce the bandwidth and result in suboptimal load-balancing



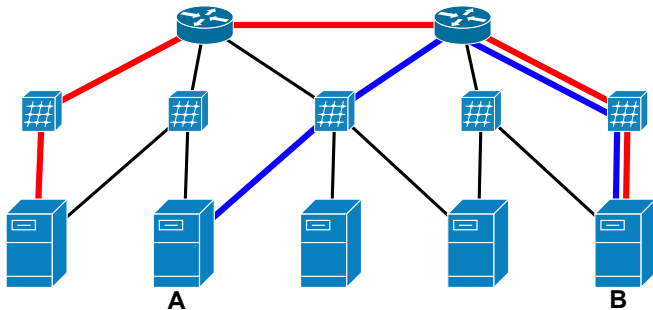
End-hosts don't use the multipath network

Collisions in data-center reduce the bandwidth and result in suboptimal load-balancing



End-hosts don't use the multipath network

Collisions in data-center reduce the bandwidth and result in suboptimal load-balancing



Mismatch between the **multipath network**
and the **single-path transport** protocol.

Transport layer

SCTP

- Needs modifications in the applications
- Does not pass by most middleboxes/firewalls

Network layer

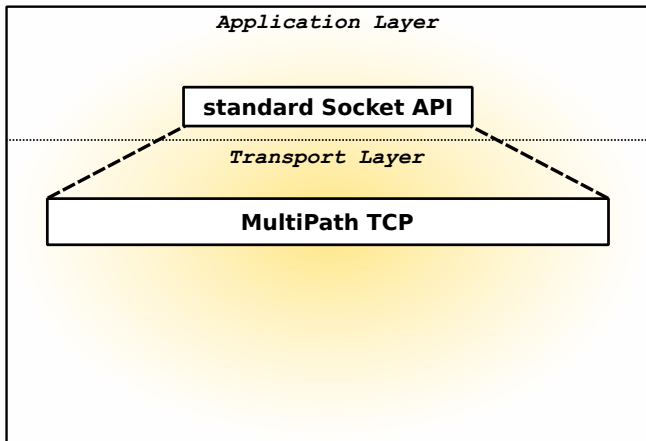
Mobile IP(v6), shim6, HIP,...

- Some are only designed for IPv6
- Do not pass by middleboxes/firewalls
- Are hard to deploy

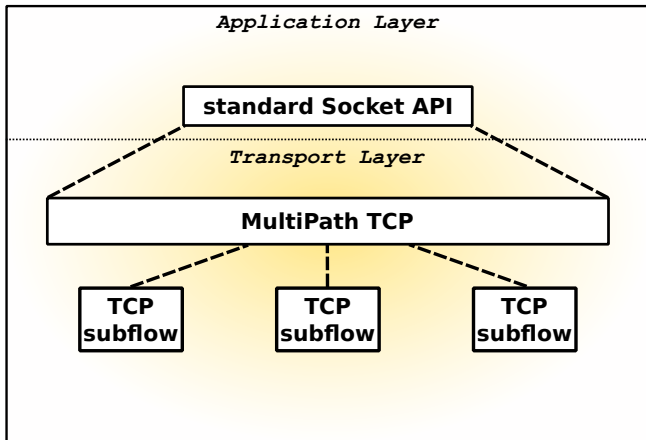
MultiPath TCP

- Runs with **unmodified applications**
- Works over **today's Internet**
- **IPv4/IPv6** are both supported (even simultaneously)

Standard Stream Socket API

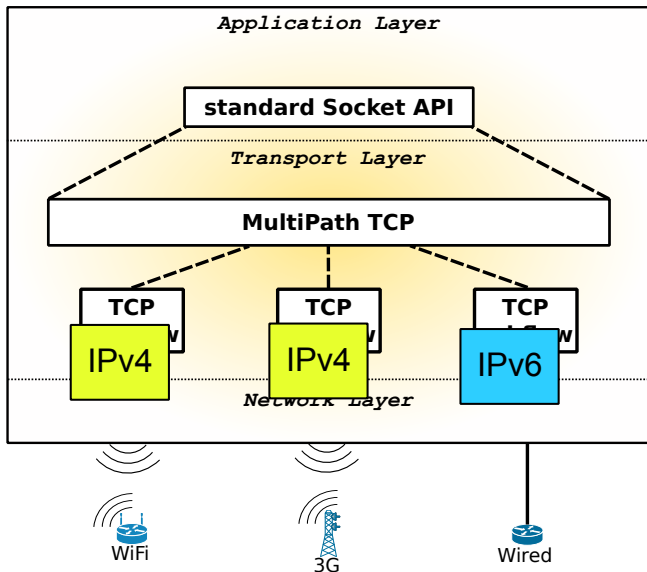


Multiple TCP subflows to pass middleboxes

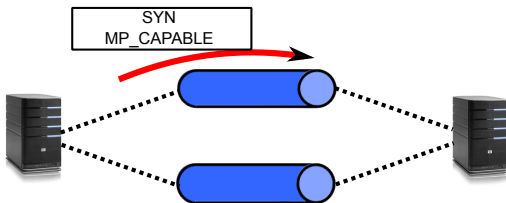


MultiPath TCP

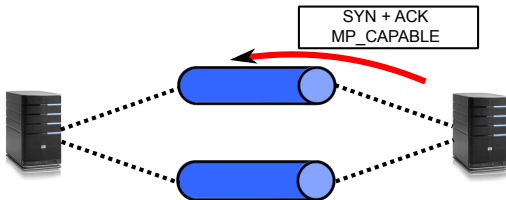
IPv4/IPv6 capable



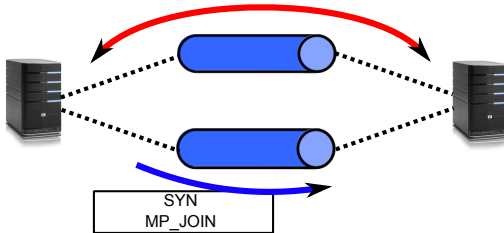
Is the server MPTCP-capable?



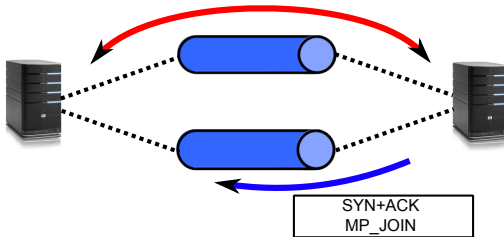
Is the server MPTCP-capable?



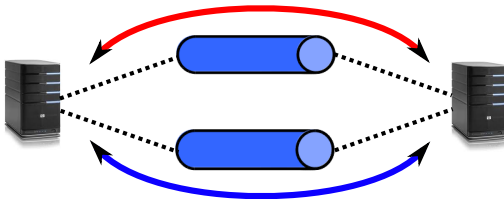
Create new subflows



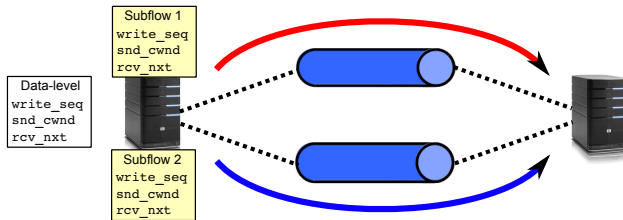
Create new subflows



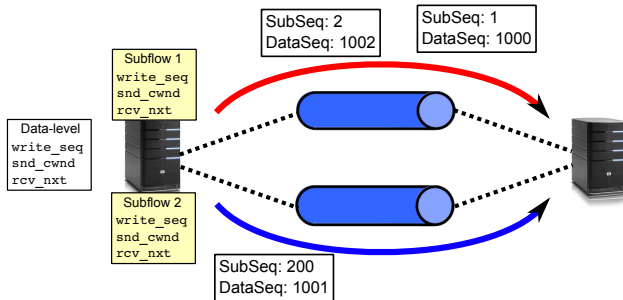
Create new subflows



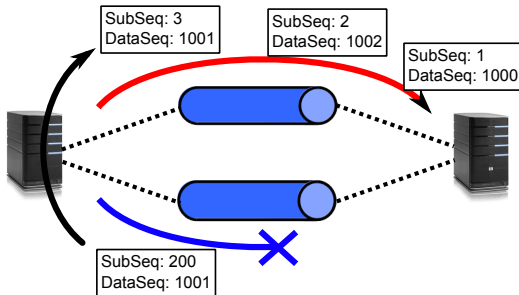
Separate sequence-number space



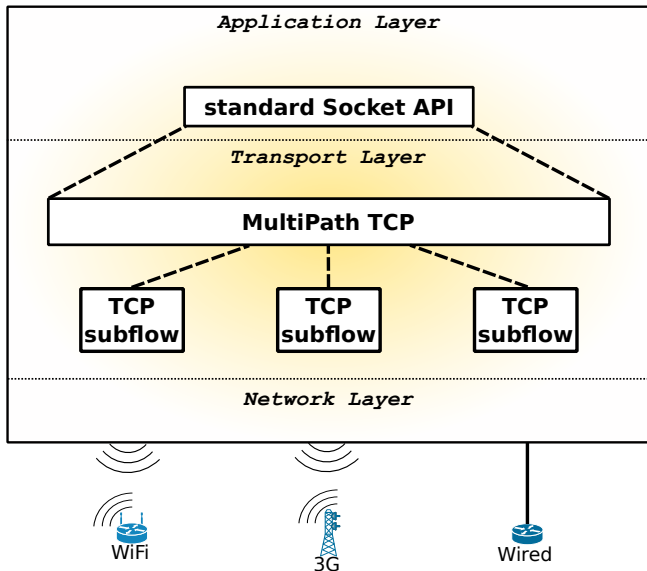
Separate sequence-number space



Cross-subflow reinjection



MultiPath TCP

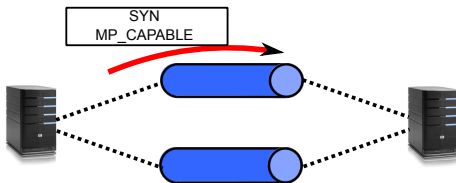


MultiPath TCP

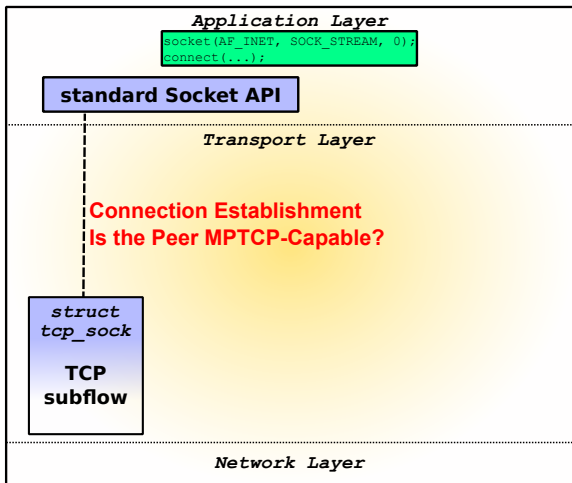
Linux Kernel Implementation

Available at <http://mptcp.info.ucl.ac.be>

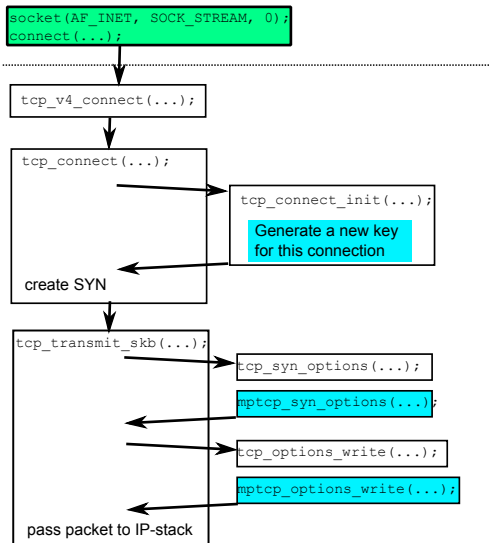
Exchanged Messages



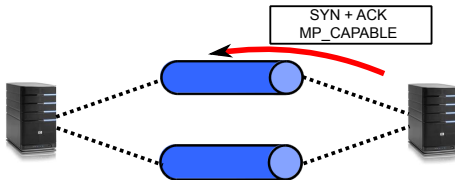
High-Level Kernel design



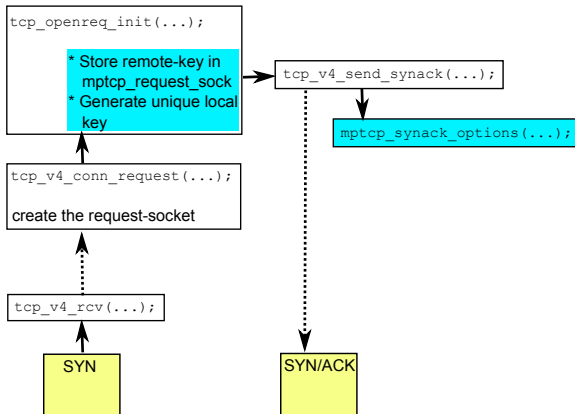
In-depth call-stack



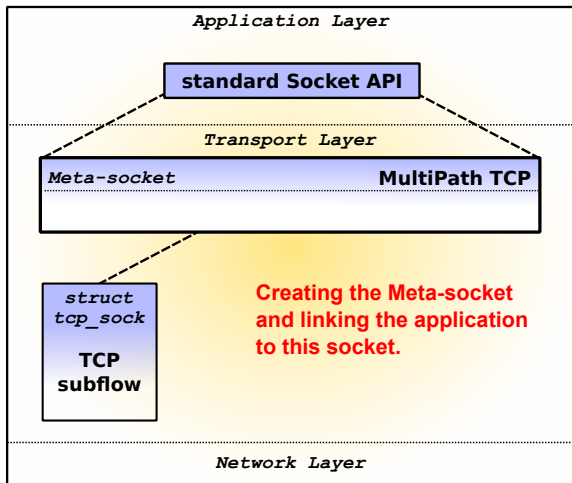
Exchanged Messages



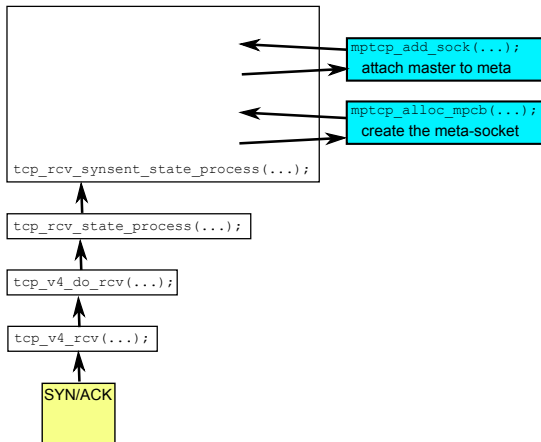
In-depth call-stack



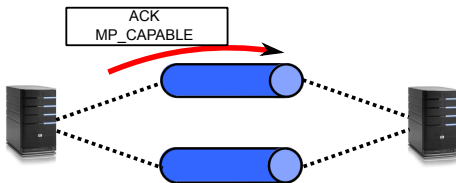
High-Level Kernel design



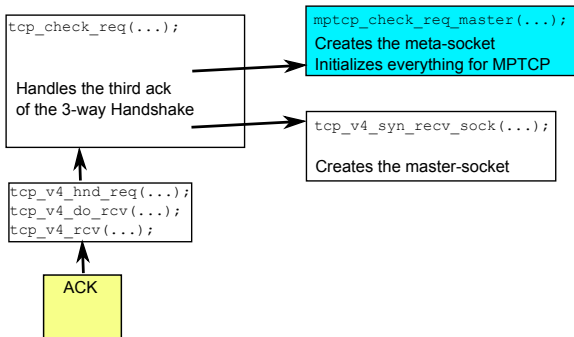
In-depth call-stack



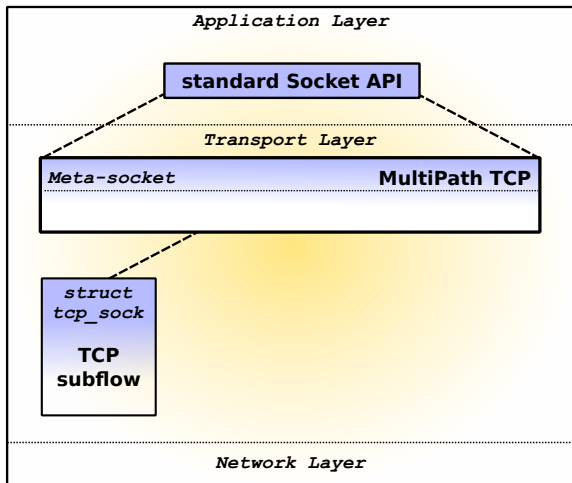
Exchanged Messages



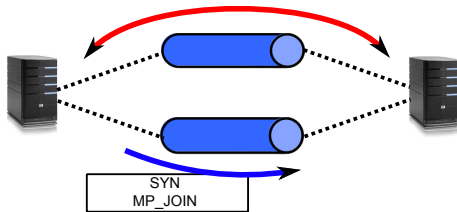
In-depth call-stack



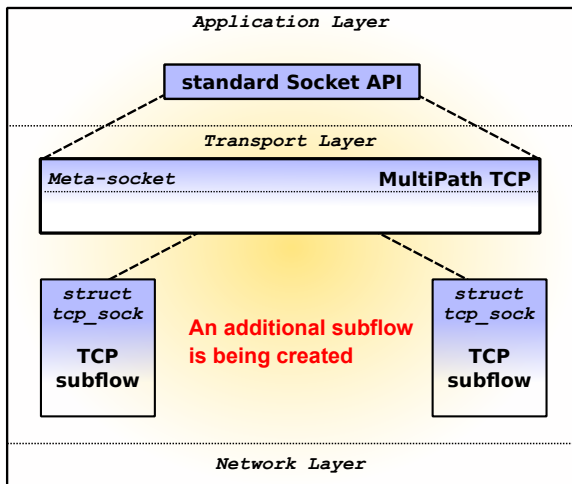
High-Level Kernel design



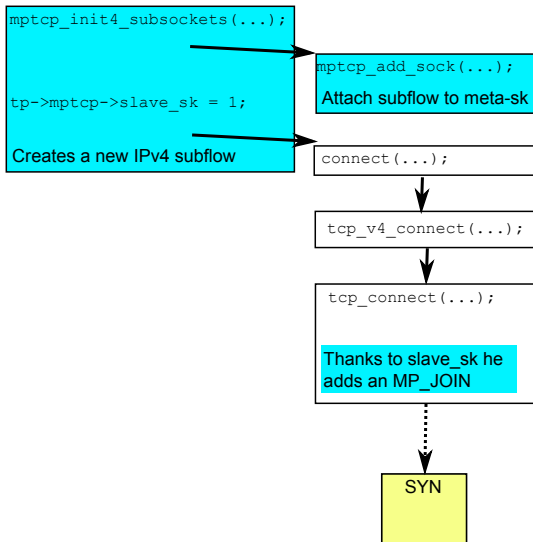
Exchanged Messages



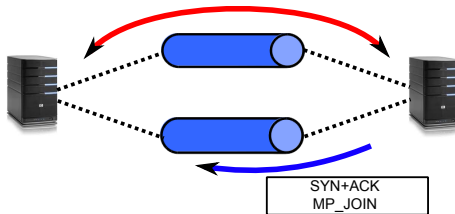
High-Level Kernel design



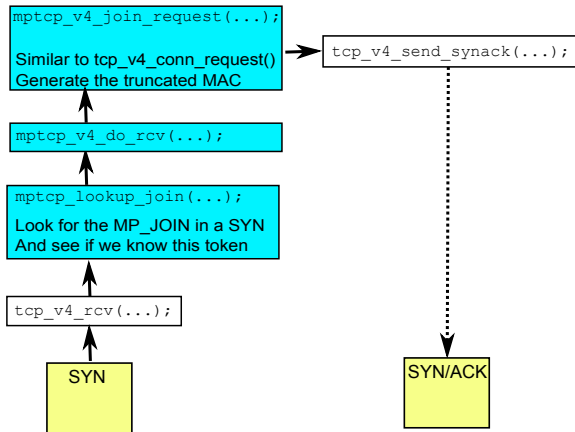
In-depth call-stack



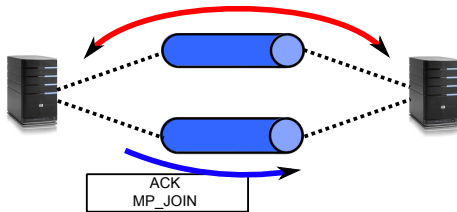
Exchanged Messages



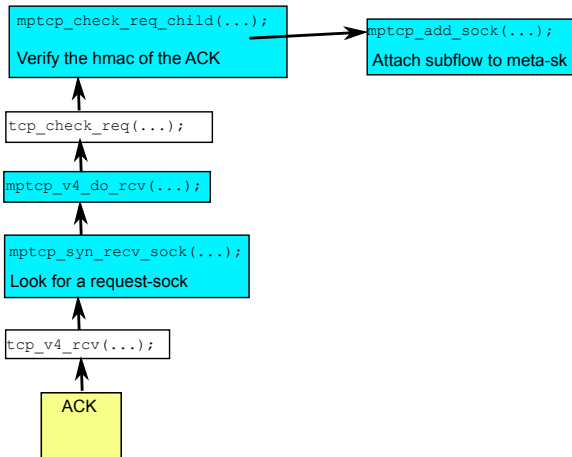
In-depth call-stack



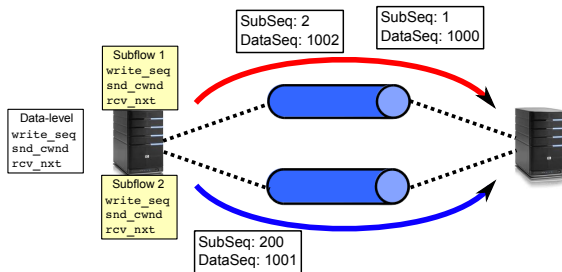
Exchanged Messages



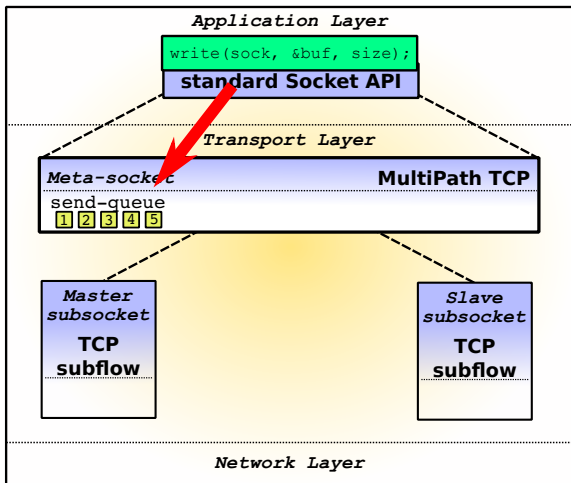
In-depth call-stack



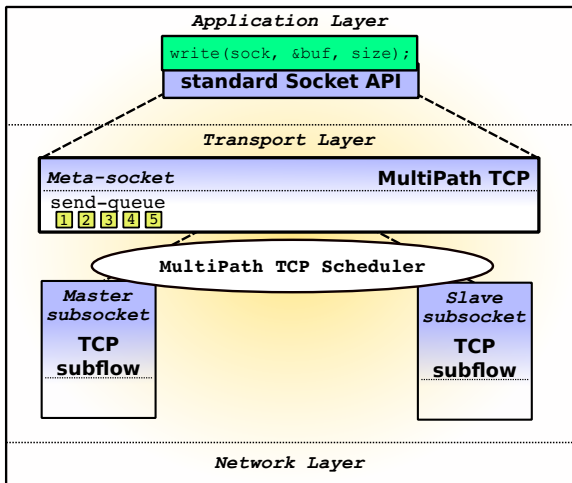
Exchanged Messages



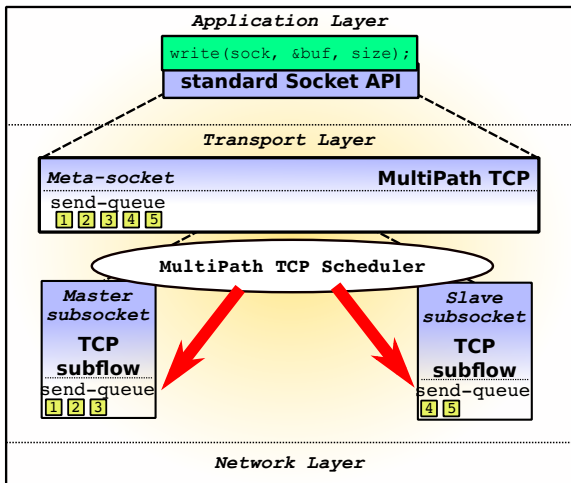
High-Level Kernel design



High-Level Kernel design

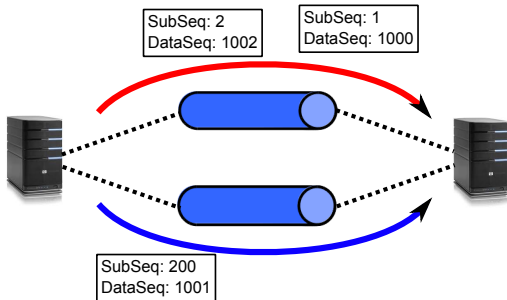


High-Level Kernel design



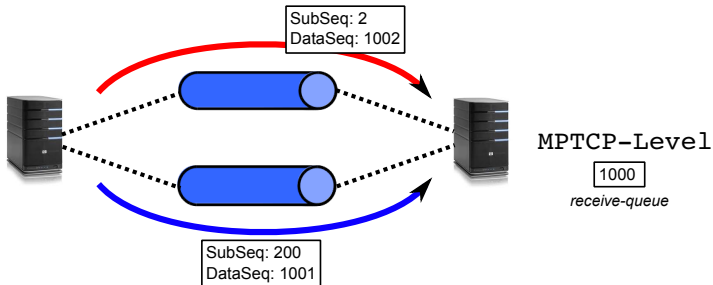
Receiving Data

Packets can be reordered at the data-level due to delay-differences.



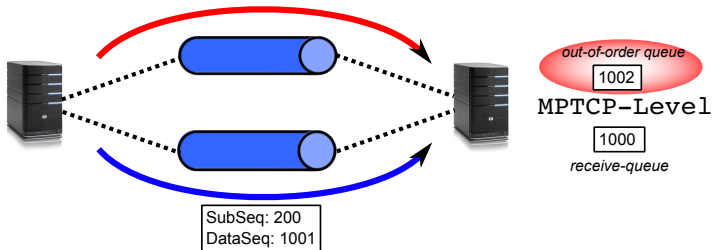
Receiving Data

Packets can be reordered at the data-level due to delay-differences.



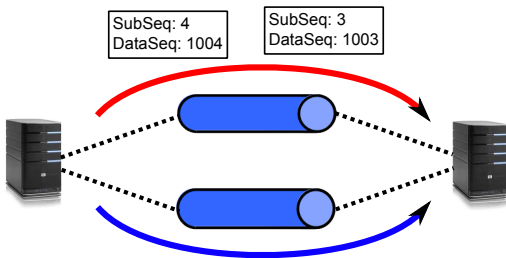
Receiving Data

Packets can be reordered at the data-level due to delay-differences.



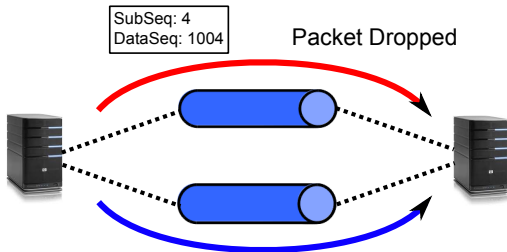
Receiving Data

A loss at the subflow-level (or network-reordering) can also cause reordering at the subflow-level

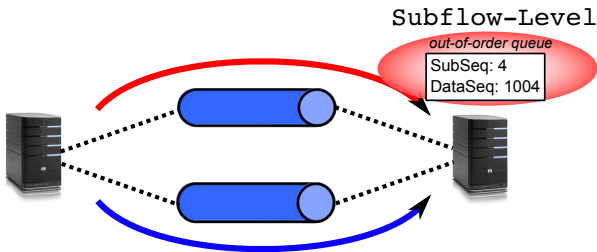


Receiving Data

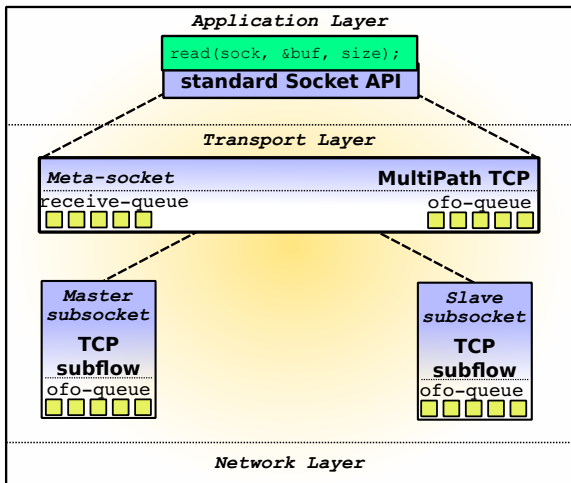
A loss at the subflow-level (or network-reordering) can also cause reordering at the subflow-level



Subflow-level out-of-order queues are necessary to handle the retransmission at the subflow-level



High-Level Kernel design

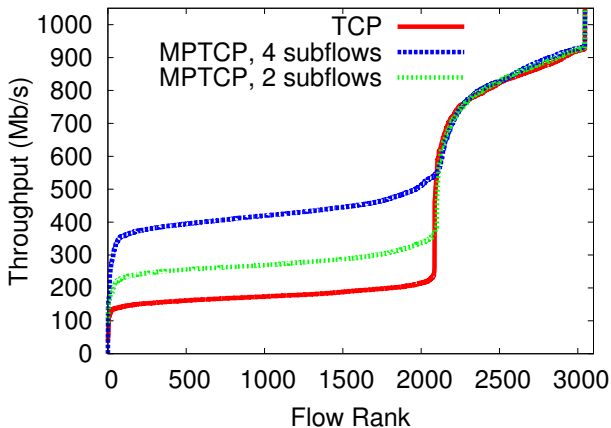


MultiPath TCP

Performance Evaluation

- 2-8 paths available between hosts not on the same machine/rack
- ECMP load-balancing
- 40 medium CPU instances running MPTCP
- During 12 hours, all-to-all iperf with:
 - TCP
 - MPTCP (2 subflows)
 - MPTCP (4 subflows)

MPTCP on Amazon EC2

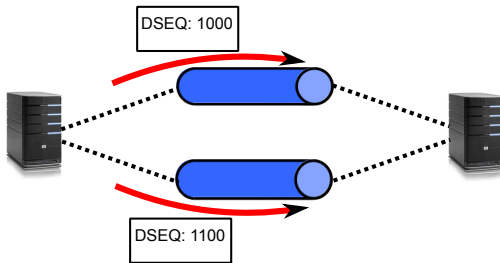


MultiPath TCP on a Smartphone/Mobile Node

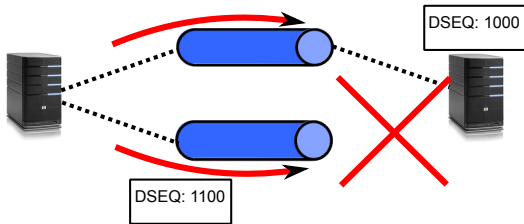


- Smartphones have multiple interfaces (WiFi/3G)
- MultiPath TCP can be a benefit for these devices
 - Increased Bandwidth
 - Increased Resilience

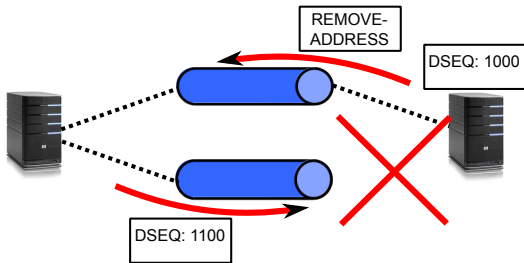
MultiPath TCP handover from WiFi to 3G



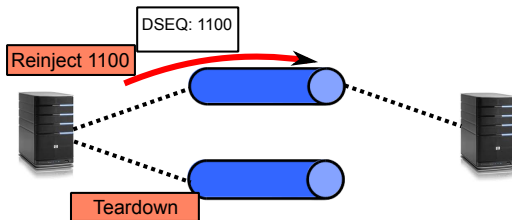
MultiPath TCP handover from WiFi to 3G



MultiPath TCP handover from WiFi to 3G

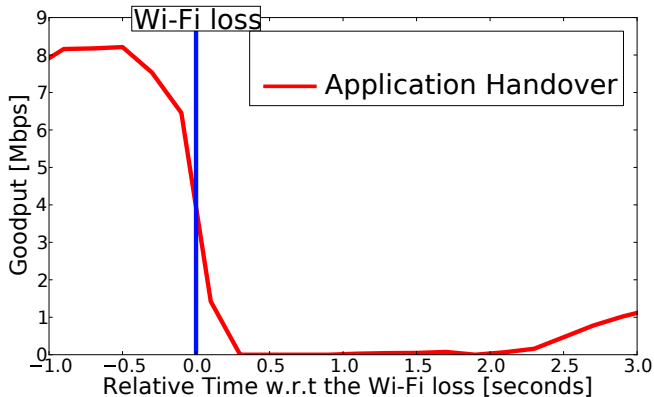


MultiPath TCP handover from WiFi to 3G

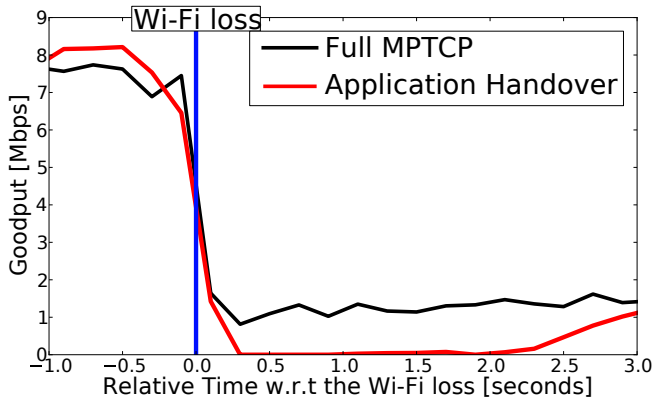


- Regular TCP would break!
- Some applications support recovering from a broken TCP (HTTP-Header Range)
- Thanks to the *REMOVE_ADDR*-option, MPTCP is able to handle this without the need for application support.

MultiPath TCP handover from WiFi to 3G

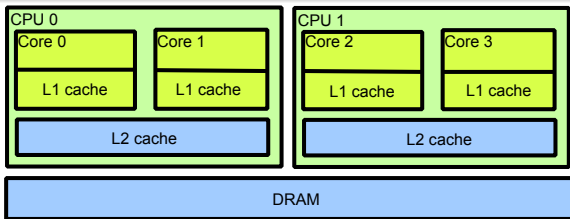


MultiPath TCP handover from WiFi to 3G

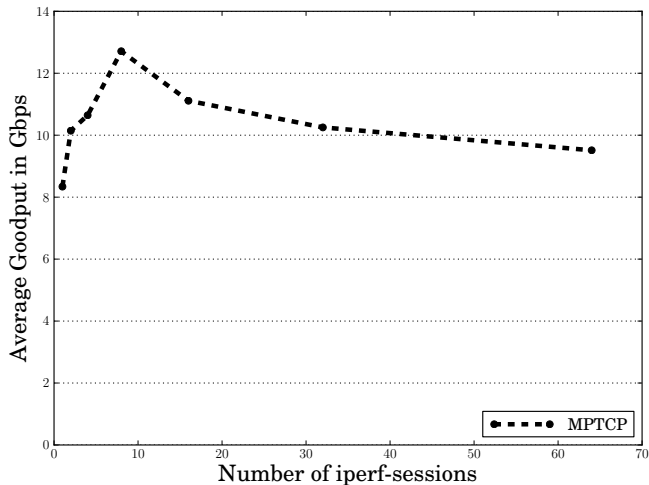


Flow-to-core affinity

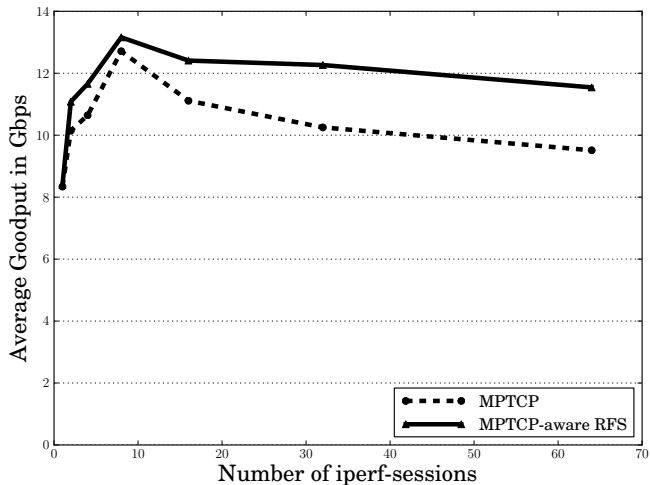
- Individual TCP-flows are steered to the same CPU-core to avoid reordering inside the receive-code.
- MPTCP has lots of L1/L2 cache-misses because the individual subflows are steered on different CPU-cores.
- MPTCP-aware Receive-Flow-Steering sends all subflows on the same CPU-core.



Flow-to-core affinity - 10 Gbps interfaces



Flow-to-core affinity - 10 Gbps interfaces



MultiPath TCP **increases the bandwidth** and allows seamless **mobile handover**.

MultiPath TCP can be used with **unmodified applications**, over **today's Internet**.

Freely available at **<http://mptcp.info.ucl.ac.be>**
Download it, try it out, contribute!

UCLouvain MPTCP-Team:

Christoph Paasch

Gregory Detal

Fabien Duchene

Prof. Olivier Bonaventure

Thanks to our previous and present partners/contributors:

