MPTCP : Linux Kernel implementation status

Presenter : Christoph Paasch
IP Networking Lab
Université catholique de Louvain

28 mars 2012

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

Outline

1. Implementation status
2. Implementation challenges
3. Performance results

Implementation status

Compared to last IETF-80 presentation (cfr. Sébastien Barré at IETF 80 - Prague March 2011)

- MPTCP security (draft v07)
- IPv6
- Fully support all kind of middleboxes (segment-splitting/coalescing, payload-modifying, pro-actively acking middleboxes,. . .)
- Support reception of 64-bit data-sequence-numbers
- Mobility supported with `REMOVE_ADDR`
- Forced closure supported with `MP_FASTCLOSE`
- Support for api-draft is ongoing
- SMP is supported (new locking architecture)
- MPTCP is on Linux kernel version 3.0 (soon 3.2)

Linux MPTCP commmunity

Total contributions from all people (ordered by number of commits) :

- **Sébastien Barré (UCLouvain - now Thelis)**
- **Christoph Paasch (UCLouvain)**
- Jaakko Korkeaniemi (Aalto)
- **Gregory Detal (UCLouvain)**
- **Fabien Duchêne (UCLouvain)**
- Andreas Seelinger (RWTH-Aachen)
- Andreas Ripke (Neclab)
- Vlad Dogaru (Intel)
- Lavkesh Lahngir (Kanpur University)
- John Ronan (TSSG)
- Brandon Heller (Stanford University)

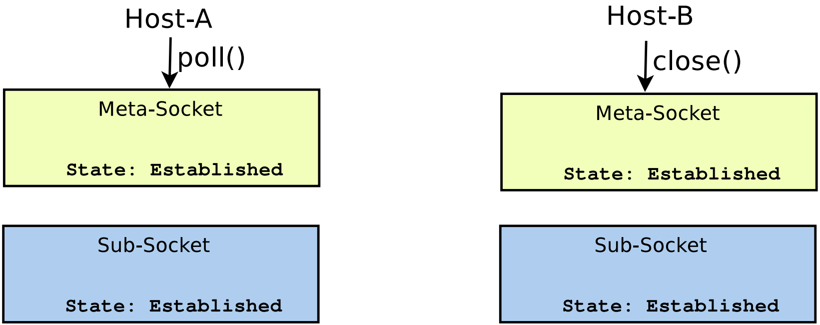
Outline

MPTCP - Avoiding TIME-WAIT

- Applications are able to avoid TIME-WAIT (e.g., apache2, apachebenchmark,...)
- On the data-level this works.
- But not at the subflow-level ...
- Many subflows are lingering around in TIME-WAIT state although the application tried to avoid it.

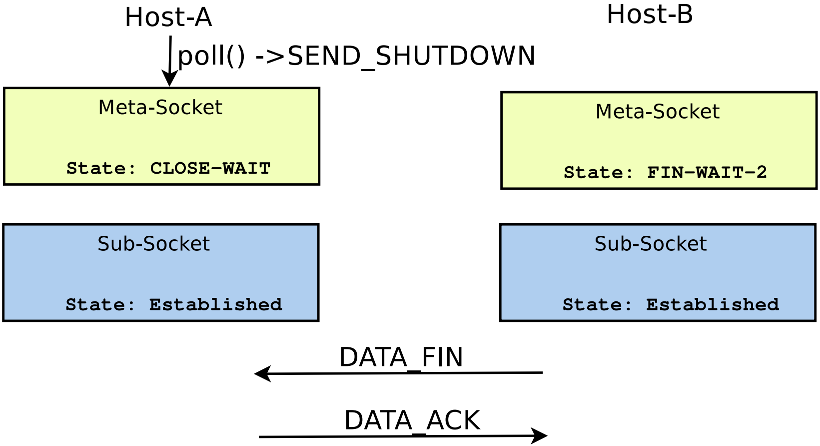
MPTCP - Avoiding TIME-WAIT

Applications poll the socket to do passive closing



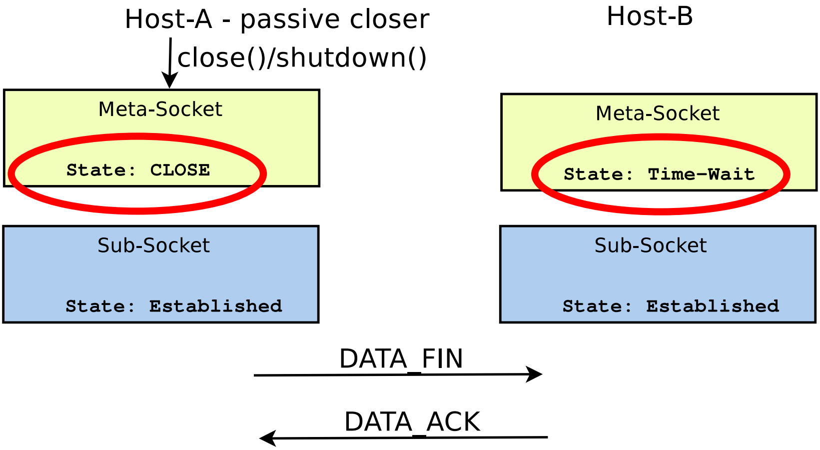
MPTCP - Avoiding TIME-WAIT

Applications poll the socket to do passive closing



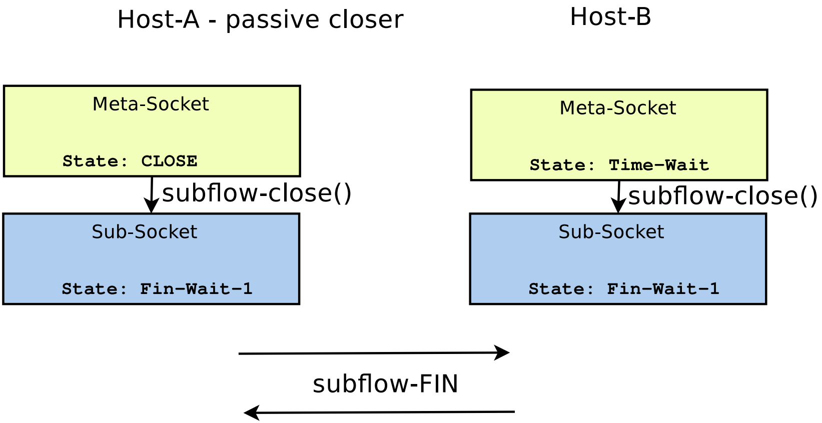
MPTCP - Avoiding TIME-WAIT

After the *DATA_FIN* the application does a passive close



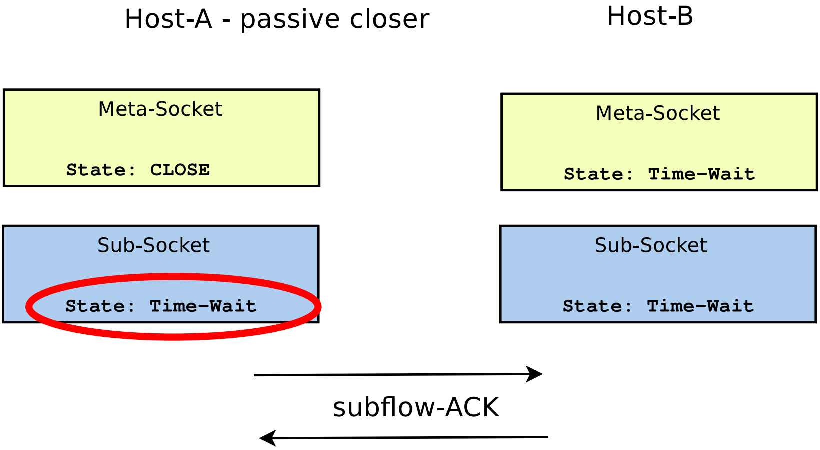
MPTCP - Avoiding TIME-WAIT

However, subflow close does not respect the passive close



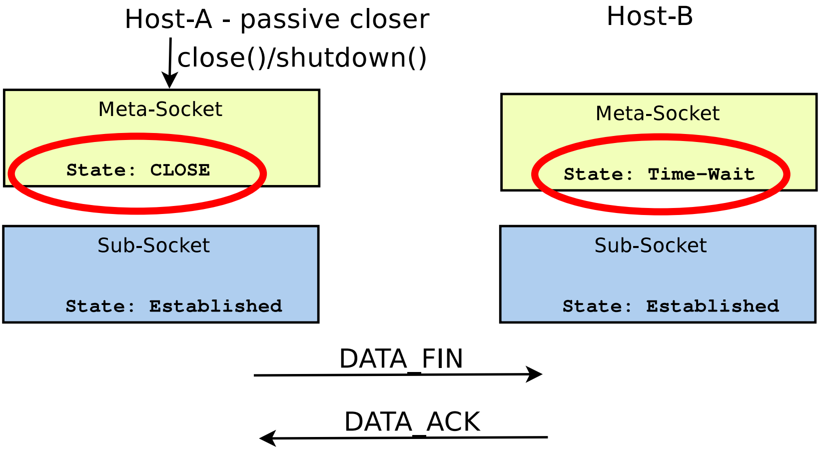
MPTCP - Avoiding TIME-WAIT

However, subflow close does not respect the passive close



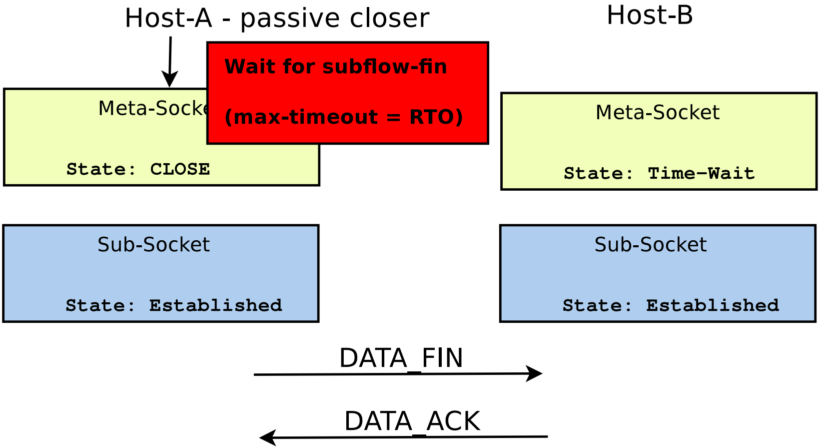
MPTCP - Avoiding TIME-WAIT

How to continue after closing the meta-sockets ?



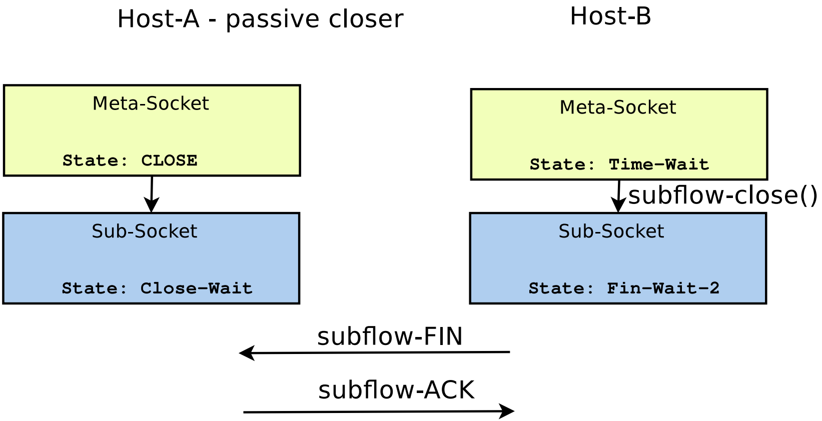
MPTCP - Avoiding TIME-WAIT

Don't close the subflow, wait for the subflow-fin !



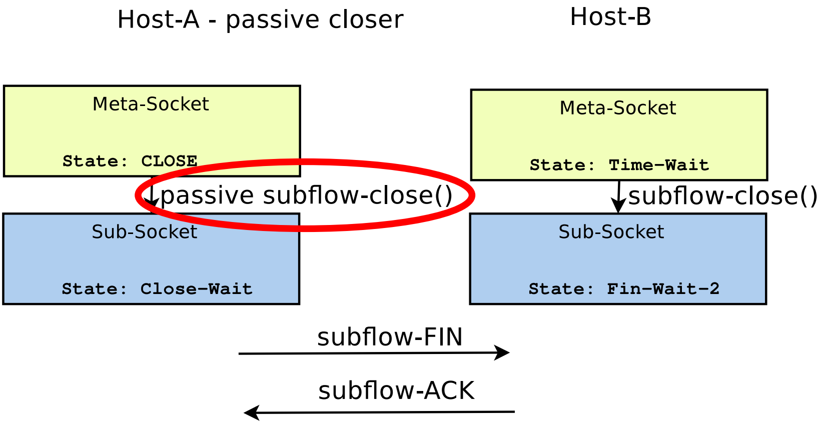
MPTCP - Avoiding TIME-WAIT

Don't close the subflow, wait for the subflow-fin !



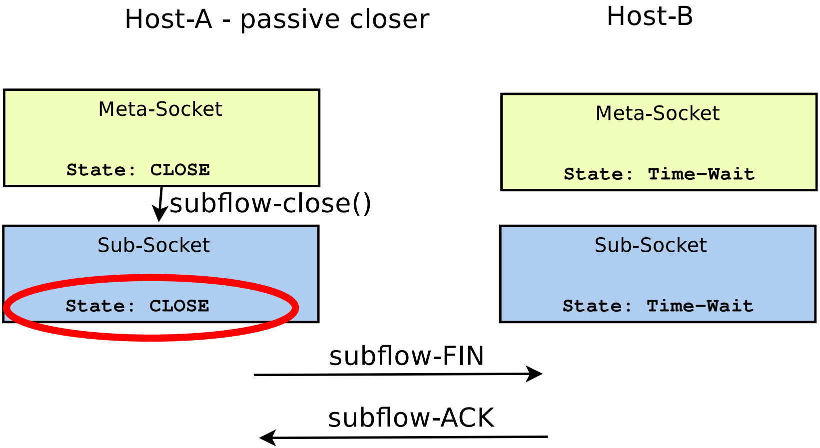
MPTCP - Avoiding TIME-WAIT

Enforced passive-close on the subflow



MPTCP - Avoiding TIME-WAIT

Enforced passive-close on the subflow

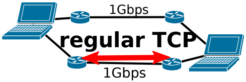


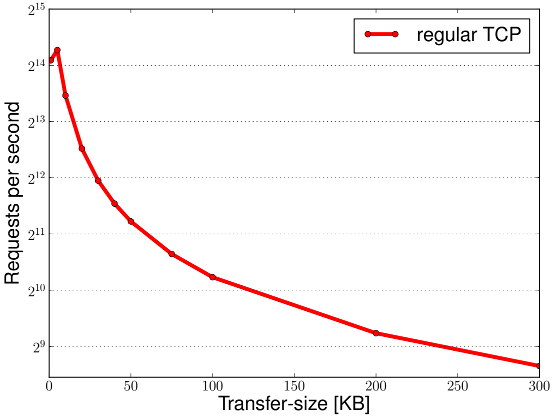
Outline

1. Implementation status
2. Implementation challenges
3. Performance results

Apachebenchmarking micro-flows

100 simultaneous requests, for a total of 100000 requests of varying size

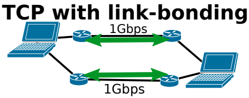


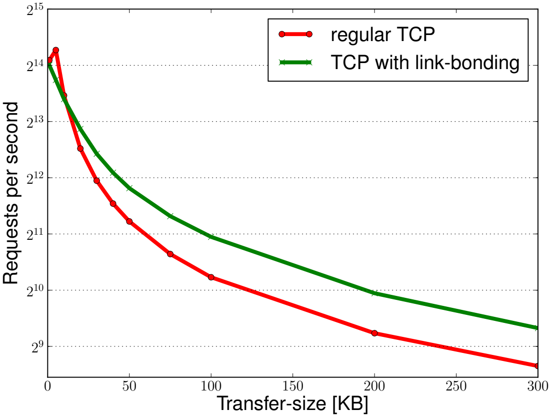


[1] "How Hard Can It Be ? Designing and Implementing a Deployable Multipath TCP" - C. Raiciu, C. Paasch, S. Barré, A. Ford, M. Honda, F. Duchêne, O. Bonaventure, M. Handley. USENIX NSDI. 2012. San Jose (CA).

Apachebenchmarking micro-flows

100 simultaneous requests, for a total of 100000 requests of varying size

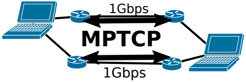


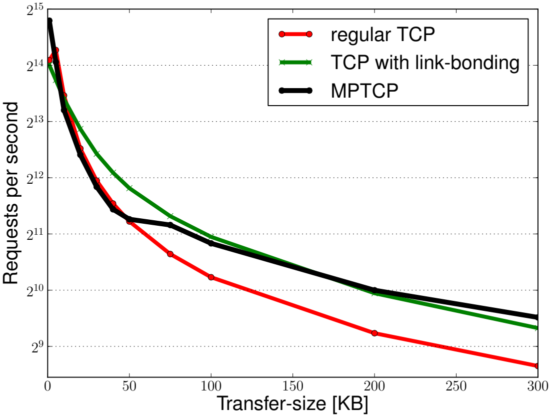


[1] "How Hard Can It Be ? Designing and Implementing a Deployable Multipath TCP" - C. Raiciu, C. Paasch, S. Barré, A. Ford, M. Honda, F. Duchêne, O. Bonaventure, M. Handley. USENIX NSDI. 2012. San Jose (CA).

Apachebenchmarking micro-flows

100 simultaneous requests, for a total of 100000 requests of varying size



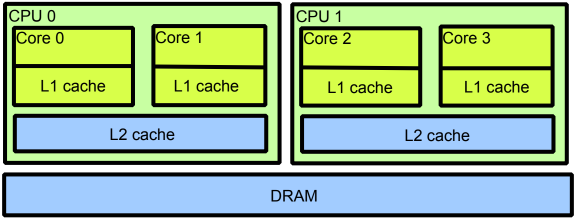


[1] "How Hard Can It Be ? Designing and Implementing a Deployable Multipath TCP" - C. Raiciu, C. Paasch, S. Barré, A. Ford, M. Honda, F. Duchêne, O. Bonaventure, M. Handley. USENIX NSDI. 2012. San Jose (CA).

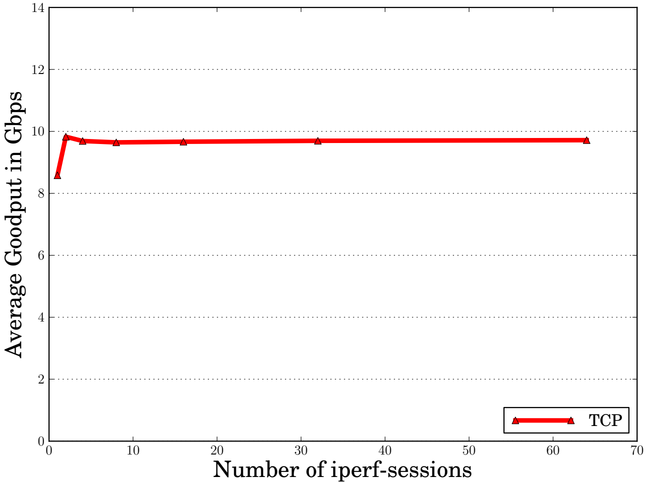
Flow-to-core affinity

Flow-to-core affinity

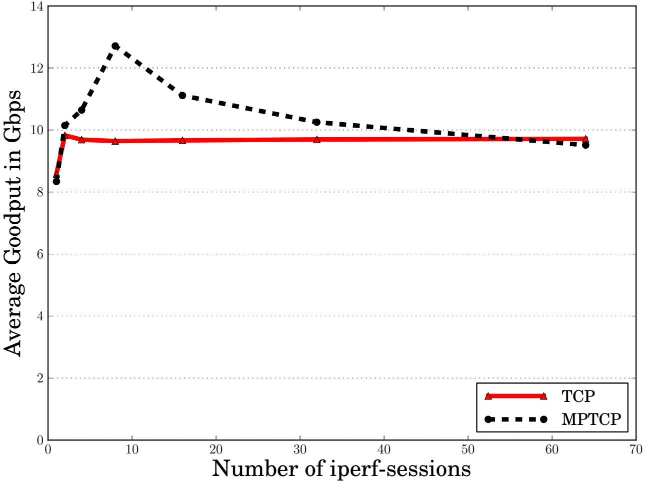
- Individual TCP-flows are steered to the same CPU-core to avoid reordering inside the receive-code.
- MPTCP has lots of L1/L2 cache-misses because the individual subflows are steered on different CPU-cores.
- MPTCP-aware Receive-Flow-Steering sends all subflows on the same CPU-core.



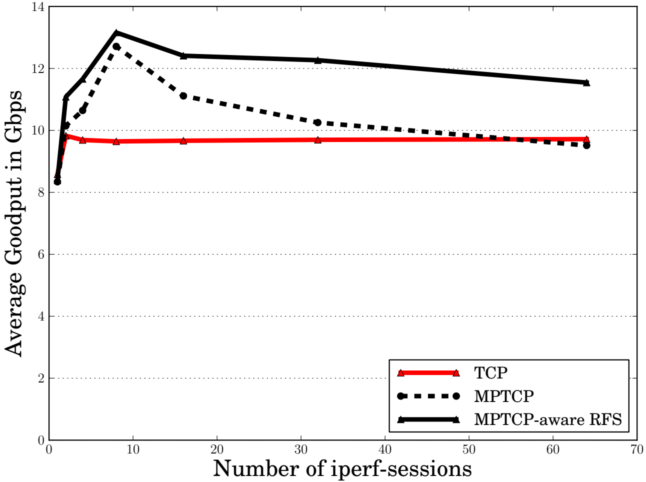
Flow-to-core affinity - 10 Gbps interfaces



Flow-to-core affinity - 10 Gbps interfaces

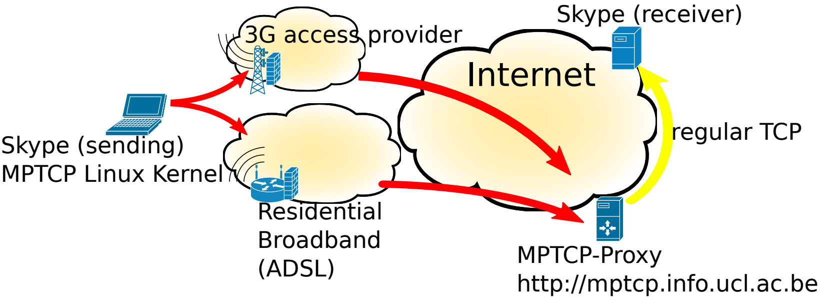


Flow-to-core affinity - 10 Gbps interfaces



Vertical Handover with MPTCP

Skype-call from MPTCP-enabled host via MPTCP-Proxy to regular TCP.
Vertical Handover from WiFi to 3G during the Skype-call.



Next Steps ?

What remains to be done before proposing something to netdev ?

- Minor missing pieces (e.g., sending 64-bit DSN,. . .)
- A cleaner separation between layers to avoid increasing the size of `struct sk_buff`
- Support of TCP SYN-Cookies
- Support of NET-DMA
- Support of TSO
- More cleanup,. . .

Readings :

"How Hard Can It Be ? Designing and Implementing a Deployable Multipath TCP" C. Raiciu, C. Paasch, S. Barré, A. Ford, M. Honda, F. Duchêne, O. Bonaventure, M. Handley. USENIX NSDI'12. San Jose (CA). 2012.

"Implementation and assessment of Modern Host-based Multipath Solutions" S. Barré. PhD Thesis. Université catholique de Louvain. 2011.

"Improving Datacenter Performance and Robustness with Multipath TCP" C. Raiciu, S. Barré, C. Pluntke, A. Greenhalgh, D. Wischik and M. Handley. ACM SIGCOMM 2011. Toronto (Canada). August 2011.

"MultiPath TCP : From Theory to Practice" S. Barré, C. Paasch and O. Bonaventure. IFIP Networking. Valencia (Spain). 2011.

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

Install MPTCP and use it ! ! ! :)