Implementation and Assessment of Modern Host-based Multipath Solutions

Sébastien Barré

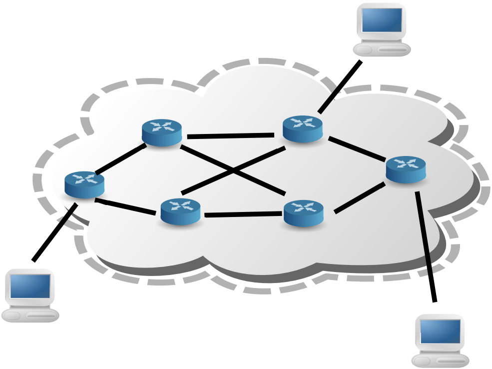
ICTEAM, Université catholique de Louvain
http://inl.info.ucl.ac.be

Examining board: Marcelo Bagnulo, Olivier Bonaventure, Christophe De Vleeschouwer, Mark Handley, Marc Lobelle, Rolf Winter

Nov. 2nd, 2011

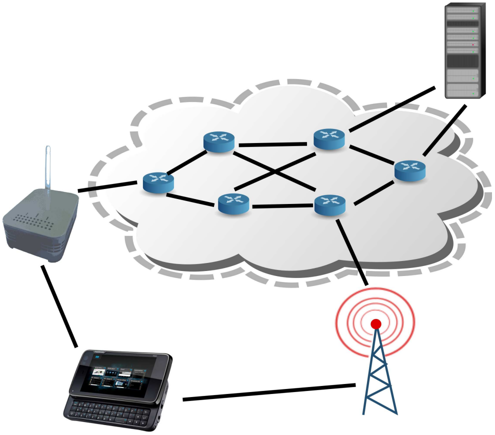
phD public defense

Motivations - Multipath in the past



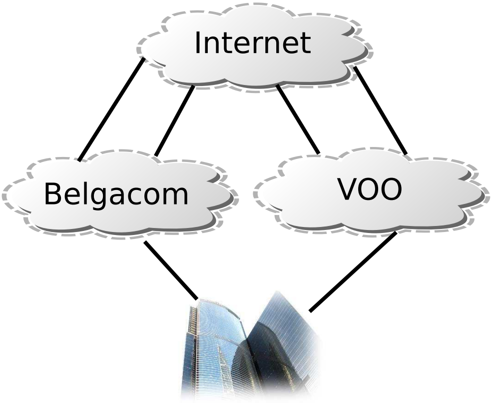
- One link connects a host to the network
- Multiple links are used inside the network

Motivations - Multipath today



- Still multiple links inside the network, but...
- Several links connect the client to the network (e.g. mobile phones)
- The same holds for servers (e.g. datacenters)

Motivations - Multipath today



- Companies also connect to multiple providers
- this improves the availability of the company services
- but moving connections across providers still breaks communications. . .

Motivations - new protocols: Shim6

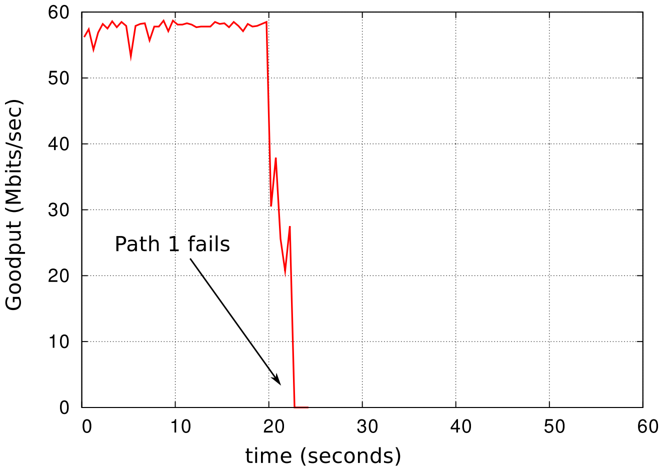


Figure: Path failure with TCP

Motivations - new protocols: Shim6

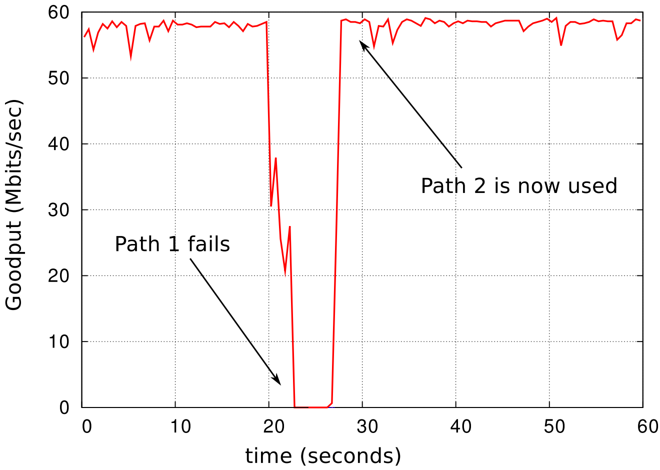


Figure: Path failure with Shim6

Motivations - new protocols: MPTCP

- Assume two hosts are connected with two 1Gbps links
  - Current connections (regular TCP) will get 1Gbps and use only one of the links
  - MPTCP can achieve **2Gbps**, without any change to the application (only to the operating system)

Motivations - The problems

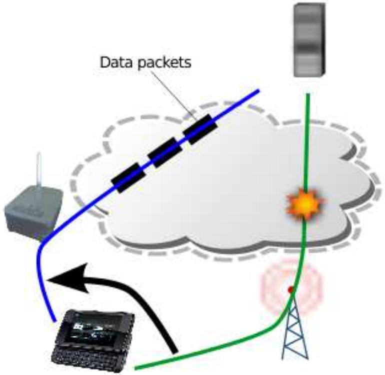
- Several protocols have been proposed (HIP[MN06], **Shim6**[NB09], **MPTCP**[FRHB11], LISP[FFML11], ILNP[Atk11], SCTP-CMT[IAS06] etc.)
- Due to their novelty, their impact is not widely understood
- **Focus of the thesis: The host networking stacks now have to deal with multiple paths**
- There is no host-based analysis of those new approaches

Motivations - Our goal

- Understand the implications of new multihoming protocols on the end-hosts.
  - Usability
  - Performance
  - System integration
- Improve the protocols accordingly

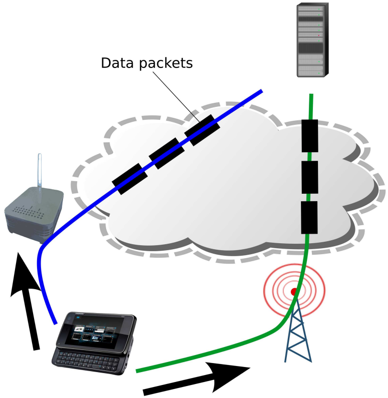
Agenda

End-host multipath control: high-level goals



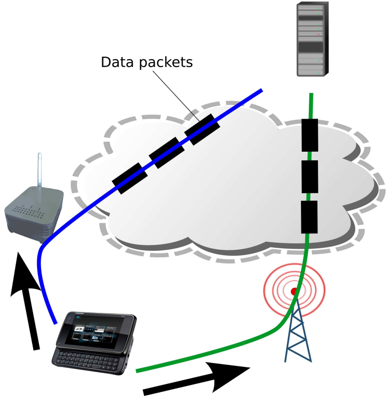
- Today: the user needs to restart the communication manually through the 3G interface
- Goal: we should preserve the communication even upon failure of a path
  - We cannot upgrade all applications

End-host multipath control: high-level goals



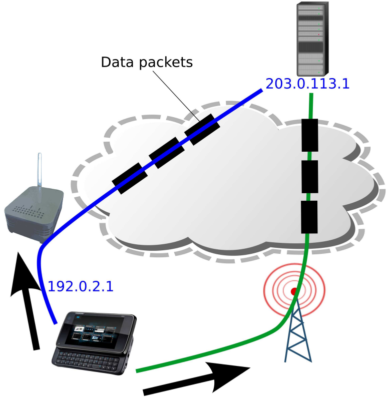
- Today: The user can use one interface at a time
- More ambitious goal: Achieve high resource utilisation
    - Use them all simultaneously

How to achieve session survival ?



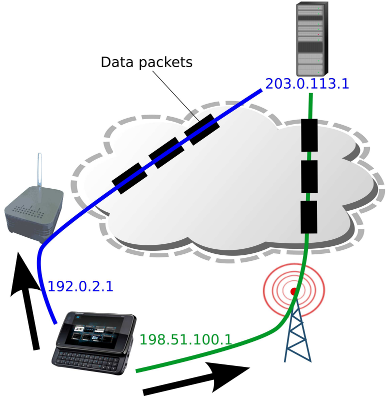
- **Locator/ID separation principle (with routeable ID)**
  - Identifier: first address used for the session
  - Locator: address used to forward the packet
  - Locator: the Identifier is also a locator
  - The applications only see the ID

How to achieve session survival ?



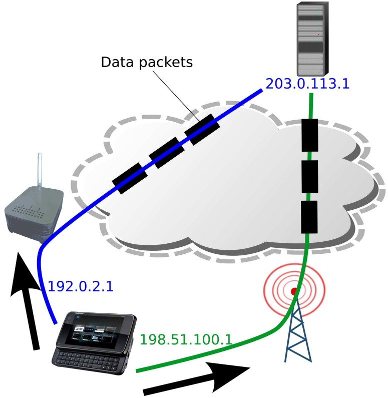
- **Locator/ID separation principle (with routeable ID)**
  - Identifier: first address used for the session
  - Locator: address used to forward the packet
  - Locator: the Identifier is also a locator
  - The applications only see the ID

How to achieve session survival ?



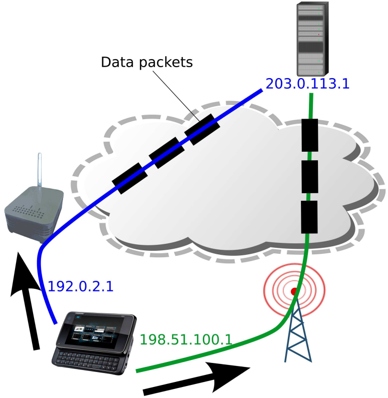
- **Locator/ID separation principle (with routeable ID)**
  - Identifier: first address used for the session
  - Locator: address used to forward the packet
  - Locator: the Identifier is also a locator
  - The applications only see the ID

How to achieve session survival ?



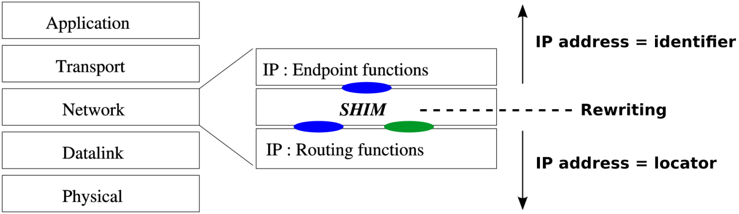
- **Locator/ID separation principle (with routeable ID)**
  - Identifier: first address used for the session
  - Locator: address used to forward the packet
  - Locator: the Identifier is also a locator
  - The applications only see the ID

How to achieve session survival ?



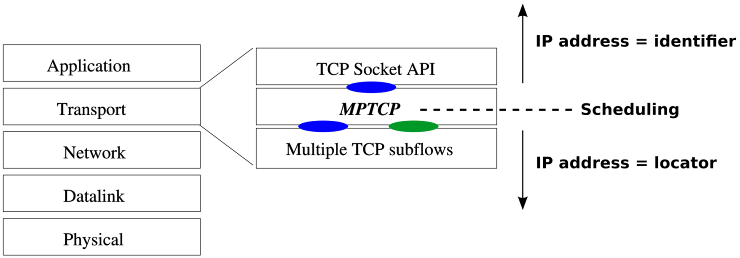
- **Locator/ID separation principle (with routeable ID)**
  - Identifier: first address used for the session
  - Locator: address used to forward the packet
  - Locator: the Identifier is also a locator
  - The applications only see the ID

Doing it in the network layer (Shim6)



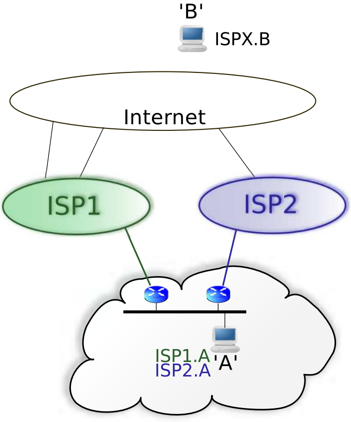
- Covers session survival across failures
- Cannot support packet level load balancing

Doing it in the transport layer (MPTCP)

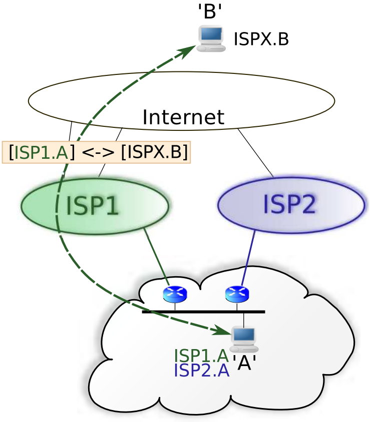


- Covers session survival across failures
- Does support packet level load balancing

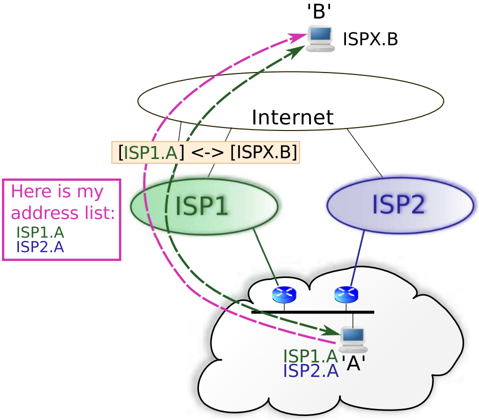
Shim6/MPTCP: common functions



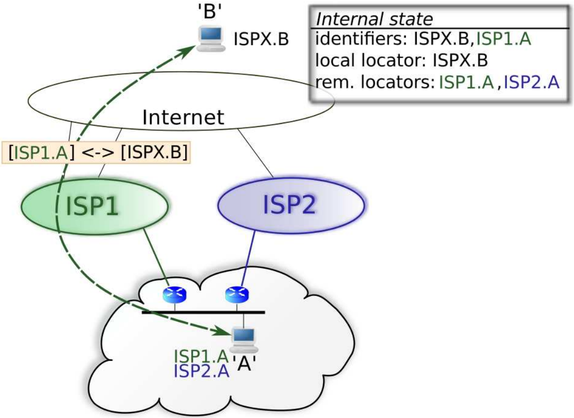
Shim6/MPTCP: common functions



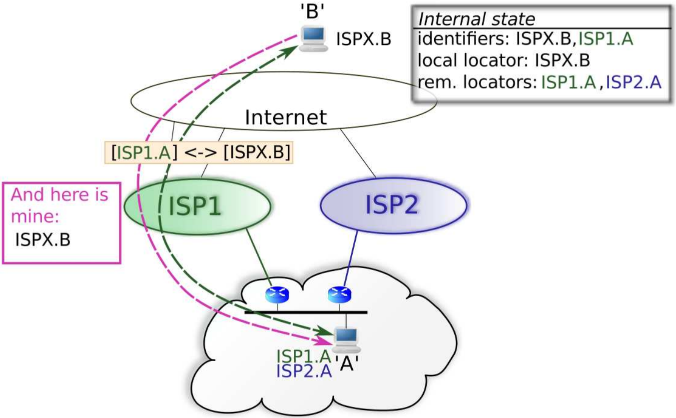
Shim6/MPTCP: common functions



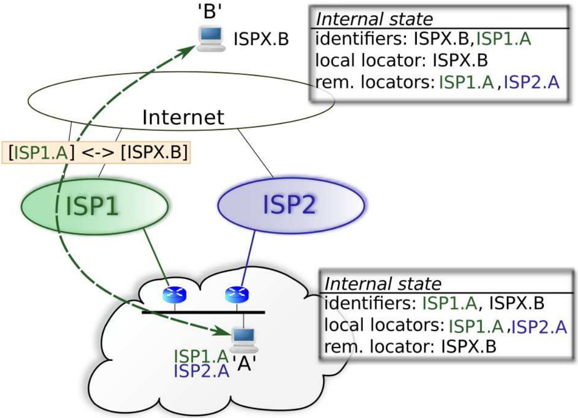
Shim6/MPTCP: common functions



Shim6/MPTCP: common functions

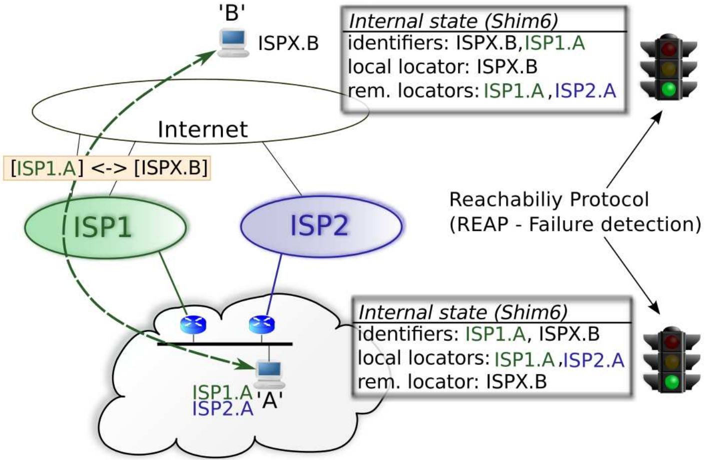


Shim6/MPTCP: common functions

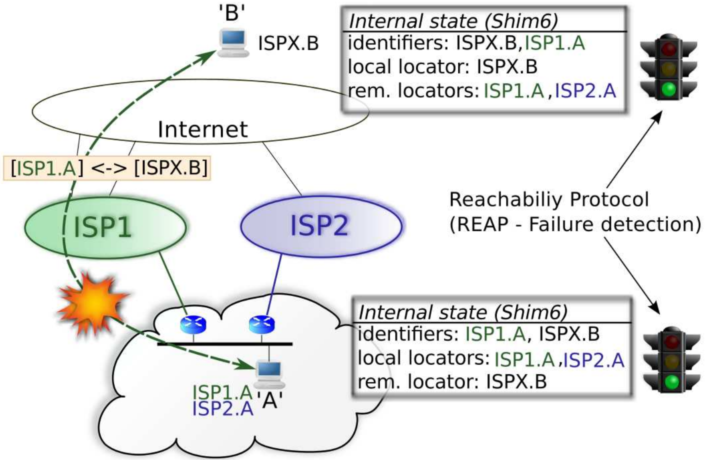


Shim6

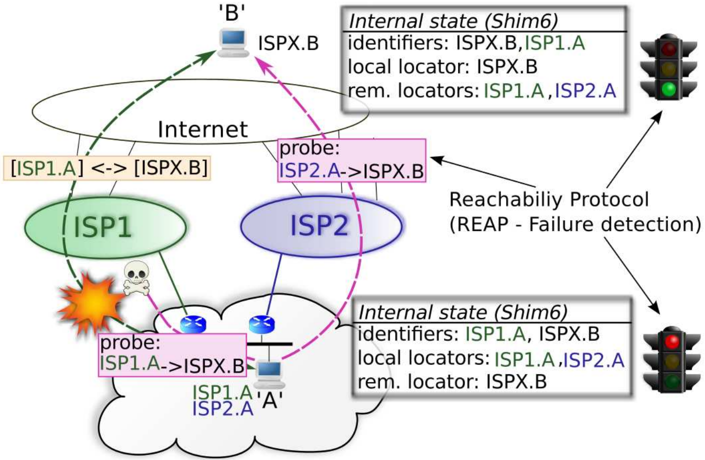
Shim6 operation



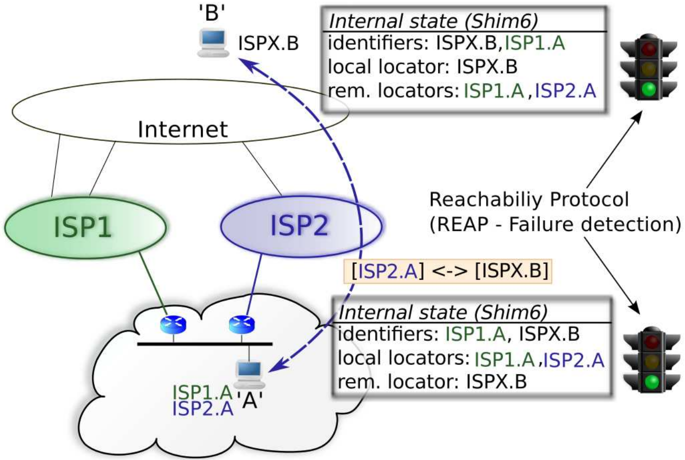
Shim6 operation



Shim6 operation



Shim6 operation



Contributions

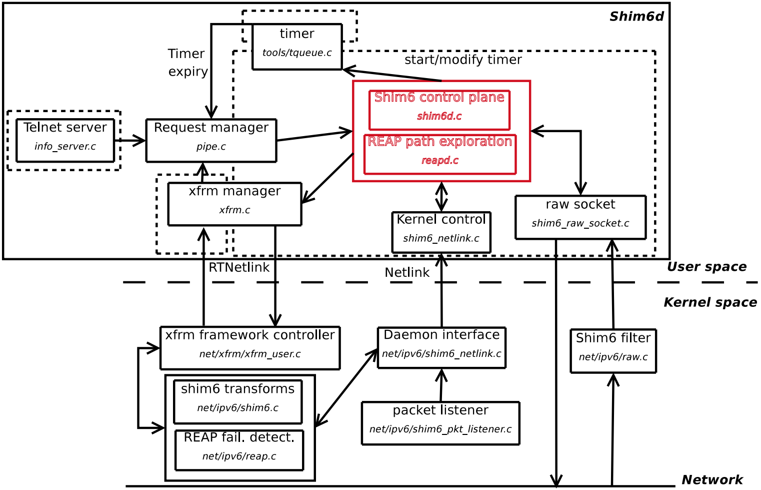
LinShim6 - A Shim6 implementation

- Many research prototypes done in high level frameworks
  - +: quick implementations
  - +: can test external protocol behaviour (e.g. middleboxes)
  - -: cannot test **internal** behaviour (system-level)
- Our choice: kernel-level implementation
  - +: can test external protocol behaviour (e.g.middleboxes)
  - +: can test internal behaviour
  - +: can be maintained to reach production-quality
  - +: reusable by others [MKS^{+}07, Mek07, RBKY08, DM08, RA08, DM09, RM10, AKP11]
  - -: slower development

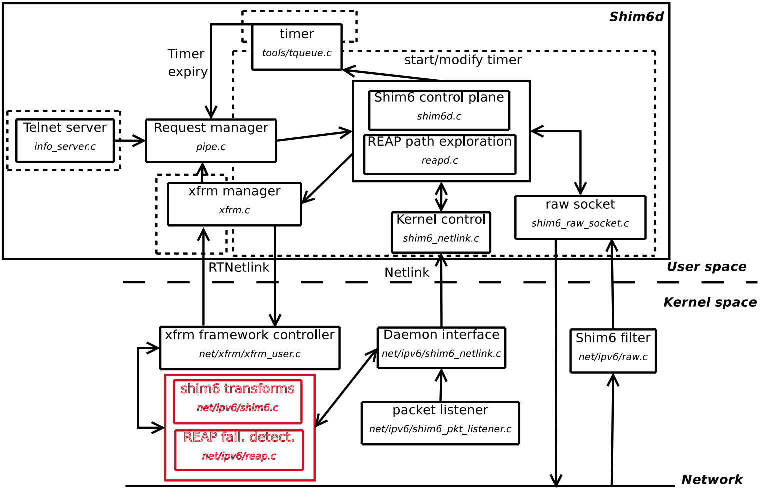
LinShim6 - design goals

- Maximum efficiency: user/kernel space separation
  - Kernel handles per-packet processing
  - userspace handles protocol control
  - minimizes context switches
  - maximizes amount of userspace code
- Interaction with other protocols, e.g. Mobile IP

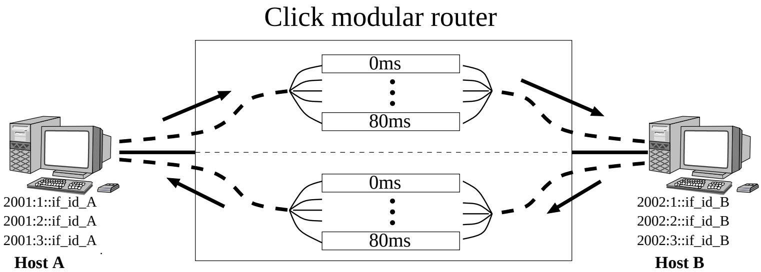
LinShim6 - implementation architecture



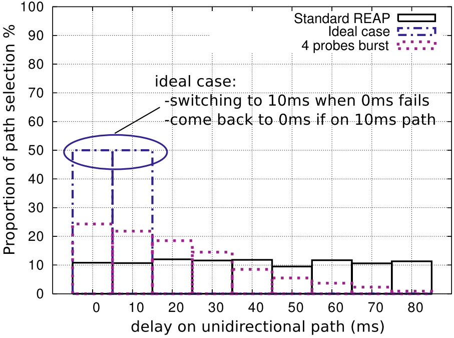
LinShim6 - implementation architecture



Can Shim6/REAP find a low-delay path ?



Yes, with burst probing



- +: higher probability to select better path
- -: higher probability to generate probe storm

Shim6: summary

- We provide an efficient, modular implementation of Shim6
  - $\sim$ 30000 lines in the daemon, $\sim$ 3500 in the kernel.
  - Widely used by other researchers [MKS+07, Mek07, RBKY08, DM08, RA08, DM09, RM10, AKP11]
- We evaluate and improve the recovery time of Shim6
- (in the thesis): we built a MipShim6 prototype, combining Mobile IPv6 with Shim6, with an architecture similar to [BGMA07]
  - Routing Optimization now secured with Shim6 CGAs
  - MIPv6 required to handle the double jump
  - Integrated implementation available

Multipath TCP

MPTCP - concepts

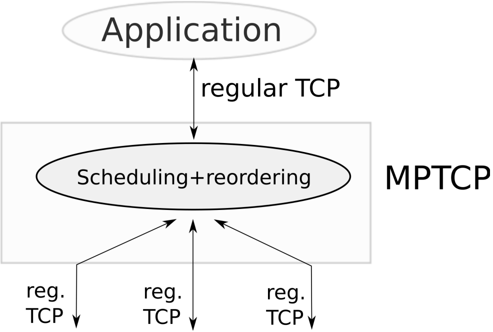


Figure: MPTCP is transparent to both the network and the applications

MPTCP - concepts

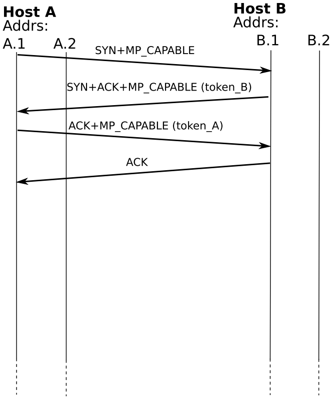


Figure: MPTCP subflow initiation

MPTCP - concepts

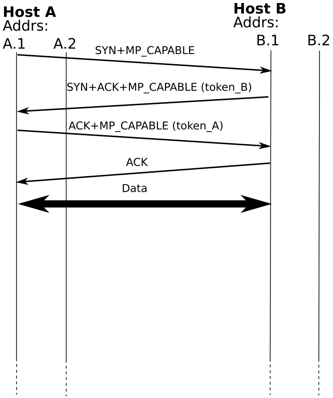


Figure: MPTCP subflow initiation

MPTCP - concepts

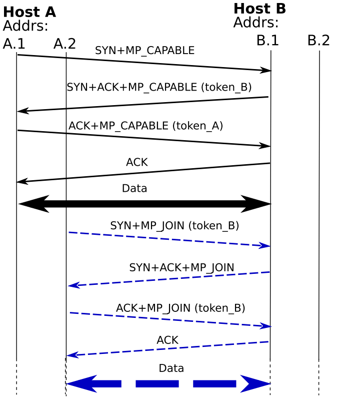


Figure: MPTCP subflow initiation

MPTCP - concepts

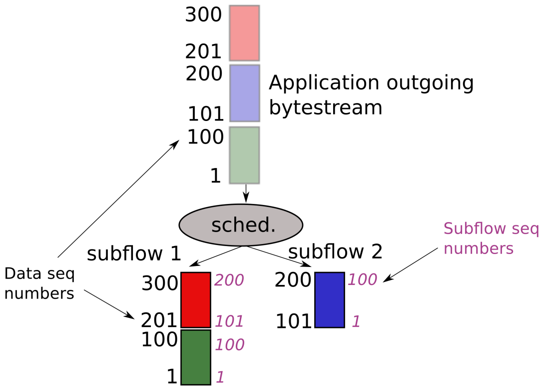


Figure: MPTCP Data Sequence Numbers (DSNs)

MPTCP - concepts

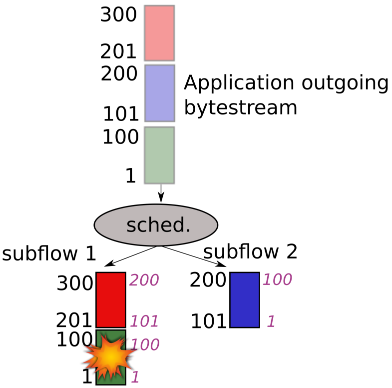


Figure: MPTCP retransmission

MPTCP - concepts

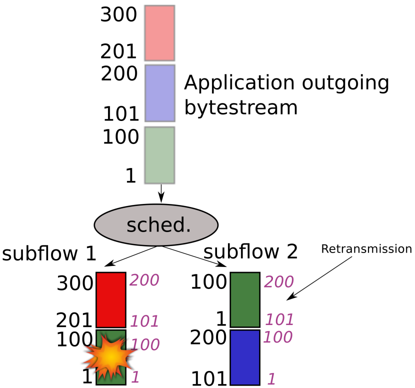
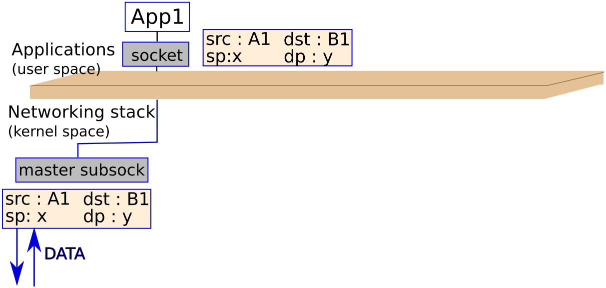


Figure: MPTCP retransmission

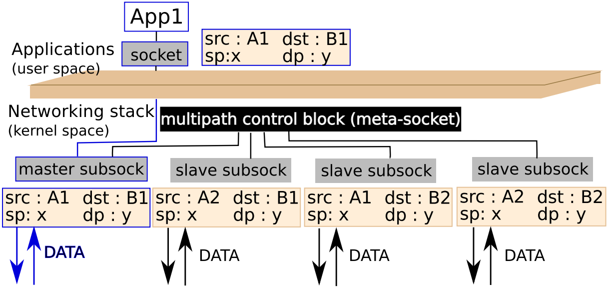
Contributions

Linux MPTCP



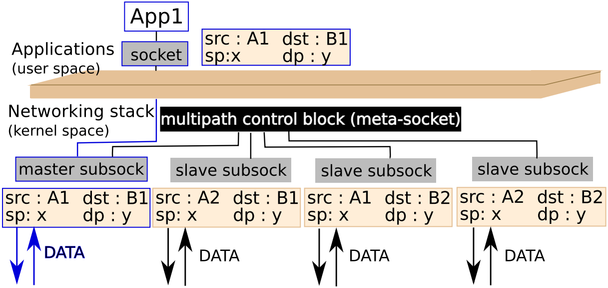
[BPB11a] S. Barré, C. Paasch, and O. Bonaventure. MultiPath TCP - Guidelines for implementers. draft-barre-mptcp-impl-00.txt, March 2011.

Linux MPTCP



[BPB11a] S. Barré, C. Paasch, and O. Bonaventure. MultiPath TCP - Guidelines for implementers. draft-barre-mptcp-impl-00.txt, March 2011.

Linux MPTCP



- ∼ 10000 lines in the Linux kernel
- Already used by other researchers: [SBS+10, NZNP11]

[BPB11a] S. Barré, C. Paasch, and O. Bonaventure. MultiPath TCP - Guidelines for implementers. draft-barre-mptcp-impl-00.txt, March 2011.

Linux architecture - Keeping Path Management appart

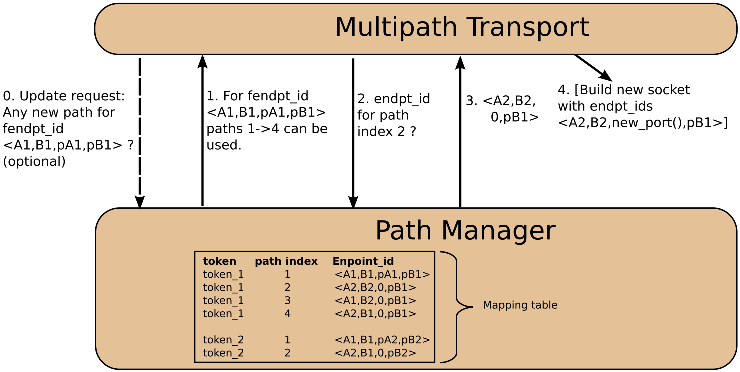
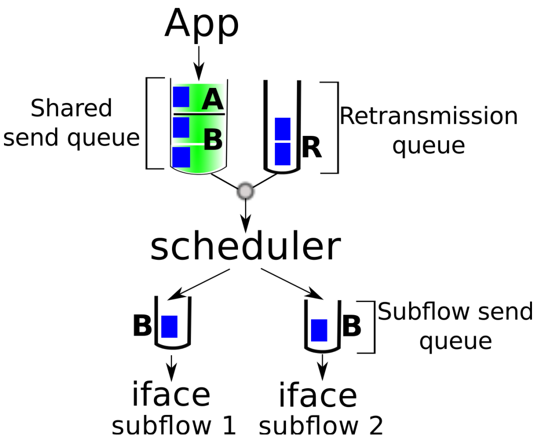
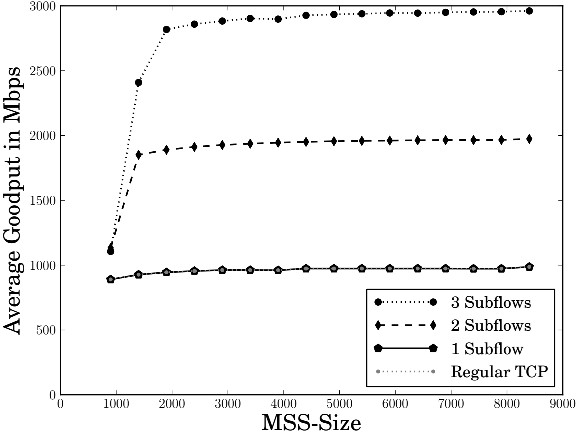


Figure: Functional separation of MPTCP in the transport layer

Linux architecture - Scheduling/retransmitting data



MPTCP performance - MSS impact



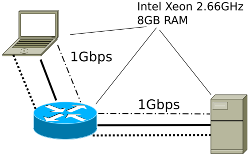
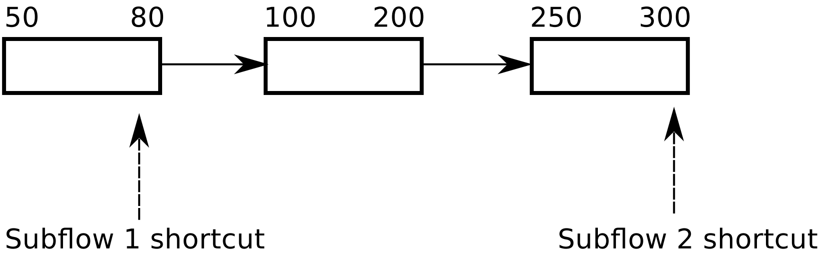


Figure: testbed

MPTCP performance - CPU consumption

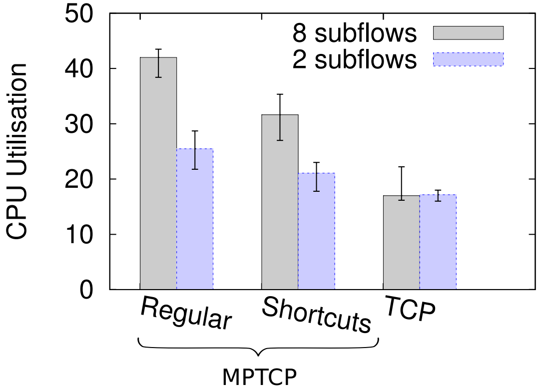
- **The MPTCP receiver requires special optimisations compared to regular TCP**
  - Regular TCP: segments arrive mostly in sequence, reordering only happens upon a loss event
  - MPTCP: subflow segments arrive in sequence, but MPTCP-level data does not arrive in sequence

MPTCP performance - CPU consumption



- Shortcuts were successful in 80% of the out-of-order packet receptions in our testbed.
  (2 hosts, Intel Xeon 2.66Ghz, 8GB RAM, 2 links with capacity 1Gbps)

MPTCP performance - CPU consumption

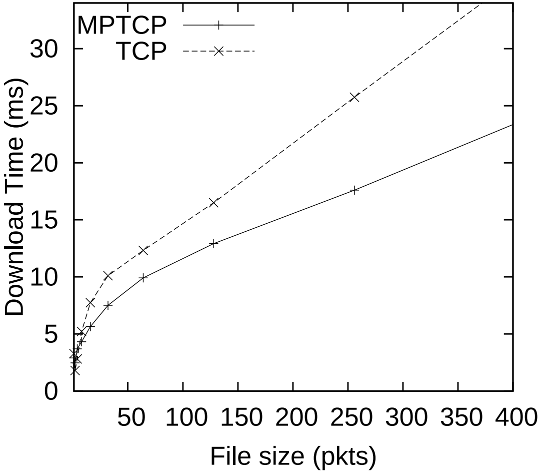


MPTCP performance - When to start MPTCP ?

- Datacenter communications may be short (a few packets)
- MPTCP negotiation has a cost (one handshake per subflow, second subflow established *after* the first one)
- We run TCP and MPTCP for short flows, testing download times for a set of transfer sizes

[RBP+11] C. Raiciu, S. Barré, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley. Improving data center performance and robustness with multipath TCP. *SIGCOMM*, Toronto, 2011.

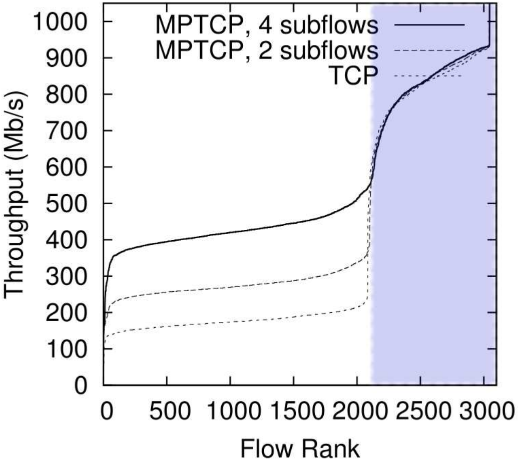
MPTCP performance - When to start MPTCP ?



MPTCP in action in a real datacenter

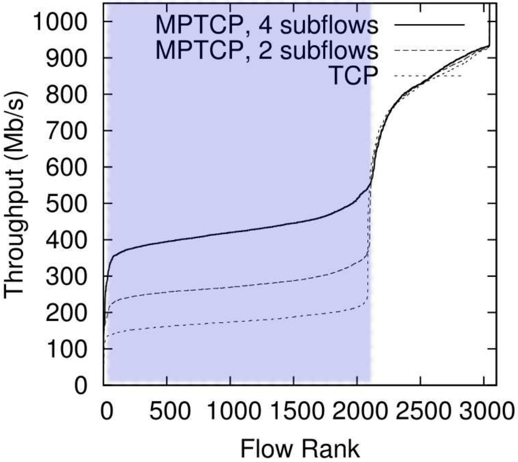
- Linux MPTCP has been run in the Amazon EC2 testbed
- 12 hours, sequentially measuring bandwidth between 40 nodes
- Linux MPTCP configured to use random ports with constant addresses
    - goal: take benefit from load-balancing

MPTCP in action in a real datacenter



- 1/3 of paths share the same switch or physical machine
- 2/3 of paths are balanced according to traceroute.

MPTCP in action in a real datacenter



- 1/3 of paths share the same switch or physical machine
- **2/3 of paths are balanced according to traceroute.**

Conclusion

Shim6 vs MPTCP: Shim6 strengths

- Security:
  - protects against time-shifting attack
  - can use the longer IPv6 addresses to encode security information
- State:
  - One state, one negotiation per address pair
  - MPTCP has one state, one negotiation per TCP socket
- Supports all transports protocols

Shim6 vs MPTCP: MPTCP strengths

- Major strength: supports simultaneous use of paths
- Consequences:
  - Better use of resources
  - Faster reaction to failures (timescale of the TCP timeout)
- Supports both IPv4 and IPv6

Putting them together ?

- How to do it:
  - MPTCP dictates its path choices to Shim6 (per segment)
  - REAP disabled (Failure detection by MPTCP)
  - MPTCP security ignored (Shim6 stronger security is used)
- If 10 connections established between same hosts, only one address exchange happens
- Proof of concept available (was Linux MPTCP 0.1)
- Not useful now. . . But the benefits will appear when IPv6 will be widely deployed

Putting them together ?

- How to do it:
  - MPTCP dictates its path choices to Shim6 (per segment)
  - REAP disabled (Failure detection by MPTCP)
  - MPTCP security ignored (Shim6 stronger security is used)
- If 10 connections established between same hosts, only one address exchange happens
- Proof of concept available (was Linux MPTCP 0.1)
- Not useful now... But the benefits will appear when IPv6 will be widely deployed

Putting them together ?

- How to do it:
  - MPTCP dictates its path choices to Shim6 (per segment)
  - REAP disabled (Failure detection by MPTCP)
  - MPTCP security ignored (Shim6 stronger security is used)
- If 10 connections established between same hosts, only one address exchange happens
- Proof of concept available (was Linux MPTCP 0.1)
- Not useful now. . . But the benefits will appear when IPv6 will be widely deployed

Putting them together ?

- How to do it:
    - MPTCP dictates its path choices to Shim6 (per segment)
    - REAP disabled (Failure detection by MPTCP)
    - MPTCP security ignored (Shim6 stronger security is used)
- If 10 connections established between same hosts, only one address exchange happens
- Proof of concept available (was Linux MPTCP 0.1)
- Not useful now... But the benefits will appear when IPv6 will be widely deployed

Conclusion

- We performed an in-depth study of the Shim6 and MPTCP protocols
- We showed how they can be combined to take advantage of strengths in both protocols
- We showed that MPTCP in particular is not only an ambitious protocol change, but also involves a number of Operating System challenges, some of them being solved today

Conclusion

- Our implementations of LinShim6 and MPTCP are both the most complete and efficient prototypes available to researchers.
- Our work lead to several improvements that are now reflected in the protocol specifications

External publications citing LinShim6

- M. Mekking. Formalization and verification of the shim6 protocol. Master's thesis, Radboud University - NLnet Labs, 2007
- K. Mitsuya, R. Kuntz, S. Sugimoto, R. Wakikawa, and J. Murai. A policy management framework for flow distribution on multihomed end nodes. SIGCOMM MobiArch Workshop, 2007
- J. Ronan, S. Balasubramaniam, A. K. Kiani, and W. Yao. On the use of SHIM6 for mobility support in IMS networks. TRIDENTCOM, ICST, 2008
- A. Dhraief and N. Montavont. Toward Mobility and Multihoming Unification- The SHIM6 Protocol: A Case Study. WCNC 2008

External publications citing LinShim6

- M.S. Rahman and M. Atiquzzaman. SEMO6 - a multihoming-based seamless mobility management framework. MILCOM 2008
- A. Dhraief and N. Montavont. Rehoming decision algorithm: design and empirical evaluation. International Conference on Computational Science and Engineering, IEEE, 2009
- J. Ronan and J. McLaughlin. An empirical evaluation of a Shim6 implementation. ICST Conference, 2010
- A. Achour, B. Kervella, and G. Pujolle. Shim6-based mobility management for multi-homed terminals in heterogeneous environment. WOCN 2011

External publications citing MPTCP

- M. Scharf, T.R. Banniza, P. Schefczik, A. Singh, and A. Timm-Giel. Evaluation and prototyping of multipath protocol mechanisms. 2010
- S.C. Nguyen, X. Zhang, T.M.T. Nguyen, and G. Pujolle. Evaluation of throughput optimization and load sharing of multipath tcp in heterogeneous networks. WOCN 2011

The End