MultiPath TCP - Guidelines for implementers draft-barre-mptcp-impl-00.txt

Sébastien Barré
Christoph Paasch
Olivier Bonaventure

March 31st, 2011

IETF 80 - Prague

Intention of this draft

- Gather in one document the experience gained from implementing the protocol.
    - At the moment, one Linux implementation.
    - Hope to include others (BSD, MAC, Windows ?) when they appear.
- Put together configuration guidelines (guidelines for *admins/users*)
- When appropriate, include results from interoperability tests.

About the Linux MPTCP implementation

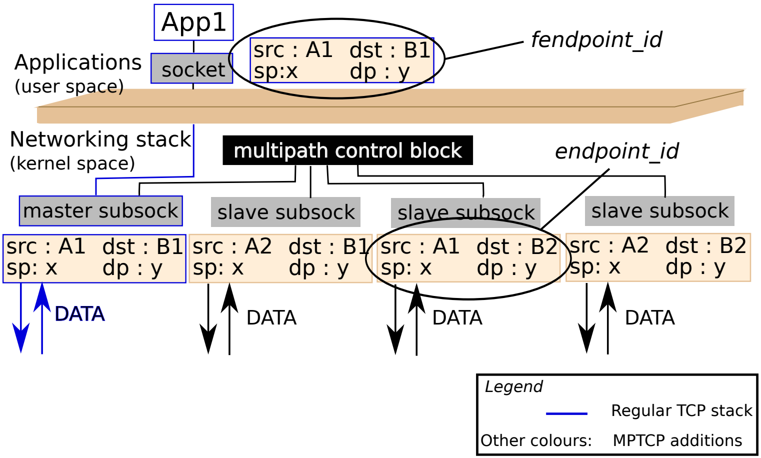
- IETF release (0.6) available since Monday !
- All information here: `http://inl.info.ucl.ac.be/mptcp`
- Try it with a live-CD: either
  - Get one from us (Olivier distributes).
  - download it locally (regular download or torrent): `https://noc.meeting.ietf.org/wiki`
  - or after the IETF: `http://inl.info.ucl.ac.be/mptcp`

Try it on your smartphone !

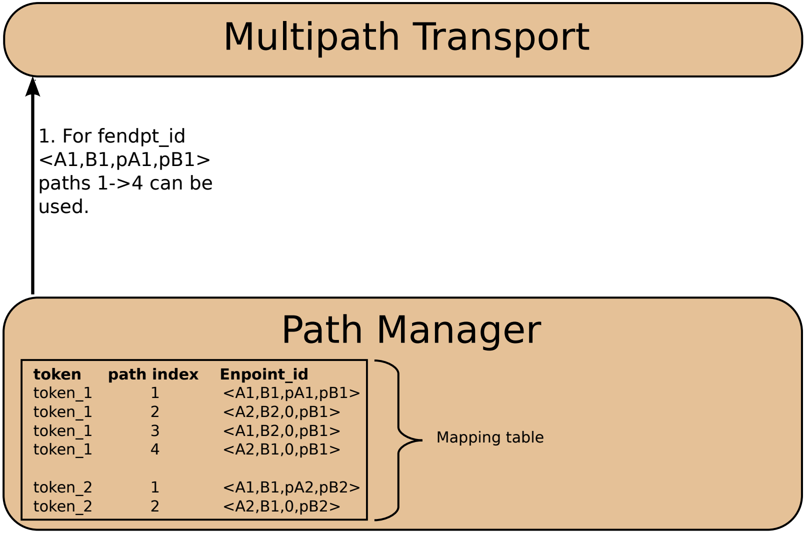
- Nokia N900. Runs a Linux 2.6.28, MPTCP 0.6, demo on request.

- Android on Nexus One phone. See `http://www.nishida.org/nexusone_mptcp/` (Thanks Yoshifumi !)

Architecture

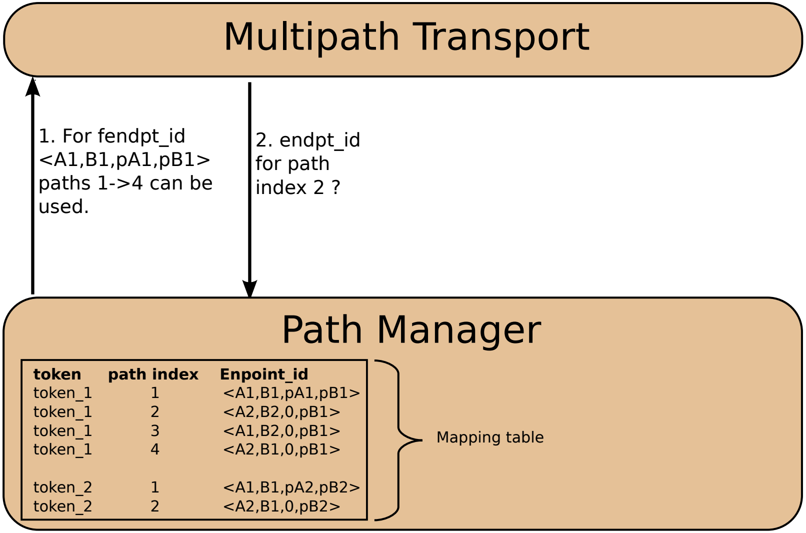


Path management as a separate entity



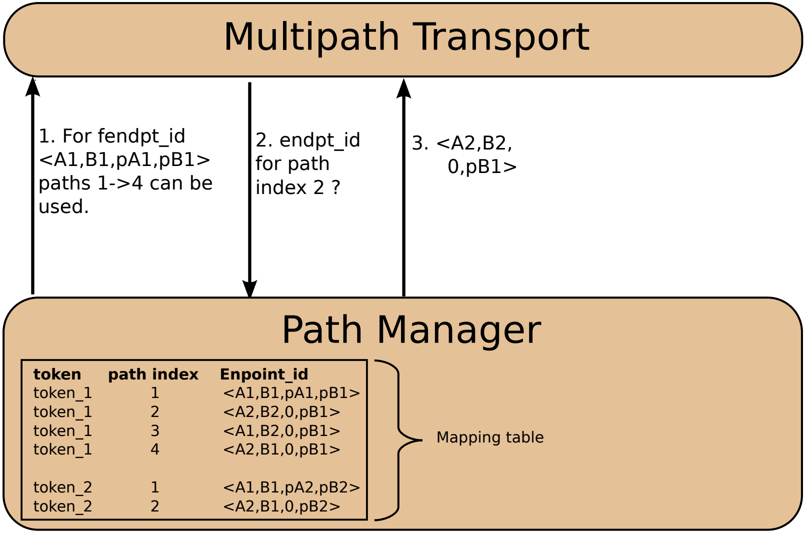
token	path index	Enpoint_id
token_1	1	<A1,B1,pA1,pB1>
token_1	2	<A2,B2,0,pB1>
token_1	3	<A1,B2,0,pB1>
token_1	4	<A2,B1,0,pB1>
token_2	1	<A1,B1,pA2,pB2>
token_2	2	<A2,B1,0,pB2>

Path management as a separate entity


Multipath Transport
1. For fendpt_id <A1,B1,pA1,pB1> paths 1->4 can be used.
2. endpt_id for path index 2 ?
Path Manager
Mapping table

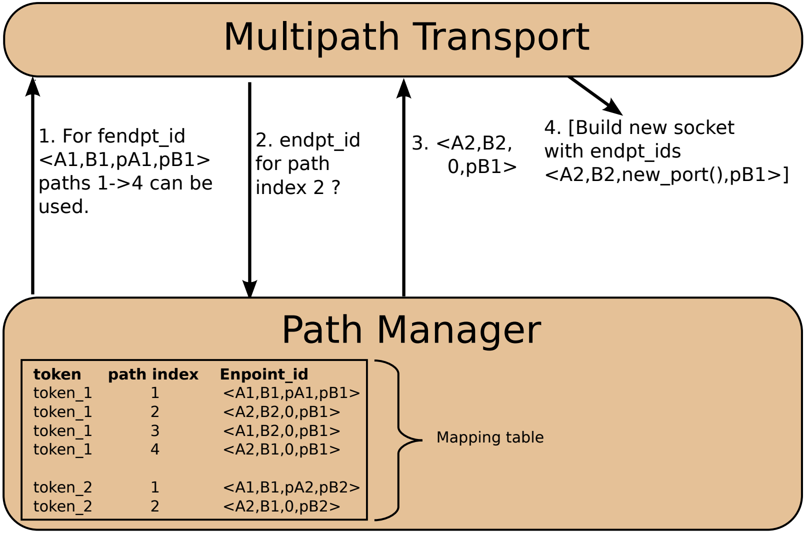
token	path index	Enpoint_id
token_1	1	<A1,B1,pA1,pB1>
token_1	2	<A2,B2,0,pB1>
token_1	3	<A1,B2,0,pB1>
token_1	4	<A2,B1,0,pB1>
token_2	1	<A1,B1,pA2,pB2>
token_2	2	<A2,B1,0,pB2>

Path management as a separate entity


Multipath Transport
1. For fendpt_id <A1,B1,pA1,pB1> paths 1->4 can be used.
2. endpt_id for path index 2 ?
3. <A2,B2, 0,pB1>
Path Manager
Mapping table

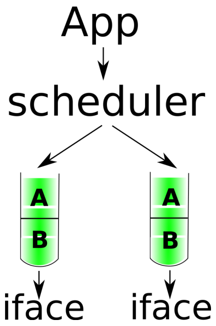
token	**path index**	**Enpoint_id**
token_1	1	<A1,B1,pA1,pB1>
token_1	2	<A2,B2,0,pB1>
token_1	3	<A1,B2,0,pB1>
token_1	4	<A2,B1,0,pB1>
token_2	1	<A1,B1,pA2,pB2>
token_2	2	<A2,B1,0,pB2>

Path management as a separate entity


Multipath Transport
1. For fendpt_id <A1,B1,pA1,pB1> paths 1->4 can be used.
2. endpt_id for path index 2 ?
3. <A2,B2, 0,pB1>
4. [Build new socket with endpt_ids <A2,B2,new_port(),pB1>]
Path Manager
Mapping table

token	path index	Enpoint_id
token_1	1	<A1,B1,pA1,pB1>
token_1	2	<A2,B2,0,pB1>
token_1	3	<A1,B2,0,pB1>
token_1	4	<A2,B1,0,pB1>
token_2	1	<A1,B1,pA2,pB2>
token_2	2	<A2,B1,0,pB2>

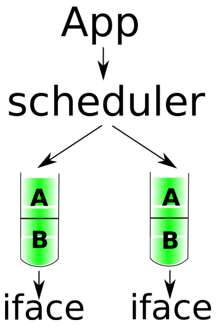
Send queue management (IETF79)



- **A**: Send buffer
- **B**: Data sent but not yet acknowledged

- Each subflow has its own send buffer
- Scheduling may be done long before a packet is sent
- . . . too sensitive to path variations

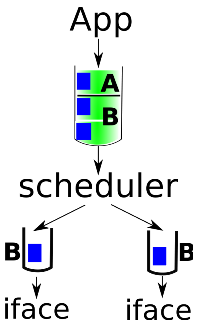
Send queue management (IETF79)



- **A**: Send buffer
- **B**: Data sent but not yet acknowledged

- Each subflow has its own send buffer
- Scheduling may be done long before a packet is sent
- . . . too sensitive to path variations

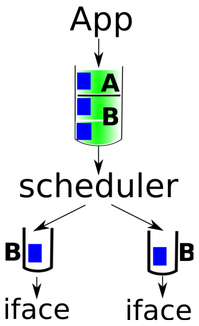
Send queue management (IETF80)



- **A**: Send buffer
- **B**: Data sent but not yet acknowledged

- The send buffer is shared by all subflows
- Allows more accurate scheduling. . .
- . . . but requires all subflows to use the same MSS.

Send queue management (IETF80)



- **A**: Send buffer
- **B**: Data sent but not yet acknowledged

- The send buffer is shared by all subflows
- Allows more accurate scheduling. . .
- . . . but requires all subflows to use the same MSS.

Receive queue management

- No subflow-specific receive queue
- When in-order at subflow level, data is immediately handed to Multipath Transport.
- Allows quick sending of data acks.

Others

- Multiport capability
- Improved connection termination
- Coupled Congestion Control (impl by Christoph Paasch)
- /proc instrumentation:
  - configure how many paths to use with multiport
  - configure the shared MSS
  - MPTCP switch (globally disable/enable)
  - Report on internal state (by Andreas Ripke)

TODO list

- Implement security mechanisms
- optimize (SMP capability, fast path, . . .)
- Allow more flexible MSS handling (path-specific MSS, TSO)
- Implement and evaluate heuristics (multipath trigger, number of subflows, . . .)
- contributions welcome !
  (see `http://inl.info.ucl.ac.be/mptcp`)