Performance of multipath HIP vs MPTCP

Dmitriy Kuptsov, Ilya Nikolaevskiy, Tatiana Polishchuk, Andrei Gurtov
HIIT, Aalto University
CWC, University of Oulu
Finland

Background and Motivation

- Multiple number of network interfaces per device
- Bandwidth aggregation and more reliable communication for multihomed hosts
- Several transport layer solutions: MP-TCP, MP-SCTP, MRTP
- HIP provides abstraction between transport and network layers:
  - “Optimal” place to implement generic multipath routing

Advantages of multipath HIP

- Multipath functionality for all transport layer protocols
- Support for legacy applications
- Middle box traversal
- Mobility support
- Security is available out-of-the-box

Multipath HIP implementation

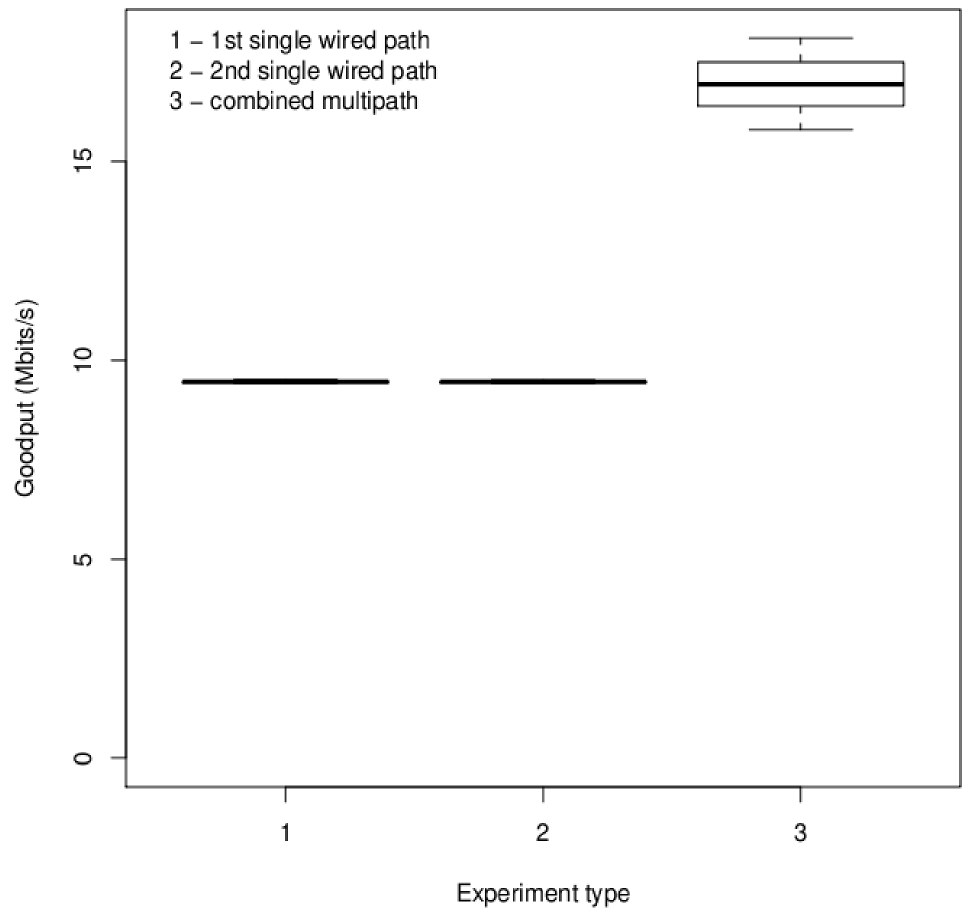
- Per path SAs are established using HIP multihoming extension
- Forwarding
  - Fastest path first forwarding rule
  - Buffer support to minimize reordering
  - Periodic path probing and statistics aggregation
- Prototype implementation in HIPL

Evaluation. Setup 1

- Small testbed:
  - Host with 2 wireless interfaces
  - Host with 2 wired interfaces
  - Both connected to a server with a single interface
- TCP bulk transfer
- MPTCP Linux implementation as comparison point (MPTCP version 0.6, kernel version 2.6.36)

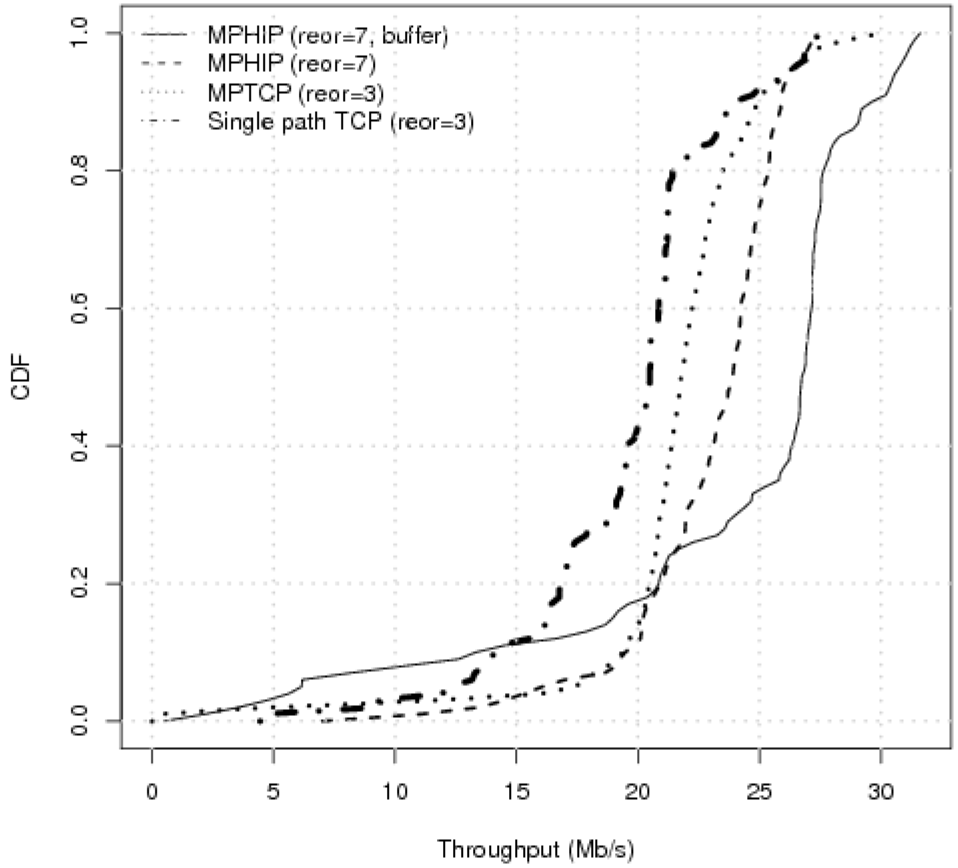
Wired experiment (mHIP w/o buffer)

- We achieved almost ~80% of the aggregated bandwidth: single wired path - 9.3 Mb/s, 2 paths - 16.5 Mb/s
- Almost no variance in throughput



Wireless experiment

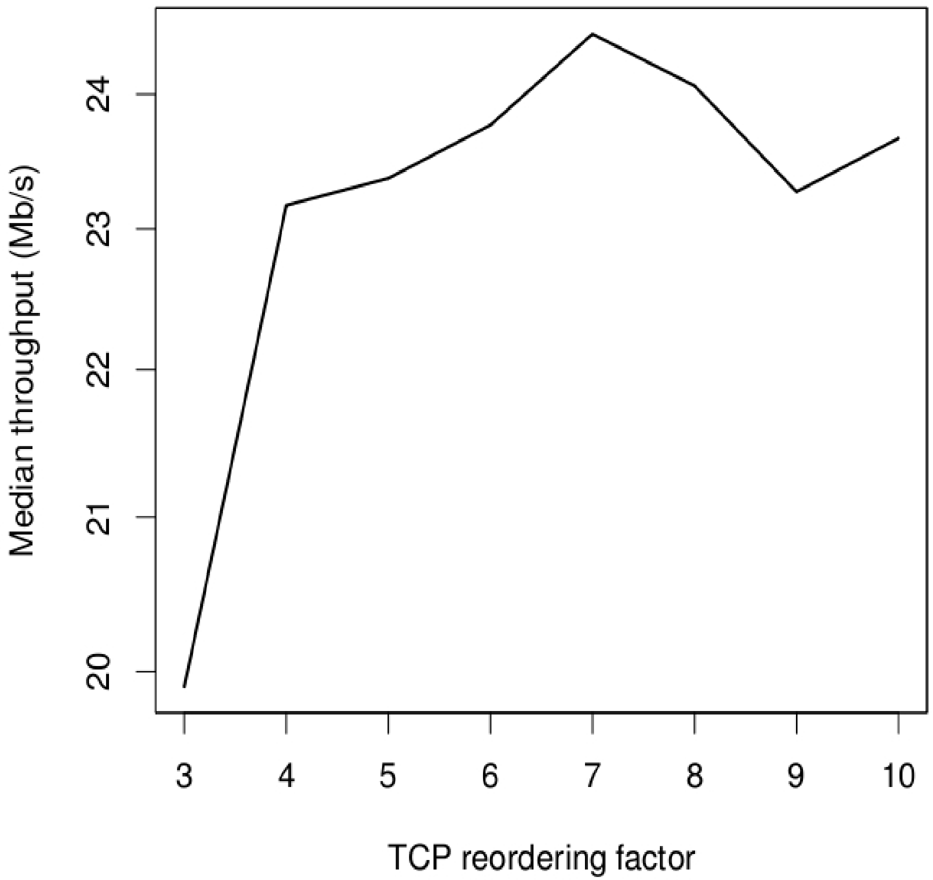
- mHIP (w/o buffer): Aggregated bandwidth increases steadily by >20% with tweaked TCP dupthresh parameter
- mHIP (w buffer): median throughput increased by > 35%
- MPTCP: Merely shows a marginal improvement (~10%)



Both MPTCP and mHIP use dupthresh that gives best results

MHIP w/o buffer. TCP dupthresh tweaks

- The highest median throughput (>24Mb/s) is achieved with TCP reordering factor set to 7
- Tweaking dupthresh is useful depending on network conditions



Evaluation. Setup 2. Synthetic tests

- Goal: Controlled experiments
  - Emulate specific loss, delay and jitter
- Emulate wide spectrum of network conditions: from mild to harsh
- Observe the trends for mHIP and MPTCP

Results

delay(ms)/jitter(ms)/loss(%)		Goodput (Mbits/s) median/std.dev.			
1st path	2nd	TCP (1st)	TCP (2nd)	mHIP	MPTCP
0/0/0.1	0/0/0.1	8.7	8.7	13.1/0.25	13.7/0.6
5/3/0	5/0/0	8.1	8.2	11.4/0.65	11/0.01
5/3/0	5/3/0	8.1	8.1	10.8/0.5	10.2/0.01
5/2/0.1	5/2/0.1	7.45	7.45	10.4/0.7	8.6/0.15
20/10/0	20/10/0	7	7	9.2/0.64	4.5/0.01
100/50/0	100/0/0	2.2	5.2	4.6/0.35	1.2/0
100/50/0	100/50/0	2.3	2.3	2.15/0.02	1/0.01
100/30/0	5/2/0	4.9	8.1	5.4/1.2	1.8/0.01
50/30/5	0/0/0.1	0.55	8.7	7.55/0.08	2.7/0.04
100/0/0	5/0/0	5.2	8.2	7.9/0.7	2/0.01
100/5/0	5/5/0	4.45	7.9	5.1/1.5	1.9/0.01
100/5/0.1	5/5/0.1	3.1	7.3	3.9/0.6	1.9/0.12
0/30/5	50/30/5	0.55	0.55	0.45/0	0.65/0.07
100/5/0.1	50/30/5	3	0.55	1.1/0.02	1.5/0.82

- Light gray: both mHIP and MPTCP perform well in almost ideal networks.
- Gray: mHIP performs relatively well in networks with heterogeneous links with high delays and jitter. MPTCP is not even close to single path TCP
- Dark gray: under sever loss neither mHIP nor MPTCP can perform well

Conclusions

- Multipath HIP:
  - Works perfectly well in networks that have low jitter
  - In networks with high jitter, multipath HIP achieves > 20% gain when TCP reordering factor is tuned properly and no additional buffer is used
  - In networks with high jitter multipath HIP achieves >35% gain when buffer is used and no additional tweaks to TCP parameters is needed
  - Poorly performs under sever loss

Conclusions

- MPTCP (or rather its particular implementation):
  - Works perfectly well in almost ideal networks (small delay and almost no jitter)
  - Performs awfully when links are heterogeneous, e.g., have different delays
  - Does not show good results when paths have high delays and jitter
  - Poorly performs under sever loss
- TCP reordering factor can be tuned in flight to adapt TCP to channel conditions

Future work

- We still have to experiment with a network which has high jitter and high loss characteristics
- Optimal parameters for buffer (timeout, size, etc), and TCP reordering factor (depending on channel jitter and loss) are still under question
- Some minor bug fixes in HIPL code