A Proactive Approach to Avoid Performance Degradation of MPTCP

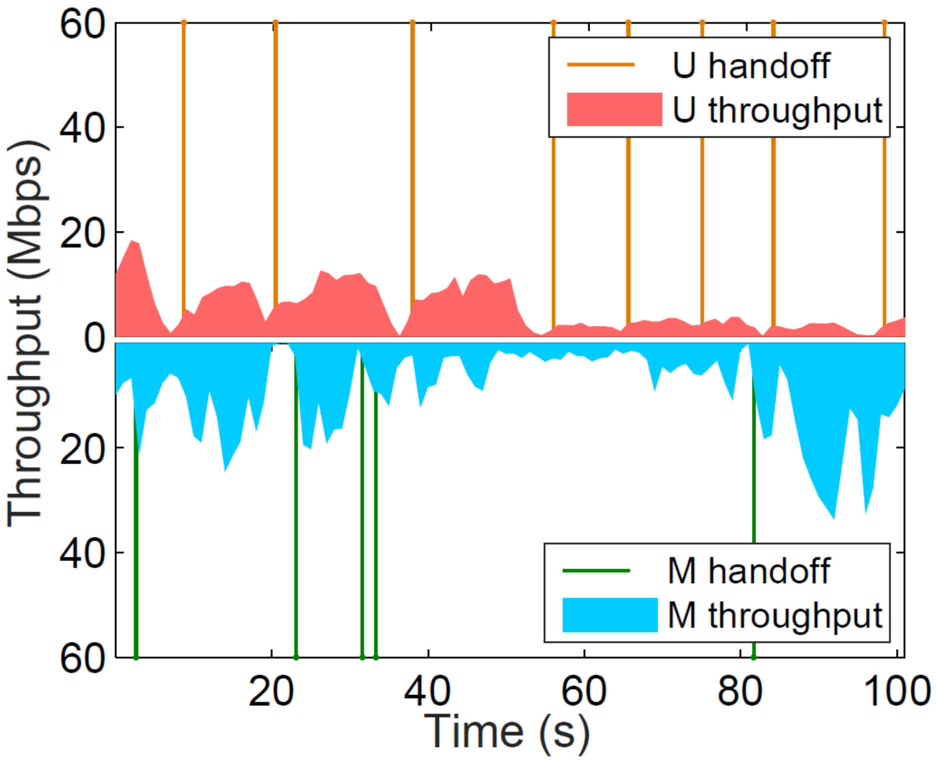
Draft-zuo-mptcp-degradation-00.txt

Authors: F. Wang, Jing Zuo , Z. Cao, K. Zheng, Huawei

MPTCP on high-speed rails

- Two LTE (4G) on high-speed rails
  - ✓One LTE in smart phone
  - ✓ Another LTE is connected through Wi-Fi hotspots
  - ✓ISPs: China Unicom (U), China Mobile (M), China Telecom (T)

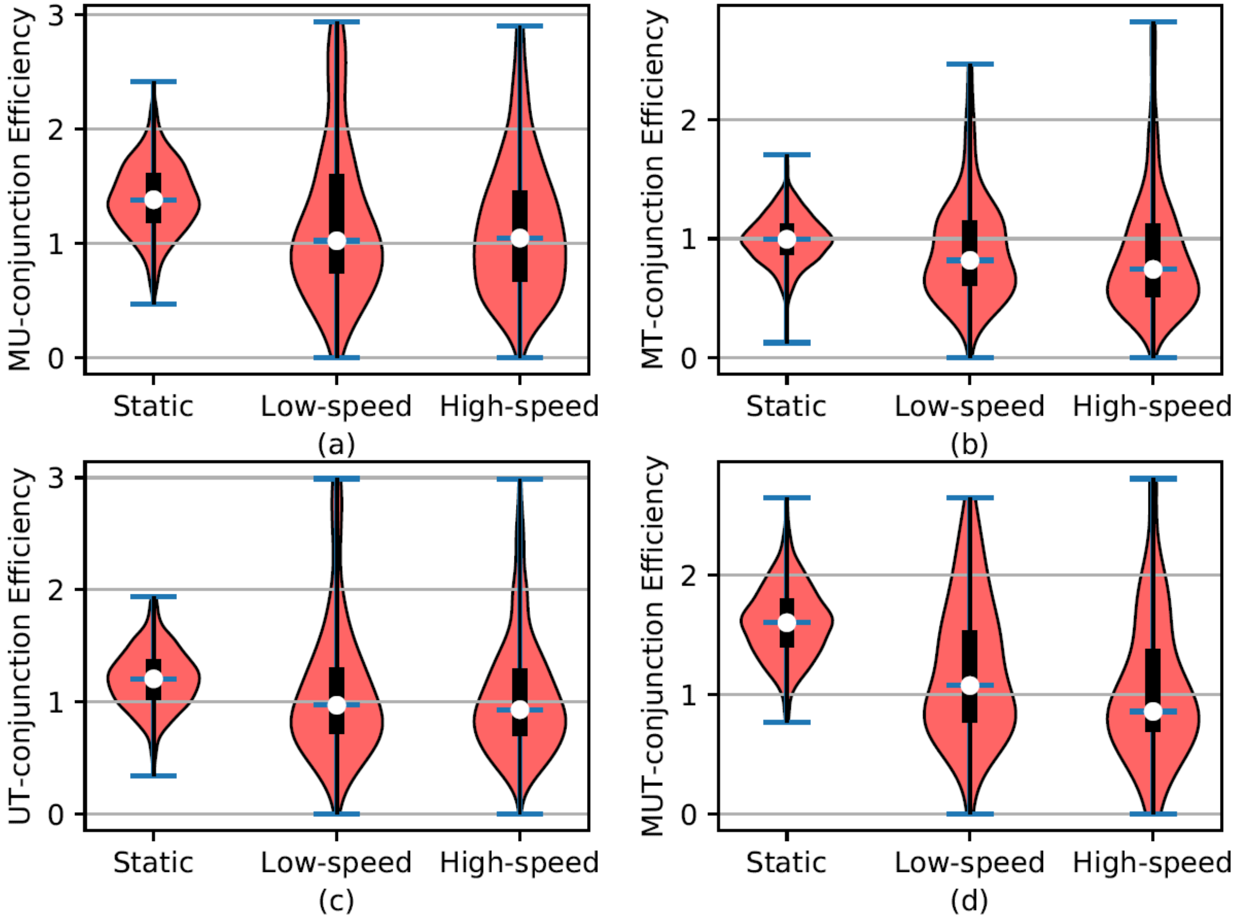
- Attributes of paths
  - ✓Frequent handoff
    - Variant RTT
    - Severe random packet loss
  - ✓Throughput variance

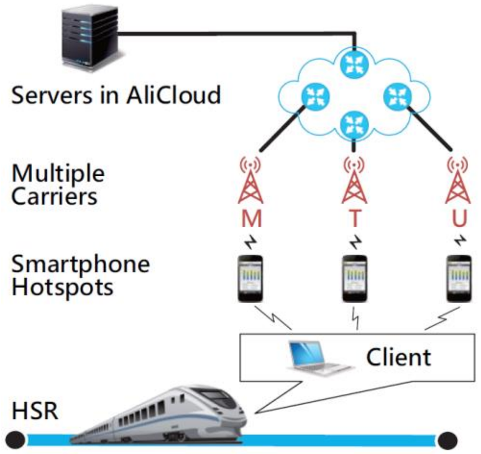


MPTCP on High-speed rails

Aggregation Efficiency: $U_c = \frac{T_{\mathrm{mp}}}{T'}$

- T_{mp}: Aggregate throughput
- T': the highest throughput of all paths





- Scenarios
  - High Speed: 150~310 km/h
  - Low Speed: < 150 km/h
  - Static: park at stations
- Experiments
  - Distance: 66,344 km
  - Data: 3.31 TB
  - Period: 6 months
- Results
  - Performance degradation ($U_c < 1$) in many cases
  - Aggregation is not always efficient

Existing work: Opportunistic retransmission and penalization (OR&P)

What is it?

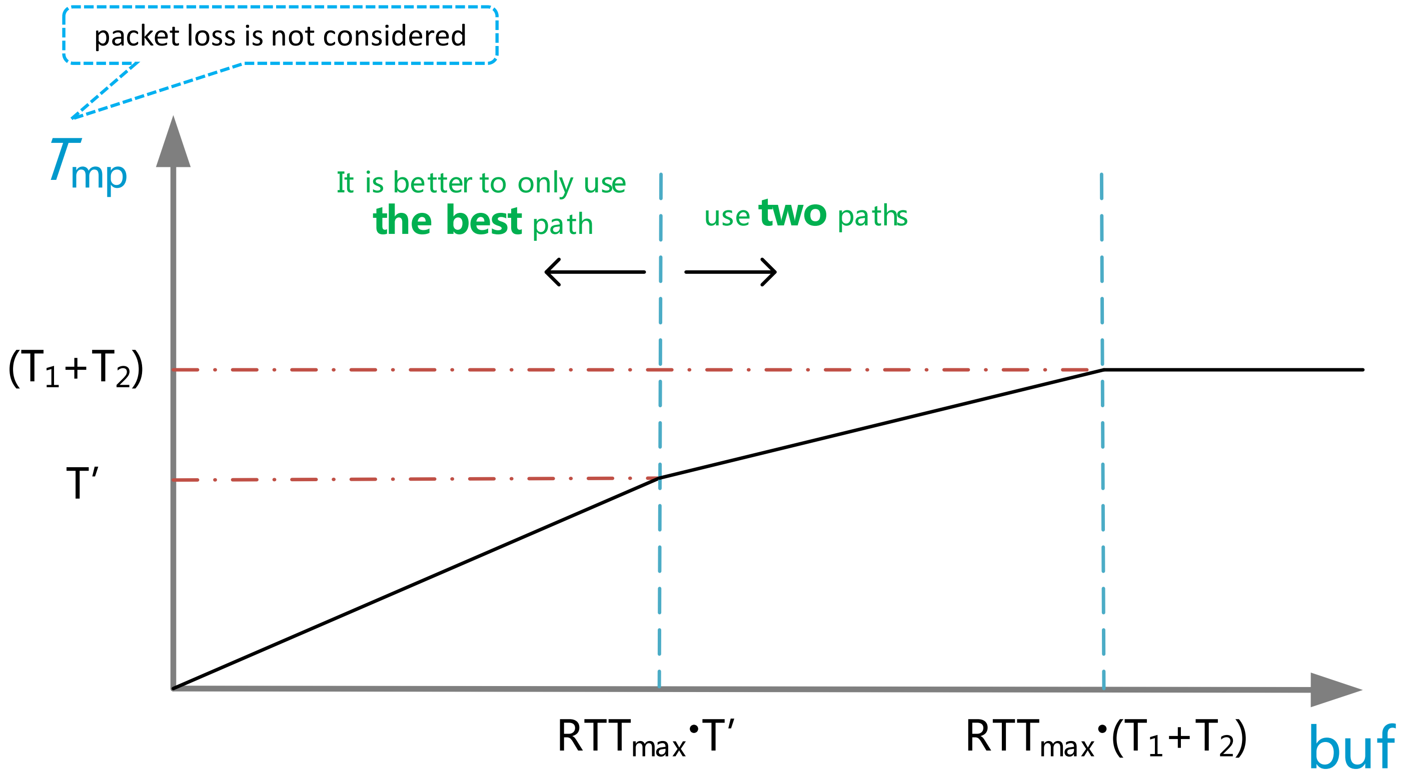
- Reinjection
- Halve the slow-path CWND
- Aim to ensure that $U_c >= 1$

However, $U_c < 1$ still exists, because it is

- Reactive: triggered when the performance has degraded
- Always trying to aggregate: this can be a problem !

One path may be better

- Achievable aggregate throughput
  - ✓ T_{mp} = buf/RTT_{max}, where
    - buf denotes the size of buffer
    - RTT_{max} = max(RTT_i)
  - ✓ Specially, T_{mp} <= T', when
    - buf <= $RTT_{max} \cdot T'$.
    - T' = max(T_1,T_2)
- If buf <= $RTT_{max} \cdot T'$ and bonding two paths
  - ✓ T_{mp} <= T'
  - ✓ Serious HoL blocking



Only use the best path may be better in some cases

Need a new solution

It should be:

- take **proactive actions** based on path attributes
- Adaptively employ both paths or only the best one according to the attributes

However, proactive actions can be counter-productive, because

- Throughputs of every single path and aggregated paths are needed
- Often estimated as CWND/RTT, which is not accurate due to severely variant CWND (caused by random packet losses)

BBR helps

BBR is part of mptcp since v0.93

BBR helps proactive measurement by offering:

- Stable throughput
  - ✓The throughput estimation of BBR is not a loss based cc.
- Stable RTT
  - ✓The pacing of BBR reduces the buffer bloat

Our solution with BBR

Challenge:

- How to get the throughput of each path and MP

Existing solutions:

- Modelling
  - ✓ Modelling with path attributes, e.g. RTT, PLR, BW, etc.
- Measurement
  - ✓ One-by-one throughput measurement of each path and MP
  - ✓ Out-of-band measurement, e.g. PCP (draft-wing-mptcp-pcp-00)

Overview

- Directly measure the throughputs of each path and MP
- Modification only in the sender
- Simultaneously measure the throughput of each path
- Periodically measure

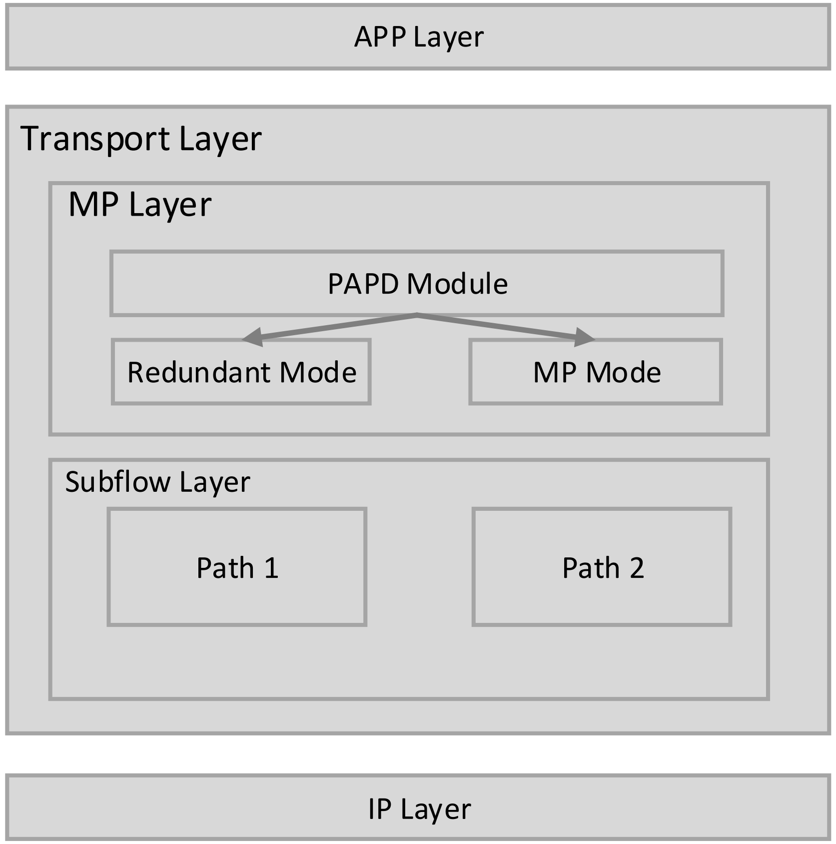
Proactive approach to Avoid the Performance Degradation (PAPD)

Throughput measurement

- Redundant mode
  - ✓ Simultaneously measure the throughputs of each path and the best path
- MP mode
  - ✓ Measure the aggregate throughput of MP layer

Mode Selection (Redundant vs MP)

- For the redundant mode, select the best path accordingly



Two stages of PAPD

- Slow-start stage → only use redundant mode
  - Unknown attributes of the paths
  - CWND increases twice after each RTT
  - The CWND of each path increases isolated, due to different RTT and delayed subflow connection

- Congestion-avoidance stage → the better mode wins
  - Fully utilized paths
  - Time-variant networks

Results – with large and small buffers

- When Buffer is large enough, minRTT is good enough
- PAPD performs equally well

- When Buffer is small, PAPD outperforms minRTT

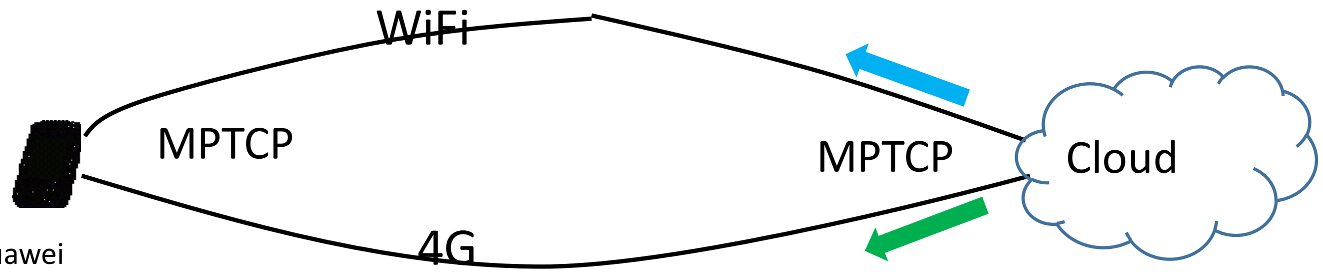
RTT_1	p_1	RTT_2	p_2	minRTT	PAPD	The best path
10	0.0001	10	0.0001	183(Mbps)	**173**	91.4
10	0.0001	10	0.001	183	**173**	91.4
10	0.0001	10	0.01	182	**172**	91.4
10	0.001	10	0.0001	183	**173**	91.4
10	0.001	10	0.001	182	**173**	91.3
10	0.001	10	0.01	182	**172**	91.2
10	0.1	10	0.0001	173	**165**	91.4
10	0.1	10	0.001	171	**165**	91.3
10	0.1	10	0.01	170	**163**	90.5

RTT_1	p_1	RTT_2	p_2	minRTT	PAPD	The best path
50	0	100	0.001	72.8	**88.5**	90.5
50	0	100	0.01	45.1	**85.8**	90.3
50	0	150	0	36.8	**84.6**	90.1
50	0	150	0.0001	51.3	**86.9**	90.5
50	0	150	0.001	47.7	**86.3**	90.2
50	0	150	0.01	30.6	**84.7**	89.9
50	0	200	0	26.9	**83.4**	89.9
50	0	200	0.0001	39	**85.2**	90.2
50	0	200	0.001	36.8	**84.8**	90
50	0	200	0.01	23.4	**83.7**	90

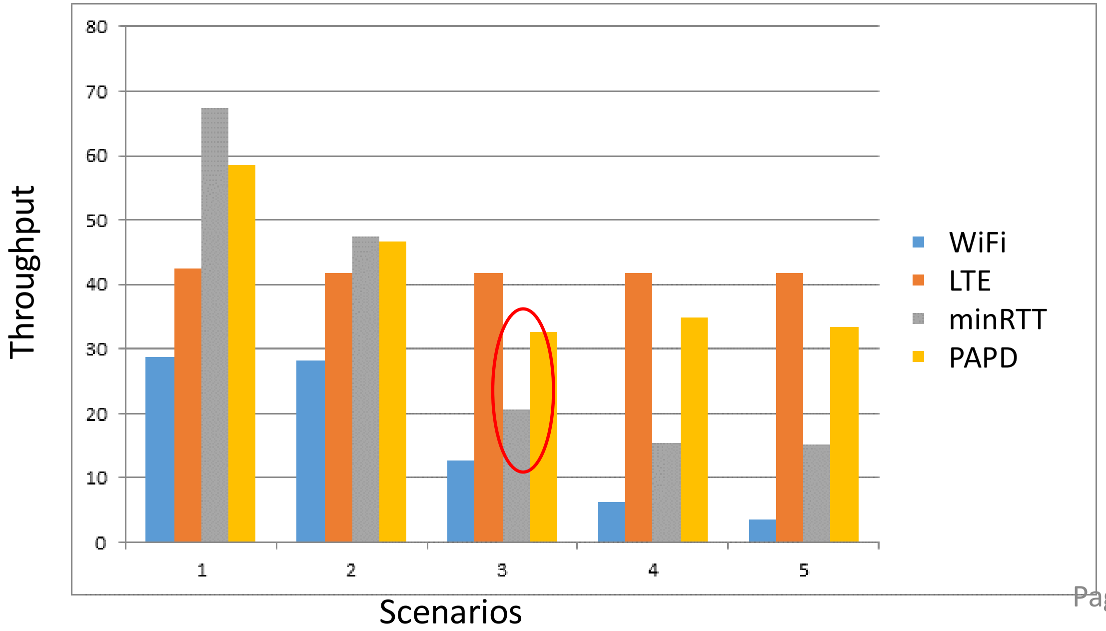
MinRTT is 26% of the best path

Mobile-Cloud Scenario: PAPD performs better than min-RTT when RTT variance increases

Testing scenario： Huawei Mate 9 downloads a large file (380M) from Huawei Cloud.



Scenario	WiFi RTT(Avg./Range)	Delay (jitter)ms	WiFi PLR	4G RTT(Avg./Range)	4G PLR
1	54ms/49-167ms	0	0%	96ms/67-336	0%
2	58ms/48-185ms	0	0.10%	96ms/67-336	0.10%
3	165ms/146-165ms	100(10)	2.00%	96ms/67-336	0.10%
4	371ms/235-625ms	200(20)	2.00%	96ms/67-336	0.10%
5	435ms/324-655ms	300(30)	2.00%	96ms/67-336	0.10%



Thank you
Questions?

Any interests in continuing this work/direction?