MPTCP - Scheduling
A brief overview of algorithms, their strengths and areas of application.

Henrik Gerdes

hegerdes@uos.de

September 23, 2021

UNIVERSITÄT OSNABRÜCK

Table of Contents I

Outline

MPTCP IS HERE!

Korean & Hrvatski Telecom enabled MPTCP

- Android smartphones can bond WiFi and LTE
- Allows download speeds of 800 Mbps

French ISP OVH 2015

- OverTheBox service is MPTCP capable
- Bond several DSL cable links

Other

- Apples SIRI
- Several MPTCP Linux implementations
- Deployable for everyone with SOCKS

A First Analysis of Multipath TCP on Smartphones

- Catholic University of Louvain
- Real word usage with a dozen of users
- Linux MPTCP v0.89.5 and Android 4.4
- Android with ShadowSocks
- SOCKS proxy with tcpdump
- Belgium from March 8th to April 28th 2015
- Total traffic of 25.4 GB in test period
- Over 390,782 MPTCP connections

Data analysis

Number of subflows	1	2	3	4	5	>5
Percentage of connections	67.75%	29.96%	1.07%	0.48%	0.26%	0.48%

Table: Number of subflows per MPTCP connection, Source:[DCBHB16]

Port	# connections	% connections	Bytes	% bytes
53	107,012	27.4	17.4 MB	<0.1
80	103,597	26.5	14,943 MB	58.8
443	104,223	26.7	9,253 MB	36.4
4070	571	0.1	91.7 MB	0.4
5228	10,602	2.7	27.3	0.1
8009	10,765	2.8	0.97	<0.1
Others	54,012	13.8	1.090 MB	4.4

Table: Statistics about destination port, Source: [DCBHB16]

MPTCP promises to:

Offer resilience

Increases bandwidth

Provide seamless handovers

Backwards compatible and deployable

...

Outline

Asymmetric paths

Multiple subflows have different characteristics in terms of:

- Round Trip Time (Latency)
- Bandwidth (Throughput)
- Error rate (Reliability)

Blocking

Figure: Head-of-line blocking and receive buffer blocking, Source: [YFD+16]

Outline

MPTCP scheduler

1. Lowest-RTT-First (LRF)
2. BLock ESTimation (BLEST)
3. Shortest Transfer Time First (STTF)
4. Earliest Completion First (ECF)
5. Additional schedulers

Lowest-RTT-First

1. Uses the subflow with the lowest RTT
2. Until congestion window is filled
3. Use next lowest RTT
4. Standard Linux scheduler

$$
\begin{aligned}
err &= measured_{RTT} - predic \\
predic_{new} &= predic_{old} + \frac{1}{8} * err \\
variation_{new} &= \frac{3}{4} * variation_{old} + \frac{1}{4} * \|err\| \\
RTO &= predic + 4 * variation
\end{aligned}
$$

Table: Jacobson's Algorithm

BLock ESTimation scheduler

- Specifically designed for asymmetric paths
- Estimate the amount of blocking
- Additional send window on the connection level
- Added to Linux MPTCP in June 2019

Skips subflow if:

$$X \times \lambda > |MPTCP_{SW}| - MSS_S * (inflight_S + 1) \tag{1}$$

$$X = MSS_F * \left(CWND + \left(\frac{RTT_S}{RTT_F} - 1 \right) /2 \right) \times \frac{RTT_S}{RTT_F} \tag{2}$$

Shortest Transfer Time First

- Similar approach as LRF
- Considers enqueued packets
- Quick response to network changes

- Fast rescheduling
- Latency orientated scheduler
- Resource heavy and complex implementation

Earliest Completion First

- Similar approach as LRF and STTF
- Is able to skip subflows for small amounts of data transmissions
- Prioritizes throughput over latency
- Can also combine subflows for big transmissions

Skips subflow if:

$$RTT_F + \frac{k}{CWND_F} \times RTT_F \leq RTT_S \tag{3}$$

Inner workings

Figure: Scheduling decisions for a burst of 15 segments, using the LRF, BLEST, and STTF schedulers, Source: [YFD+16]

Additional schedulers

- Out-of-Order Transmission for In-Order Arrival Scheduler (OTIAS)
- Delay-Aware Packet Scheduler (DAPS)
- RoundRobin (RR)
- Redundant Scheduler

Outline

Schedulers in short

1. LRF und RR are not suitable for asymmetric paths
2. LRF does not consider the relative subflows RTT difference
3. BLEST and STTF both prioritize latency. Both are able to skip subflows
4. STTF is complex and heavy
5. ECF improves utilization of the fastest subflow. Prioritizes throughput

1. **Controlled latency analysis**
2. **Controlled bandwidth analysis**
3. **Real-World analysis**

Performance baseline

- Results for LRF, DAPS, OTIAS, ECF
- Figure a): WLAN/3G setup
- Figure b): WLAN/WLAN setup
- All schedulers perform similarly with marginal differences in a)
- Increased page load times for DAPS and OTIAS in b)

Figure: Page load time for web traffic, Source: [HGB+18]

Latency

Figure: Average transmission time for schedulers, Source: [HGB+18]

Figure: Average Download Completion Time, Source: [LNTG17]

Throughput I

- Increasing asynchrony
- WLAN 1 MBps and LTE 1 to 10 MBps

Figure: Average goodput for TCP and MPTCP (LRF, BLEST, and STTF), Source: [HGB+18]

- DAPS is always worse
- ECF fastest with BLEST second

- Shows intact of growing RTT asynchrony
- Throughput degrades drastically if $\frac{RTT_2}{RTT_1} \leq 4$
- A single TCP connection can deliver higher throughput (on highly asymmetric paths)

Real-World

- $RTT_{WLAN} \approx 25ms$ & $RTT_{3G} \approx 75ms$
- Object load times improved by up to 26% (BLEST) and 32% (STTF)
- Practical ECF test showed 26% faster load of CNN-page
- ECF improved video streaming throughput by 16% (WLAN-Spot & LTE)

Figure: Page load and object times for HTTP2, Source: [HGB+18]

Conclusion time

- Best scheduler is . . . it depends
- Know your use cases
- Default scheduler are not optimized for asymmetric paths
- BLEST is a promising scheduler for asymmetric paths
- Linux-MPTCP is flexible and easy adoptable
- Development is active and MPTCP evolves fast new MPTCP RFC 6884 revision (March 2020)

References I

- Q. De Coninck, M. Baerts, B. Hesmans, and O. Bonaventure, “A first analysis of multipath TCP on smartphones,” in *International Conference on Passive and Active Network Measurement*. Springer, 2016, pp. 57–69.
- P. Hurtig, K.-J. Grinnemo, A. Brunstrom, S. Ferlin, Ö. Alay, and N. Kuhn, “Low-latency scheduling in MPTCP,” *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 302–315, 2018.
- Y.-s. Lim, E. M. Nahum, D. Towsley, and R. J. Gibbens, “ECF: An MPTCP path scheduler to manage heterogeneous paths,” in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*. IEEE, 2017, pp. 147–159.

References II

K. Yedugundla, S. Ferlin, T. Dreibholz, Ö. Alay, N. Kuhn, P. Hurtig, and A. Brunstrom, "Is multi-path transport suitable for latency sensitive traffic?" *Computer Networks*, vol. 105, pp. 1–21, 2016.

Questions?

Thank you for your attention!
Any questions?

Shortest Transfer Time First

```
function TRANSFER_TIME:
    if cwnd_free > 0 and data_to_send < cwnd_free then
        return rtt / 2
    transfer_time = transfer_time + rtt
    cwnd = increase_cwnd(current_cc_state)

    if data_to_send <= max_segments_in_ss then
        transfer_time = transfer_time + rtt * (rounds_in_ss-1) + rtt/2
        return transfer_time
    else if cwnd < ssthresh then
        transfer_time = transfer_time + max_rounds_in_ss * rtt
    if ends_in_ss(data_to_send) then
        return transfer_time

    cwnd = ssthresh
    transmission_time += rtt * (rounds_in_ca - 1) + rtt / 2
  return transfer_time
```

Listing: STTF pseudo code, Source: [HGB+18]