Why Multipath TCP Degrades Throughput Under Insufficient Send Socket Buffer and Differently Delayed Paths

Toshihiko Kato, Adhikari Diwakar, Ryo Yamamoto,
Satoshi Ohzahata
University of Electro-Communications
Email: kato@net.lab.uec.ac.jp

Presenter: Toshihiko Kato

- Professor of University of Electro-Communications located in Tokyo, Japan
- Research interest includes communication protocols, such as TCP, Contents centric networks.
- This paper focuses on the behavior of Multipath TCP under limited send socket buffer.
    - MPTCP throughput degrades worse than single path TCP when send socket buffer size is not sufficient (we pointed out in previous paper).
    - This paper discusses why such degradation happens.

1. Introduction (1)

- Recent Mobile Terminals：　Multiple Network Interfaces (WLAN/LTE)
- TCP using Multiple Interfaces：Multipath TCP
  - Multiple TCP connections (Subflows) => One MPTCP connection
  - Application Does Not care about MPTCP
- Three RFCs
  - RFC 6182：　Guideline for Protocol Design
  - RFC 6824：　Detailed Protocol Procedures
  - RFC 6356：　Congestion Control

1. Introduction (2)

- Changing path delay and send socket buffer size (receive socket buffer large enough)
  - Send socket buffer ⇒ retransmission, not appear as protocol parameter
- Under some conditions: Throughput is lower than one TCP connection
  - Send socket buffer among subflows
  - Due to starvation of send socket buffer, data sending stops
  - A kind of Head-of-Line blocking

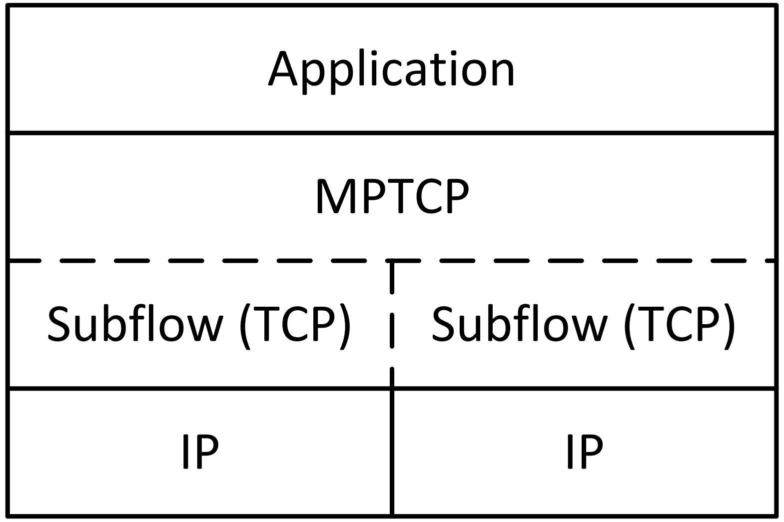
1. Introduction (3)

This paper:

- Analyze Linux MPTCP software
- Estimate the reason for throughput degradation

2. Related Work (1)

- MPTCP：　locate over TCP
- Suflows（legacy TCP connection） and MPTCP connection
  - MP_CAPABLE TCP option in first subflow
  - MP_JOIN TCP option in second subflow
    - Associate subflows and MPTCP connection



2. Related Work (2)

- MPTCP level data sequencing: Data Sequence Signal（DSS) option
  - Data Sequence Number／Data Acknowledgment（DACK）

<table>
<tr><td>Kind (= 30)</td><td>Length</td><td>Subtype (= 2)</td><td>Flags</td></tr>
<tr><td colspan="4">Data ACK (4 or 8 octets, depending on flags)</td></tr>
<tr><td colspan="4">Data sequence number (4 or 8 octets, depending on flags)</td></tr>
<tr><td colspan="4">Subflow sequence number (4 octets)</td></tr>
<tr><td colspan="2">Data-level length (2 octets)</td><td colspan="2">Checksum (2 octets)</td></tr>
</table>

2. Related Work (3)

- NO window size parameter in MPTCP
  - Share window size among MPTCP connection and subflows
- Recommended receive socket buffer size

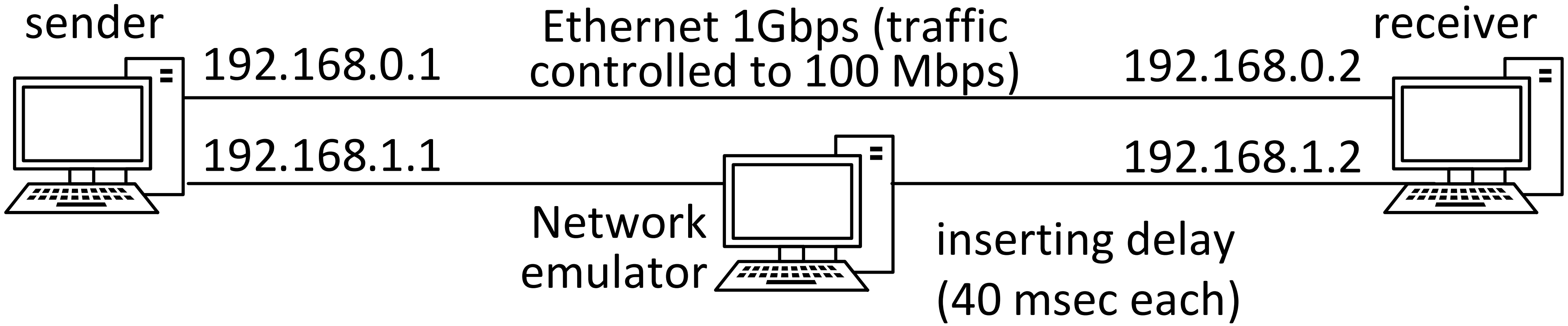
$$\text{Buffer size} = \sum_{i}^{n} bw_i \times RTT_{max} \times 2$$

2. Related Work (4)

- Scheduler: Assign data from application to subflows
- Dafault scheduler: minRTT:
  - select a subflow with smallest RTT
  - send data continuously according to advertised window and congestion window
  - opportunistic retransmission and penalization （RP） mechanism

3. Throughput Degradation due to Insufficient Send Socket Buffer

A. Experimental settings



Send socket buffer size: 1,048,576 bytes (1 Gibibytes)

Receive socket buffer size: default setting

4,096, 87380, and 6,291,456 bytes
for the minimum, default, and maximum sizes

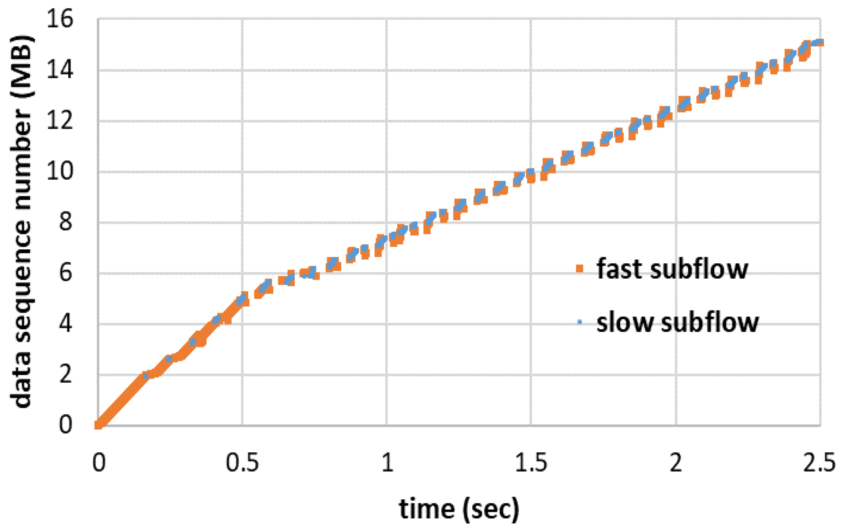
3. Throughput Degradation due to Insufficient Send Socket Buffer

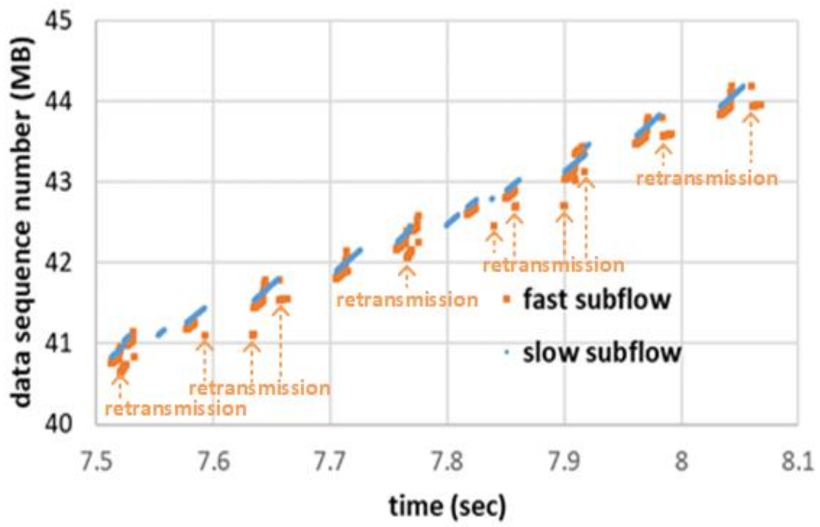
B. Results and analysis

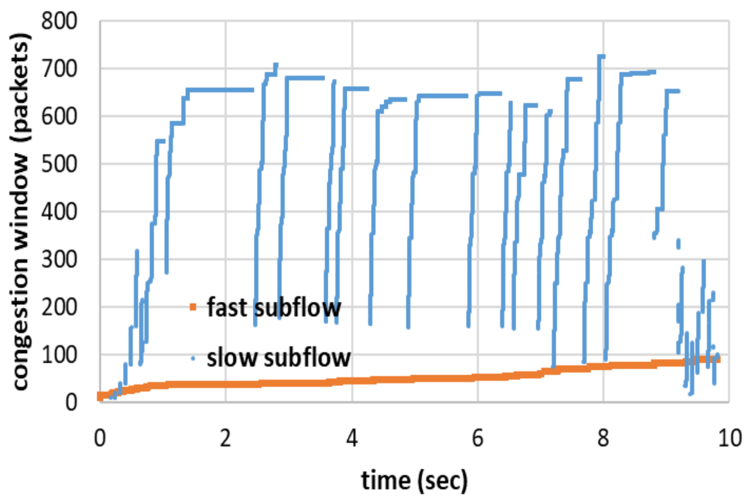
5 experiment runs

Throughput measured at receiver side: 42.4 to 49.8 Mbps

Slower than 100 Mbps







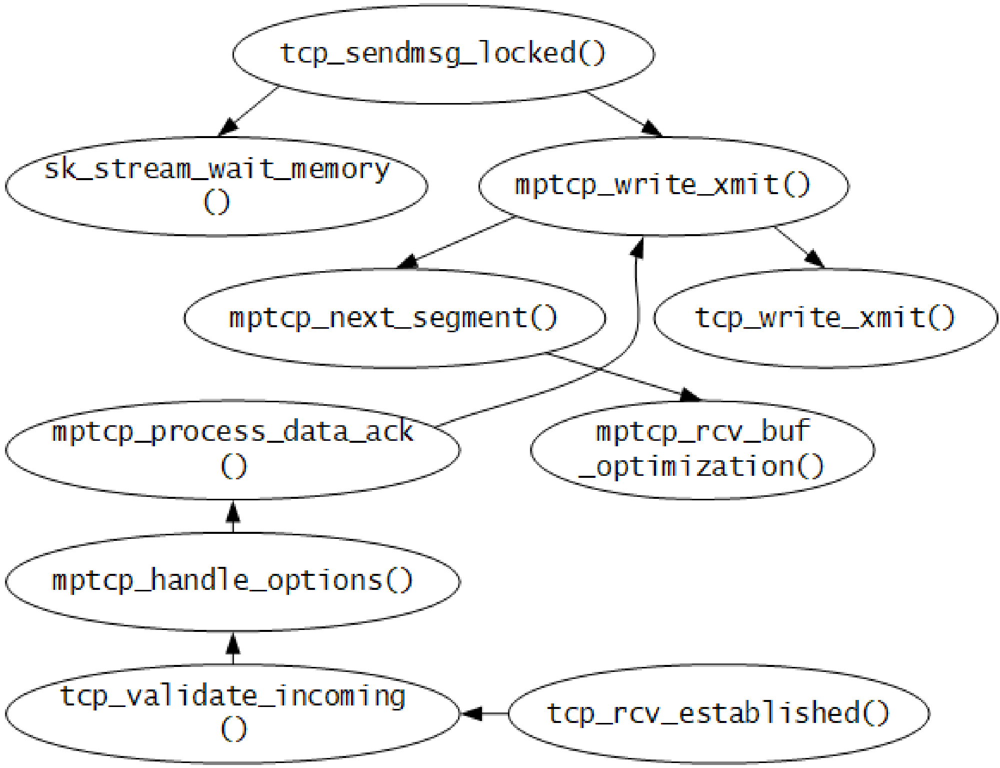
Intermittent data transfer

4. Analysis of Linux MPTCP Software
A. Internals of Linux MPTCP

Data sending from upper layer is done by tcp_sendmsg_locked()

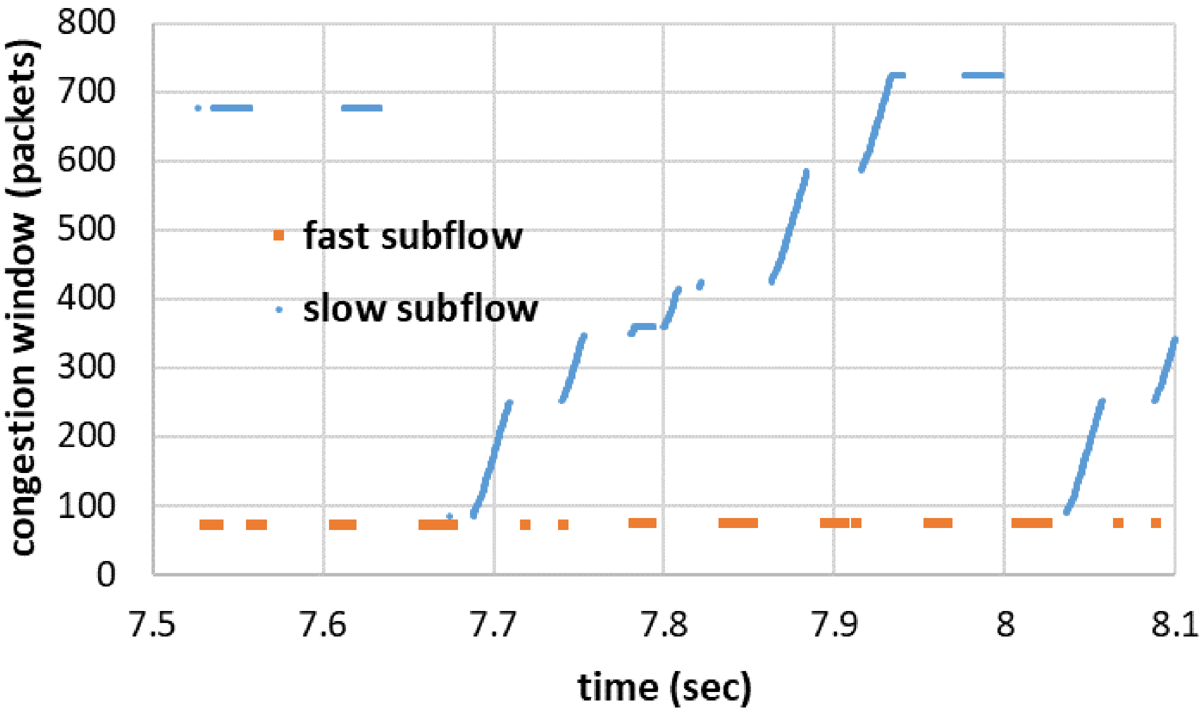
Send socket buffer starvation is handled by sk_stream_wait_memory()

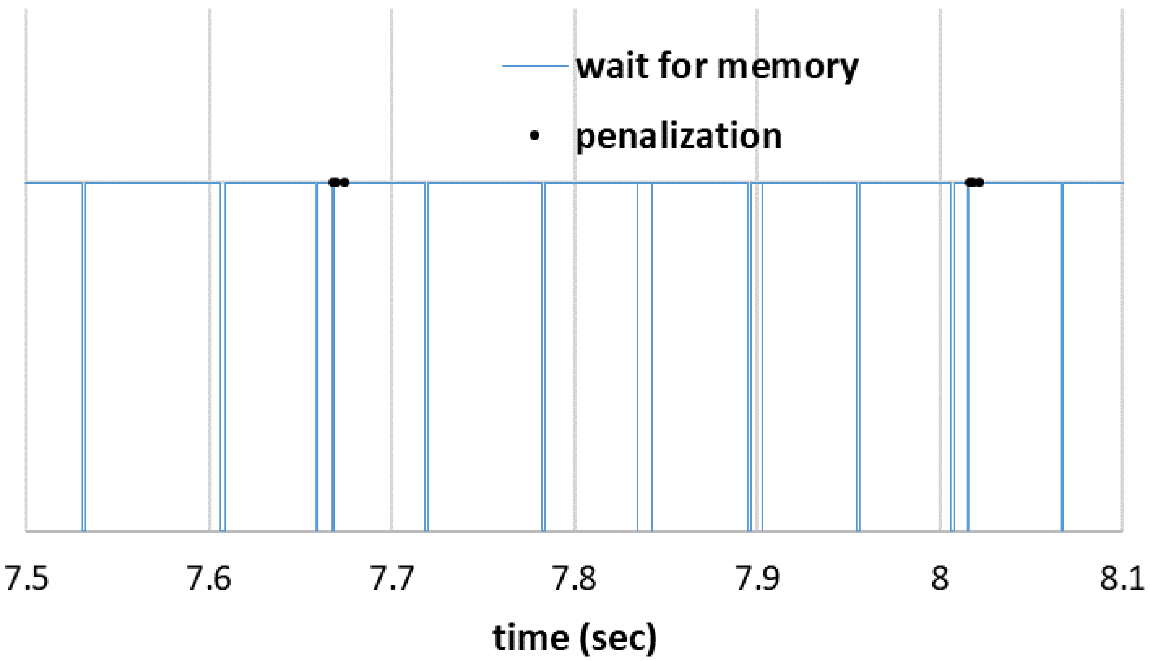
RP mechanism is handed by mptcp_rcv_buf_optimization(), independently of send socket buffer processing



4. Analysis of Linux MPTCP Software

B. Behaviors of Linux MPTCP Software





5. Conclusions

- We showed this situation by the experiments using the in-house network and discussed the details of the MPTCP parameters during the degradation.
- We also showed the internal structure of Linux MPTCP software focusing on the buffer starvation and the MPTCP scheduler.
- We showed a possible reason why the performance degradation occurs.