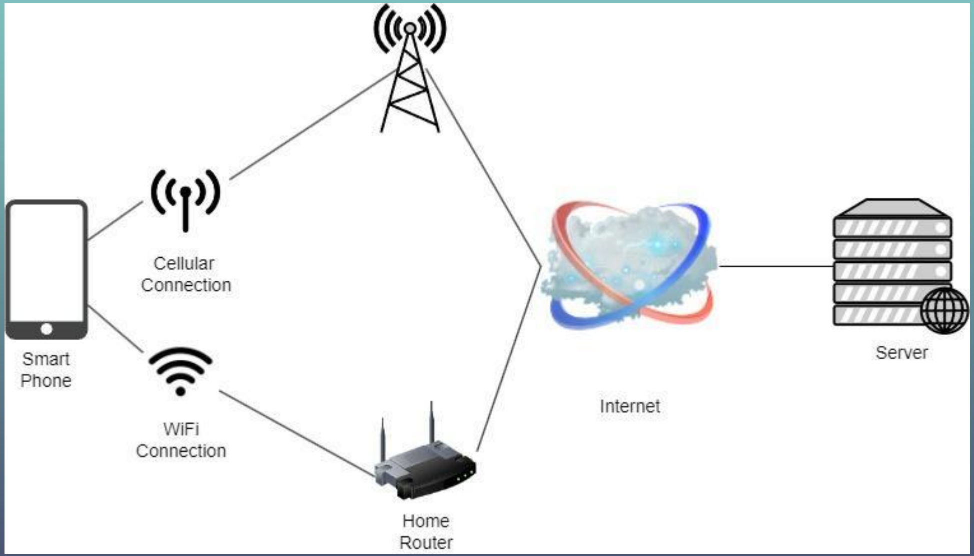
Multipath TCP Overview and Packet Scheduling Method

Cole Maxwell
University of Minnesota, Morris

Introduction

- Modern devices have multiple network connections
- Transition between connections not seamless
- More than one connection cannot be used simultaneously
- MPTCP aims to solve these problems



Outline

Background

- Computer Networks
- **T**ransmission **C**ontrol **P**rotocol (TCP)
- **M**ultipath **TCP** (MPTCP) connections
- Challenges of data scheduling

Overview of a Purposed Packet Scheduling Method

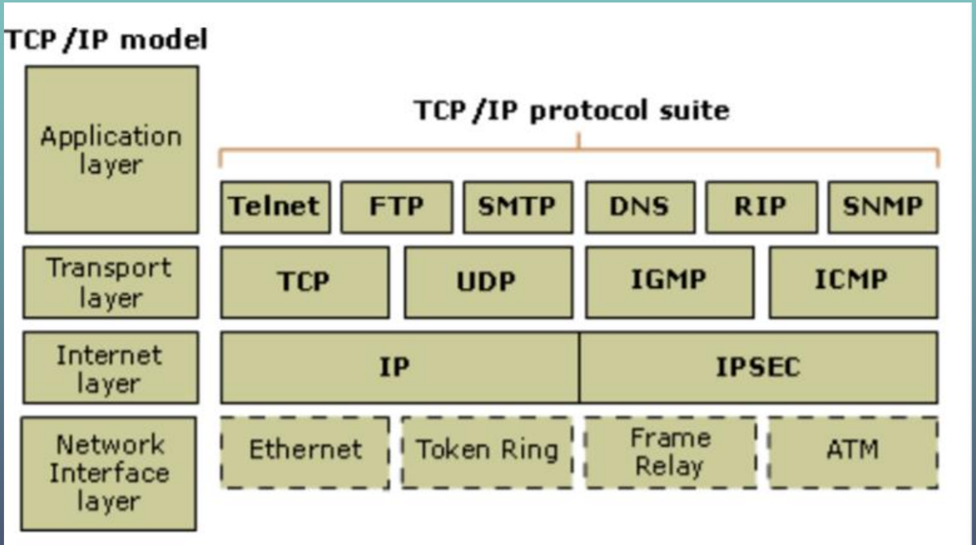
- Scheduling value estimation
- Characteristics used
- Dynamic scheduling value adjustment

Results and Performance

- Simulation results
- Performance comparison to other schedulers

Computer Networks

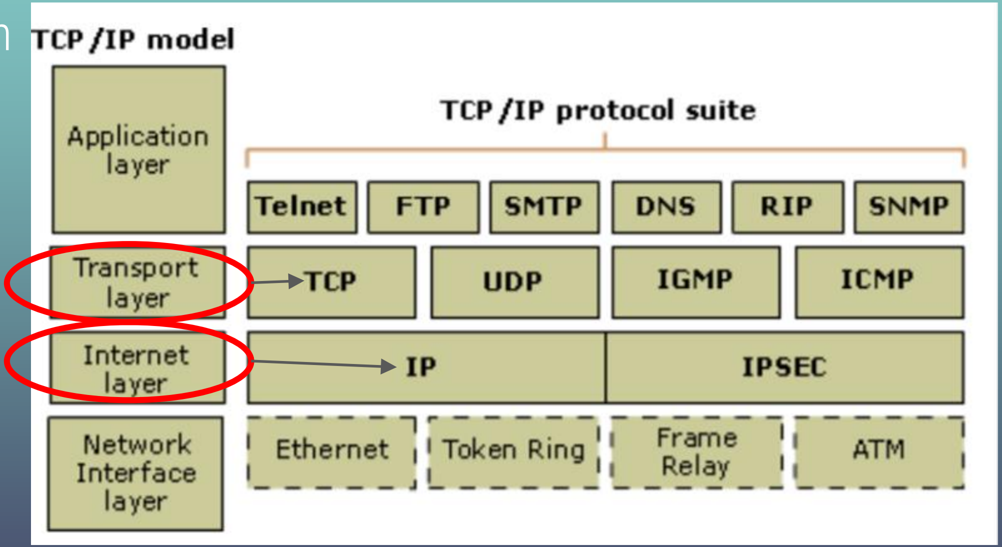
- Abstraction layers
- Layer responsible for a specific function
- Focus today
    - Network layer
    - Transport Layer



https://quizlet.com/302585362/tcpip-model-diagram/

Computer Networks

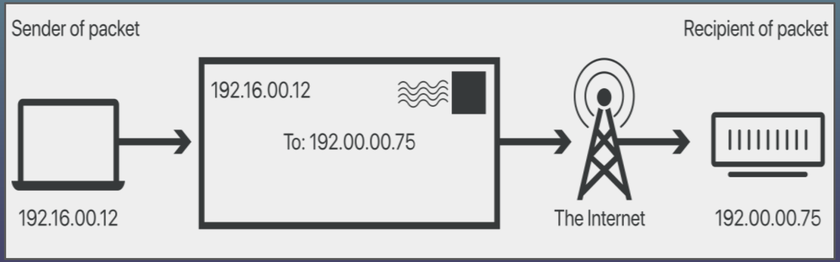
- Abstraction layers
- Layer responsible for a specific function
- Focus today
  - Network layer
  - Transport Layer



https://quizlet.com/302585362/tcpip-model-diagram/

Network Layer

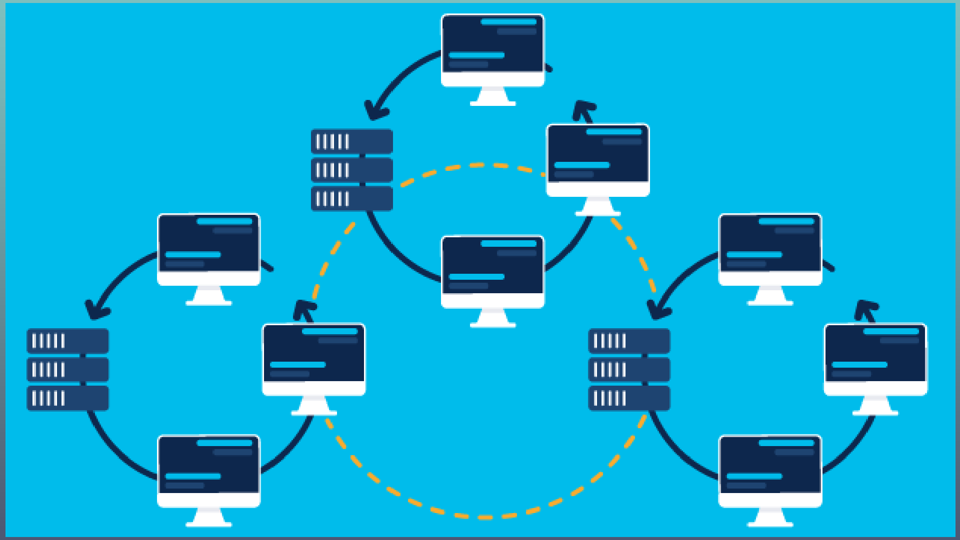
- Responsible for moving data
- Messages broken down into smaller pieces of data called ***packets***
- Computers on networks have **I**nternet **P**rotocol (IP) addresses
  - Analogous to real-world street addresses
- ***Routers*** move packets between networks



https://www.cloudflare.com/learning/network-layer/internet-protocol/

Transport Layer

- IP does **not** handle
  - packet ordering
  - error checking
- **U**ser **D**atagram **P**rotocol (UDP)
  - Data can be lost
  - Voice/Video
- **T**ransmission **C**ontrol **P**rotocol (TCP)
  - Data cannot be lost
  - File download

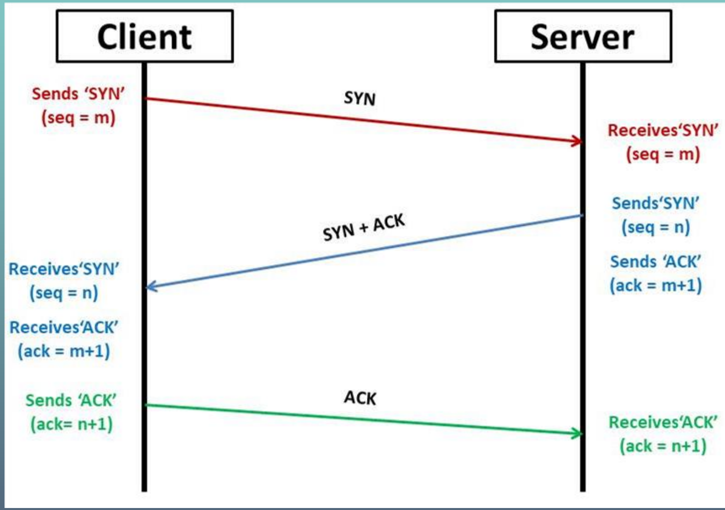


Transmission Control Protocol (TCP)

TCP 3-way Handshake

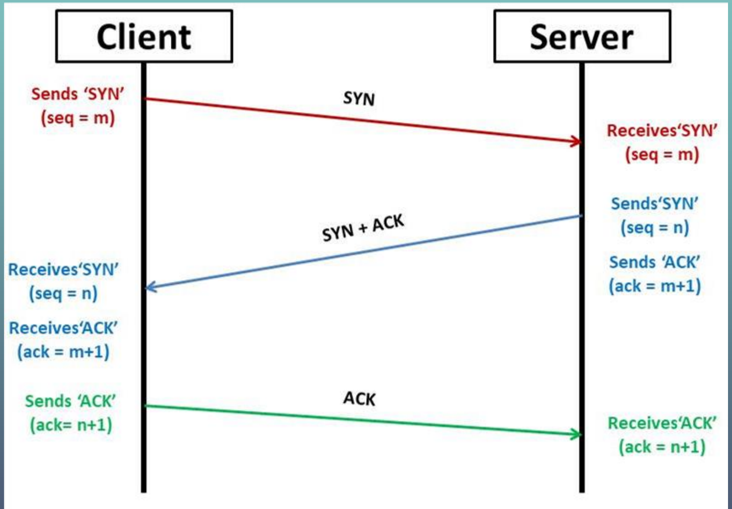
1. The sender sends an “initial request” or **synchronization** (SYN) to the receiver to start communication
2. Receiver sends **synchronization-acknowledgement** (SYN-ACK) to the sender, agreeing to connection
3. Sender sends an **acknowledgement** (ACK) to the receiver

- Messages can now be sent.

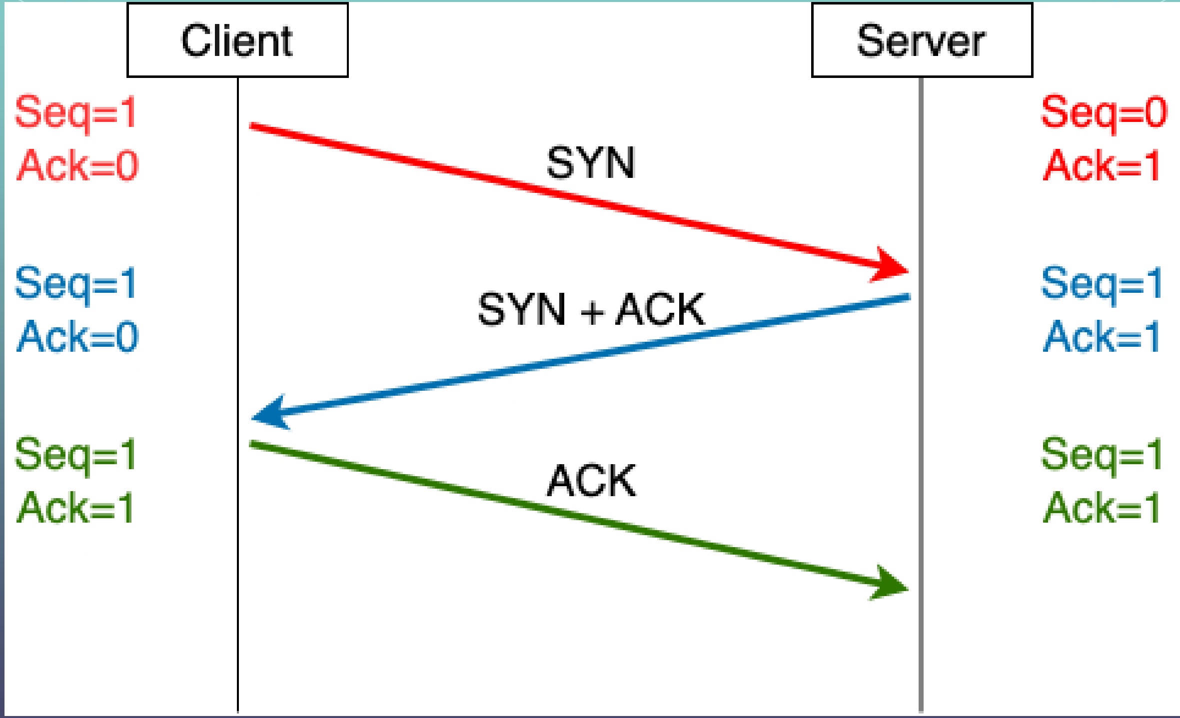


Sequence and Acknowledgement Numbers

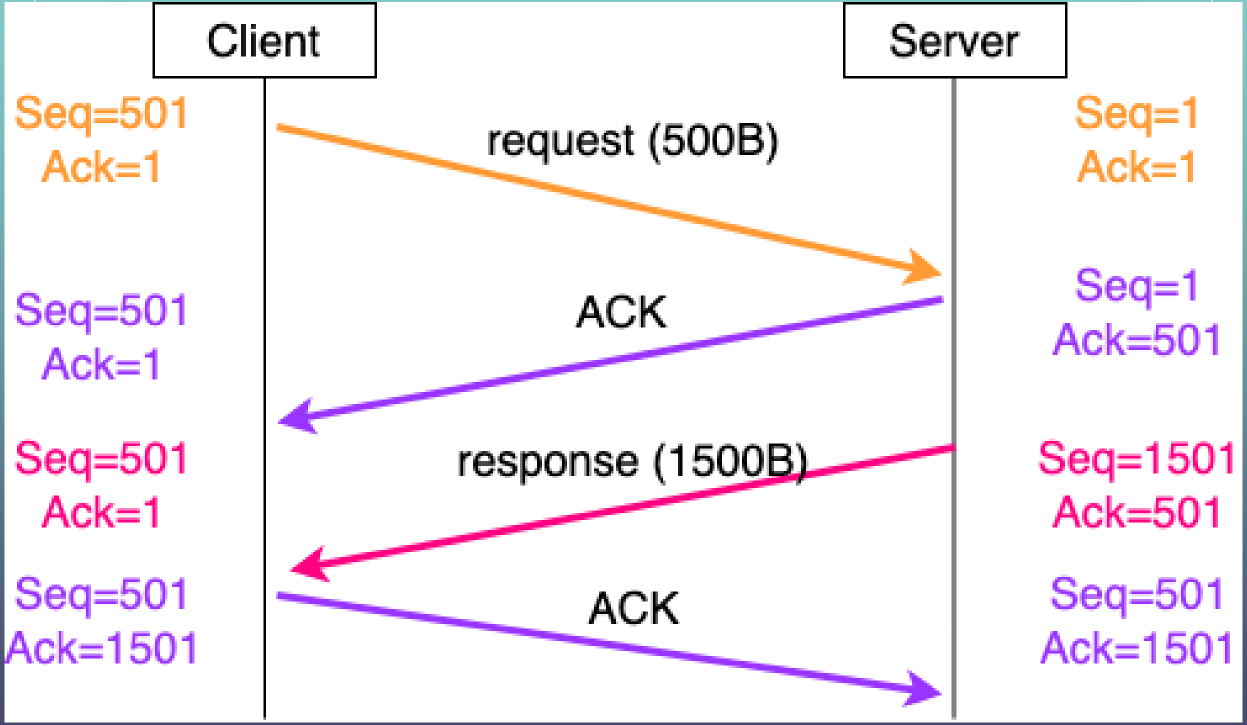
- Tacks “in flight” data to ensure delivery
- Orders received data
- Re-sends data if not delivered
- Random 32-bit numbers
    - Discussed in relative terms



Sequence and Acknowledgement Numbers



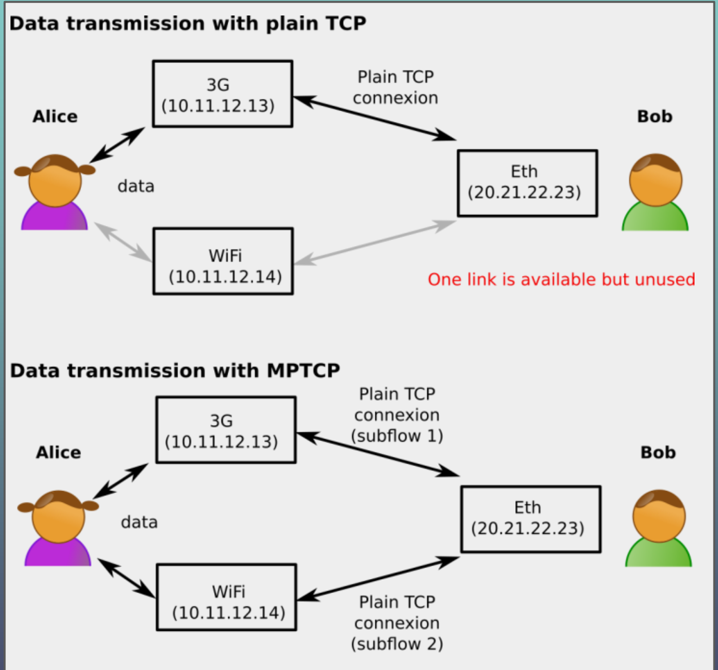
Sequence and Acknowledgement Numbers



Multipath Transmission Control Protocol (MPTCP)

The MPTCP Extension

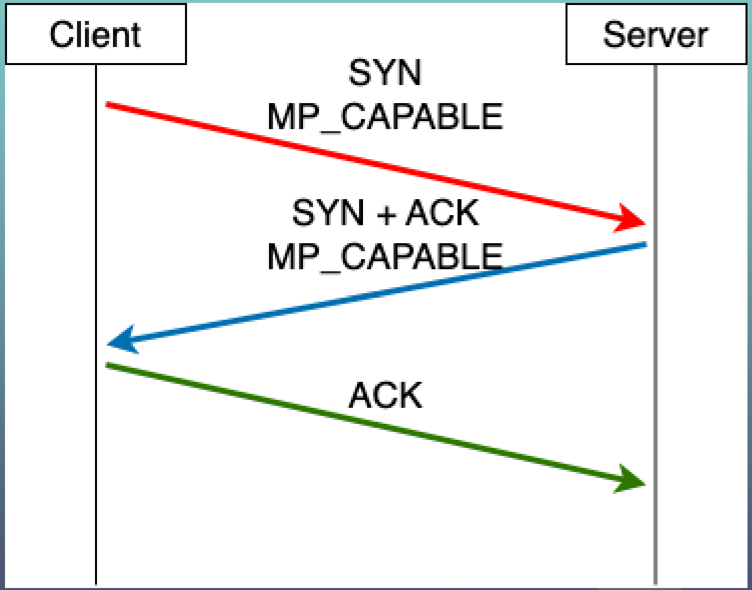
- Open standard
- Pushed by Apple in 2013, updated in 2020
  - For Siri performance
- Compatible with TCP
- Sends data simultaneously on all connections
- Each network connection is a ***subflow***
- Failover mechanisms if a subflow is lost
  - Loss of Wifi



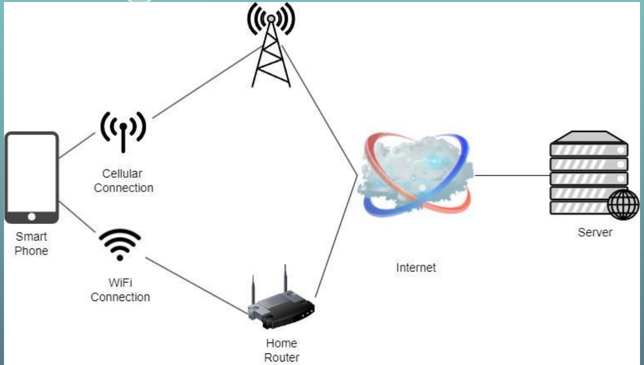
https://en.wikipedia.org/wiki/Multipath_TCP

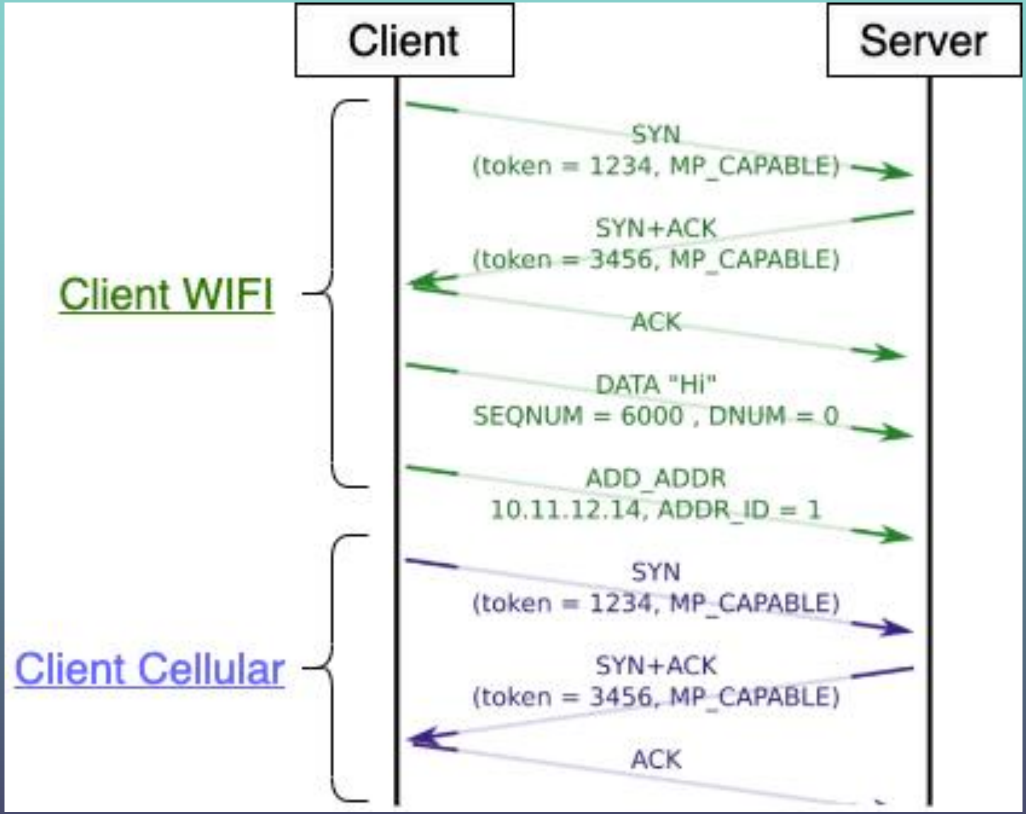
MPTCP Connection

- Subflows use same handshake as TCP
  - Allows fallback
- Additional connection parameters are exchanged in the handshake
  - **MP_CAPABLE** - specifies if a device supports MPTCP
  - **Session tokens** identify the connection



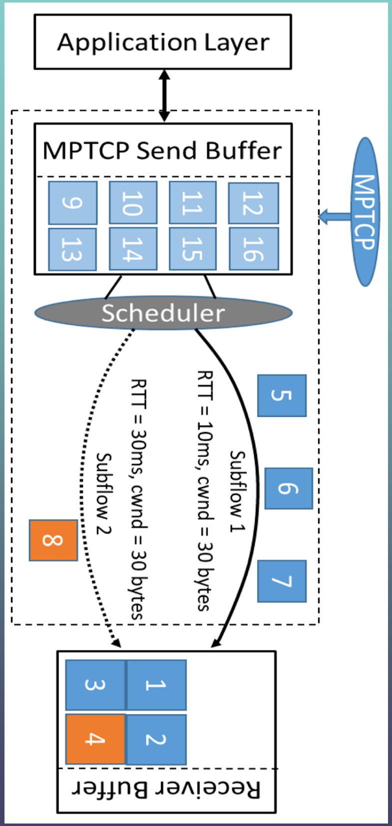
MPTCP Add Subflows



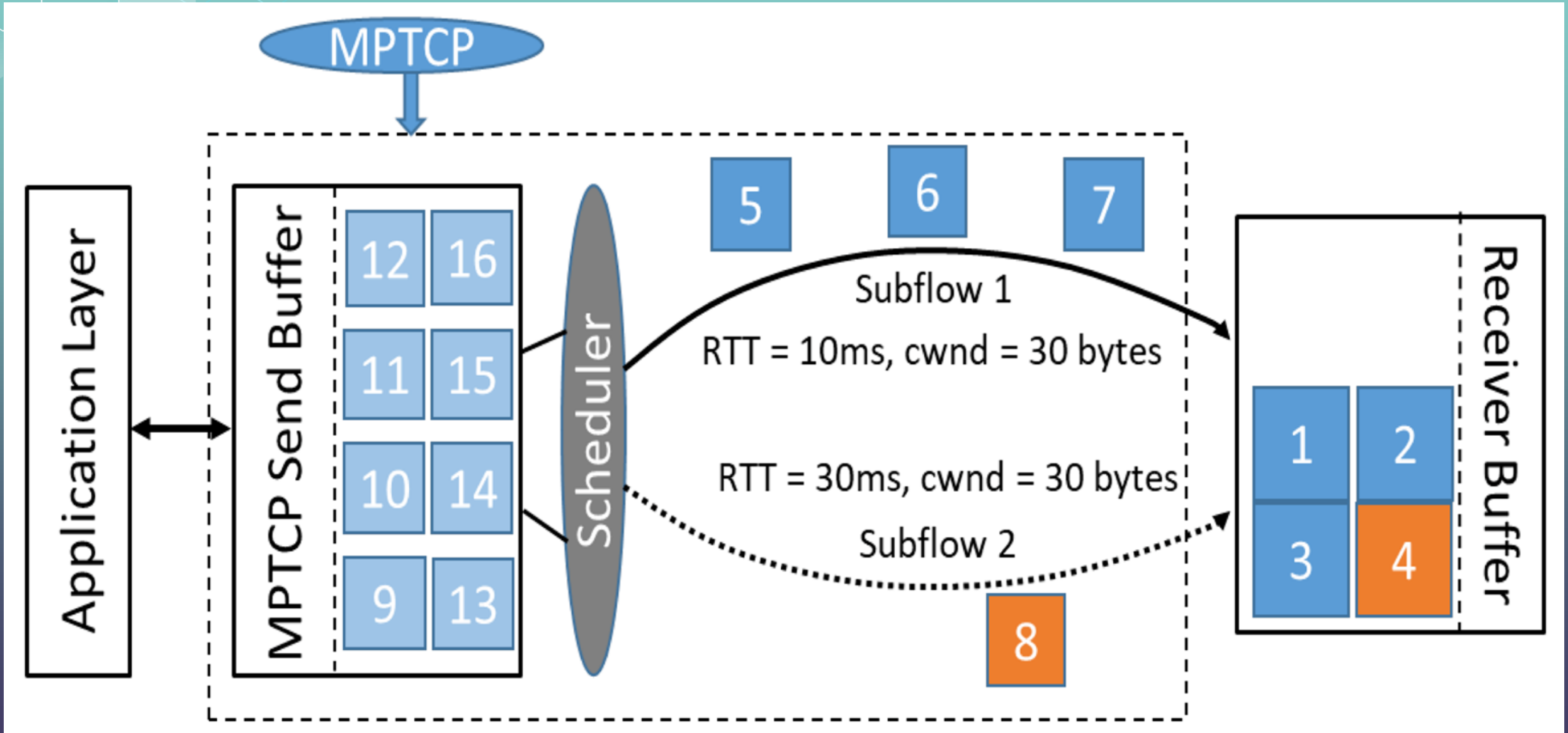


MPTCP

- Subflows maintain sequence and acknowledgment numbers.
  - **subflow sequence number (SSN)**.
- Packets ready to send put in a **sending buffer**
- **Data sequence number (DSN)** map to SSNs
  - Position in buffer -> SSN
- A scheduler allocates packets to subflows based on a <u>scheduling algorithm</u>.

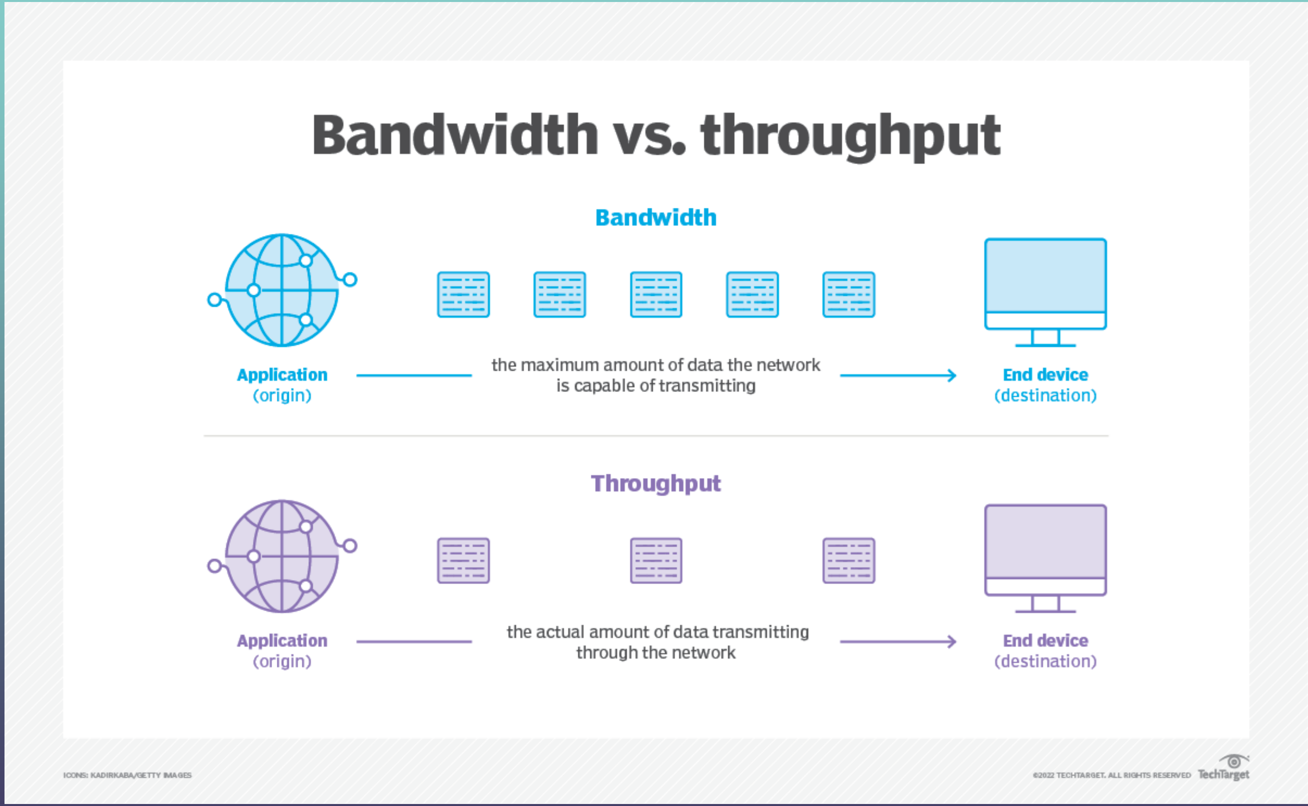


Out-of-Order Packet Problem



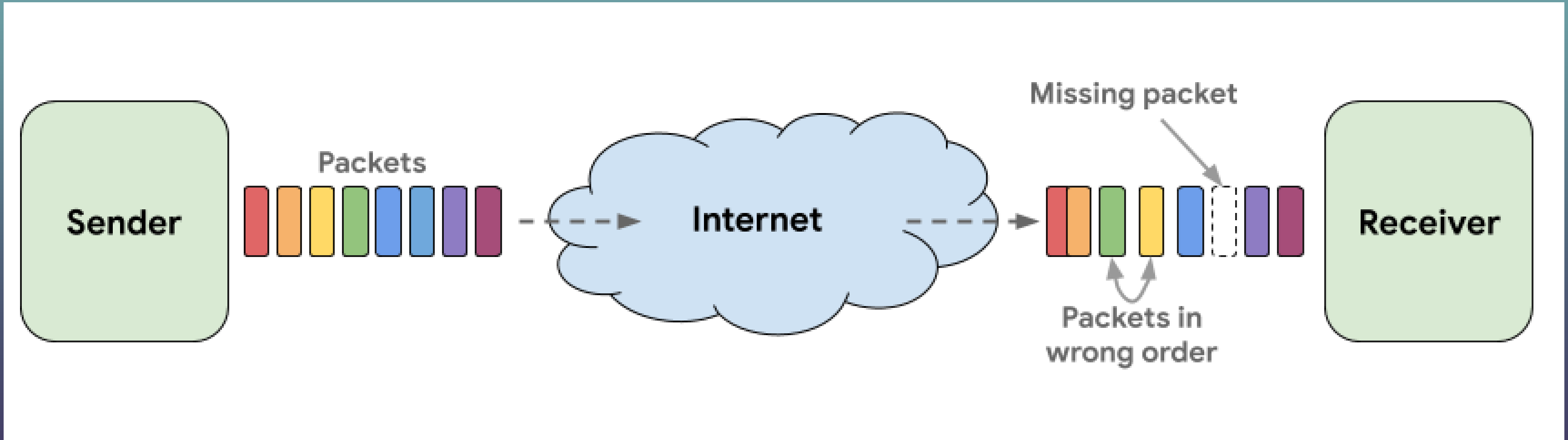
Subflow Characteristics

Subflow Characteristics



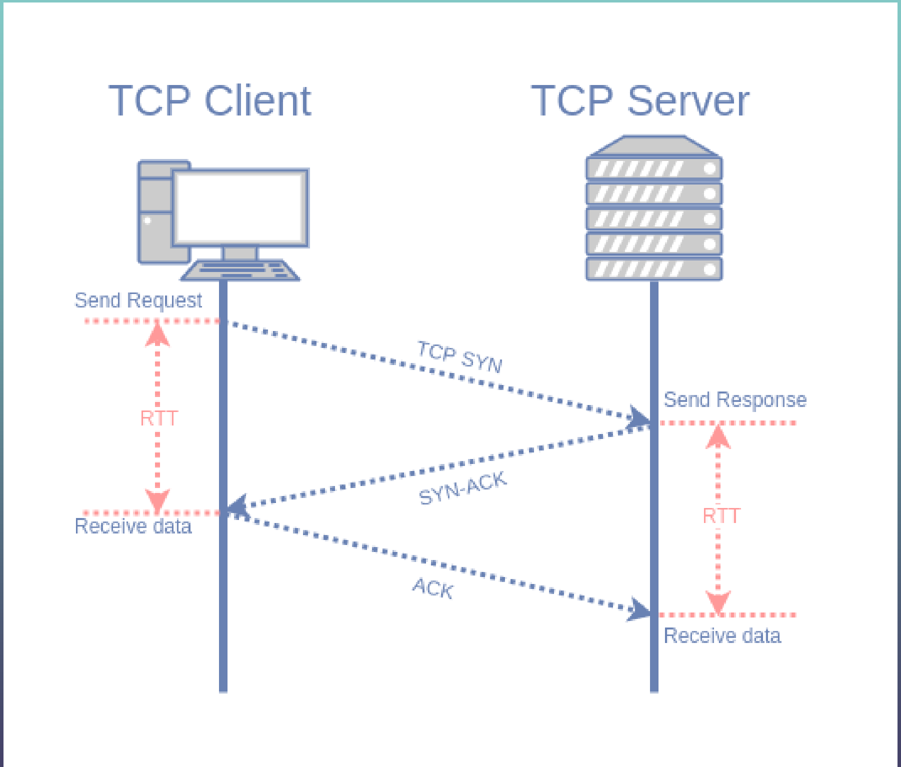
Subflow Characteristics

- **Packet loss-** when one or more transmitted data packets fail to arrive at their destination
- Can happen for many reasons, such as network congestion and hardware failure, but these reasons are **unknown to MPTCP**



Subflow Characteristics

- **Round Time Trip (RTT)-** the duration, measured in milliseconds, from when a request is made to when a response is received



A New Packet Scheduling Method

Heterogeneous Wireless Networks

- 2018 Paper: "*A Dynamic Packet Scheduling Method for Multipath TCP in Heterogeneous Wireless Networks*" from the College of Information Science and Electronic Engineering, Hangzhou, China
- One of the most challenging environments
    - Rapid shift in subflow characteristics

Method Goals

- Build on previous scheduling techniques:
  - Forward Prediction Scheduling (FPS)
  - Dynamic Packet Scheduling and Adjusting with Feedback (DPSAF)
- Use all the path characteristics of a subflow
- Improve throughput
- Minimize out-of-order packets

Stage One: Forward Prediction

- All path characteristics are used
- A prediction of the **amount of the earlier-arriving on other subflows packets**, denoted as N_i, is estimated.
- Subflows ordered by speed

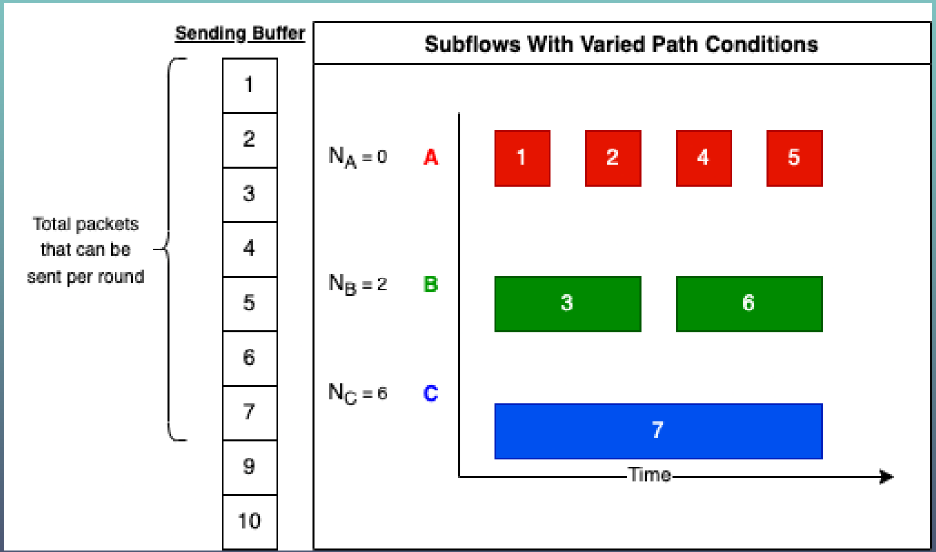
$$DATA_i = \sum_{1 \le j \le P_{RTT_j < RTT_i}}$$

$$DATA_{i,j} = \sum_{1 \le j \le P_{RTT_j < RTT_i}} BW_j \cdot \frac{RTT_i}{2} \cdot (1 - PLR_j)$$

$$N_i = \left\lfloor \frac{DATA_i}{MSS} \right\rfloor$$

Stage One: Forward Prediction

- All path characteristics are used
- A prediction of the **amount of the earlier-arriving on other subflows packets**, denoted as N_i, is estimated
- Subflows ordered by speed



Stage One: Forward Prediction

- A **simple exponential smoothing** method used to periodically estimate all the path characteristics based on their observation
- More recent observations better reflect the current state of a path
  - Smoothing parameters are set to 0.8 and 0.9
  - Weighs the **most recent more heavily** in the prediction

$$SRTT_i = \alpha \cdot SRTT_i + (1 - \alpha) \cdot RTT_i$$
$$SBW_i = \beta \cdot SBW_i + (1 - \beta) \cdot BW_i$$
$$SPLR_i = \gamma \cdot SPLR_i + (1 - \gamma) \cdot PLR_i$$

Stage Two: Dynamic Adjustment

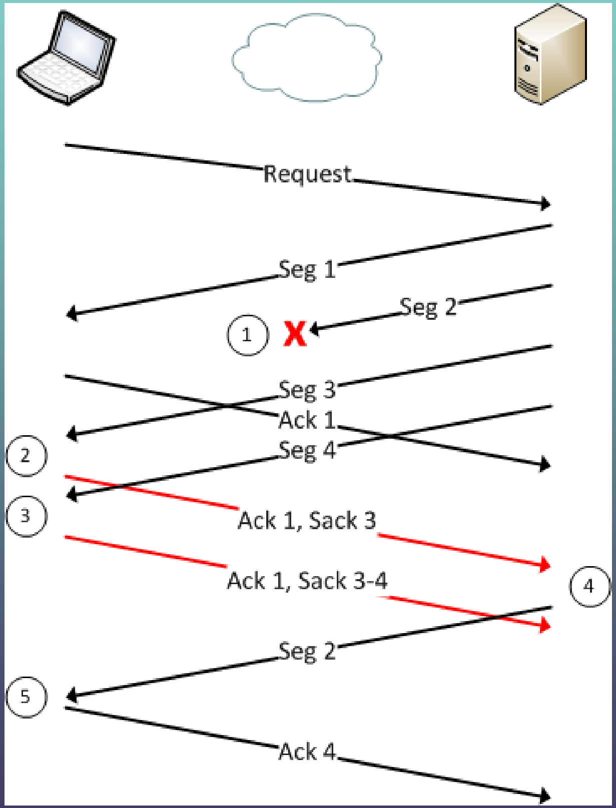
- Recall that *N* is the predicted number of packets that can be transmitted simultaneously through all the **other** subflows
- Path condition changes quickly
  - High prediction error will accumulate rapidly
- The scheduling value (*N*) is adjusted
  - Minimize deviations between the predicted and the **actual** values.
- **Actual** values are determined by an additional TCP option, TCP selective acknowledgment (SACK).

$$N_i'(n) = N_i(n) + \delta_i(n)$$

$$\delta_i(n+1) = \lfloor \delta_i(n) + \theta \cdot \sigma_i(n) \rfloor$$

TCP SACK

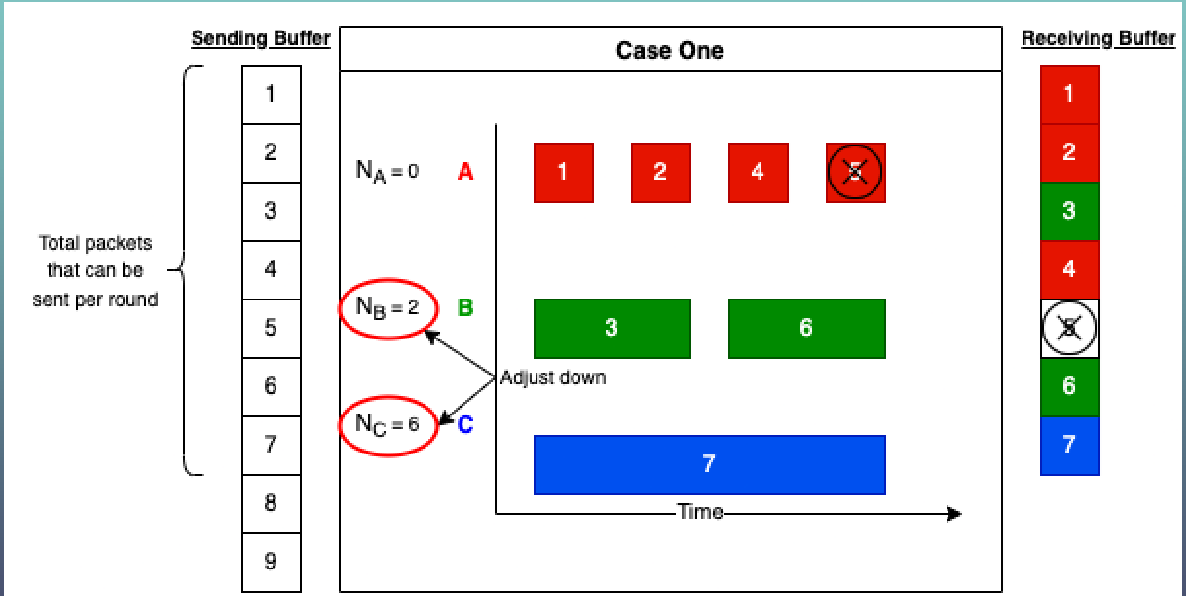
- **TCP SACK-** resend <u>only missing data</u>
- Packet specific
- Detects "**holes**" in the receiving buffer
  - Missing DSNs that were mapped to SACK SSN



Stage Two: Case One

- Subflow A is **slower**
- Packets **arrive later**
- Scheduling value **decreased**

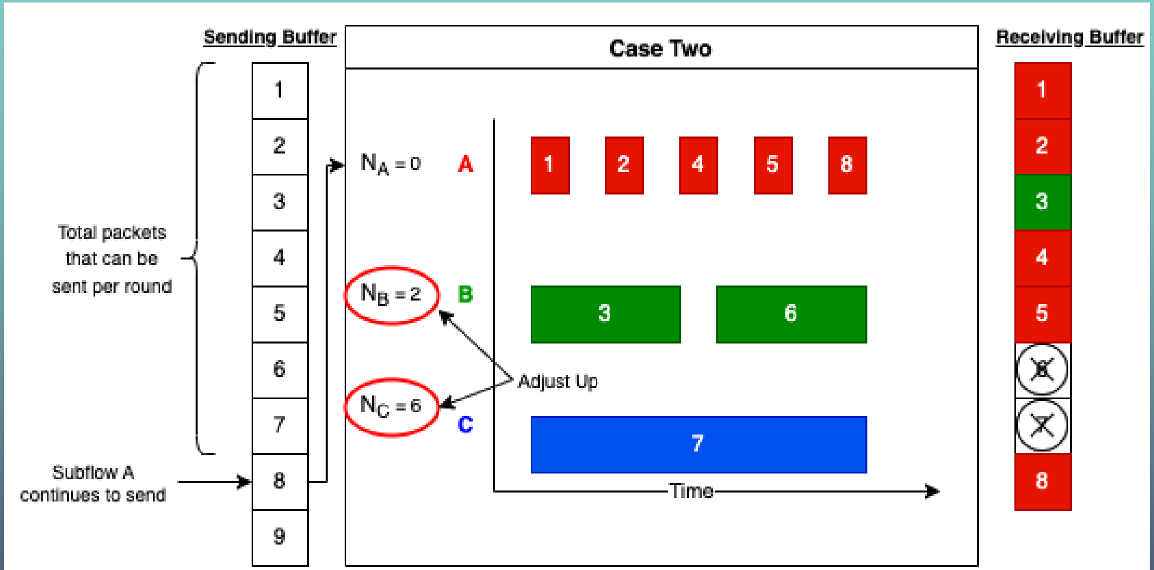
$$\sigma_i(n) \leftarrow \sigma_i(n) - \sum_{\substack{1 \le k \le P \\ RTT_k < RTT_i}} h_{i,k}(n)$$



Stage Two: Case Two

- Subflow A is **faster**
- Packets **arrive earlier**
- Scheduling value **increased**

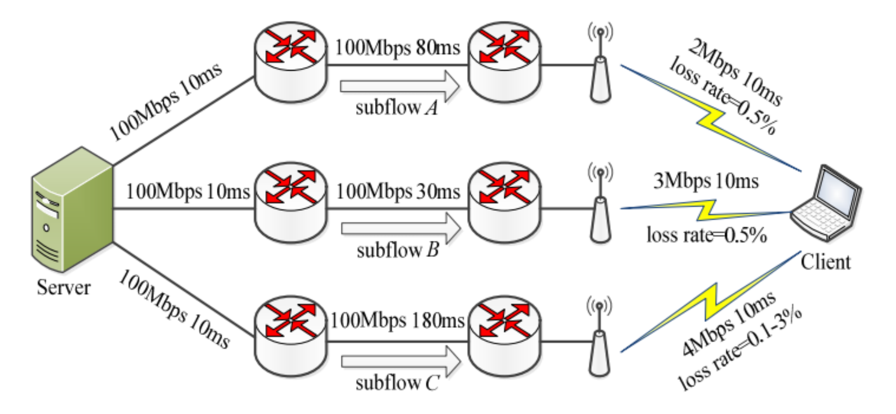
$$\sigma_i(n) \leftarrow \sigma_i(n) + \sum_{\substack{1 \le k \le P \\ RTT_m < RTT_i}} h_{i,m}(n)$$



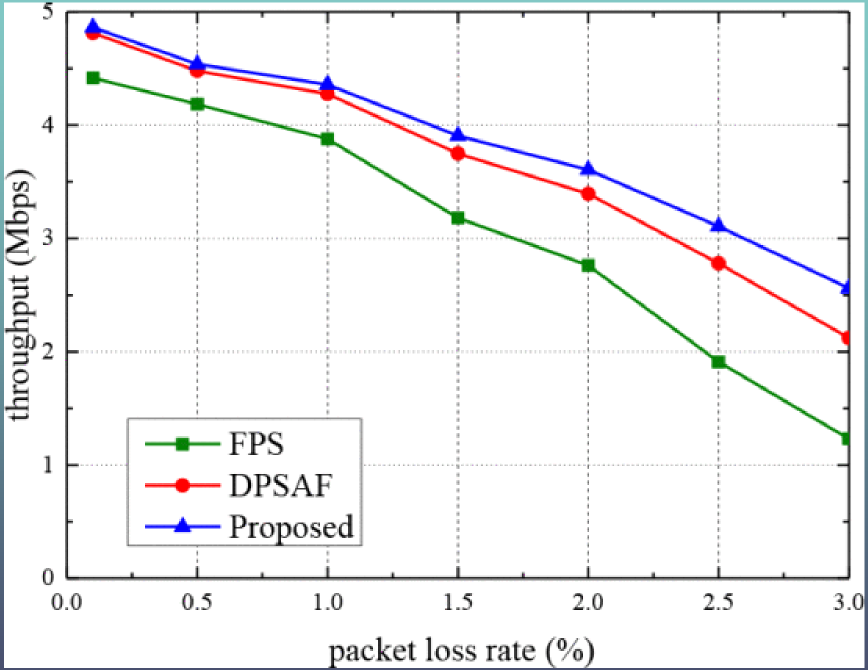
Results

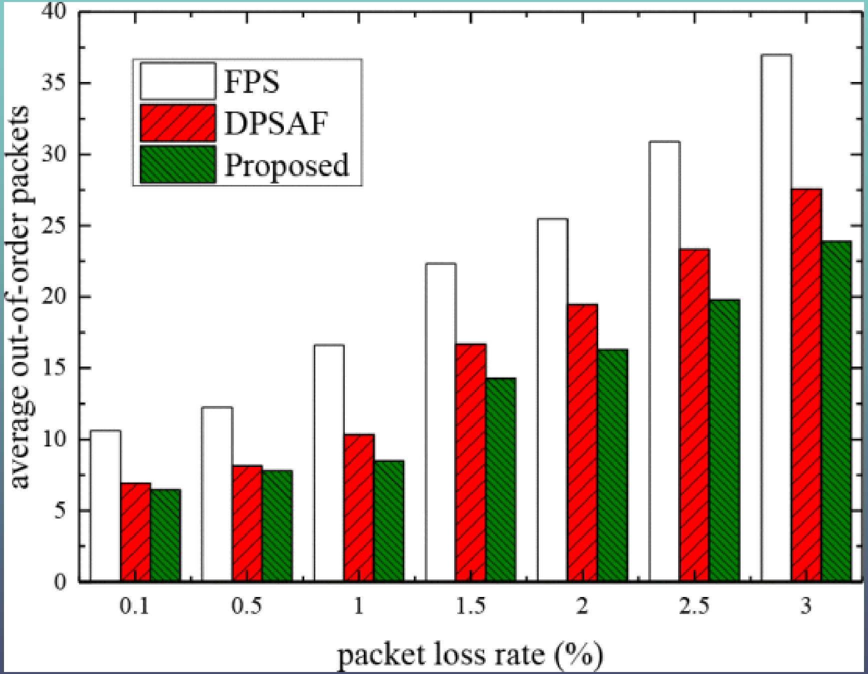
Evaluation Method

- Authors used ns-3 network simulator software to evaluate the performance
- Manipulated subflow C
  - Packet loss rate from 0.1% to 3%
  - Changed receiving buffer size from 16KB to 512KB
- Compared the algorithm with
  - Forward Prediction Scheduling (FPS)
  - Dynamic Packet Scheduling and Adjusting with Feedback (DPSAF)

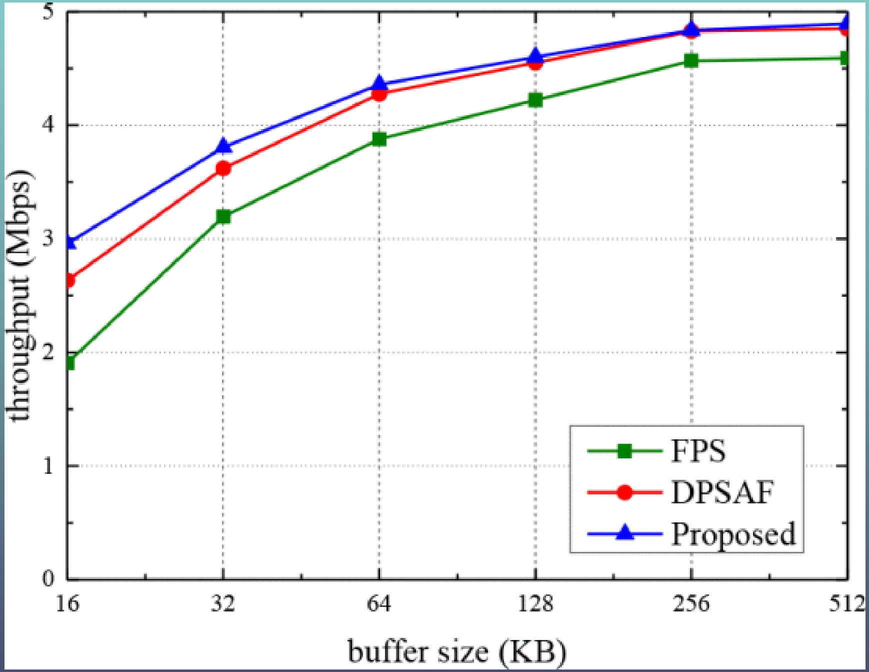


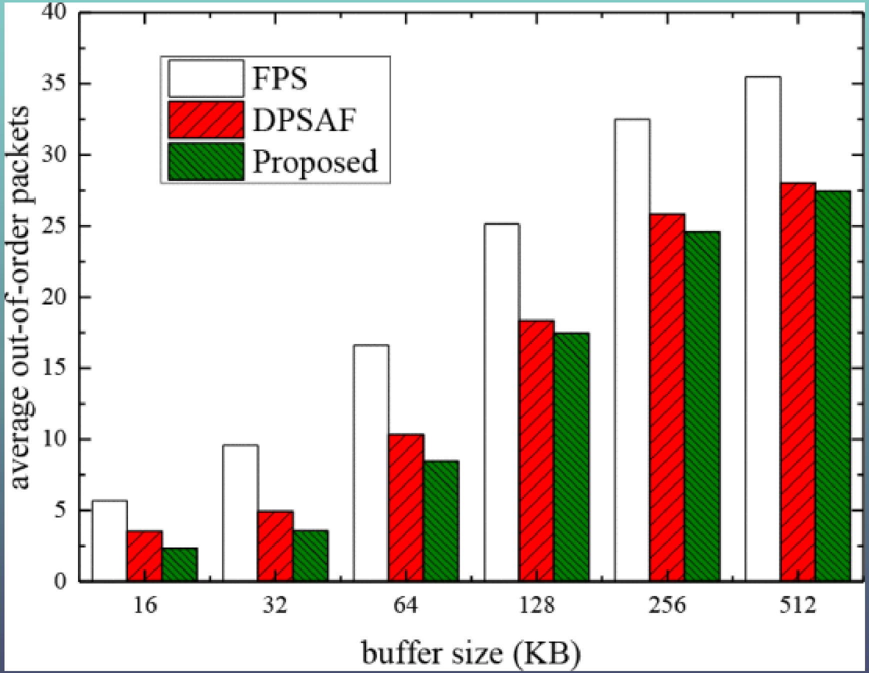
Packet loss rate from 0.1% to 3%





Increase Buffer Size





Conclusion

- Fault tolerant connections
- Levages all devices network connection
- Backward compatible with TCP mean MPTCP will be widely adopted
- Performance improvements research is active
- Performance improvement can be made to the existing scheduling algorithm implementations

Questions?

References

1. Mark ; Raiciu Costin Bonaventure, Olivier ; Handley. 2012. An Overview of Multipath TCP. login: 37, 5 (2012), 17–23. http://hdl.handle.net/2078.1/114081
2. Pingping Dong, Jingyun Xie, Wensheng Tang, Naixue Xiong, Hua Zhong, and Athanasios V. Vasilakos. 2019. Performance Evaluation of Multipath TCP Scheduling Algorithms. IEEE Access 7 (2019), 29818–29825. https://doi.org/10.1109/ACCESS.2019.2898110
3. Bruno Y. L. Kimura, Demetrius C. S. F. Lima, and Antonio A. F. Loureiro. 2021. Packet Scheduling in Multipath TCP: Fundamentals, Lessons, and Opportunities. IEEE Systems Journal 15, 1 (2021), 1445–1457. https://doi.org/10.1109/JSYST.2020.2965471
4. W. Richard Stevens and Kevin R Fall. 2011. TCP/IP Illustrated: The Protocols (2 ed.). Vol. 1. Addison-Wesley Professional, Upper Saddle River, NJ, USA.
5. Guannan Xie, Huifang Chen, Lei Xie, and Kuang Wang. 2018. A Dynamic Packet Scheduling Method for Multipath TCP in Heterogeneous Wireless Networks. In 2018 IEEE 18th International Conference on Communication Technology (ICCT). 678–682. https://doi.org/10.1109/ICCT. 2018.860017