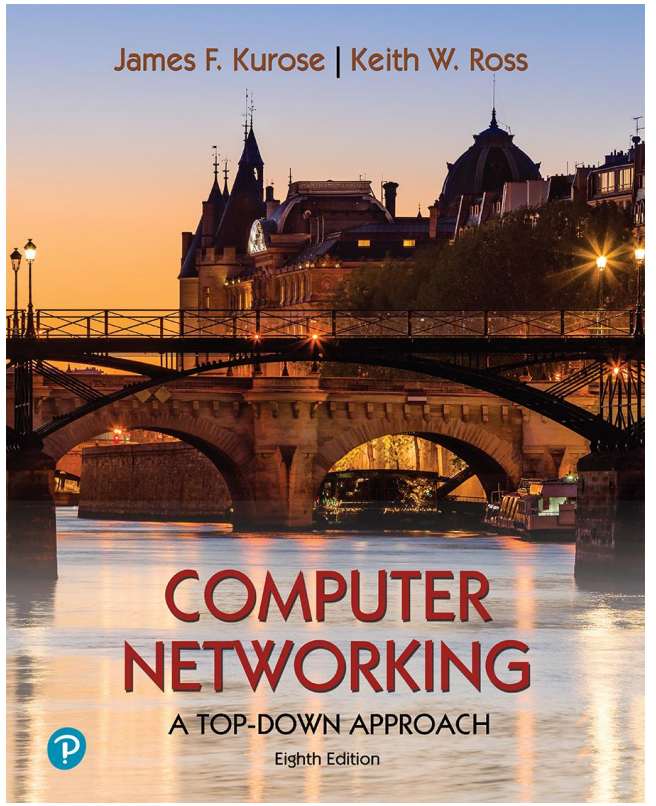
Chapter 3
Transport Layer

Yaxiong Xie

Department of Computer Science and Engineering
University at Buffalo, SUNY



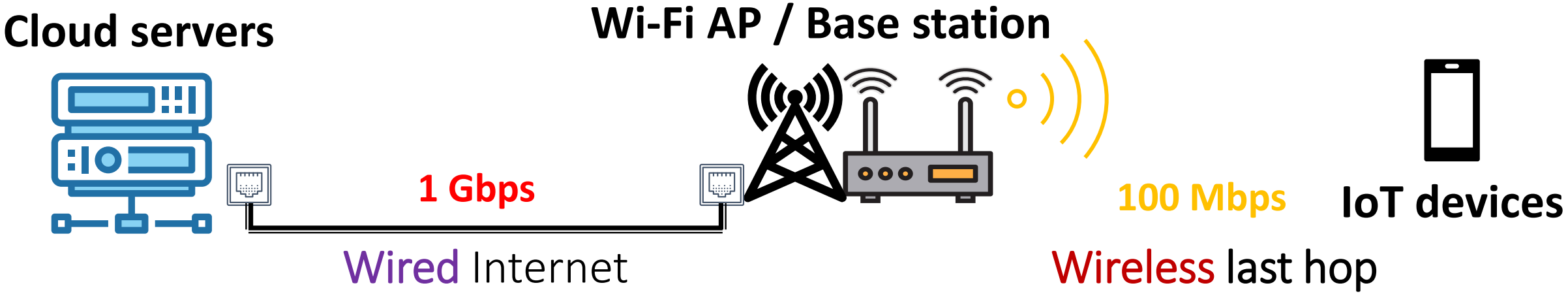
Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

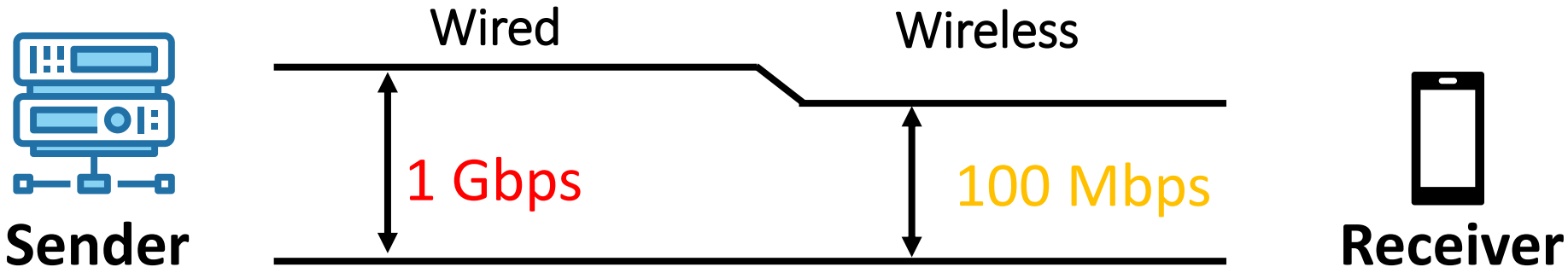
Adapted from the slides of the book's authors

Chapter 3: roadmap

- Transport-layer services
- Multiplexing and demultiplexing
- Connectionless transport: UDP
- Principles of reliable data transfer
- Connection-oriented transport: TCP
- Principles of congestion control
- TCP congestion control
- Evolution of transport-layer functionality

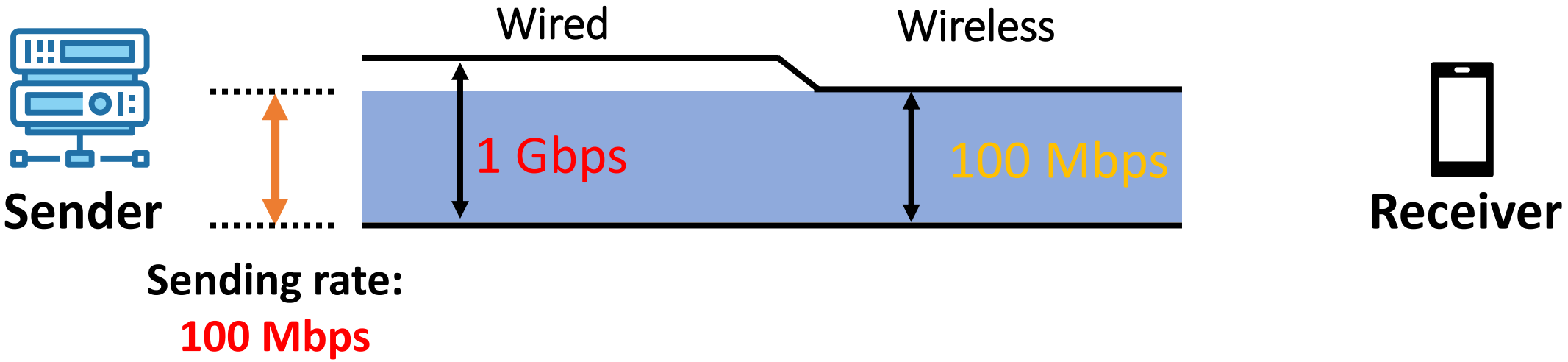
A wireless connection consists a wired Internet hop and a wireless last hop





Throughput: 100 Mbps

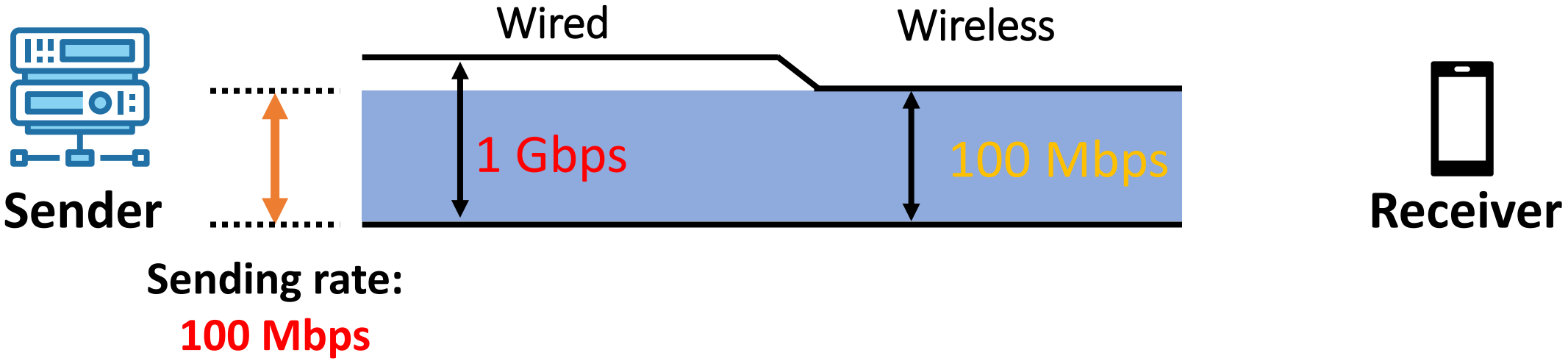
Congestion Control: Overview



Step 1: Measure the throughput of the connection

Step 2: Match the sending rate to the throughput

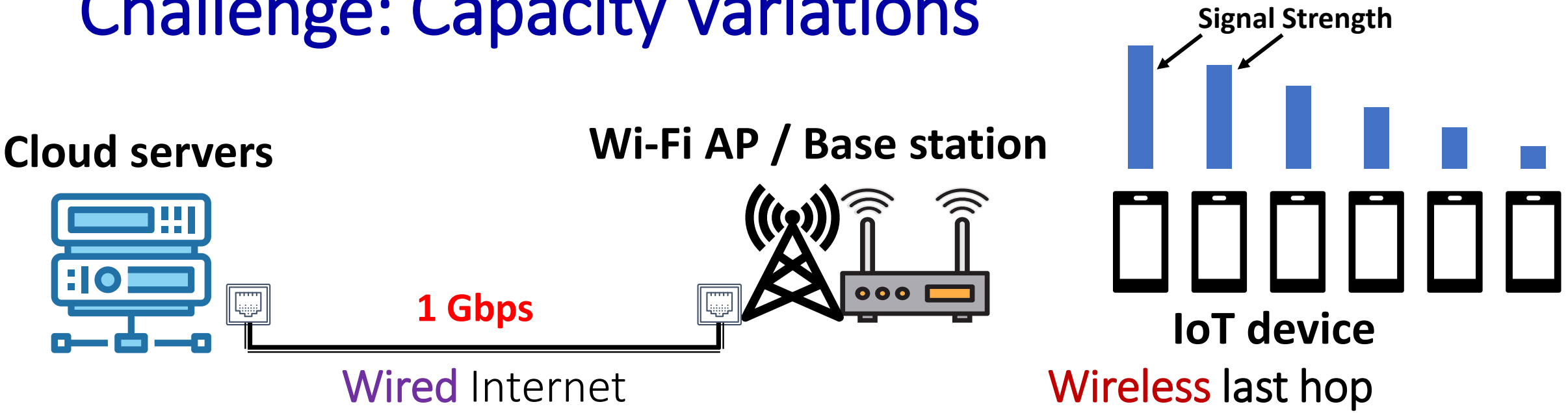
Challenge: Capacity Variations

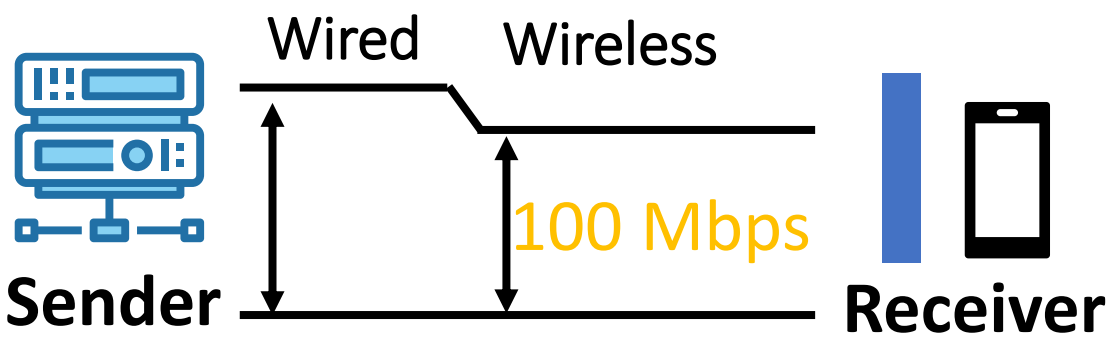


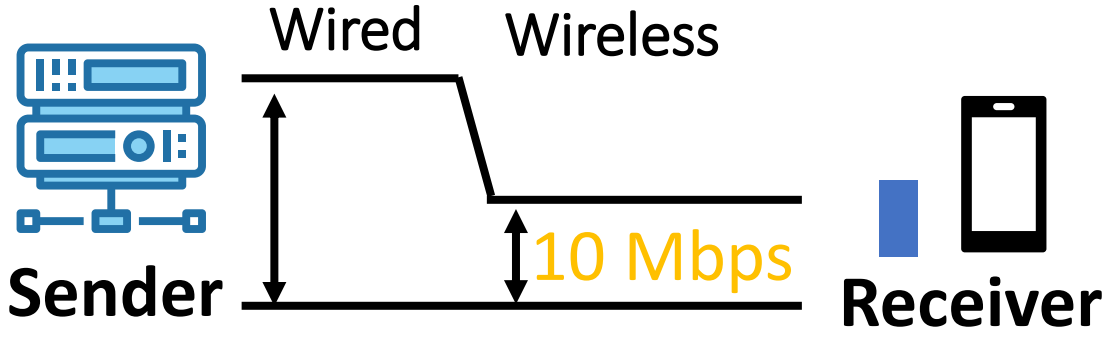
Step 1: Measure the throughput of the connection

Step 2: Match the sending rate to the throughput

Challenge: Capacity Variations

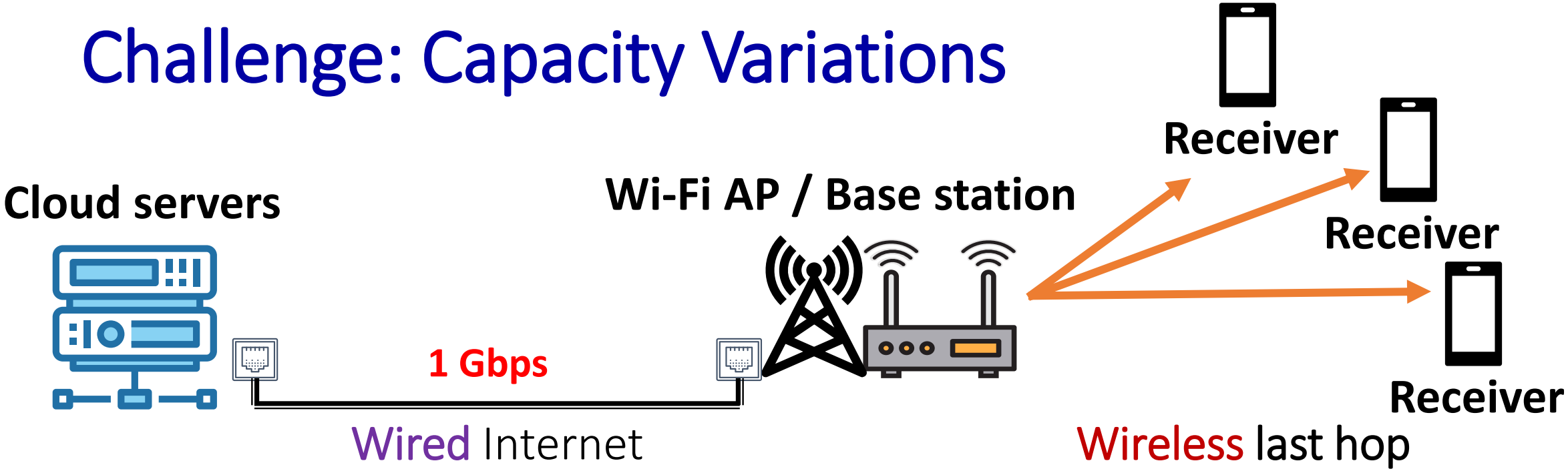


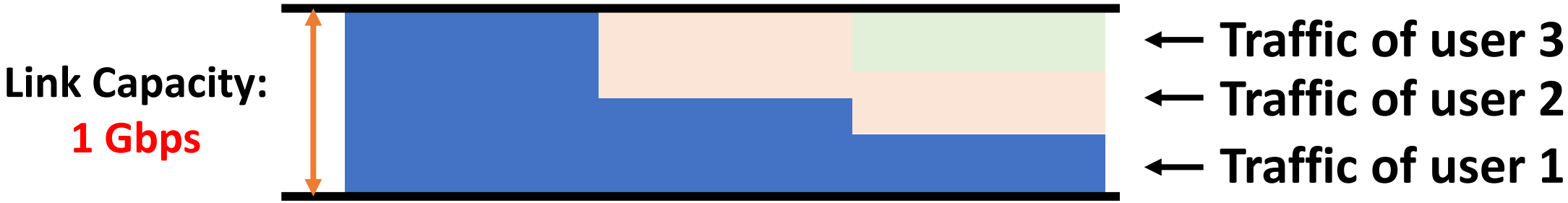




Mobility causes capacity variations

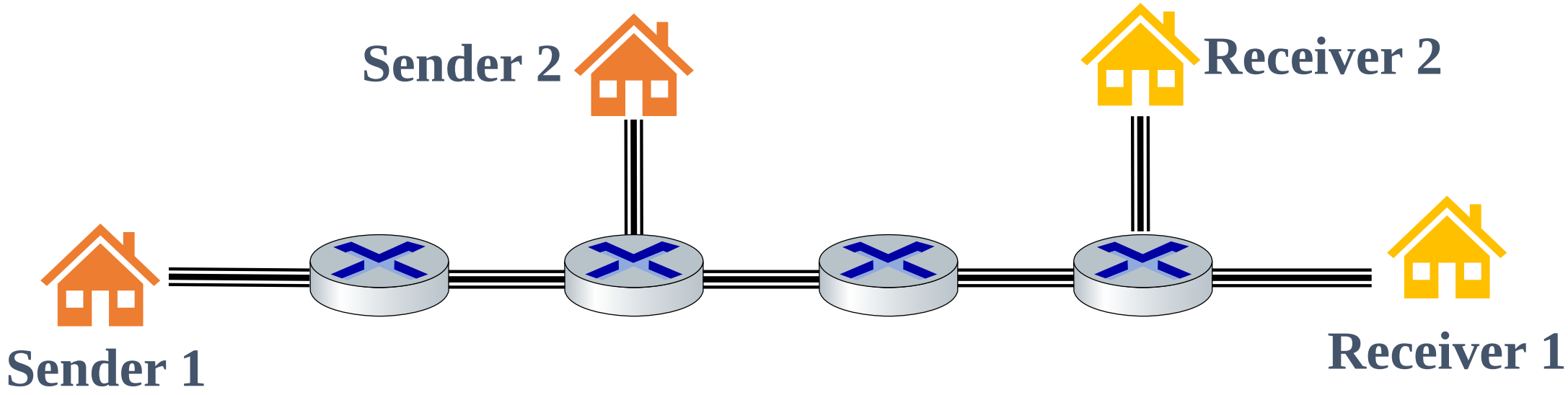
Challenge: Capacity Variations





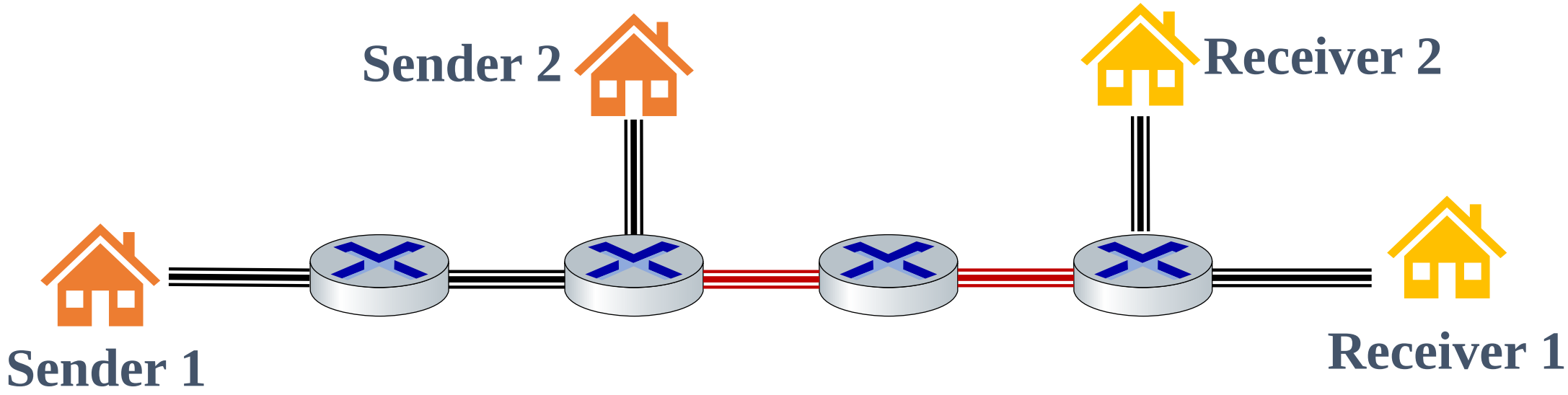
Competition between connections causes capacity variations

Challenge: Capacity Variations



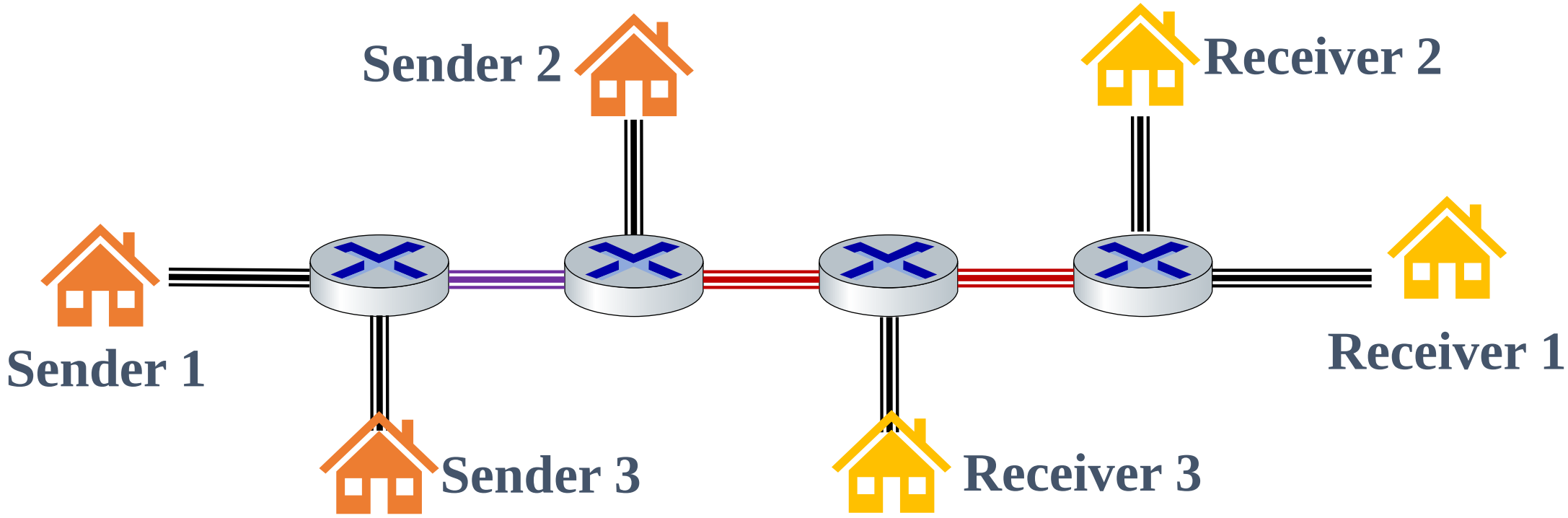
Competition between connections causes capacity variations

Challenge: Capacity Variations



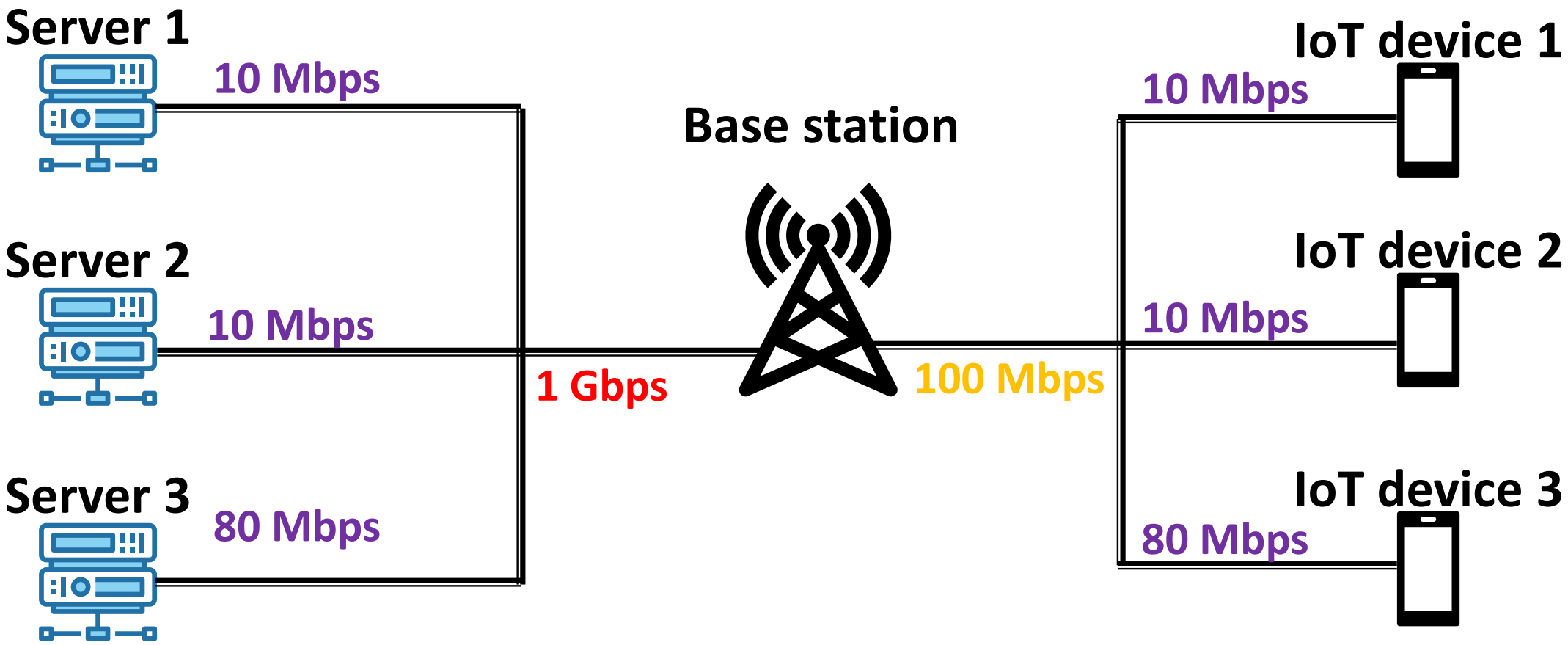
Competition between connections causes capacity variations

Challenge: Capacity Variations



Competition between connections causes capacity variations

Challenge: Fairness

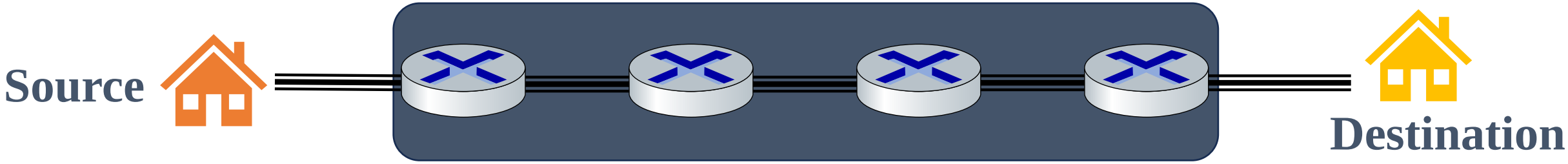


Congestion control must guarantee fairness between connections

Approaches towards congestion control

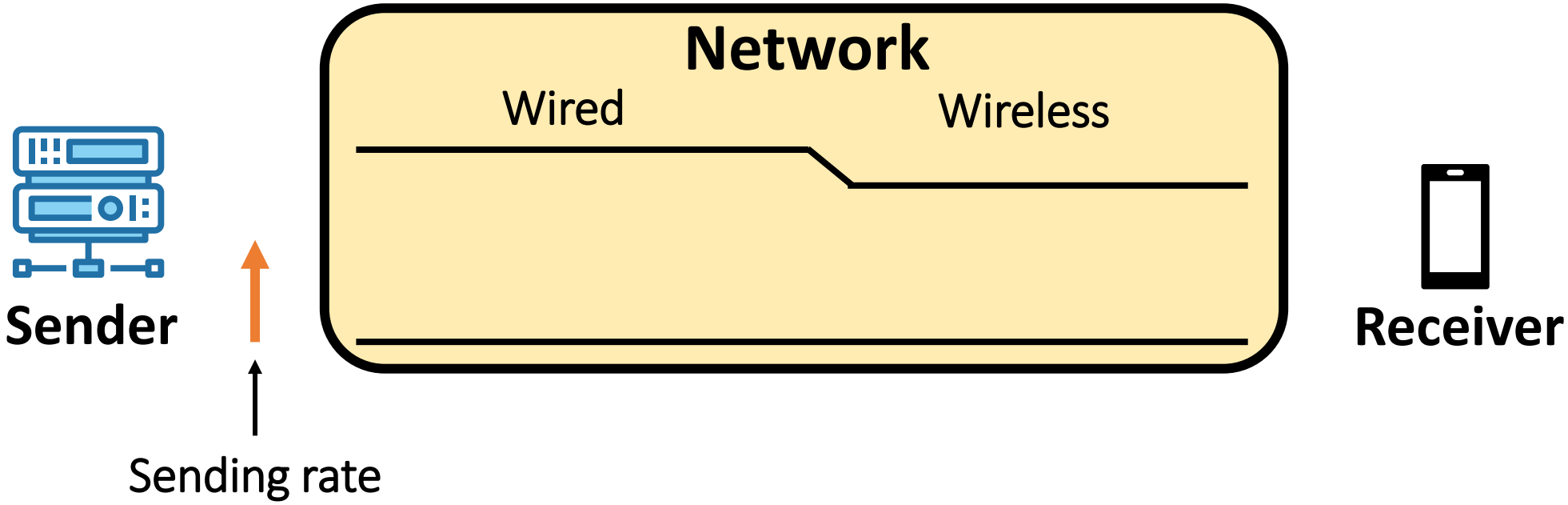
End-end congestion control:

- no explicit feedback from network
- congestion *inferred* from observed loss, delay
- approach taken by TCP

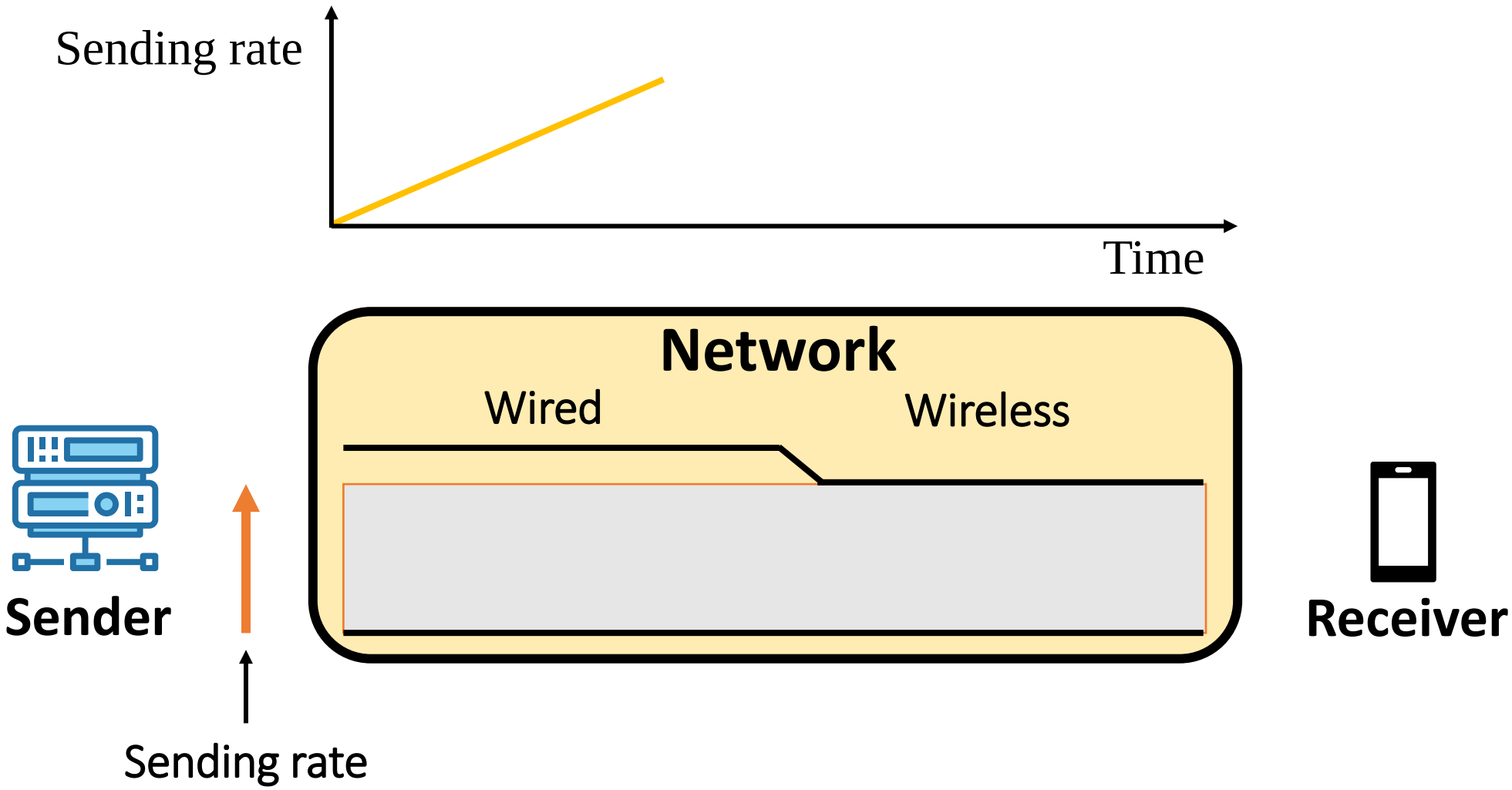


Treating network as a black box

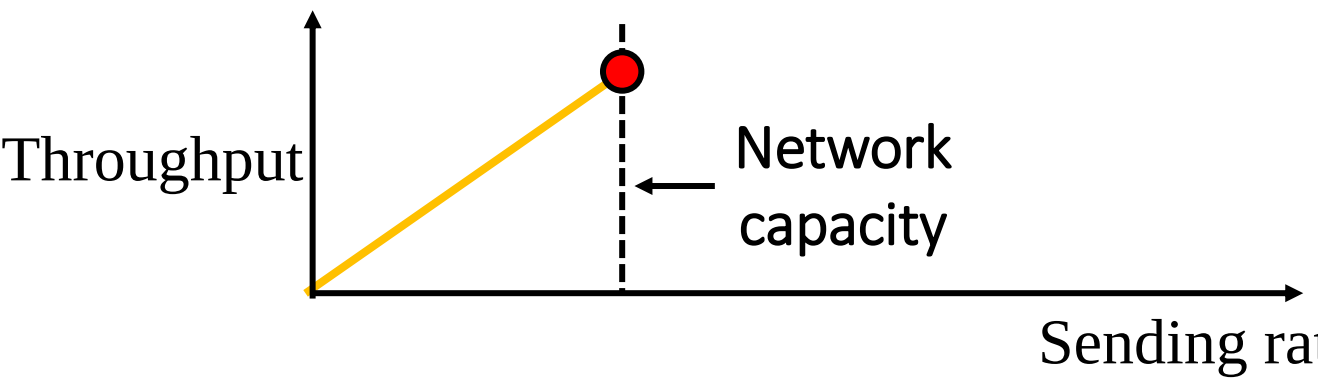
Pure end-to-end solution without cooperation from the network: probing for the capacity

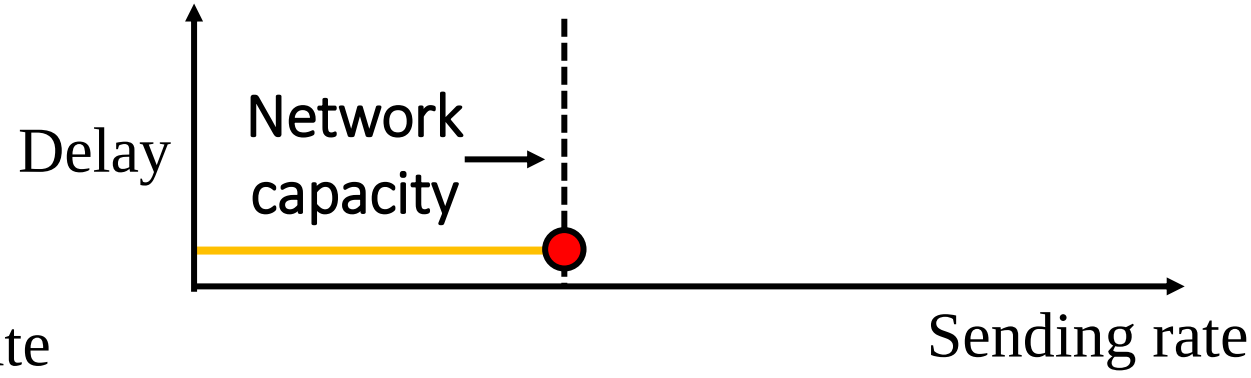


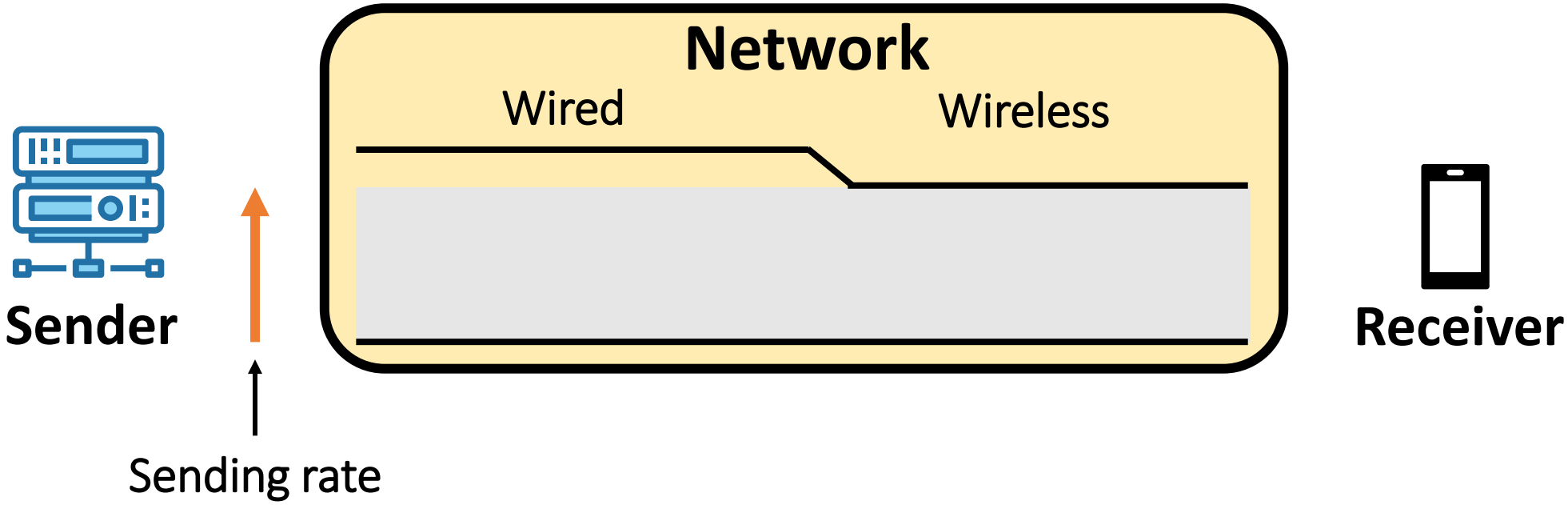
Pure end-to-end solution without cooperation from the network: probing for the capacity



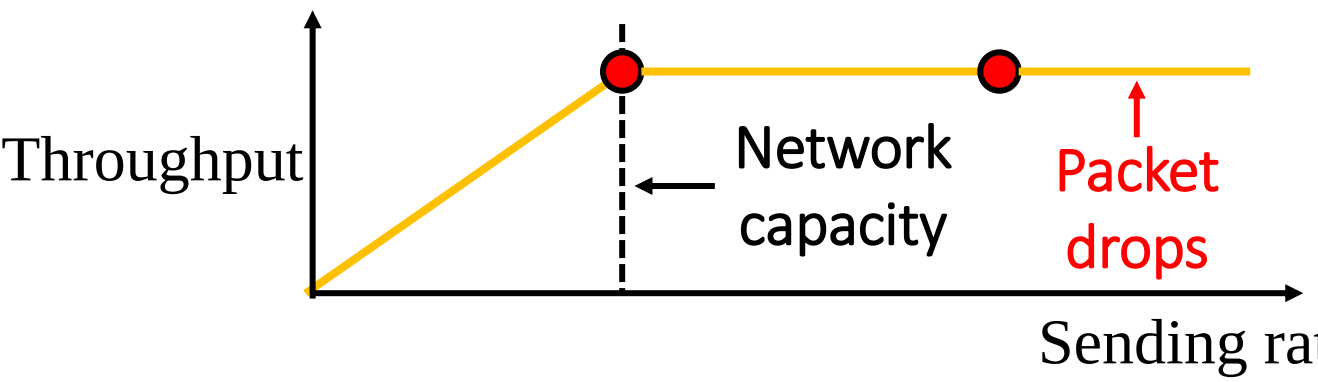
Pure end-to-end solution without cooperation from the network: probing for the capacity

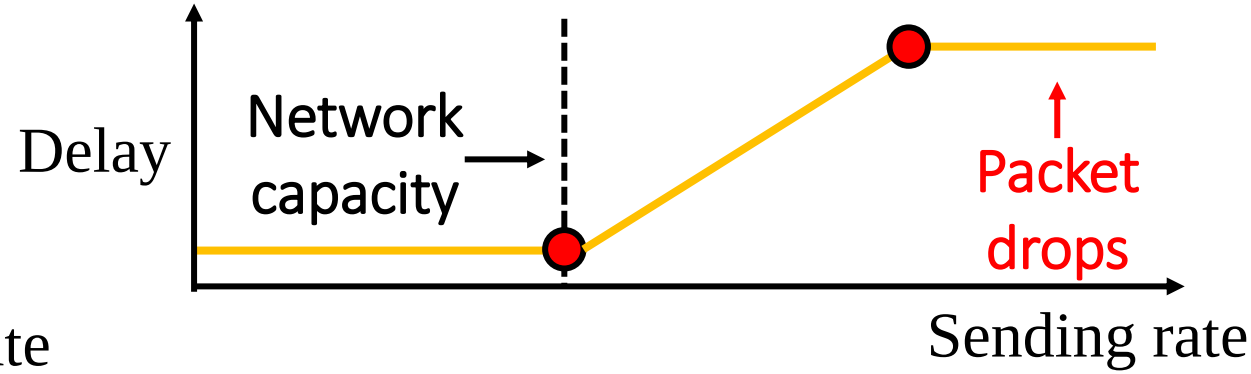


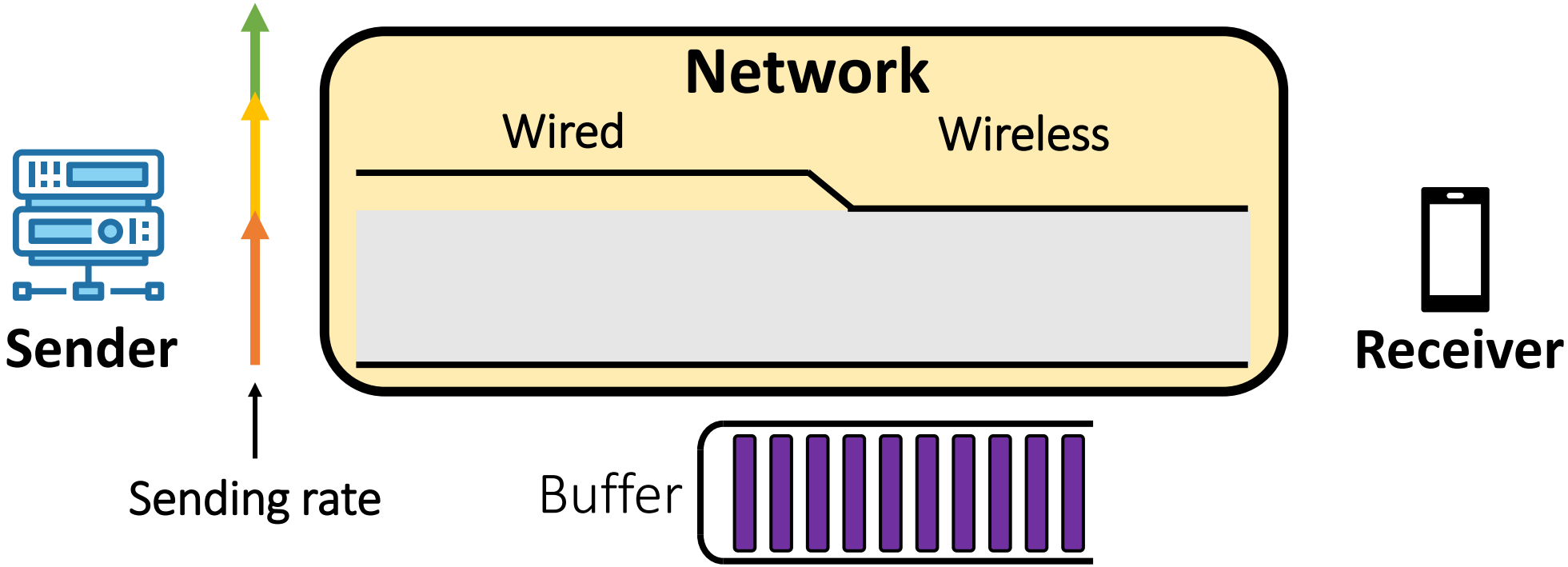




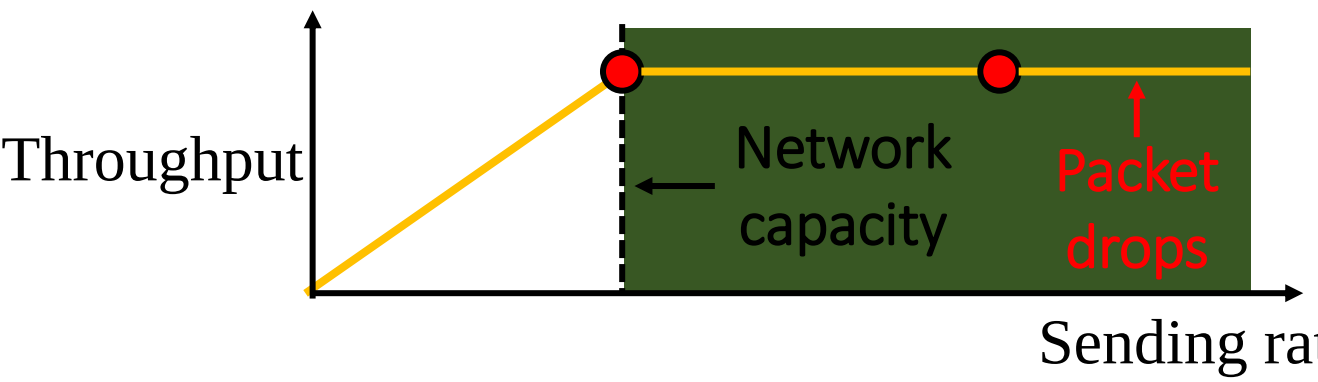
Pure end-to-end solution without cooperation from the network: probing for the capacity

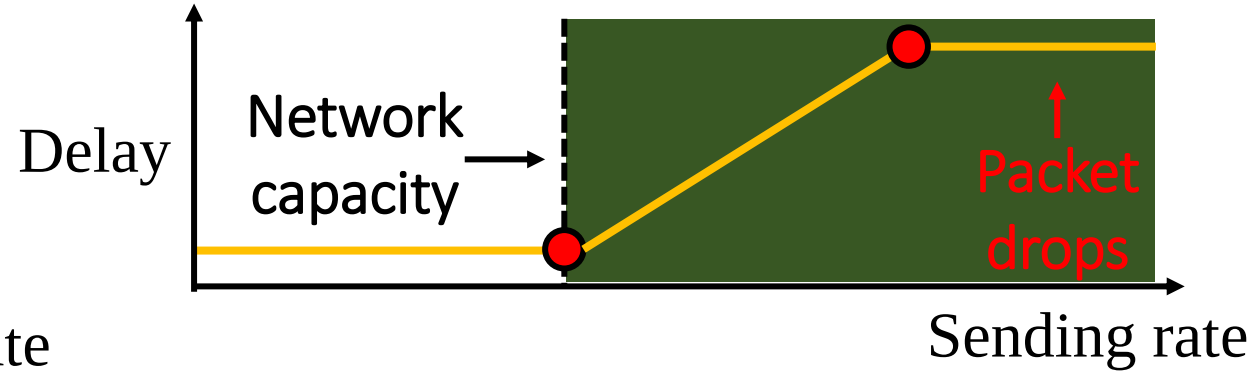


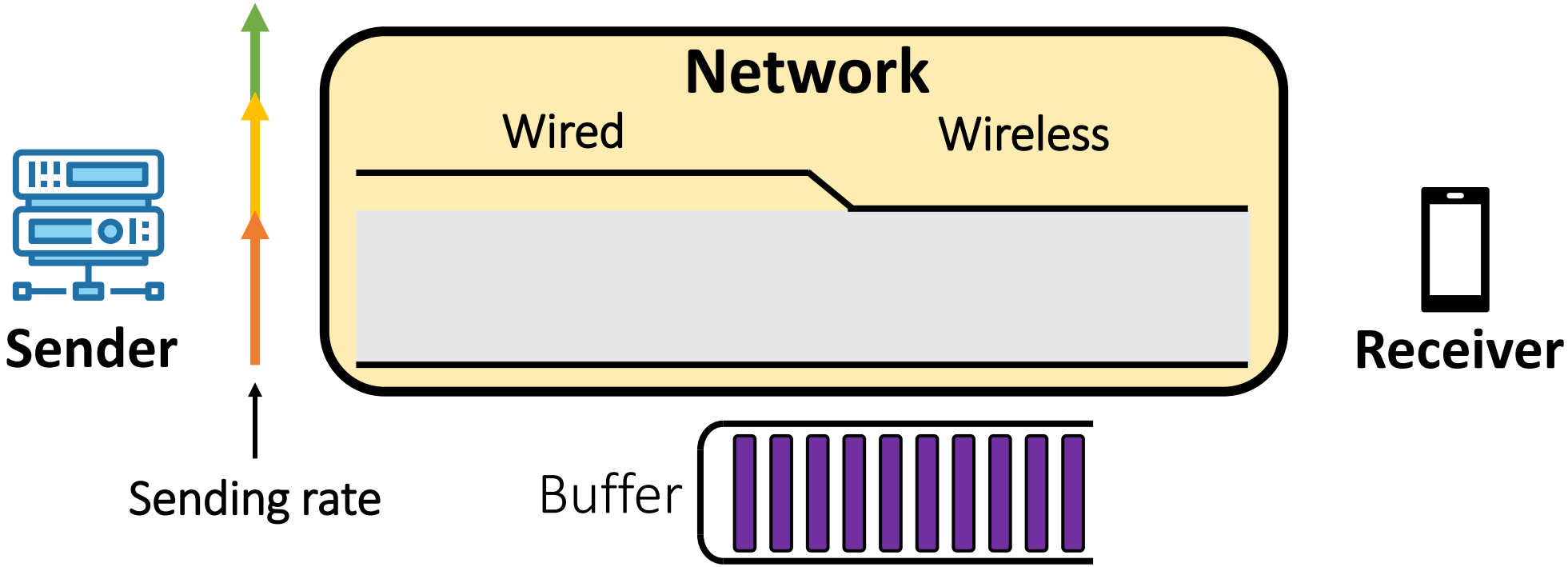




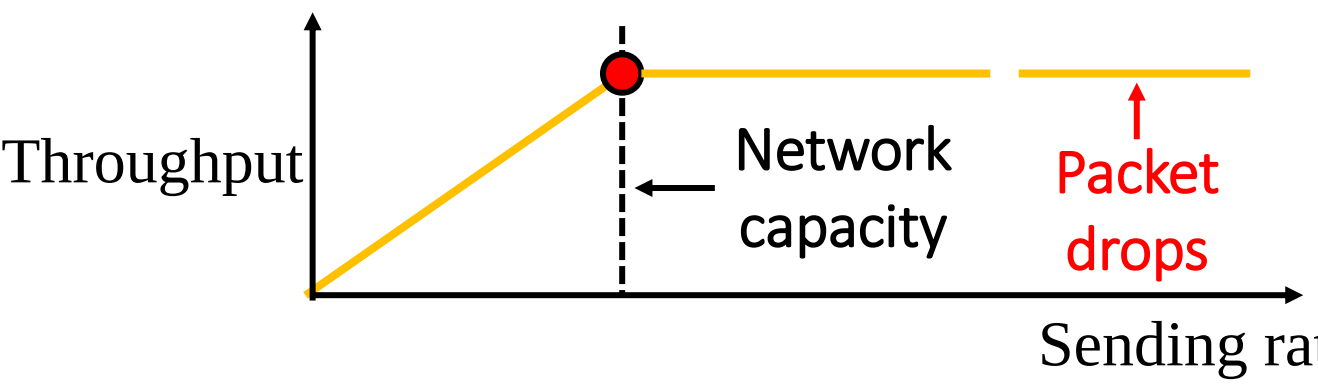
Pure end-to-end solution without cooperation from the network: probing for the capacity

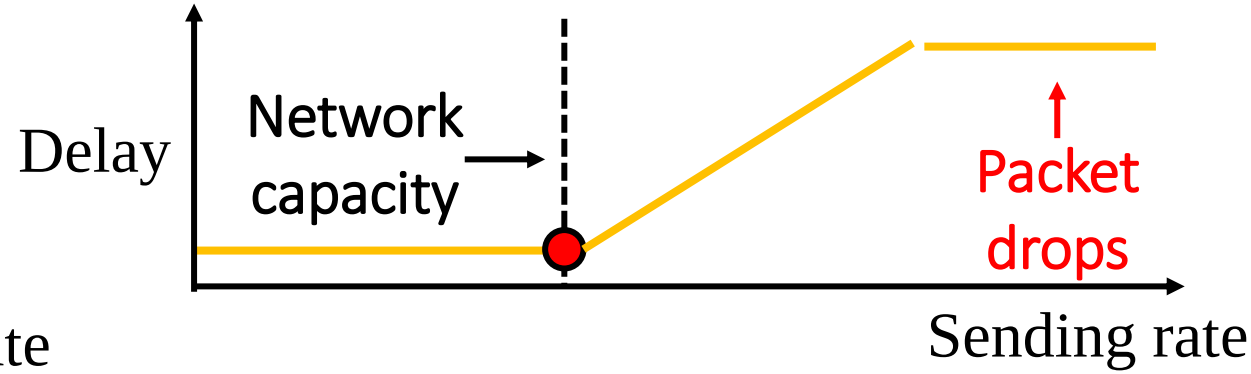


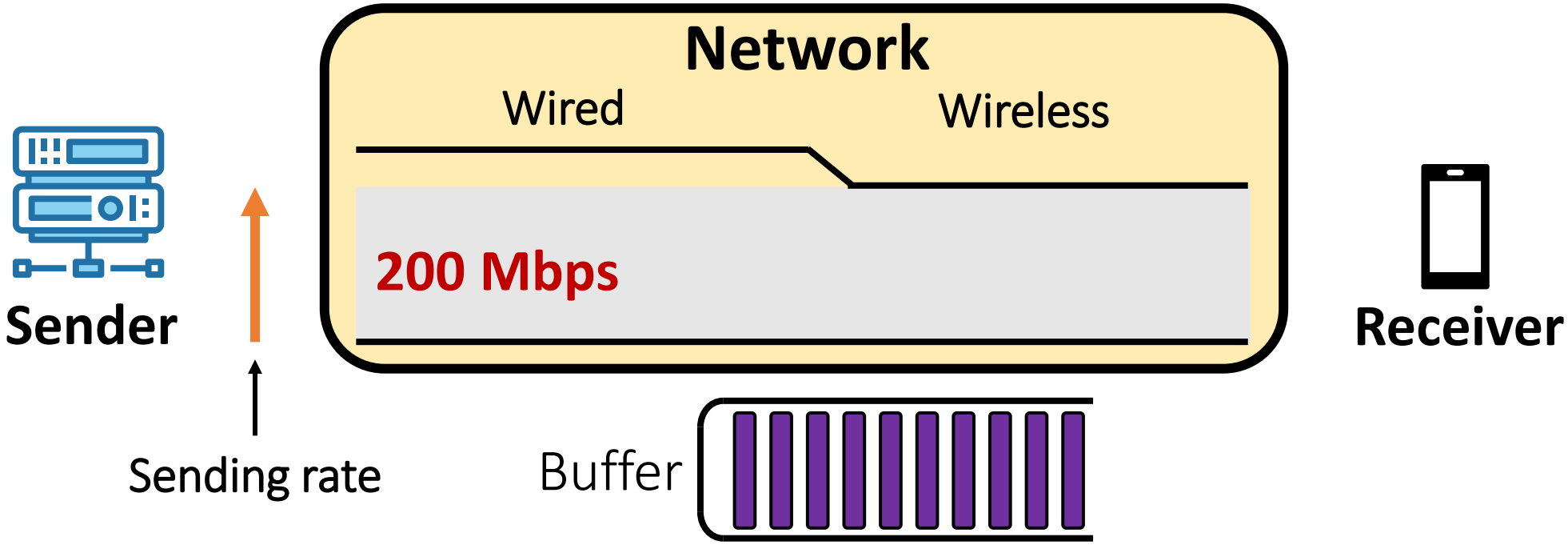




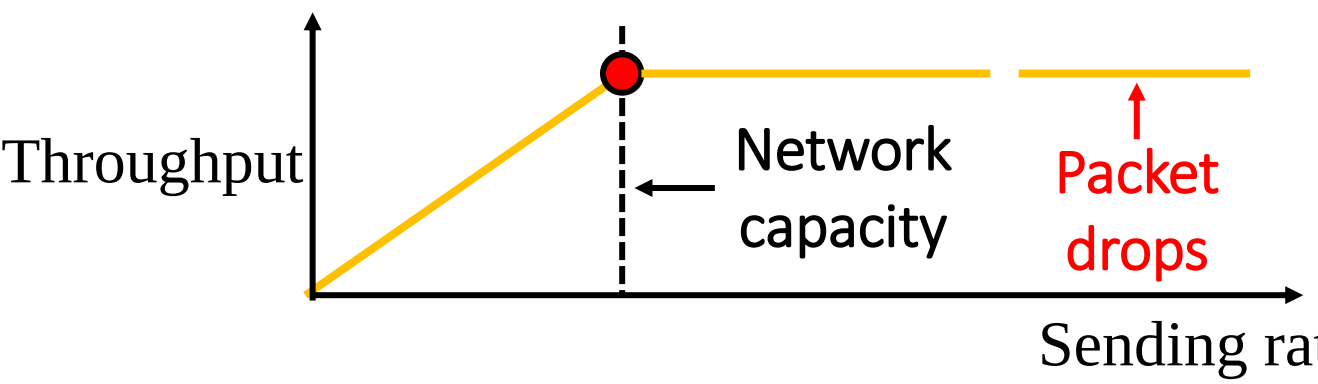
Pure end-to-end solution without cooperation from the network: probing for the capacity

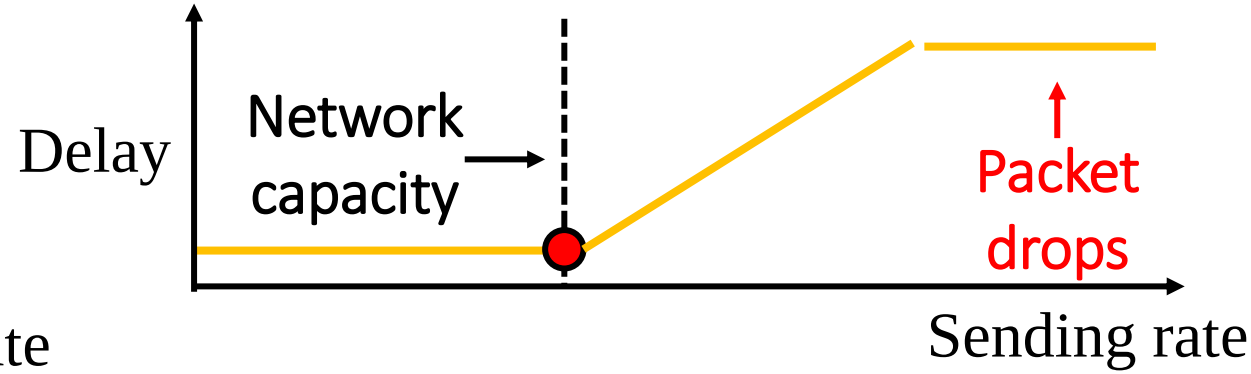


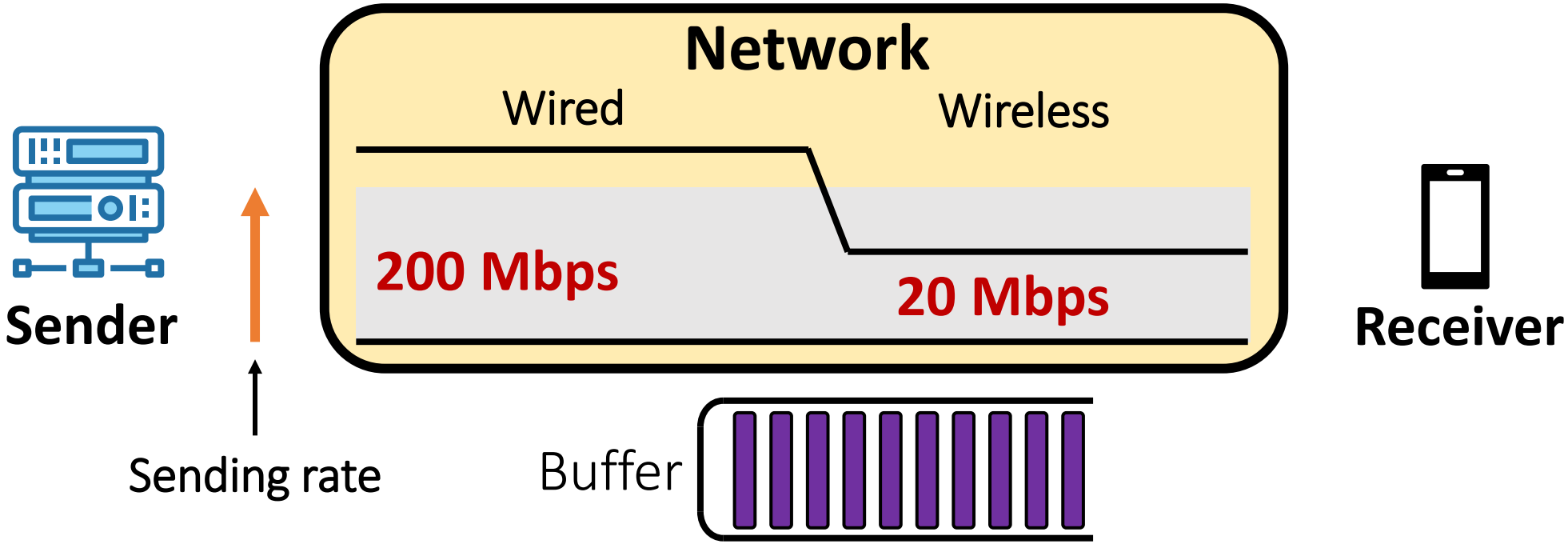




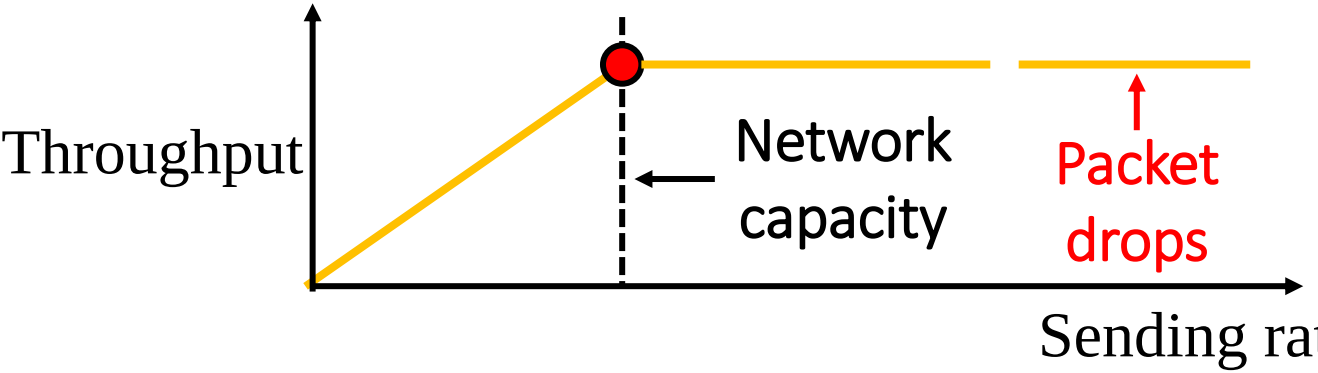
Pure end-to-end solution without cooperation from the network: probing for the capacity

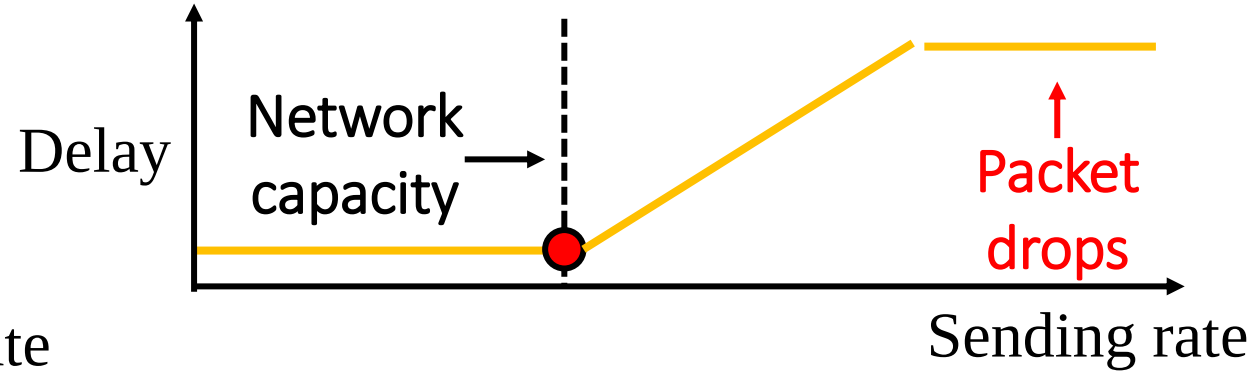


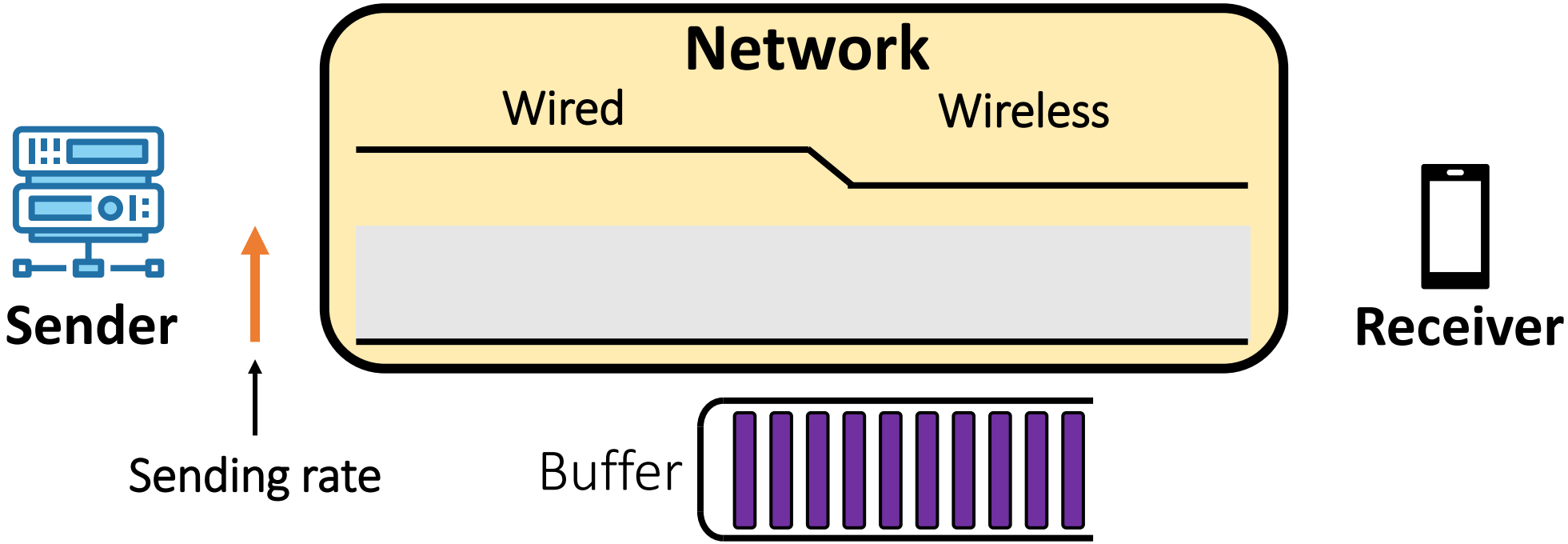




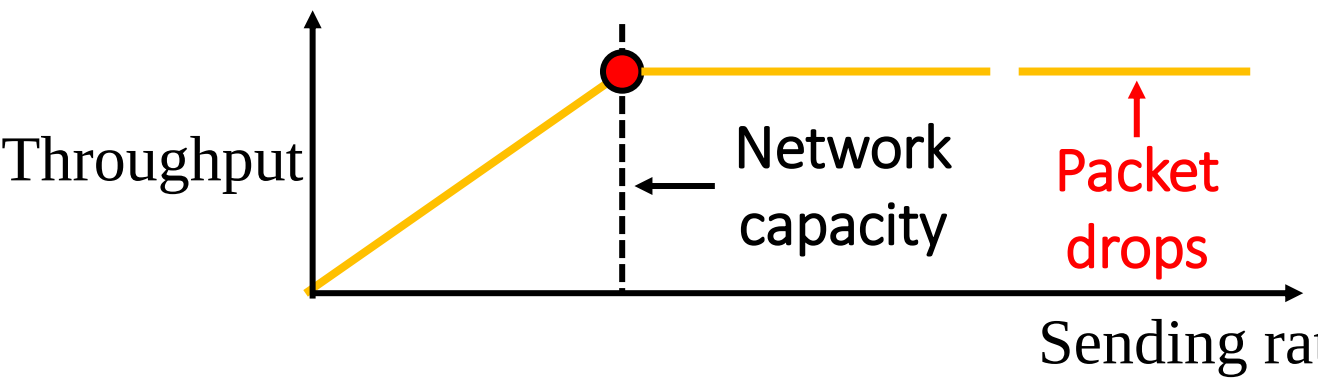
Pure end-to-end solution without cooperation from the network: continuously probing

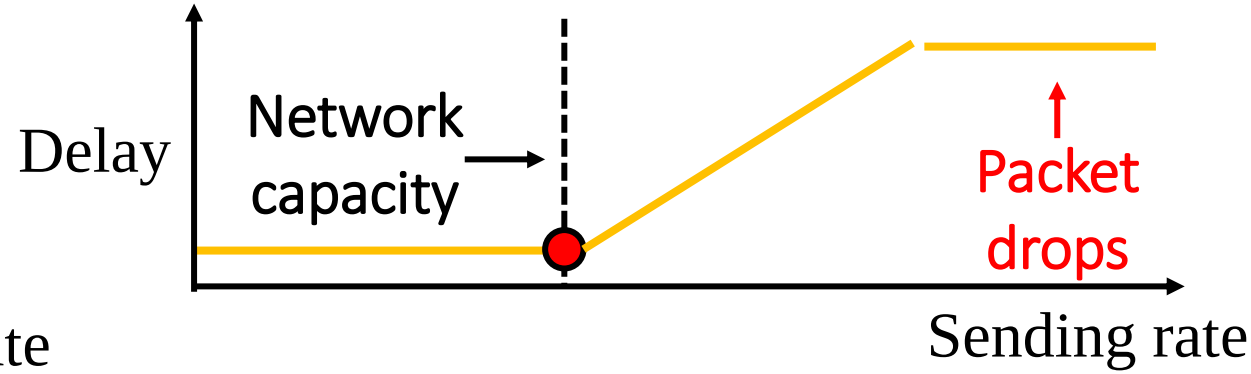


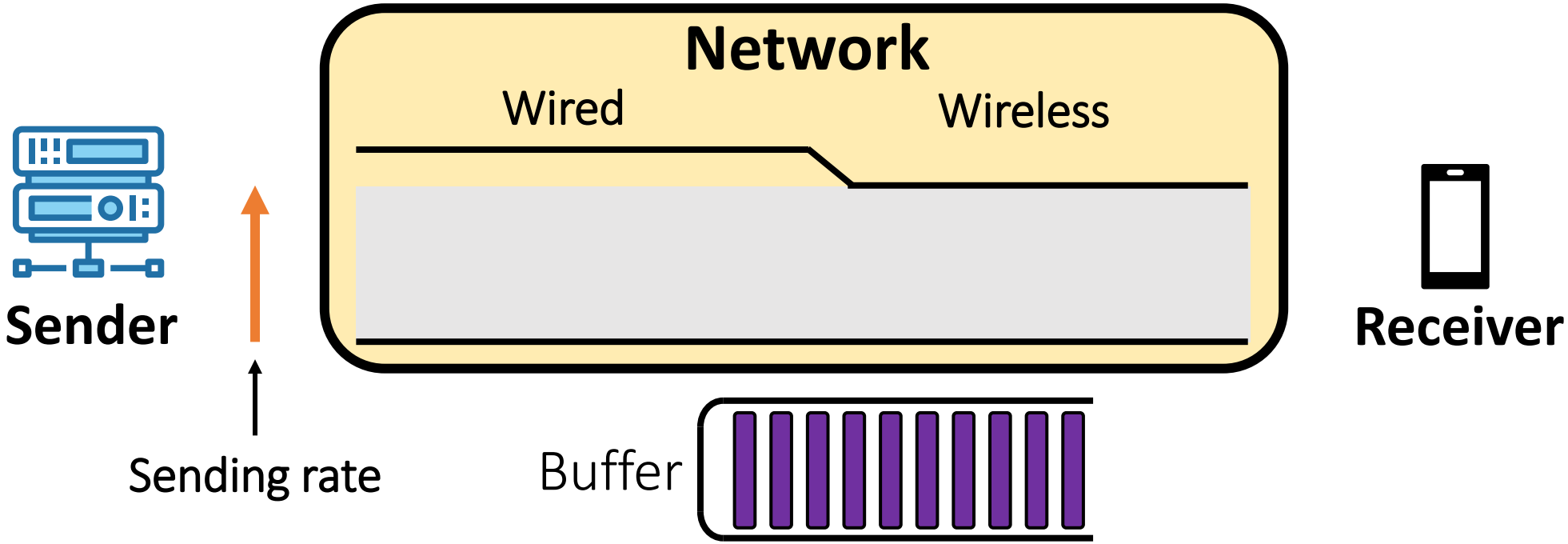




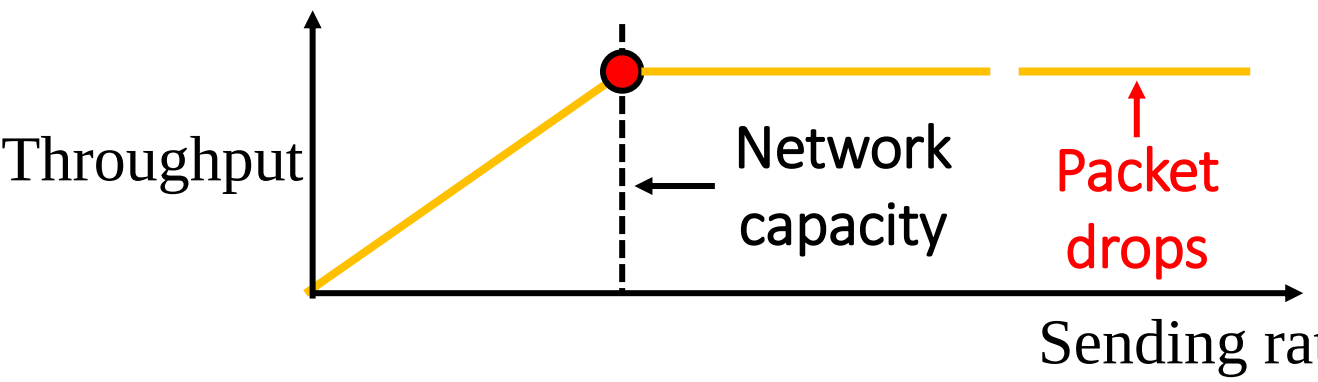
Pure end-to-end solution without cooperation from the network: continuously probing

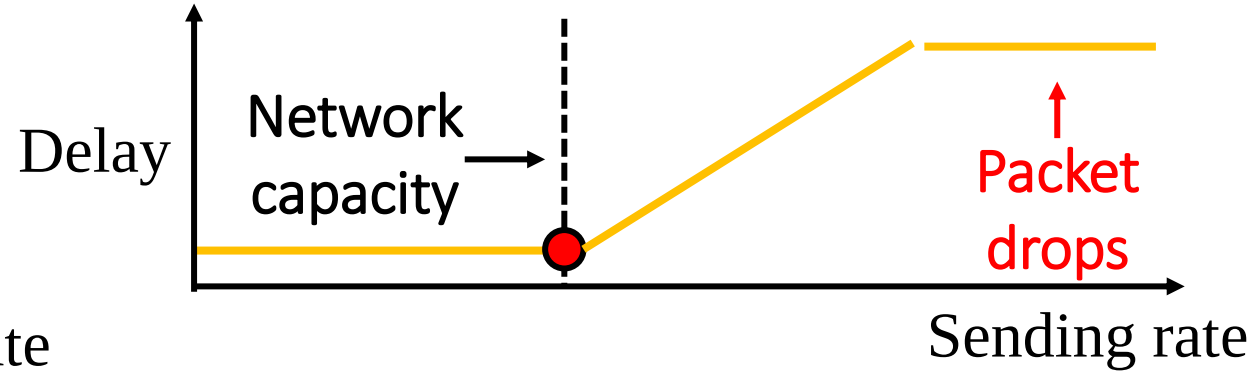


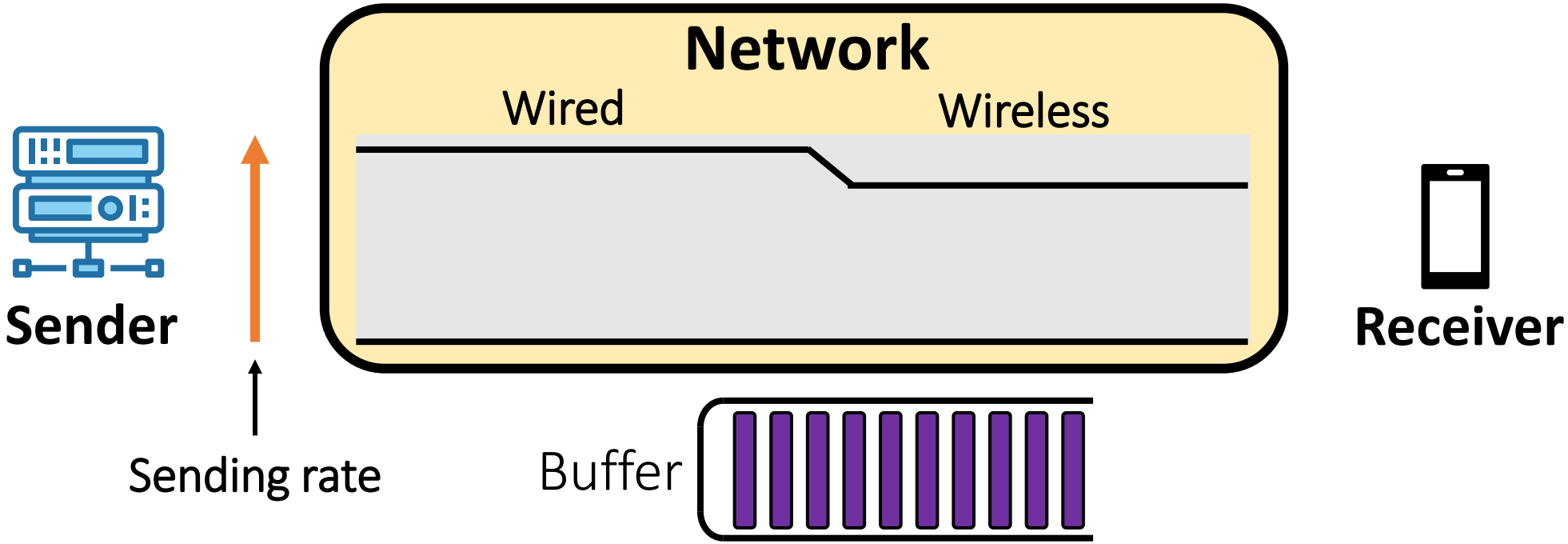




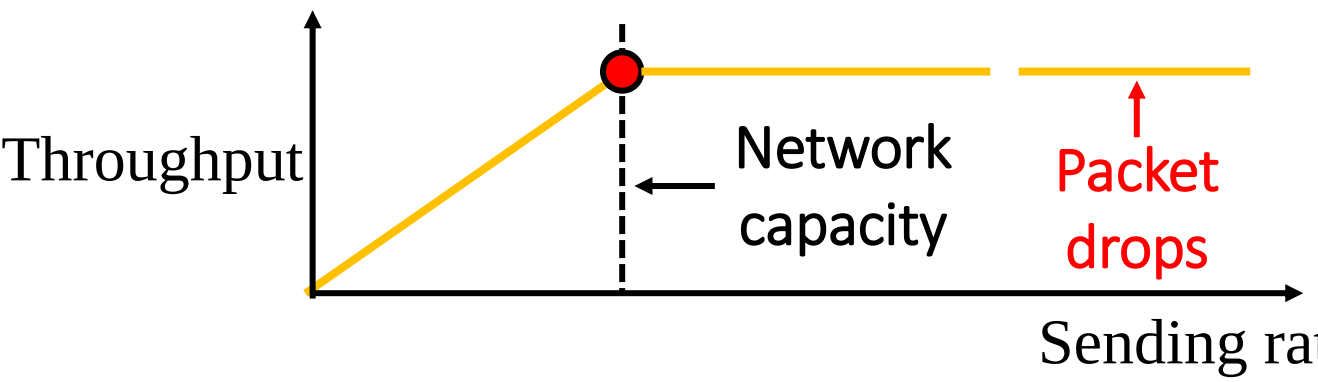
Pure end-to-end solution without cooperation from the network: continuously probing

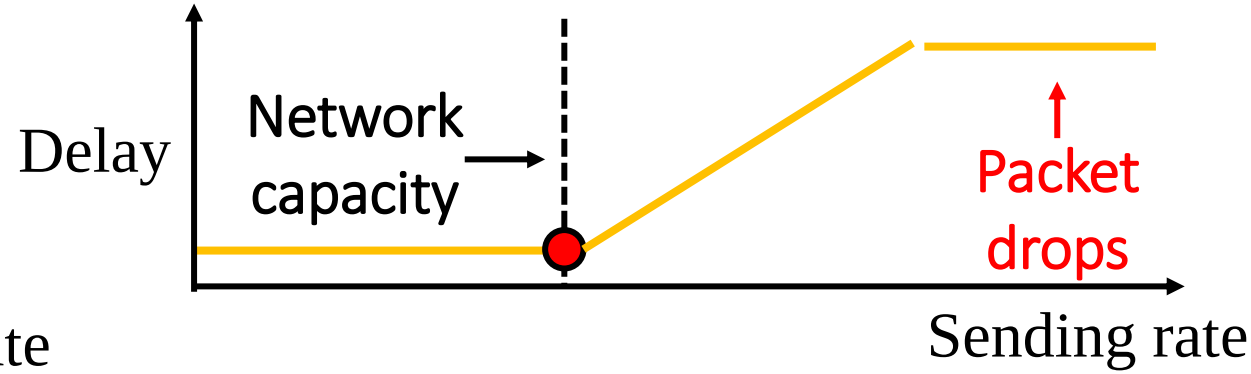


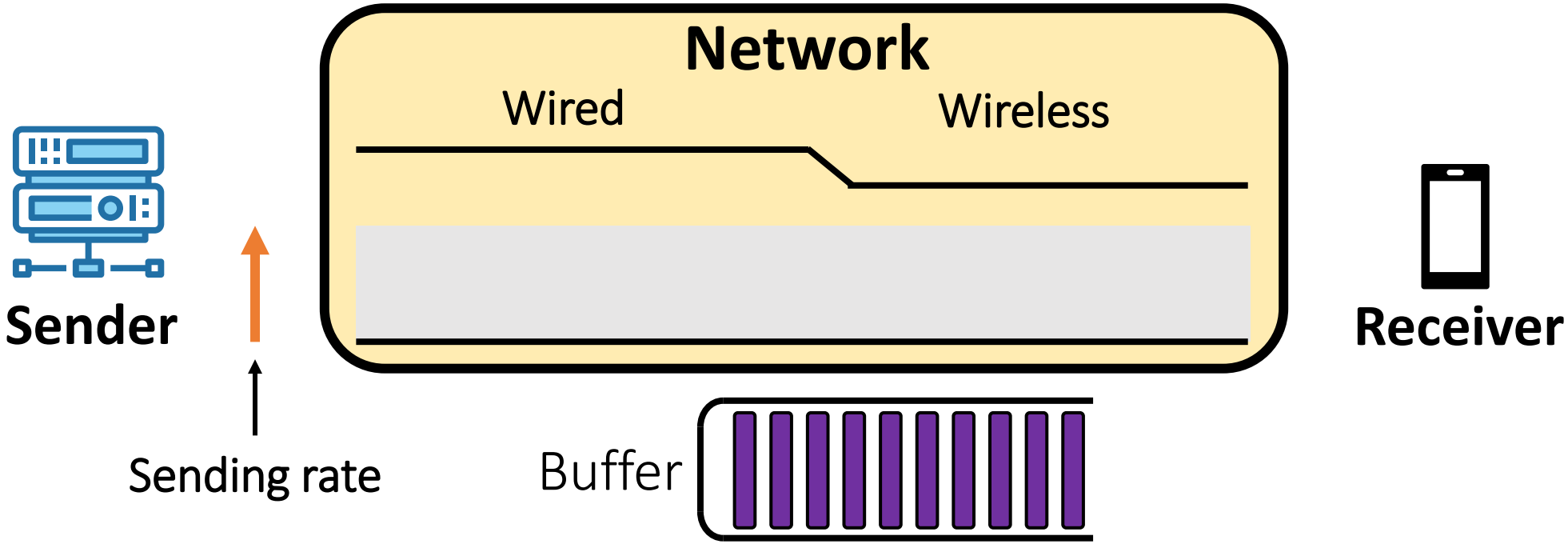




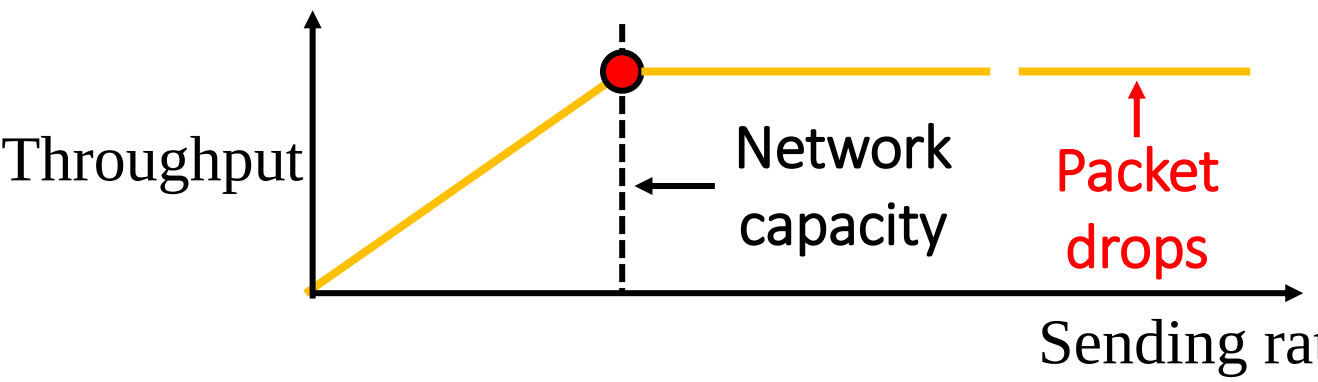
Pure end-to-end solution without cooperation from the network: continuously probing

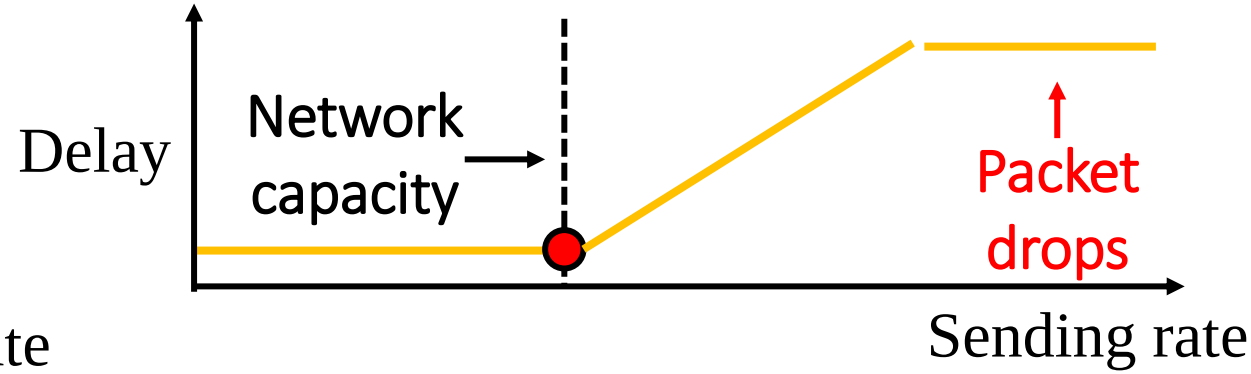


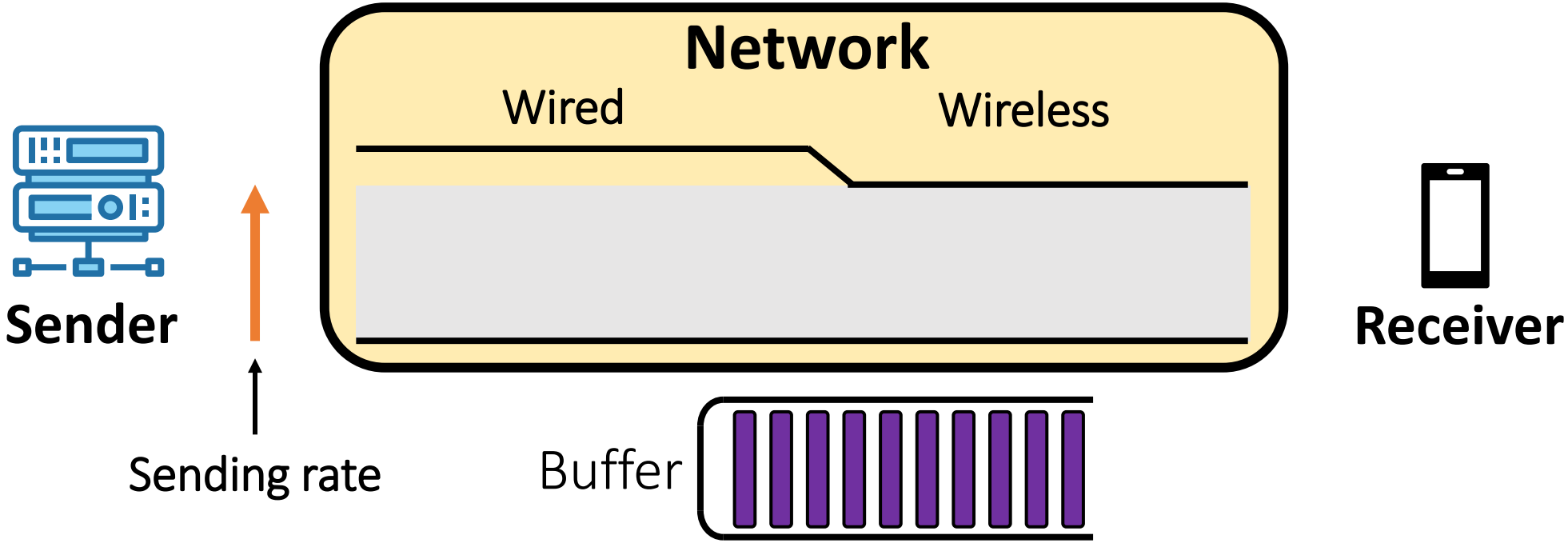




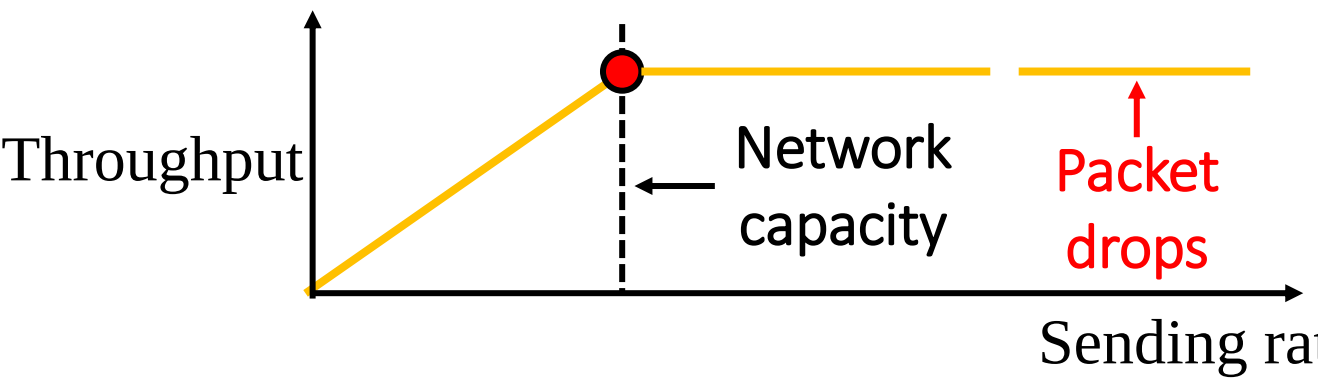
Pure end-to-end solution without cooperation from the network: continuously probing

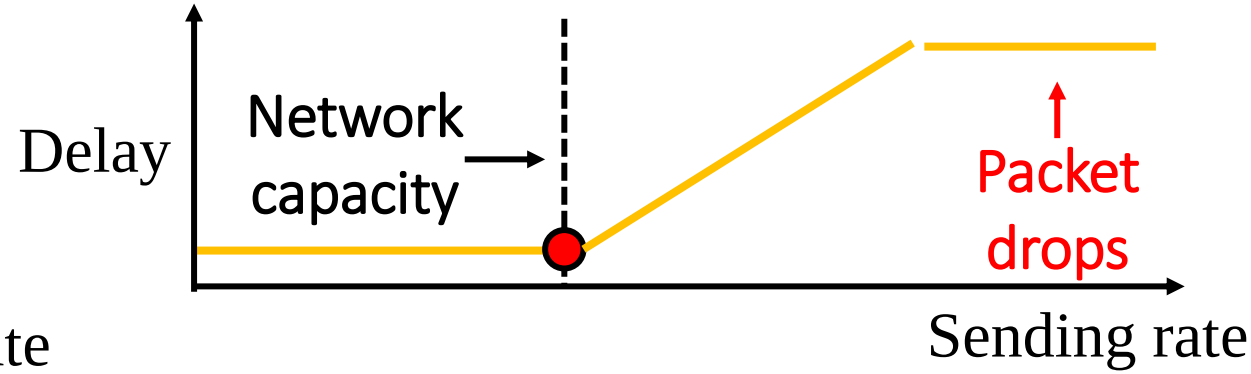


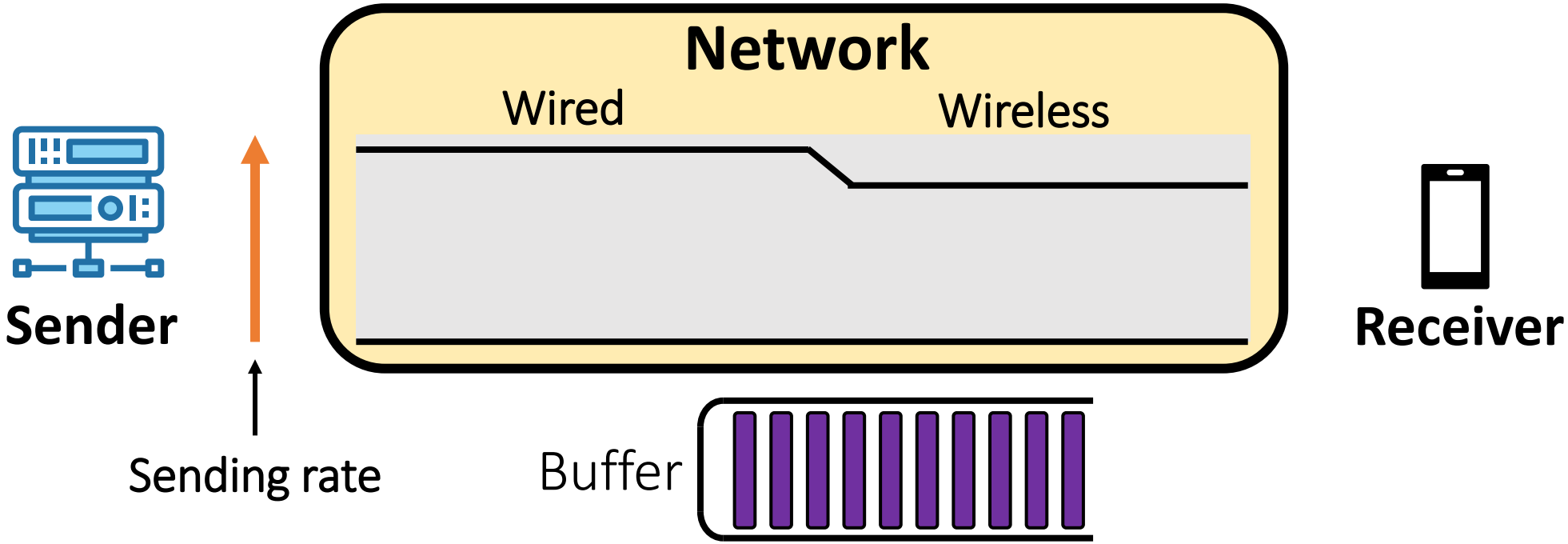




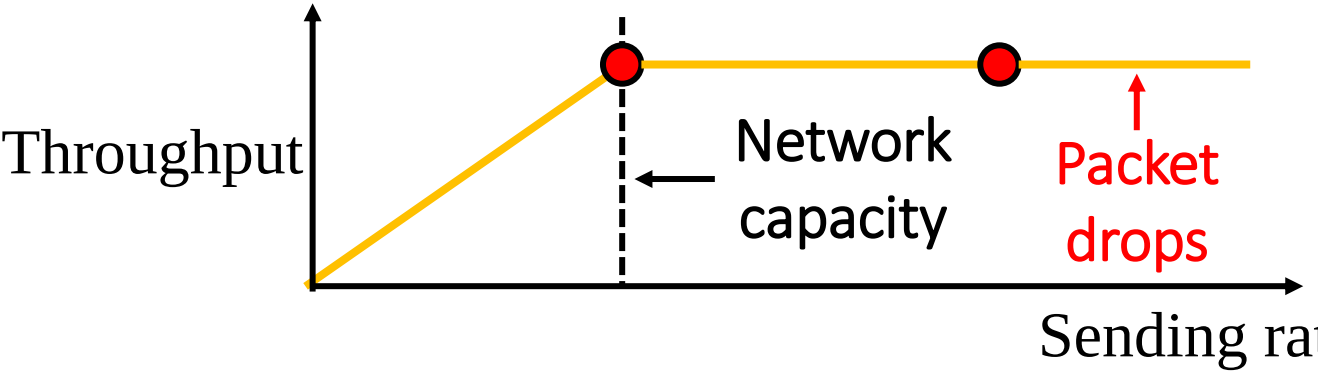
Pure end-to-end solution without cooperation from the network: continuously probing

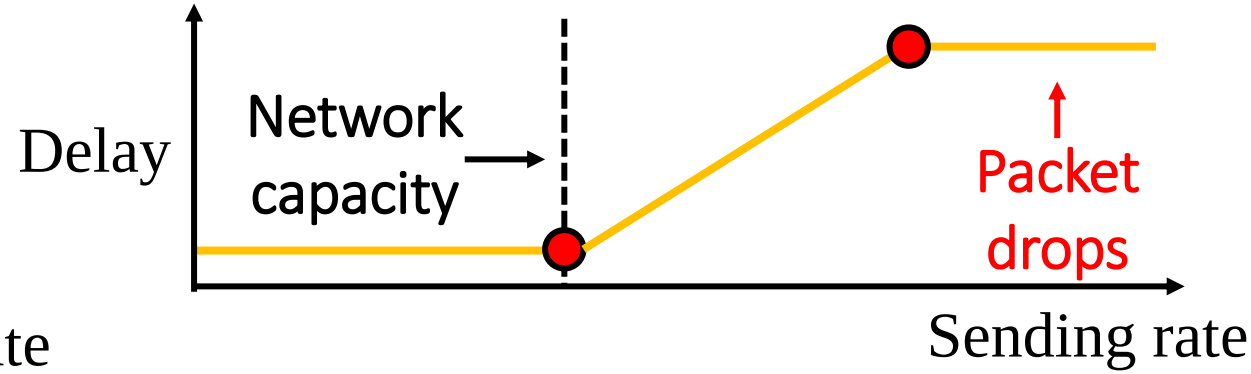


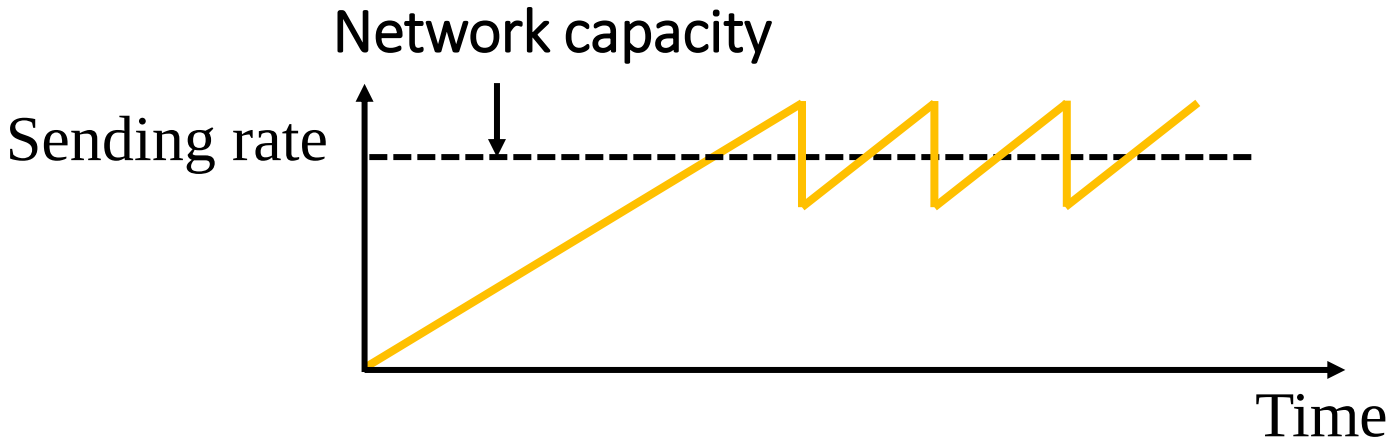




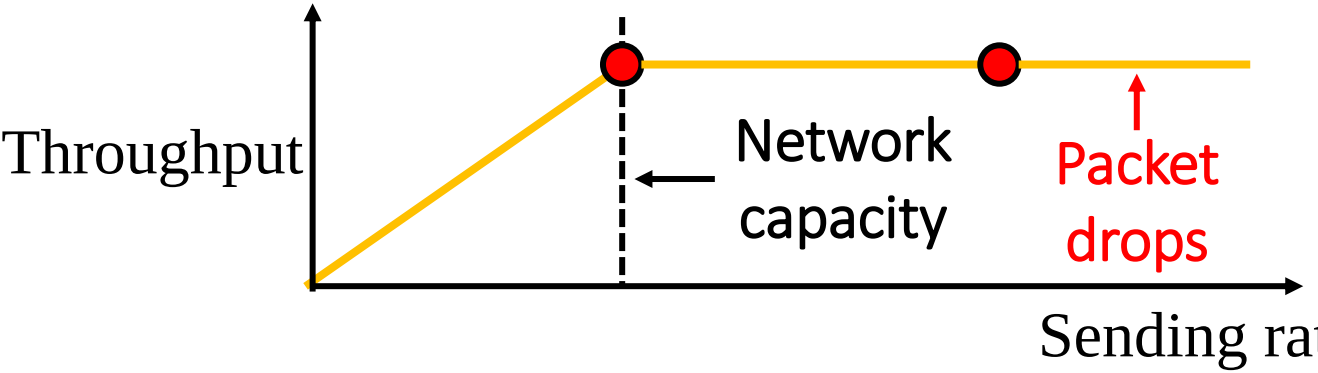
Pure end-to-end solution without cooperation from the network: continuously probing

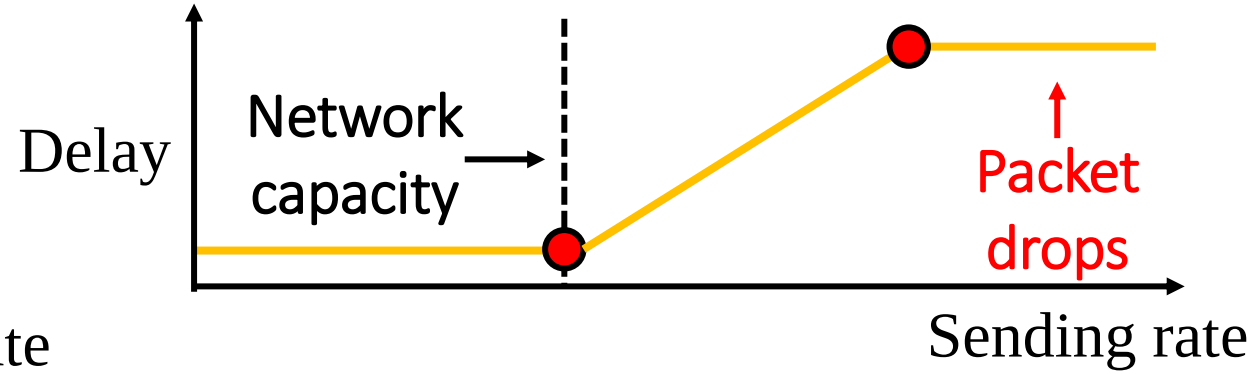


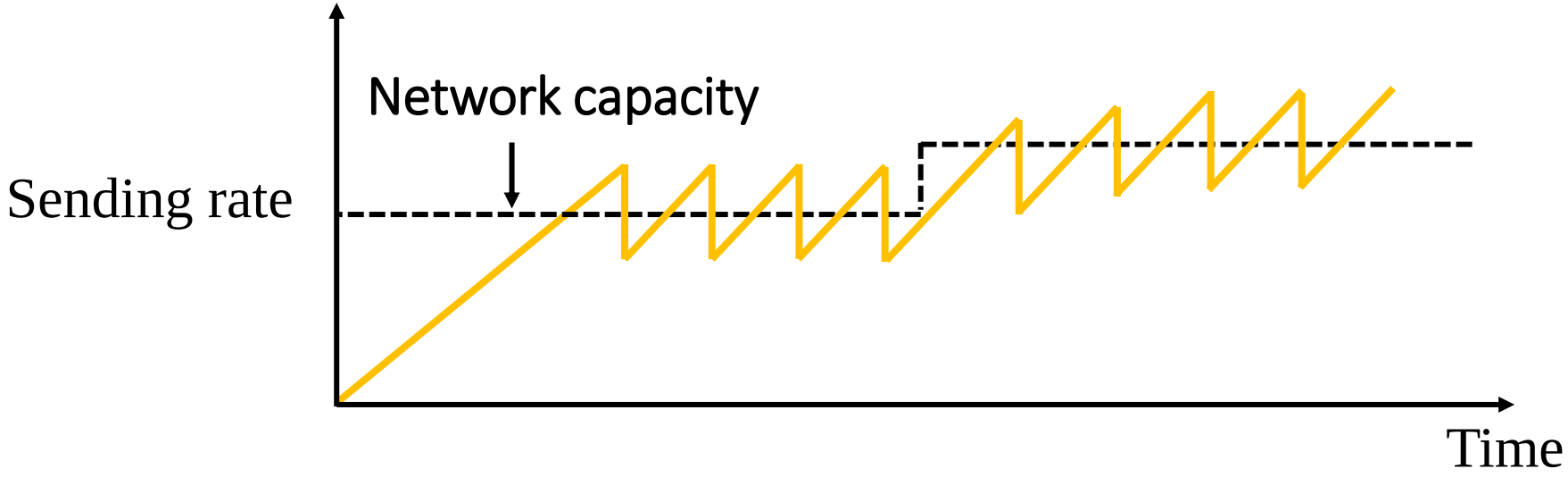




Pure end-to-end solution without cooperation from the network: continuously probing







TCP congestion control: AIMD

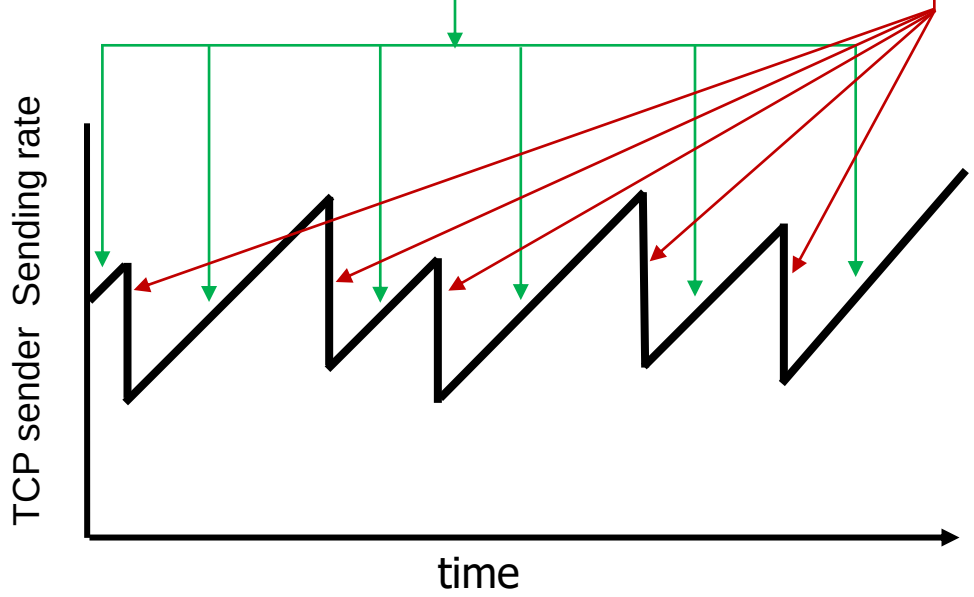
- *approach:* senders can increase sending rate until congestion occurs, then decrease sending rate on congestion

Additive Increase

increase sending rate by 1 maximum segment size every RTT until loss detected

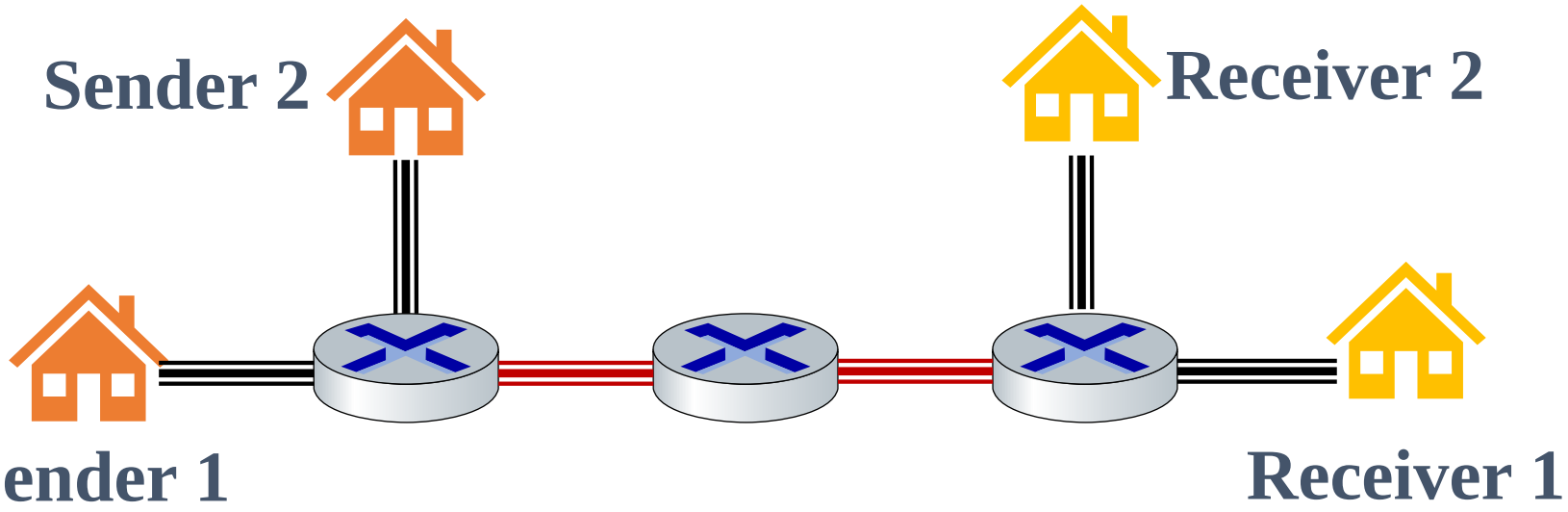
Multiplicative Decrease

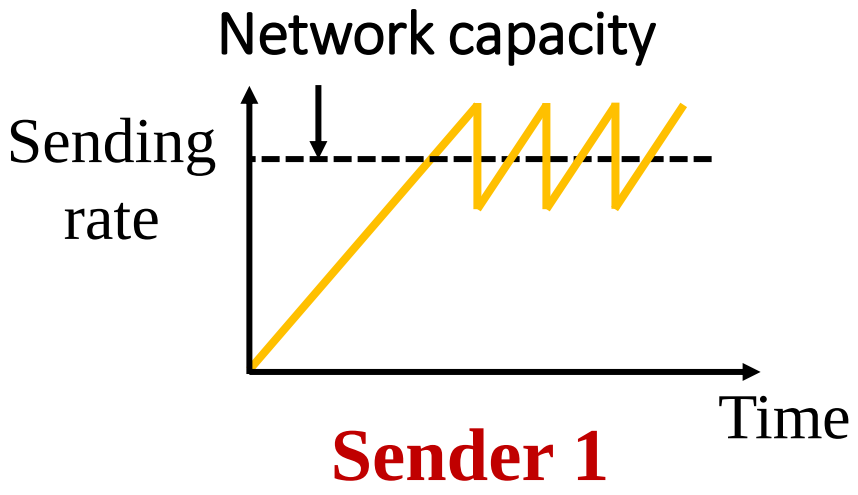
cut sending rate in half at each loss event

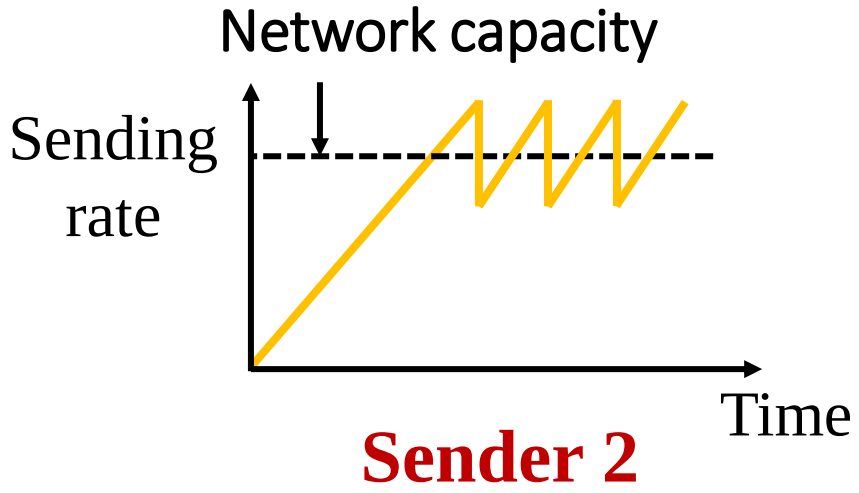


AIMD sawtooth behavior: *probing* for bandwidth

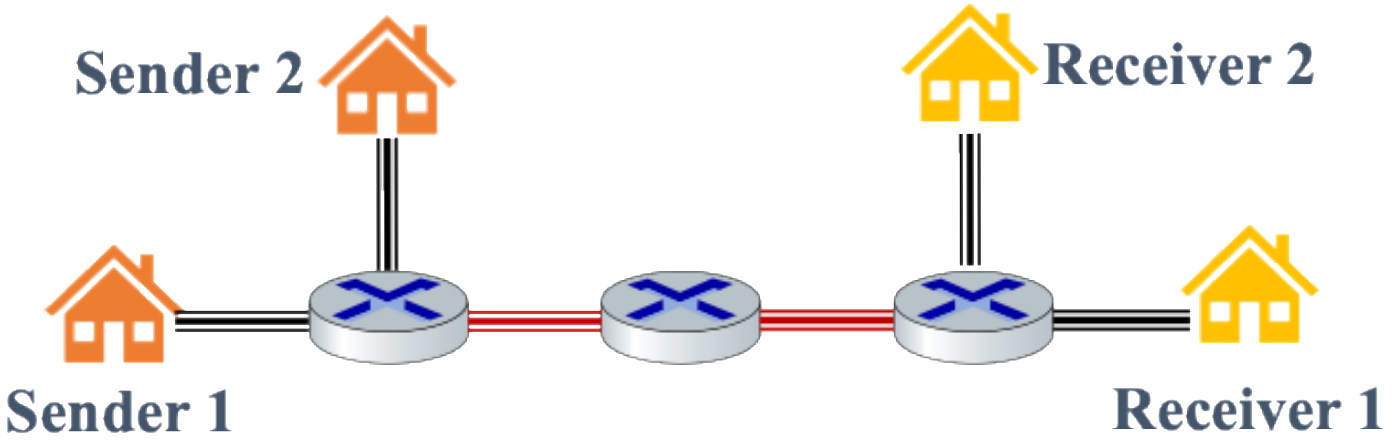
TCP congestion control: AIMD

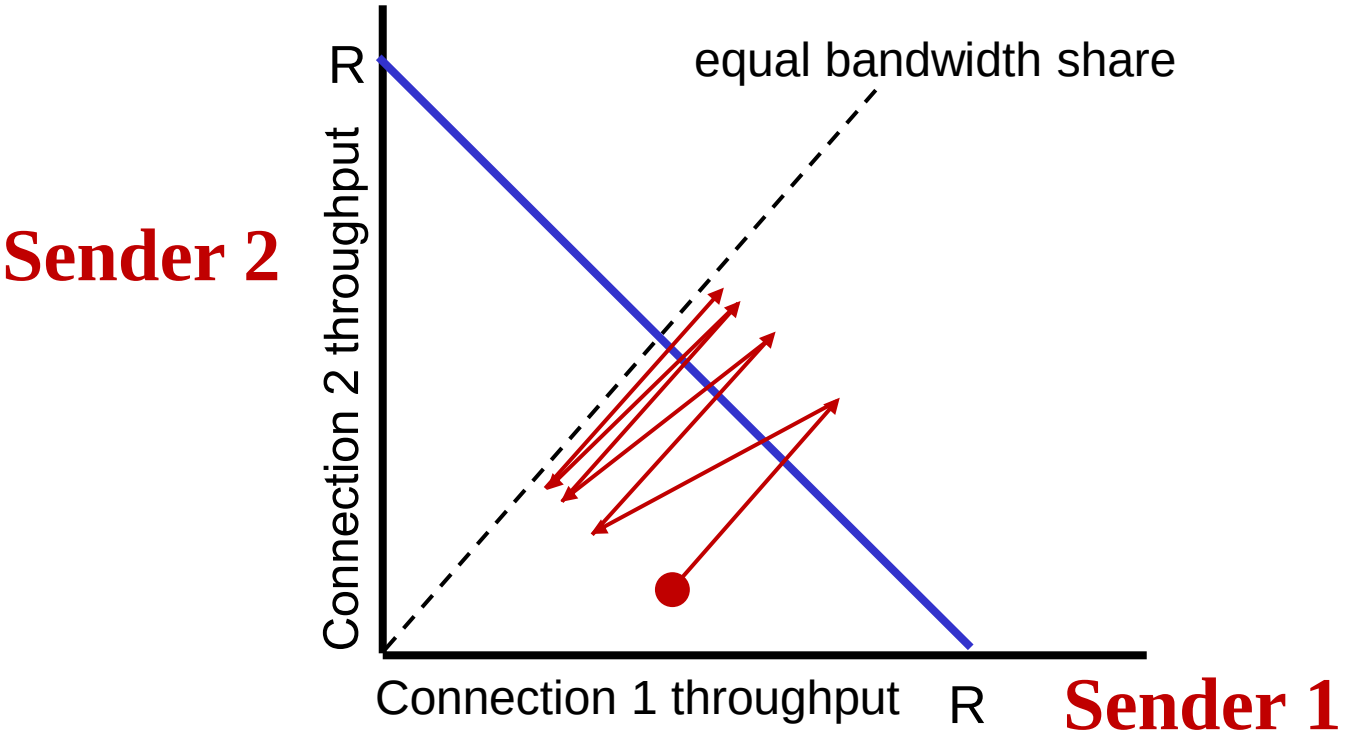






TCP congestion control: AIMD

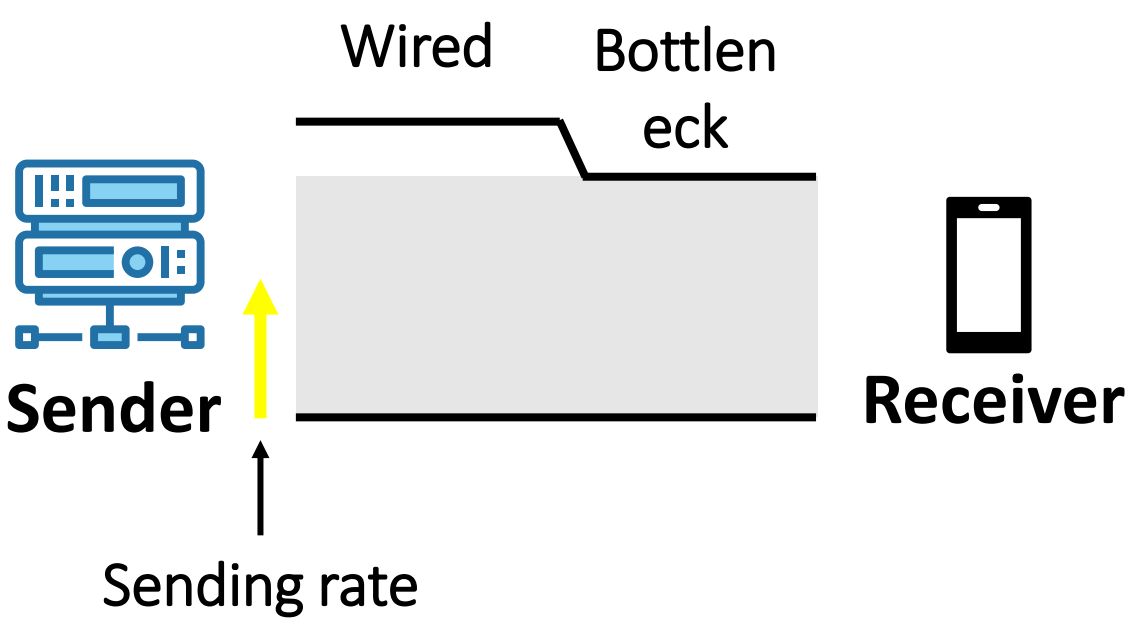




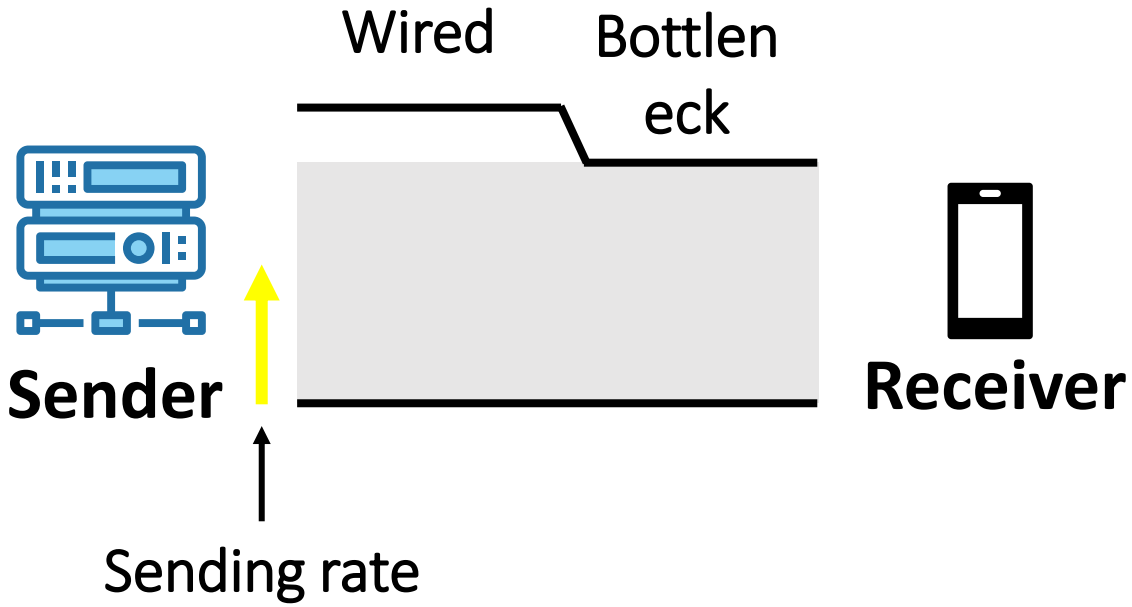
Why AIMD?

- AIMD – a distributed, asynchronous algorithm – has been shown to:
  - optimize congested flow rates network wide!
  - have desirable stability properties

Two Stage of TCP: Slow-Start and Congestion avoidance



Slow Start



Congestion Avoidance

TCP slow start

- when connection begins, increase rate exponentially until first loss event:
  - initially **`cwnd`** = 1 MSS
  - double **`cwnd`** every RTT
  - done by incrementing **`cwnd`** for every ACK received
- *summary:* initial rate is slow, but ramps up exponentially fast