This lecture is being recorded

18-452/18-750
Wireless Networks and Applications
Lecture 19: Wireless and the Internet
TCP

Peter Steenkiste

Spring Semester 2022

http://www.cs.cmu.edu/~prs/wirelessF22/

Announcements

- Midterm solutions will be posted on Wednesday
  - » Please use office hours for questions
  - » Regrade requests should be posted on gradescope (not office hours!)
- Homework 3 will also be posted later this week
- Project 2: I have given feedback on all proposals
  - » We have an interesting set of projects!
  - » However, many proposals were very vague about equipment
    - – "We were thinking about using this device"
  - » The equipment for teams that submitted proposals early has arrived – I will e-mail details

Outline

- Wireless and the Internet
- Mobility: Mobile IP
- TCP and wireless
- Applications and wireless

Main TCP Functions

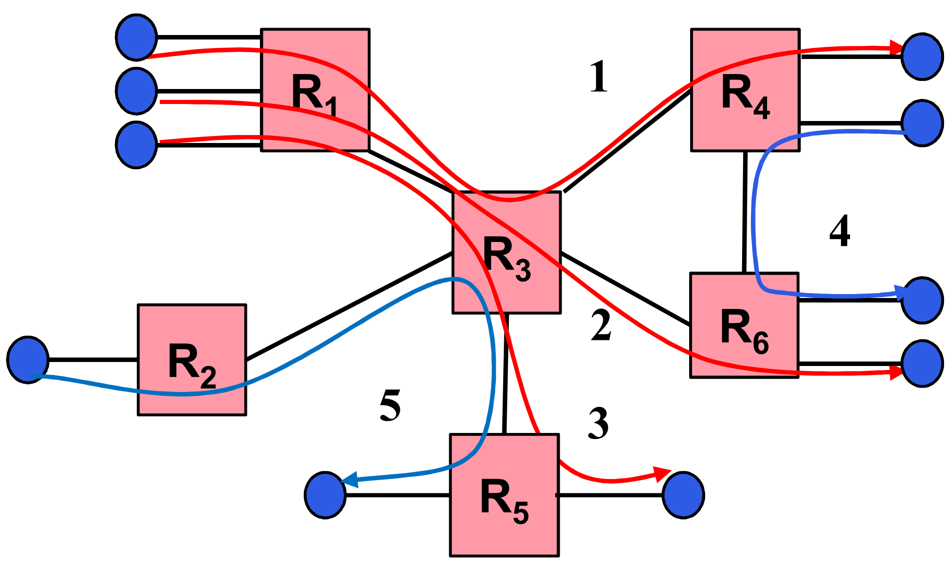
- Connection management
  - Maintain state at endpoints to optimize protocol
- Flow control: avoid that sender outruns the receiver
  - Uses sliding window protocol
- Error control: detect and recover from errors
  - Lost, corrupted, and out of order packets
- Congestion control: avoid that senders flood the network
  - Leads to inefficiency and possibly network collapse
  - Very hard problem – was not part of original TCP spec!
  - Solution is sophisticated (and complex)

Wireless Challenges for TCP

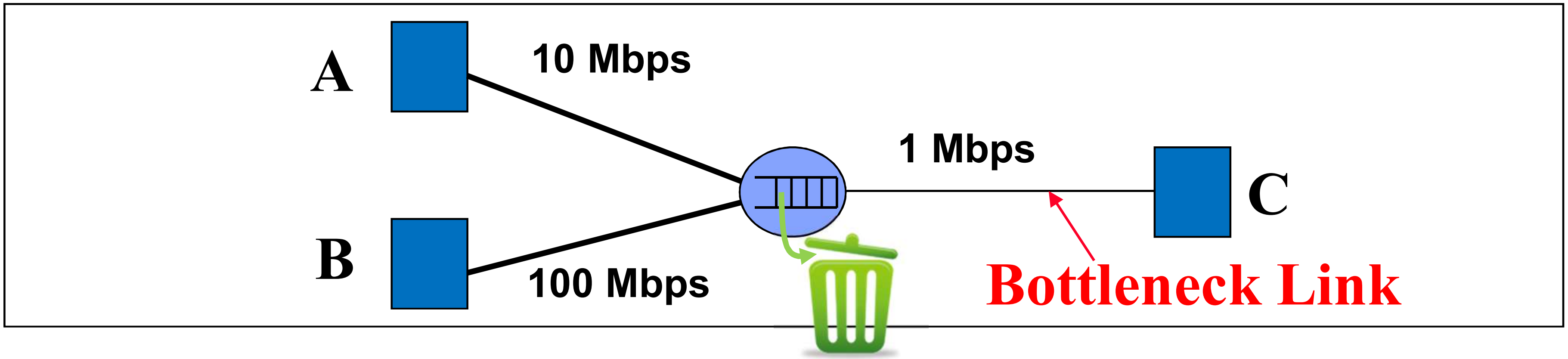
- Loss of network connectivity, e.g., due to mobility
  - » TCP session breaks if the disconnection is long enough
  - » TCP state is lost, so new connection will have to be established when reconnected
  - » For application: not clear how much data was successfully transmitted
- Variability in available bandwidth, e.g., due to changes in channel CSI, handover, …
  - » Should be handled by congestion avoidance (later)
- Increases in latency due to MAC protocol and higher packet loss rate
  - » They have a surprisingly big impact on TCP!

Goals TCP Congestion Control

- The goal of TCP congestion control is to limit the transmit rates of senders so traffic can be handled efficiently by the network
  - » Similar to traffic control on the road – avoid gridlock
- Ideally traffic will get a fair bandwidth allocation
  - » Fair = equal bandwidth under the same conditions



TCP Congestion Control 101



- The bottleneck limits the throughput of senders A and B to receiver C
  - » It is congested: there is more traffic than bandwidth

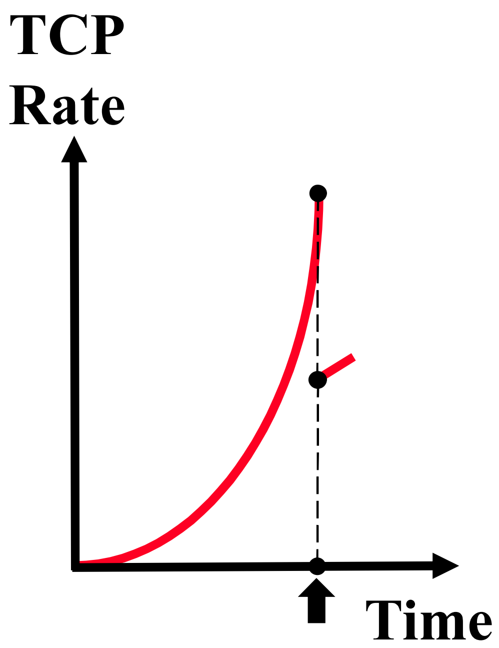
- What should the router do?
- It drops packets – what else can it
  - » Informally: when the queue is full, it overflows

- What should senders do?
- Slow down when there is congestion
  - » Congestion event = packet loss

Loss

- The Internet design and TCP specifically assumes that packet loss is a sign of congestion
  - » It is defined as a "congestion event" and TCP will reduce its transmit rate
- This is appropriate in wired networks since practically all losses are the result of queue overflow
- However, wireless channels are much more challenging resulting in higher bit error rates and thus more packet loss
  - » Especially when sender tries to maximize the bit rate
- Solution: wireless network aggressively avoids packet loss on the wireless ink
  - » To higher level protocols, the wireless link looks like a wired one!
  - » WiFi and cellular use both FEC and retransmission for error recovery

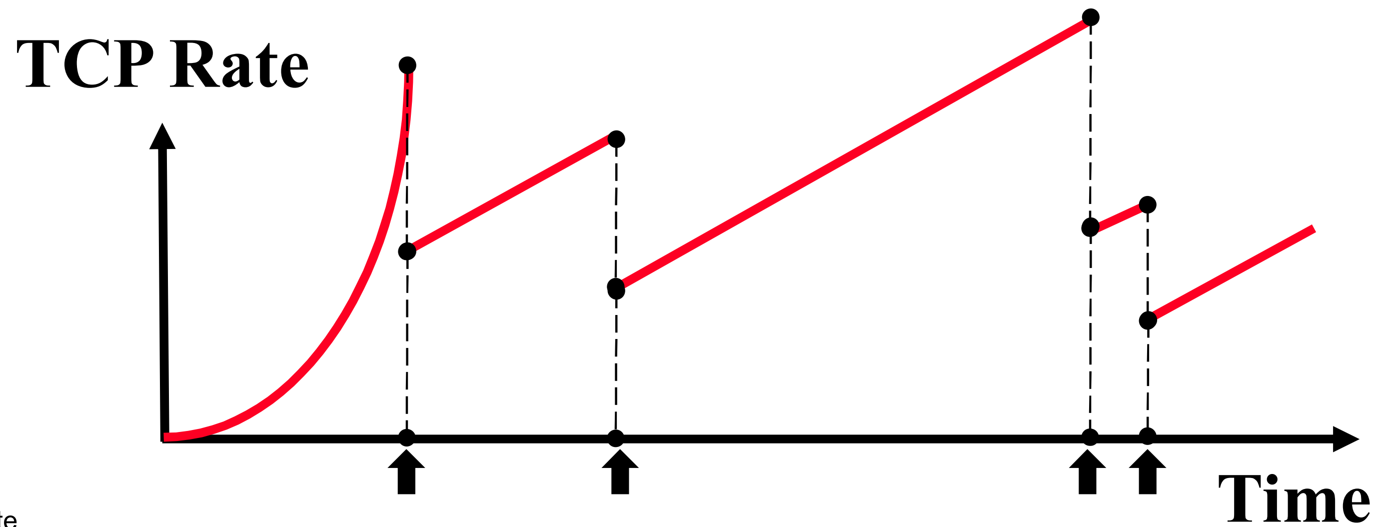
What Transmit Rate Should a new TCP Session Use?



- Analogy: suppose you want to know how long it takes to drive to a new destination?
  - » It depends on traffic!
- TCP discovers the available bandwidth by increasing transmit rate exponentially!
  - » Double the transmit rate every RTT
- This is called “Slow Start”
- Slow Start ends when the sender observes a congestion event
  - » Packet loss is the most common one

What Rate Should TCP Use after Slow Start?

- What goals and constraints should be considered?
  - » The sender wants to go as fast as possible!
  - » The network requires that the sender slows down in response to congestion
- Continuously probe for more bandwidth
  - » Increase rate (by one packet/RTT) every RTT
- Reduce rate when there is a congestion event
  - » Cut the transmit rate in half

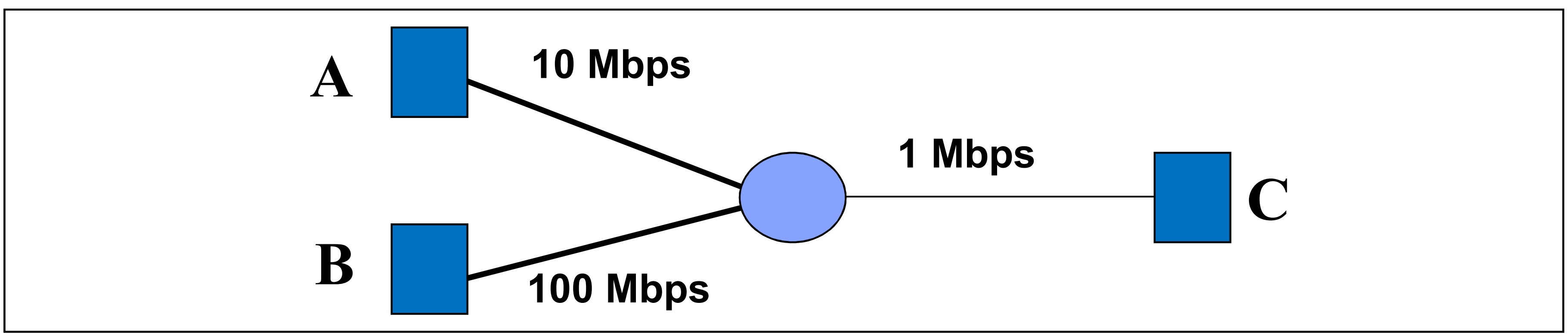


Limitations

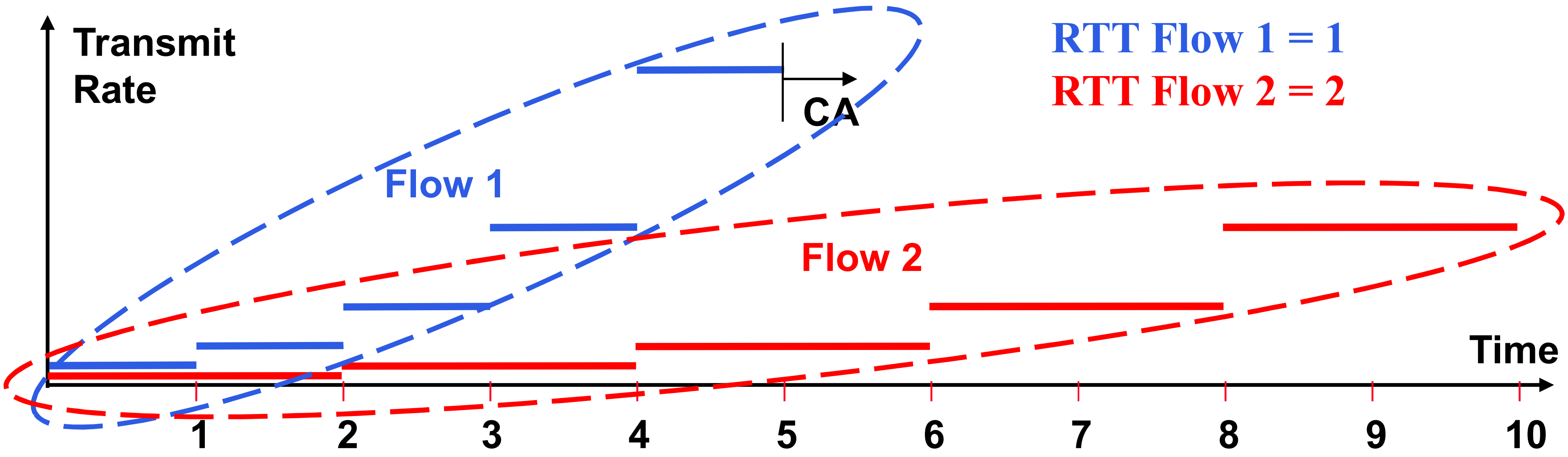
- The version of TCP congestion control describes so far is just the basics
- Today's congestion control algorithms are much more efficient (background only):
  - » They aggressively try to avoid time outs
  - » They are more careful about they increase/decrease the transmit rate
  - » They sometimes use increase in RTT as a congestion event
    - – Avoids packet loss!
  - » Etc.

Relevance to Wireless

- **More variability of the latency**
  - » **But congestion control should be able to keep up**
- **The reaction time of the network is measured in number of Round Trip Times (RTT)**
  - » **Longer RTT → slower response time**
- **This means that the RTT has an impact on throughput!**



Latency Implications for Slow Start



- Flow 2 has an RTT that is twice that of Flow 2, so …
- During SS, its rate in each RTTs is half that of Flow 1
  - » TCP controls transmit rate using packet counts, not transmit rate
- Since rate increases occur once each RTT, increases occur at only half the rate
  - » So Flow 1 is stuck sending at lower rate much longer

CloudSLAM: Edge Offloading of Stateful Vehicular Applications, Kwame-Lante Wright, Ashiwan Sivakumar, Peter Steenkiste, Bo Yu, and Fan Bai Fifth ACM/IEEE Symposium on Edge Computing, 2020.

Implications for Congestion Avoidance

$$\text{Rate} = \frac{MSS}{RTT} \times \frac{C}{\sqrt{P}}$$

MSS Maximum Segment Size
RTT = round trip time
C = constant depends on context
P = packet loss rate

- In Congestion Avoidance mode, the transmit rate is inverse proportional to the roundtrip time
- Again, flows with high RTT are at a disadvantage!
  - » Informal reason: low-RTT flows increase their rate faster, i.e., more aggressively
- Moving servers closer to clients has many advantages:
  - » Transmit rates increase much faster during Slow Start
  - » Higher throughputs during Congestion Avoidance
  - » Shorter network paths may reduce significant network bottleneck
- Builds a strong case for edge computing

The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm, Matthew Mathis, Jeffrey Semke, Jamshid Mahdavi, ACM Sigcomm, 1997

Outline

- Wireless and the Internet
- Mobility: Mobile IP
- TCP and wireless
- Applications and wireless

How Does Wireless Impact Applications

- The layered Internet protocol stack largely isolates applications from layer 1&2 details
- Except for:

1. Disconnected operation: it is impossible to hide that fact that the device is no longer connected to the network
   - » This is a big deal – not just a detail!
2. Variability in available bandwidth
   - » Due to changing channel conditions, handover, ..
3. Higher end-to-end latency (RTT)
   - » Due to the extra delay introduced by MAC mechanisms

Disconnected Operation

- Mobility means that devices will occasionally be disconnected from the network
  - Seconds ... Minutes … Hours .. Days
  - Mostly an issue for clients
- This can confuse systems and applications that assume a wired/stationary model
  - Clients cannot access servers, e.g., mail, calendar applications, back up, …
  - Distributed file systems
- Must adapt the applications so they become “disconnection aware”
  - Not always possible: watching online movie, voice/video meetings, etc.

Making Applications Disconnection-aware

- The goal is to

1. maintain as much functionality on client as possible while disconnected
2. properly update global state when reconnecting to server

- Clients that rely on long lived connections with a server are fundamentally problematic
  - » Basically assumes the network is available
  - » Session state is lost when connection breaks
- Two step solution: move away from long term connections and adapt how state is maintained

REST(full) APIs

- Application programming interface for client server interactions that is:
  - Clean separation of client and server using well defined data formats
  - Statelessness: each client request is independent, allowing the server to complete processing
  - REST APIs are viewed a good software engineering practice
- Simple example is web interface: HTTP
  - HTTP is stateless and supports a well defined set of requests clients can to servers
  - Not just for page retrieval! Much more general
- In our (limited) context: avoids the user of long lived sessions, failed requests can be retried

Two Examples

- E-mail: users must be able to "work on" e-mail offline and operations are requested from the server when the client reconnects
  - » Sends/deletes/moves e-mails
  - » Possibly others: manage folders, etc.
- Calendars and tasks are similar: operations performed offline must be executed later
  - » Adding or removing appointment and tasks, …
- Must sometimes resolve conflicts when multiple clients are used offline
  - » E.g., mail is deleted on one client and moved to another folder on another – delete or keep?
  - » Tend to be minor – ask user for help if needed

More Complex Case: Shared File System

- A distributed file system can be shared by application on many computers
  - » Files are cached in the client computers
- Applications can modify files locally and later update the server, but …
- File updates are not limited to small, well-defined number of operations
  - » Files can be changed in random places, data can be moved around in a file and between files, etc.
- The merging files can be arbitrarily complex
- AFS is an example of a FS that provides good support for managing consistency