Computer Networks (CS F303)

BITS Pilani
Pilani Campus

Virendra Singh Shekhawat
Department of Computer Science and Information Systems

innovate achieve lead

BITS Pilani

Pilani Campus

Second Semester 2020-2021

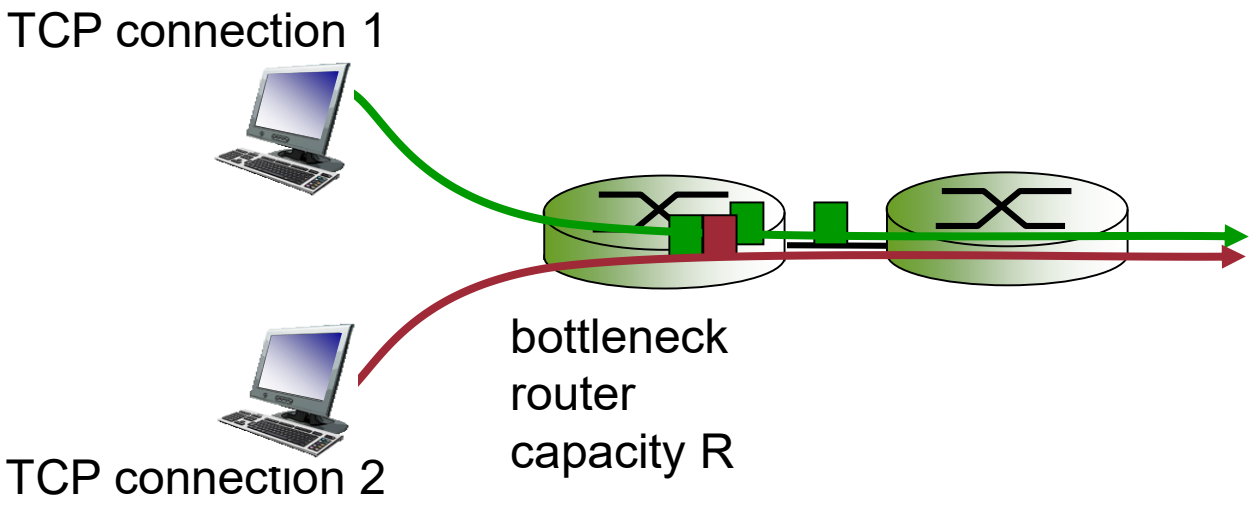
Module-3 <Transport layer>

Topics

- Transport Layer
- TCP Protocol
  - Connection Establishment
  - TCP Segment Structure
  - Reliable data transfer
  - Flow control
  - Timeout Estimation
  - Congestion control
  - **TCP Fairness**
  - **TCP Variants**

TCP Fairness

Fairness Goal: if K TCP sessions share same bottleneck link of bandwidth R, each should have average rate of R/K



Fairness and UDP

- Multimedia apps often do not use TCP
  - Do not want rate throttled by congestion control
- Instead use UDP
  - Send audio/video at constant rate, tolerate packet loss
- Multimedia applications running over UDP are not being fair. Why???

Fairness and Parallel TCP Connections

- Web browsers often use multiple parallel TCP connections
  - To transfer multiple objects within a Web page

- Application level fairness with multiple parallel TCP connections???

TCP Fairness with different RTT

- Flows sharing bottleneck link with different RTT do not get same bandwidth. Why???
  - BW is proportional to 1/RTT

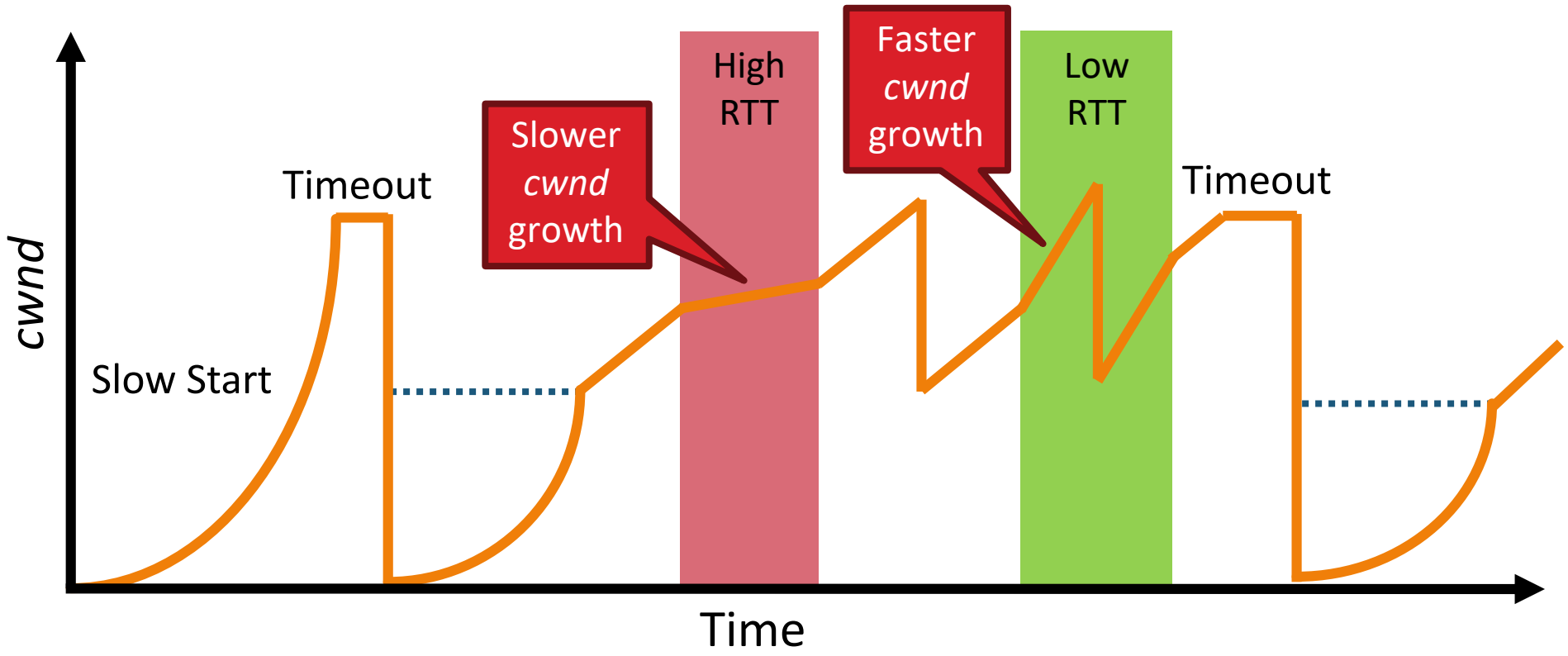
Goals for Improving TCP Fairness and Throughput

- Fast window growth
  - Slow start and additive increase are too slow when bandwidth is large
  - Want to converge more quickly
- Maintain fairness with other TCP variants
  - Window growth cannot be too aggressive
- Improve RTT fairness
  - TCP Tahoe/Reno flows are not fair when RTTs vary widely

Compound TCP Implementation

- Default TCP implementation in Windows 2008 TCP Stack
- Key idea: split *cwnd* into two separate windows
  - Traditional, loss-based window and new, delay-based window
- *wnd* = min(*cwnd + dwnd*, *adv_wnd*)
  - *cwnd* is controlled by AIMD
  - *dwnd* is the delay window
- Rules for adjusting *dwnd:*
  - If RTT is increasing, decrease *dwnd* (*dwnd* >= 0)
  - If RTT is decreasing, increase *dwnd*
  - Increase/decrease are proportional to the rate of change

Compound TCP Example

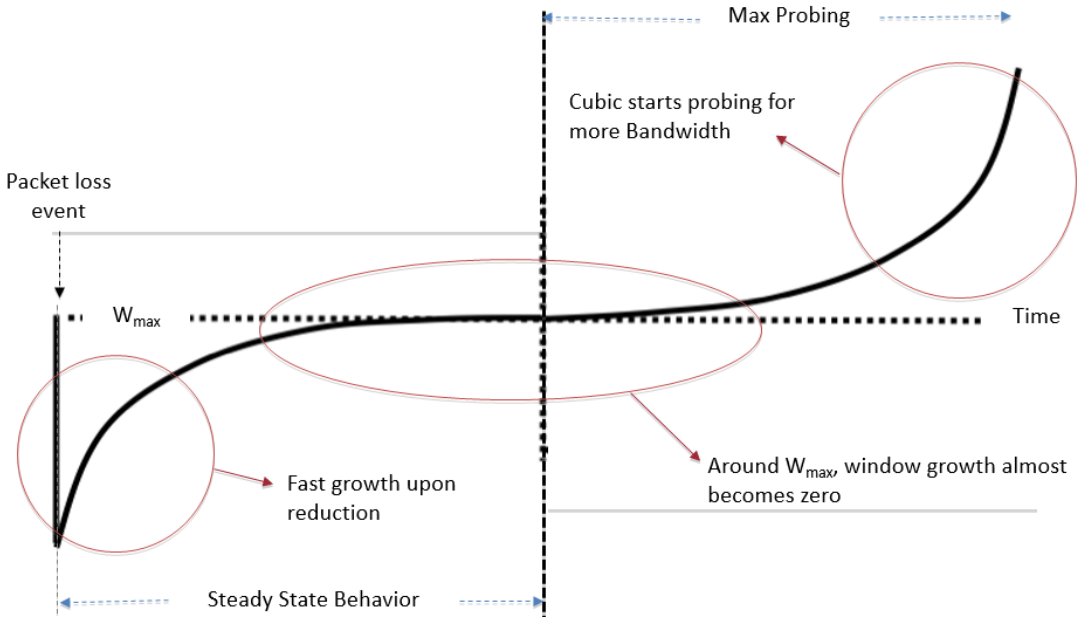


- Aggressiveness corresponds to changes in RTT
- +ive: fast ramp up, more fair to flows with different RTTs
- -ive: must estimate RTT, which is challenging

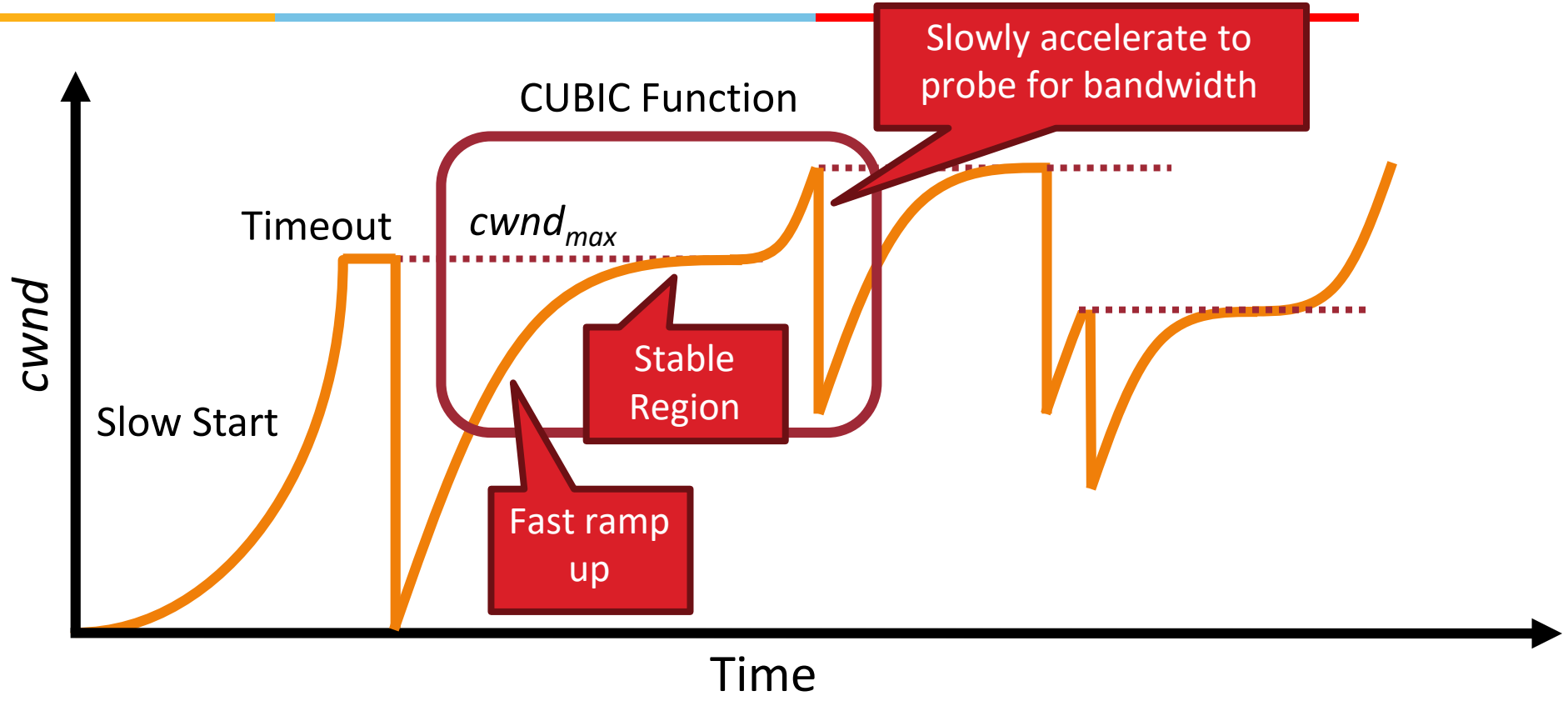
TCP Cubic

- cwnd = $C(t - K)^3 + W_{max}$
  - W_{max} = cwnd before last reduction
  - β multiplicative decrease factor
  - C scaling factor
  - t is the time elapsed since last window reduction

$$K = \sqrt[3]{W\beta / C}$$



TCP CUBIC Example



- Less wasted bandwidth due to fast ramp up
- Stable region and slow acceleration help maintain fairness
  - Fast ramp up is more aggressive than additive increase
  - To be fair to Tahoe/Reno, CUBIC needs to be less aggressive

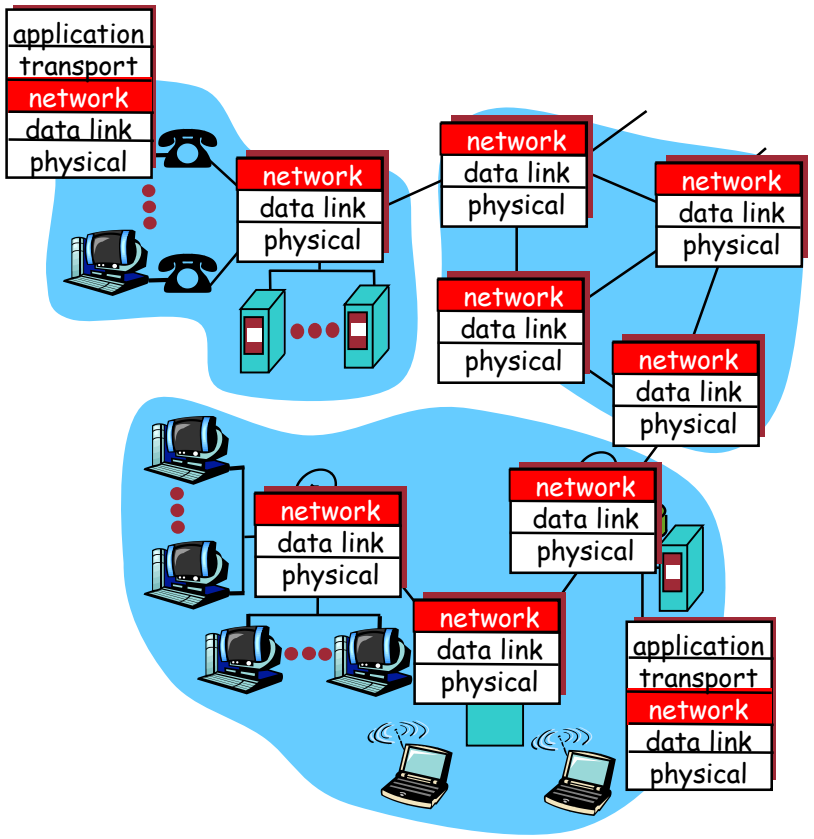
What's Next…

- Network Layer
  - Network layer service models (Datagram,MPLS and ATM)
  - Forwarding versus Routing
  - How a router works?
  - IPv4 Datagram and Fragmentation
  - IPv4 Addressing
    - Hierarchical Addressing
    - NAT, Sub Netting, IPv4 to IPv6 translation, ICMP
  - Routing Algorithms and Protocols
    - Inter-domain Routing and Intra-domain routing
  - Multicast Routing

Network layer

- Transport segment from sending to receiving host
- On sending side encapsulates **segments** into **datagrams**
- On receiving side, delivers **segments** to transport layer

- Router examines header fields in all IP datagrams passing through it



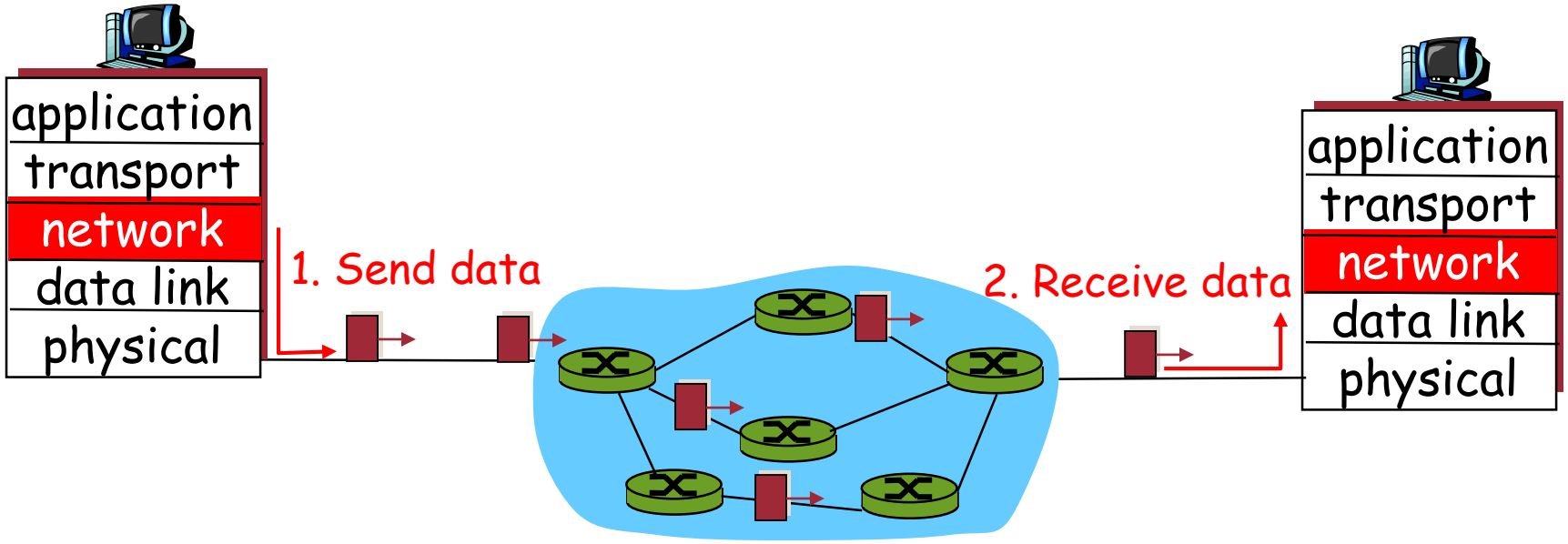
Network Layer Connection and Connection-less Service

- Datagram network
  - Network-layer connectionless service
- VC Network
  - Network-layer connection service

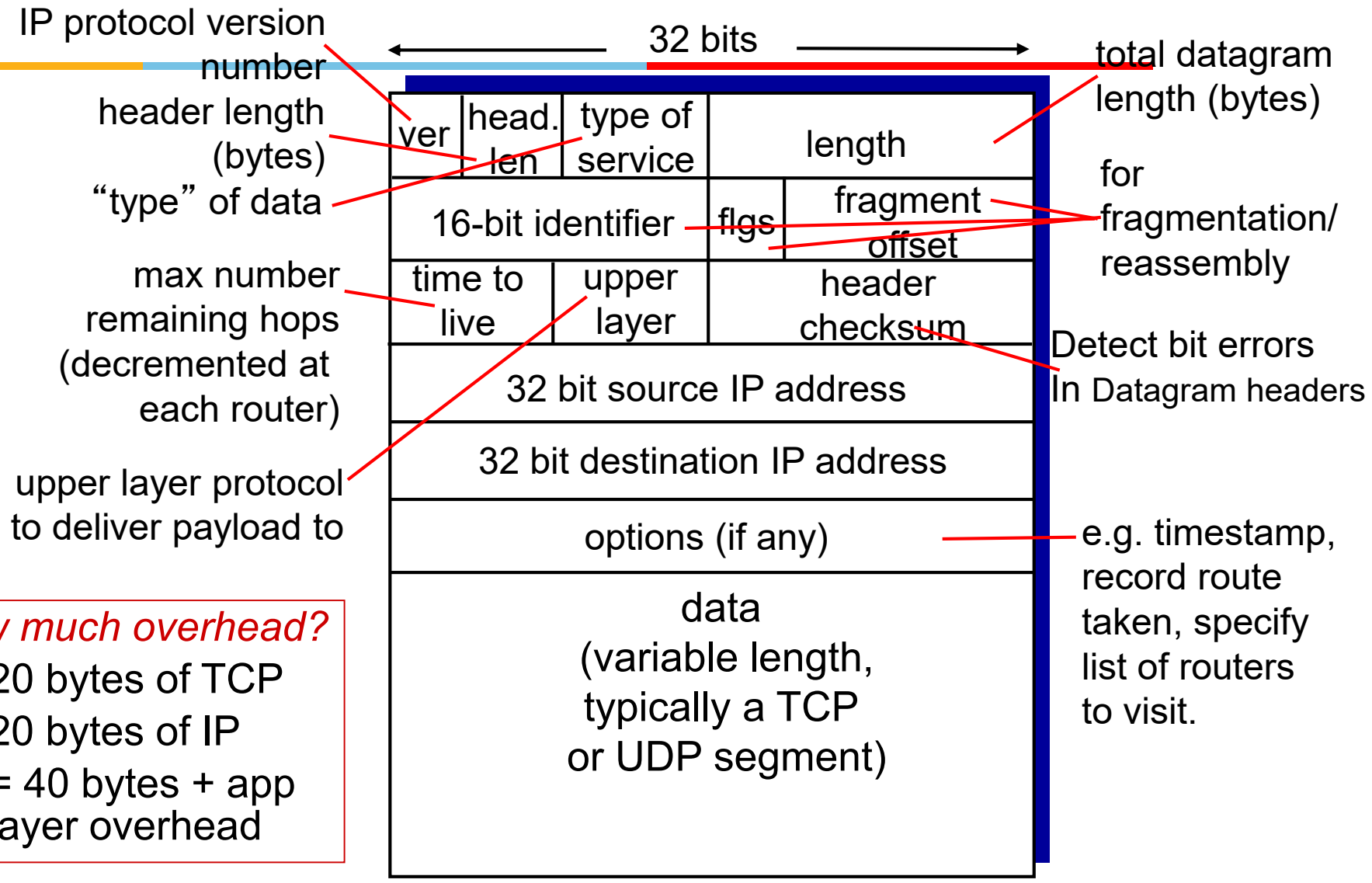
- Analogous to the transport-layer services, but
  - Service: host-to-host
  - No choice: network provides one or the other
  - Implementation: in network core

Datagram Networks

- No call setup at network layer
- Routers: no state about end-to-end connections
  - No network-level concept of “connection”
- Packets forwarded using destination host address
  - Packets between same src-dest pair may take different paths



IP Datagram Format



how much overhead?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

Thank You