Computer Networks
COL 334/672

Measuring Internet Performance and Link Layer

Slides adapted from KR

Sem 1, 2025-26

Recap

- How to measure Internet performance?
- Key metrics
  - Delay $\longrightarrow$ $d_{node} = d_{processing} + d_{queuing} + d_{transmission}$
    $+ d_{propagation}$
  - Loss
  - Throughput

Packet loss

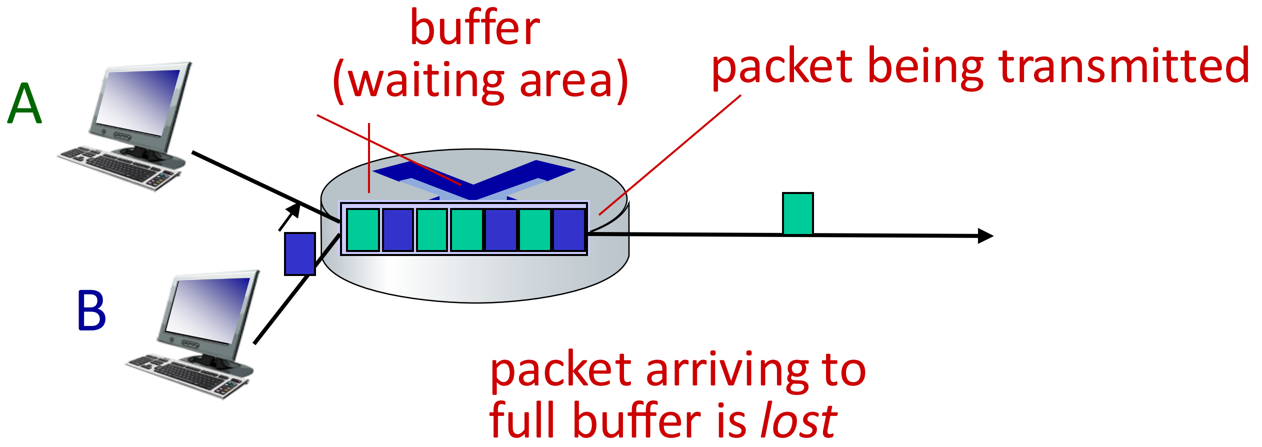
Other reason
① hard failure
② routing failure
③ bit error

How to measure packet loss?
↳ as a n/w operator
↳ as an end user

- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all
- Typically measured as the percentage of total packets sent

100 pkts , 5 lost

loss : 5%

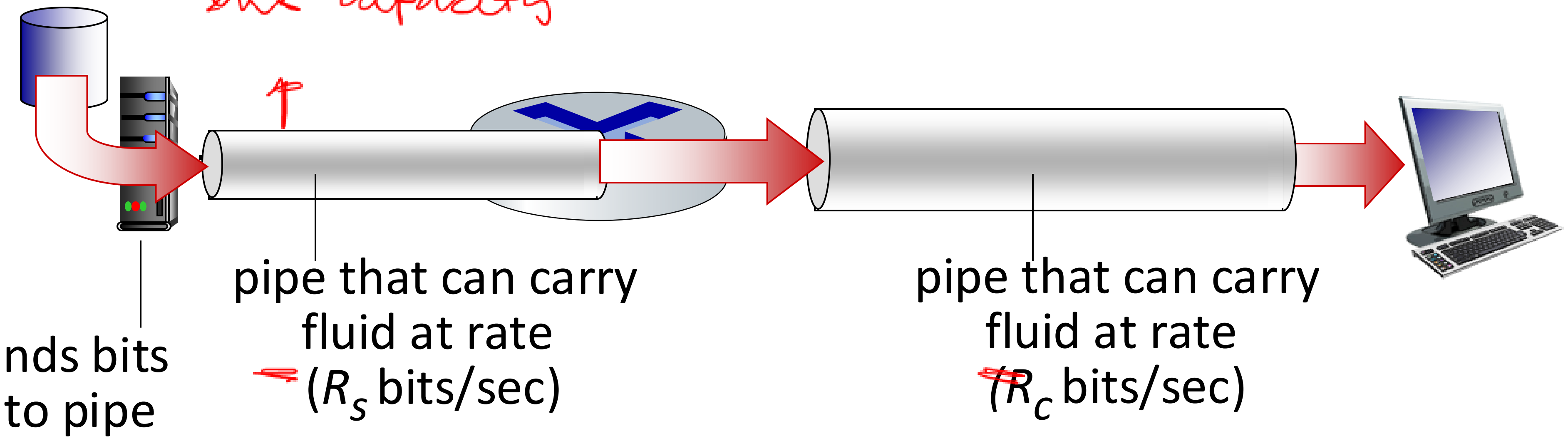


Throughput

- *throughput:* rate (bits/time unit) at which bits are being sent from sender to receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time

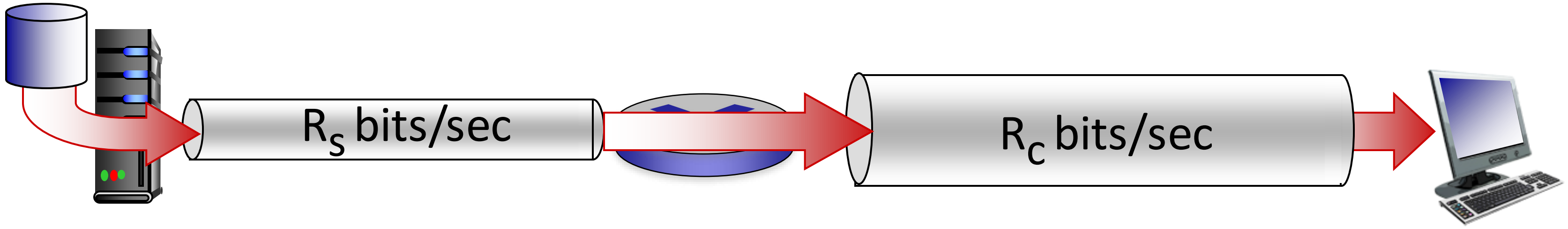
min (R_s , R_c)

link capacity

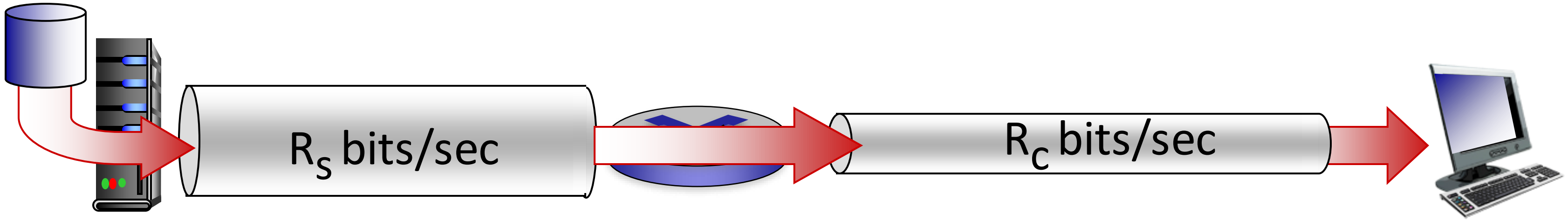


Throughput

$R_s < R_c$ What is average end-end throughput?



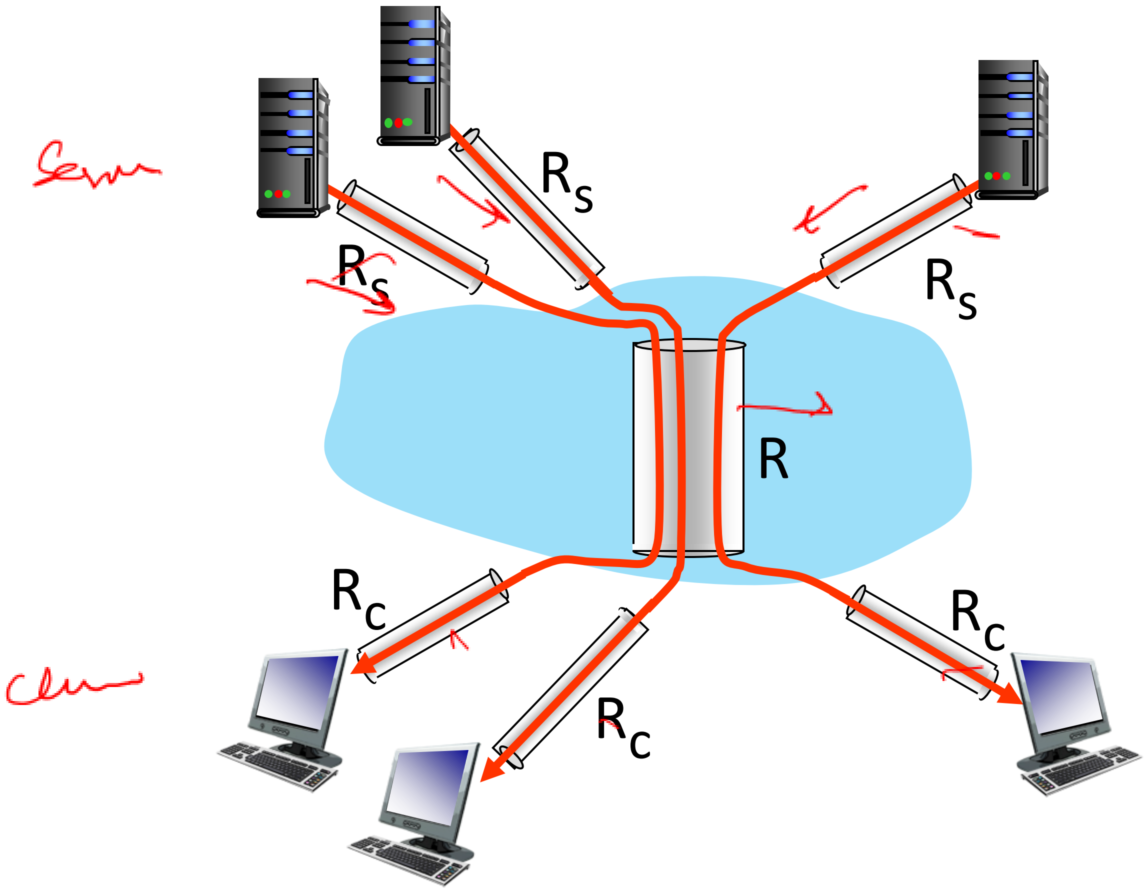
$R_s > R_c$ What is average end-end throughput?



bottleneck link

link on end-end path that constrains end-end throughput

Throughput: network scenario



10 connections (fairly) share backbone bottleneck link *R* bits/sec

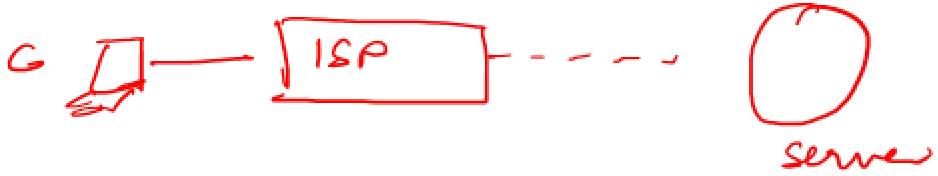
- per-connection end-end throughput: $min(R_c, R_s, R/10)$
- in practice: R_c or R_s is often bottleneck

How do you measure end-to-end throughput on network?

- Various speed test tools (e.g., ookla speedtest, mlab ndt7, iperf)
- How does a speedtest work?

Successfully send D bits in T seconds

throughput = D/T



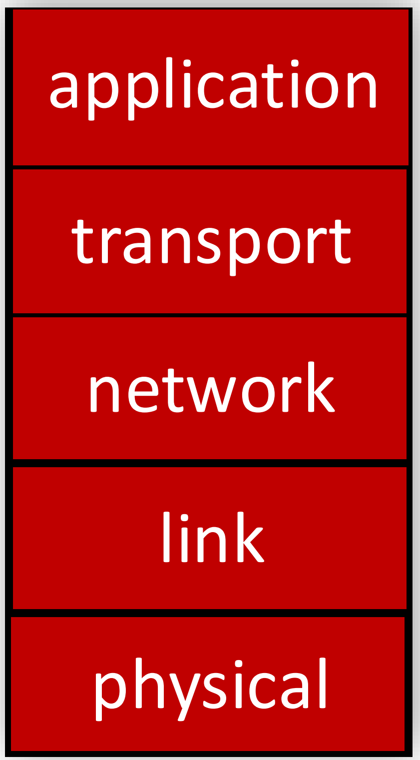
goodput: rate of successfully sending useful data

1. Only app payload
2. Excluding retransmission

- Let's take a speedtest

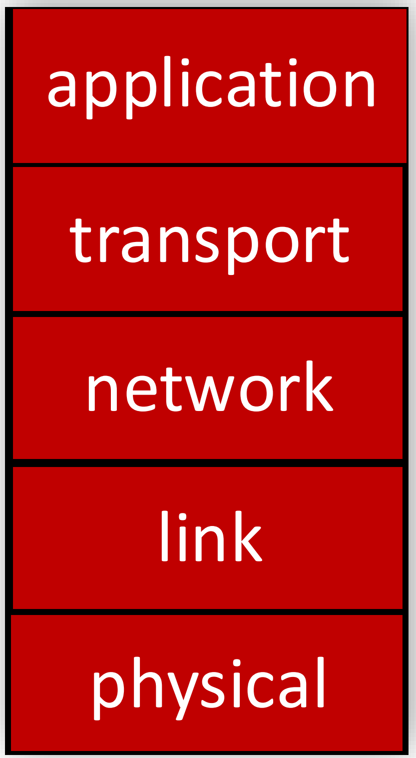
Recap

- Internet design philosophy
  - Network of networks – any network can connect
  - Packet switching for cost-effective resource sharing
  - End-to-end principle: dumb network, intelligent end-hosts
  - Layered architecture: 5-layered IP stack
- Measuring network performance
  - Metrics: throughput, latency, loss

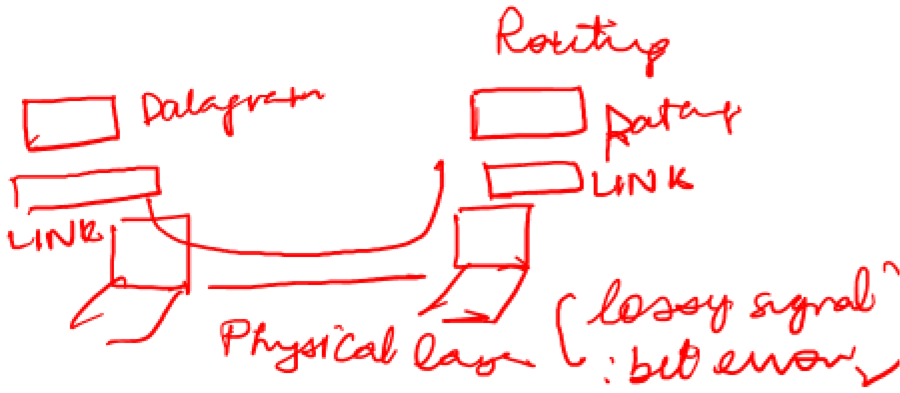


Layered Internet Protocol Stack

- *application:* supporting network applications
  - HTTP, IMAP, SMTP, DNS
- *transport:* process-process data transfer
  - TCP, UDP
- *network:* routing of datagrams from source to destination
  - IP, routing protocols
- *link:* data transfer between neighboring network elements
  - Ethernet, 802.11 (WiFi), PPP
- *physical:* bits “on the wire”



Link Layer: Services



Layer-2 packet: *frame*, encapsulates datagram

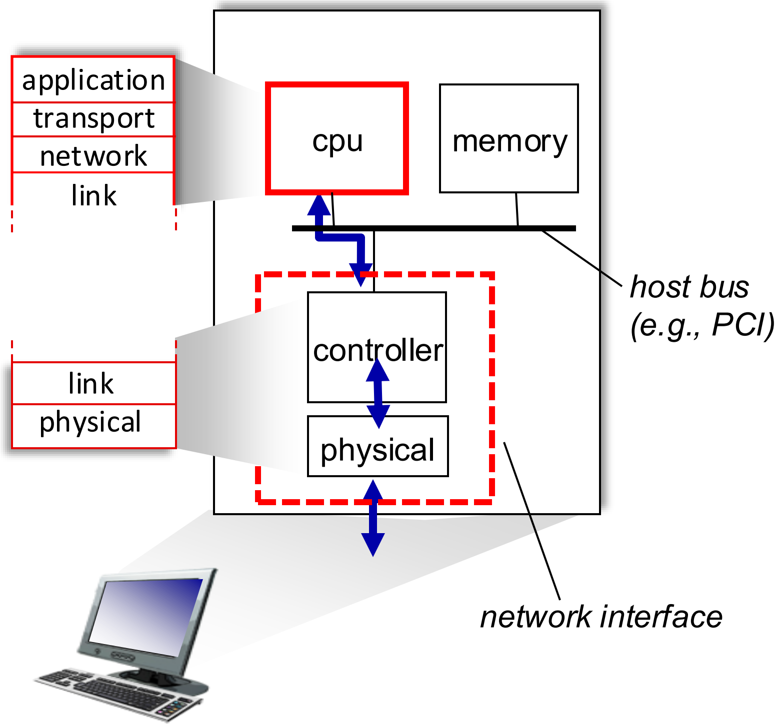
- Encoding
- Framing
- Error detection
- Link access (Medium Access Control)
- Addressing

1. Encoding : bits to signal
2. Flow control
3. Addressing
4. Error detection
8. Error correction
6. Multiple Access Control
7. Framing

link layer has responsibility of transferring datagram from one node to physically adjacent node over a link

Where is the Link Layer?

- in each-and-every node
- link layer implemented on-chip or in network interface card (NIC)
  - implements link, physical layer
- attaches into host’s system buses
- combination of hardware, software, firmware



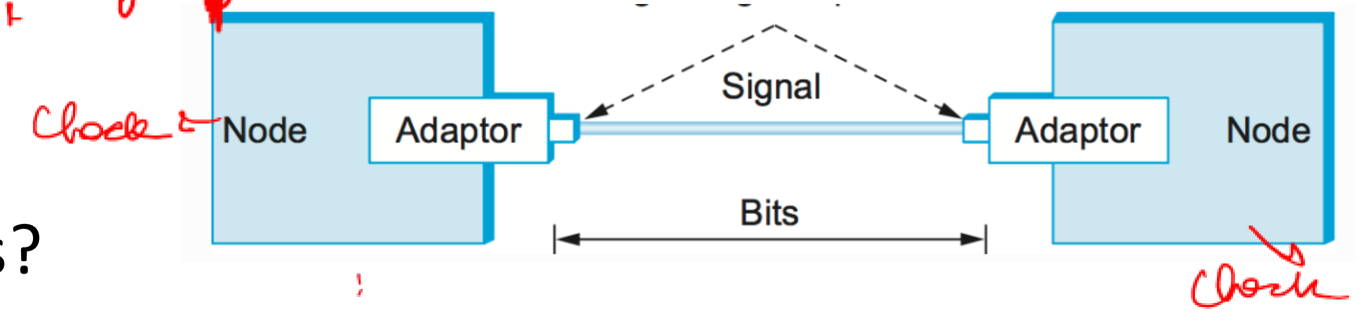
800 Series Network Adapters (Up to 100GbE)	Supports speeds up to 100GbE, available in PCI Express (PCIe) and Open Compute Project (OCP) form factors.
800 Series Controllers (Up to 100GbE)	Supports speeds up to 100GbE.
700 Series Network Adapters (Up to 40GbE)	Supports speeds up to 40GbE, available in PCI Express (PCIe) and Open Compute Project (OCP) form factors.

Encoding

0.2, 0.8, 0.3, 0.7 → 0.5

0 1 0 1

- Converting bits to signals
- How to convert bits to signals?



(1). Baseline wander

(2) clock recovery

Non-Return to Zero (NRZ)

Consecutive 1s