3. Delays, Throughput, Protocol Layers

23 January 2025 08:46

NOTE: Packet delay

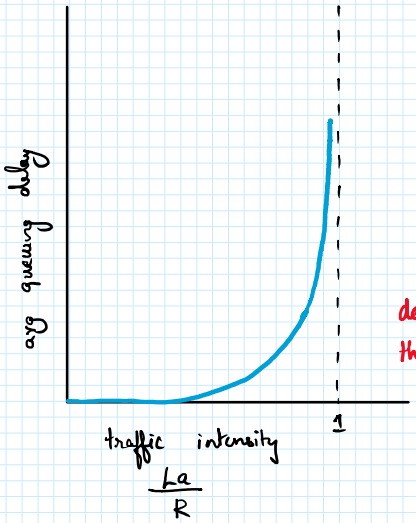
$$d_{nodal} = d_{processing} + d_{queue} + d_{transmission} + d_{propagation}$$

- $d_{processing}$: check bit errors, determining output link; < msec
- d_{queue}: waiting at o/p link for transmission; micro-milli
- $d_{transmission}$: packet length (bits) / link transmission rate $\left(\frac{L}{R}\right)$; NOTHING to do with distance
- $d_{propagation}$: physical length / prop. speed ($\sim 2 \times 10^8$ m/s) $\left(\frac{d}{s}\right)$; NOTHING to do with packet size or transmission rate

QUEUING DELAY

- Can vary from packet to packet
- Because of this, it is defined by average, variance, P(exceeding certain value) etc

depends on:
- Rate of traffic arrival at queue
- Transmission rate
- Nature of traffic (periodic/bursts)



R: link bandwidth (bps)
L: packet length (bits)
a: average packet arrival rate (pps)
La: avg rate at which bits arrive

design system so that intensity always < 1

NOTE: traceroute program
Measures delay by sending 3 probe packets to every router and measuring gap b/w transmission & reply

$\frac{La}{R} > 1$: more bits arriving than can be serviced

$\frac{La}{R} \leq 1$: nature of arriving traffic

$\frac{La}{R} \sim 0$: almost no queuing delay

Packet loss

- If buffer is full, packet is dropped/lost
- Packet may be retransmitted → previous node / source end system

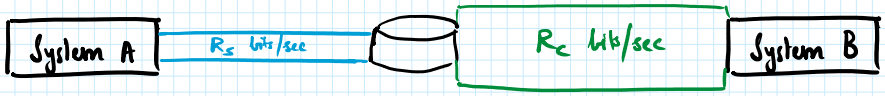
[OR]

not at all.

THROUGHPUT → instantaneous / average

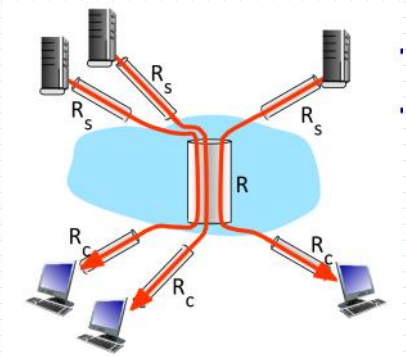
Rate at which bits are being sent from sender to receiver

Rate at which bits are being sent from sender to receiver



NOTE: Bottleneck link

Total throughput from sender to receiver → transmission rate of bottleneck link



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

Throughput = $\min\{R_s, R_c, R/n\} = \min\{R_s, R_c, R/10\}$ (these)

PRACTICE

$L = 4000$ bits

① $d_{trans} = \frac{L}{R_1} = \frac{4000}{10\ Mbps} = 4 \times 10^{-4}\ s$

② $\frac{10^8}{3 \times 10^8} = 0.33 \times 10^{-5} = 3.33 \times 10^{-6}\ s$

③ Assume $d_{processing} = 0$

$d_{queuing} = 0$

$d_{tot} = 4 \times 10^{-4} + 3.33 \times 10^{-6}$

PROTOCOL LAYERS

Complex system ⟶ modularised into layers

Open System Interconnection (OSI) Model

- Application
- Presentation
- Session
- Transport

Internet Protocol Stack

- Application
- Transport
- Network
- Link

Network

Link

Physical

Physical

Application

- End system to end system
- Packet: message
- DHCP, IMAP, SMTP, FTP, HTTPS, POP, SSH etc.

Transport

- Process-process information exchange
- TCP / UDP → connection-less → lightweight
  - → flow control
  - → congestion control
  - → connection oriented
- Packet: segment

Network

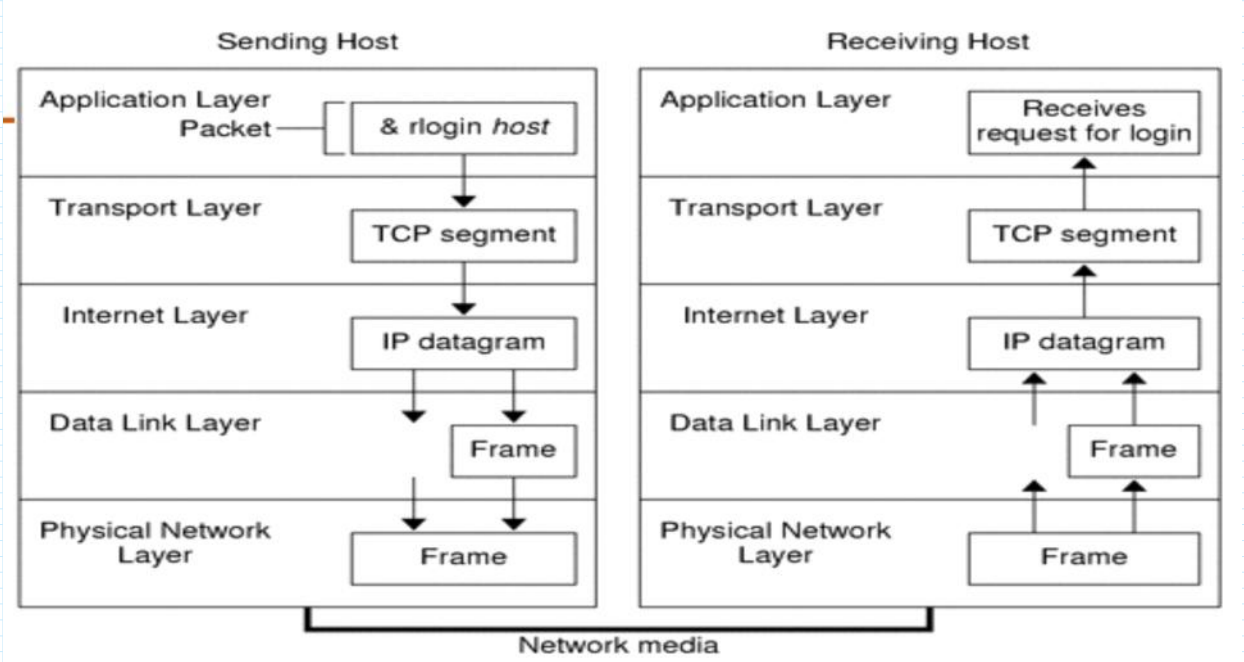
- Decides on path taken by data packets, manages routing
- IP, routing algos
- Packet: datagram
- Gets transport layer segment + destination address

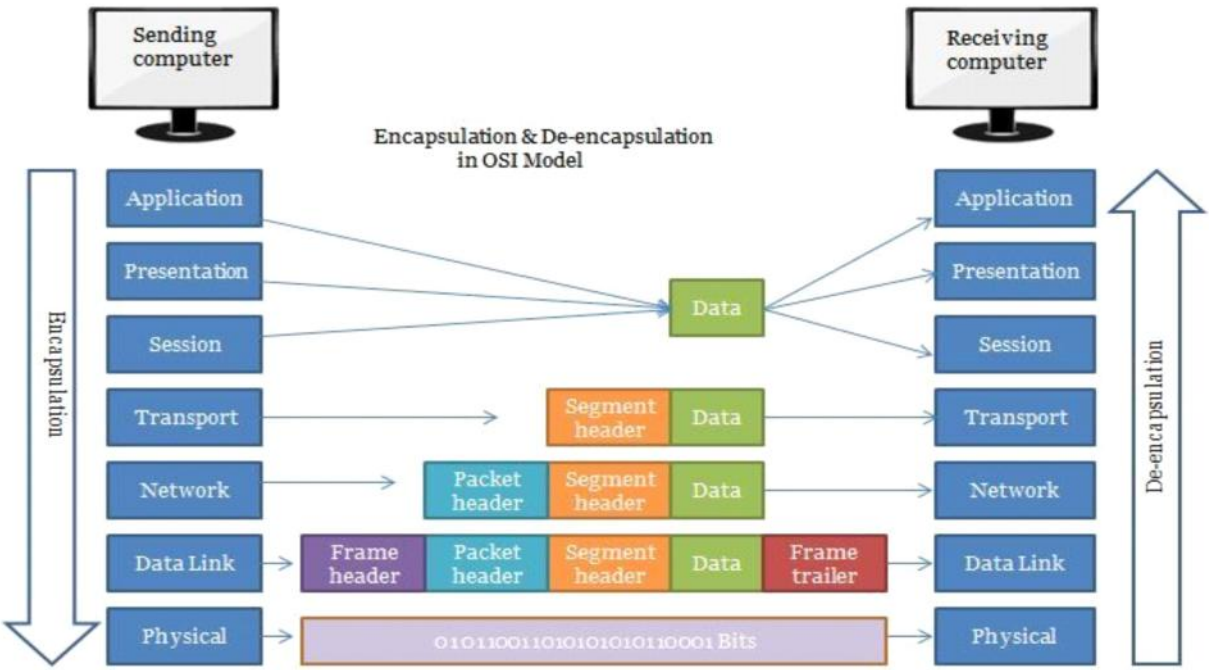
Link → uses MAC address

- Forwards packets locally from one link to another
- Ethernet, 802.11 (WiFi), PPP
- Packets: frames

Physical

- Raw bit data over physical transmission medium





ENCAPSULATION & DE-ENCAPSULATION

- Message goes through each layer
- Each layer adds a header
- Message is sent to end system
- Each layer's header is removed
- Original message recieved

} Encapsulation

} De-encapsulation

NOTE: Intermediate devices

Switch: Link layer
Operates in same network

Router: Network layer
Can modify headers to send to different network