University of Southern California

Viterbi School of Engineering

EE450
Computer Networks

Network Performance and Latency Measures

Shahin Nazarian

Spring 2013

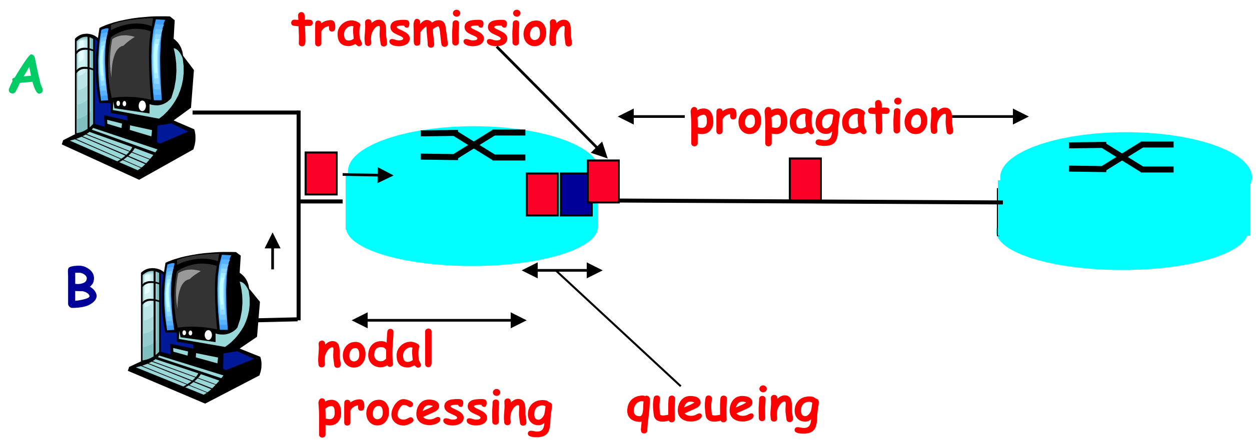
Network Performance Measures

- Performance analysis
  - Delay (latency) seconds
  - Throughput (bits per second) bps
    message unit/unit of time
- Performance optimization

Delay, throughput
Analysis: measuring delay thru

Four Sources of Packet Delay

- 1. Nodal processing delay
  - To check bit errors
  - To determine output link
- 2. Queuing delay
  - Time waiting at output link for transmission
  - Depends on congestion level of router



- There is no deterministic equation for them, therefore they are studied statistically

Delay in packet-switched networks

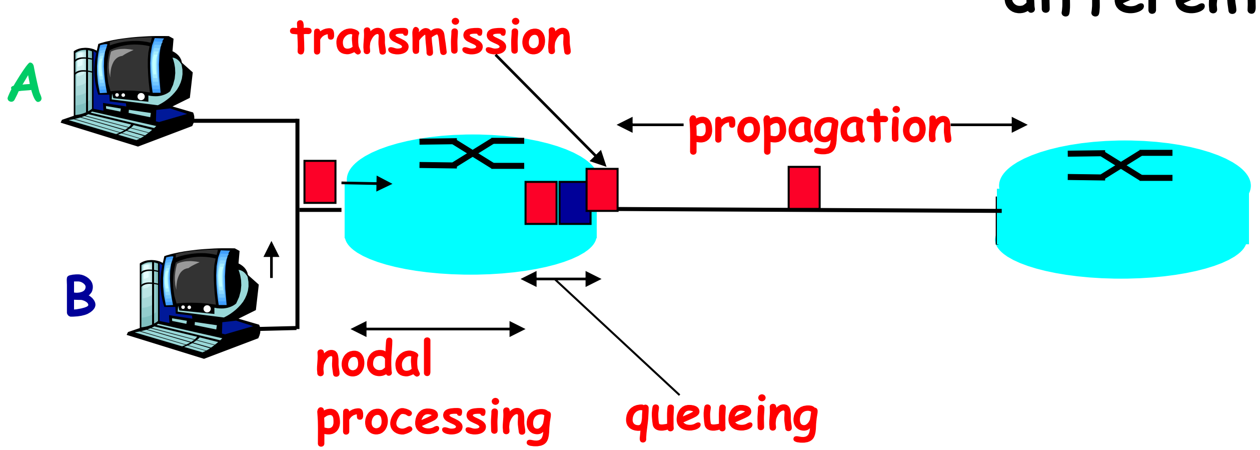
3. Transmission delay

- R = link **BW** (bps)
- L = packet length
- time to send bits into link = d_{trans}

4. Propagation delay

- d = length of physical link
- s = propagation speed in medium (2×10^8 to 3×10^8 m/s)
- propagation delay = d_{prop}

Note: s and R are *very* different quantities!

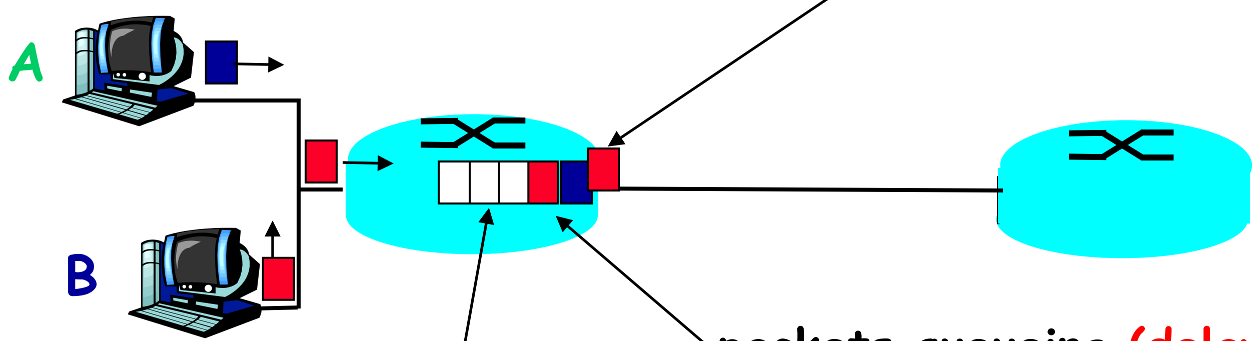


How Do Loss and Delay Occur?

Packets *queue* in router buffers

- If packet arrival rate to link exceeds the output link capacity, packets queue, and wait for their turn

0 to infinity



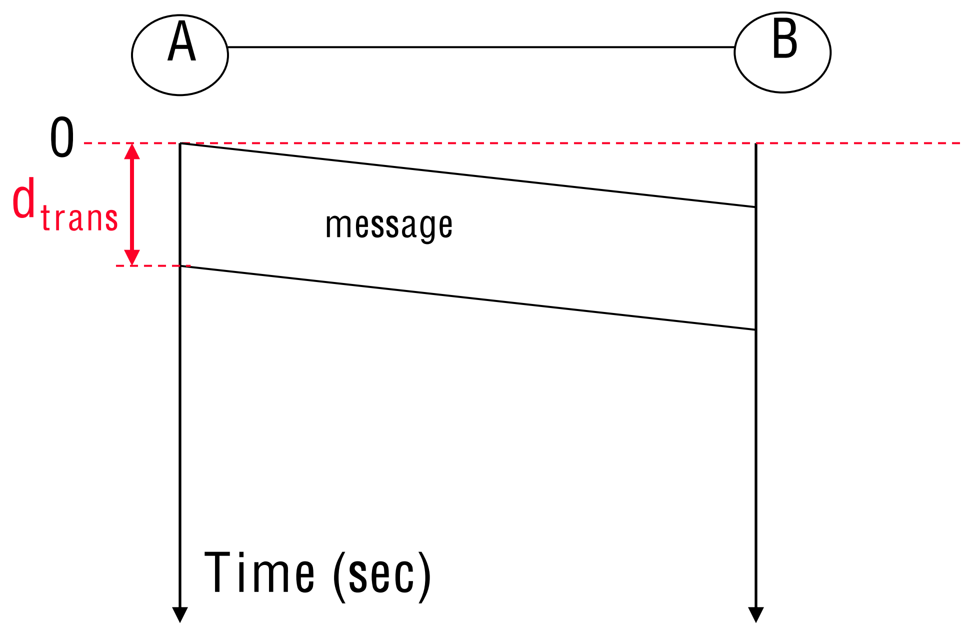
Transmission Delay, d_{trans}

- d_{trans} is time it takes to push into the link (transmit) all the message bits

d trans = L (bits)/ R (bps)

  - d_{trans} can be significant for low-speed links

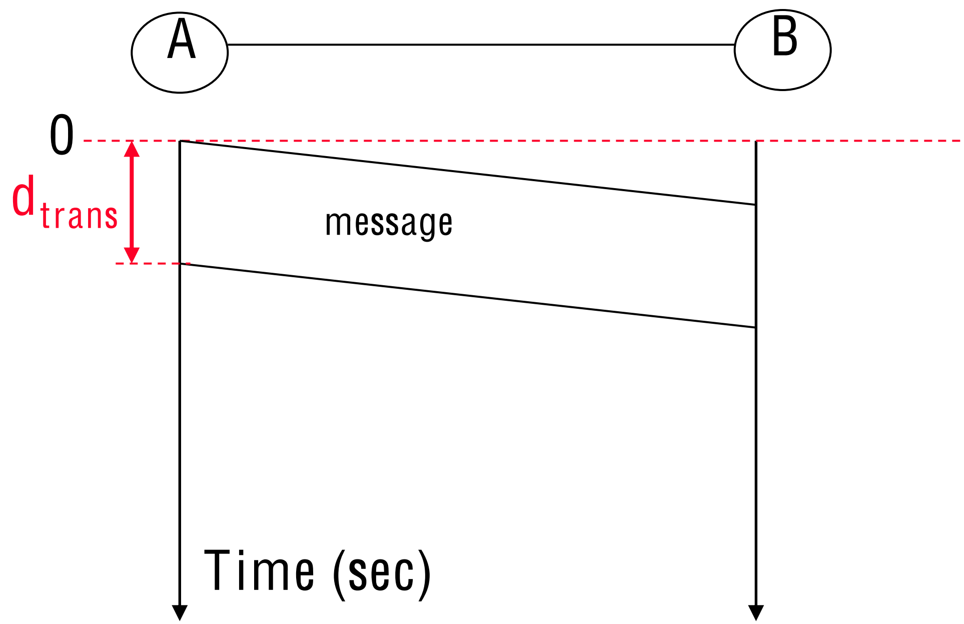
bits/meter



d_{trans} (Cont.)

- Note: d_{trans} is only about transmitting the message, i.e., sending the message to the link
- It does not include the time it takes for message to propagate and be received by the other end of the link

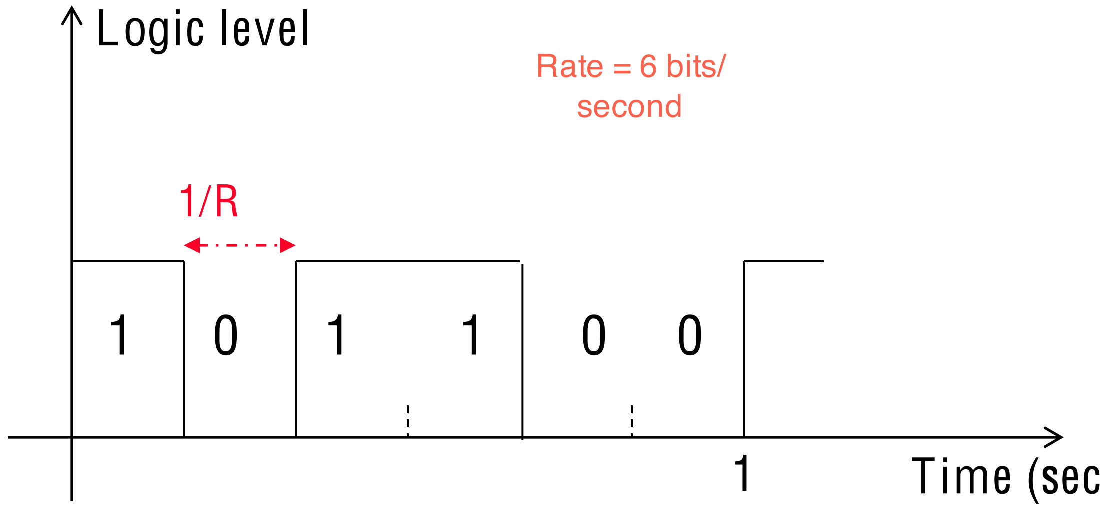
L L/R = dtrans



Transmission Delay (Cont.)

- **Example:**

transmition delay for 128 bts is 128bits/(6bits/ second) = 21.333 seconds



- **Bit rate, R**
- **The transmission delay for a message of length 128bits**
- **The transmission delay for one bit**
- **Bit duration optimization**

Transmition delay for one bit is called bit duration

Transmission Delay (Cont.)

- Example: Host H1, is downloading the season finale of Dexter from its peer, H2
  - The download time is not the same as the transmission delay of H2

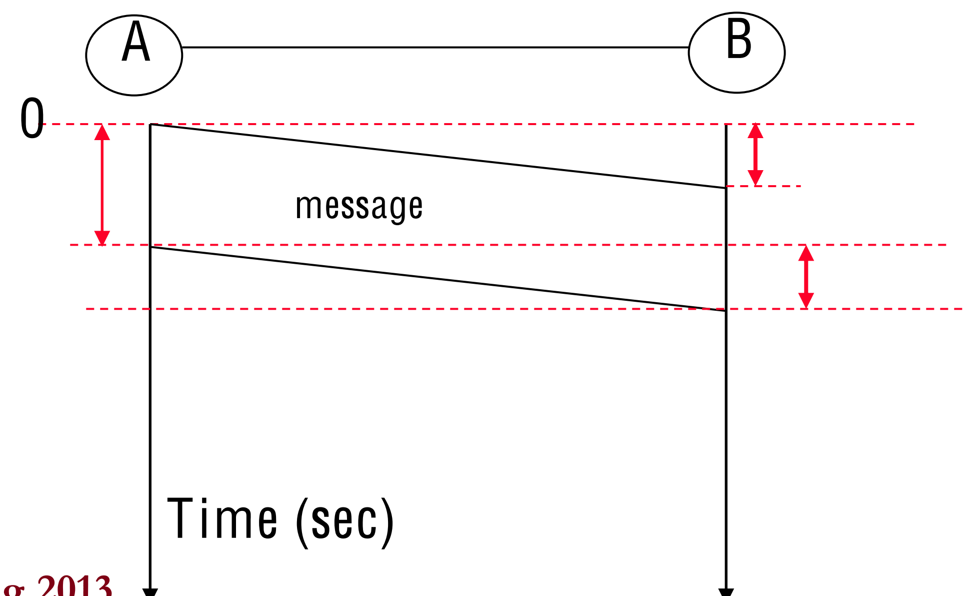
  - Reminder: d_{trans} does not include the time it takes for message to propagate and be received by the other end of the link)

Propagation Delay, d_{prop}

- d_{prop} is the difference between the time one bit is transmitted and the time, that one bit is received. In other words, the time for one bit to propagate from source to destination

  where d (m) is the link length and s (m/sec) is the propagation speed

propagation delay =

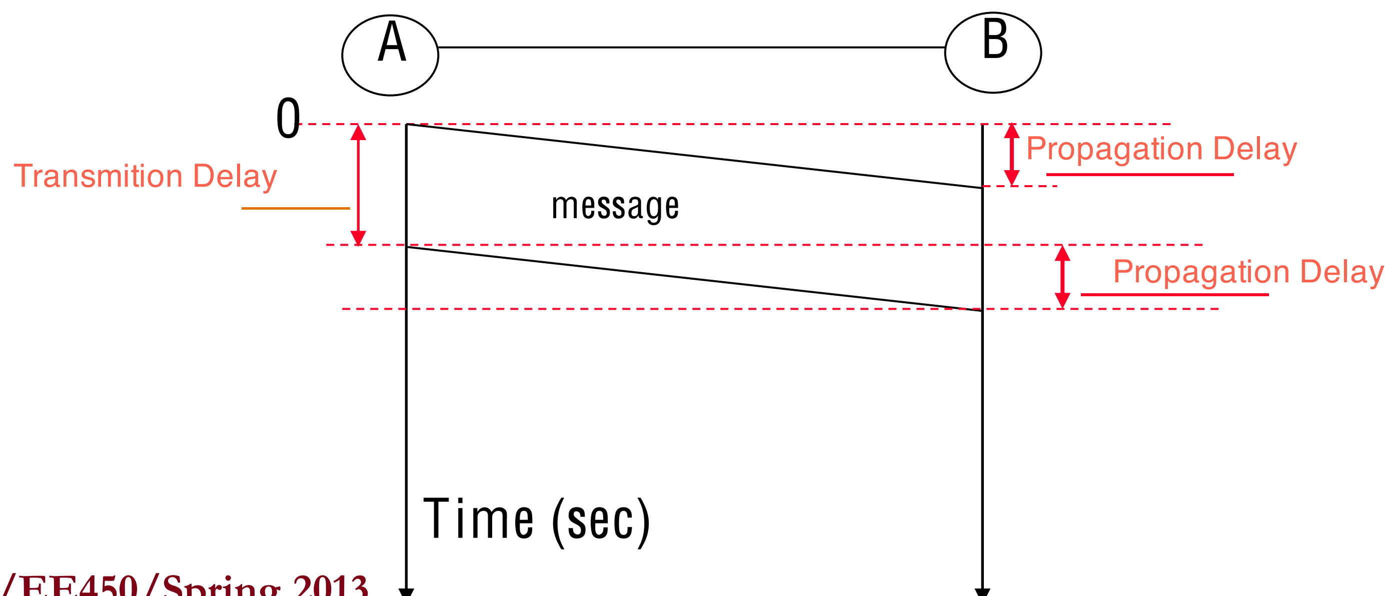


d_{prop} (Cont.)

- s depends on the physical medium of the link and ranges from 2×10^8 to 3×10^8 m/sec

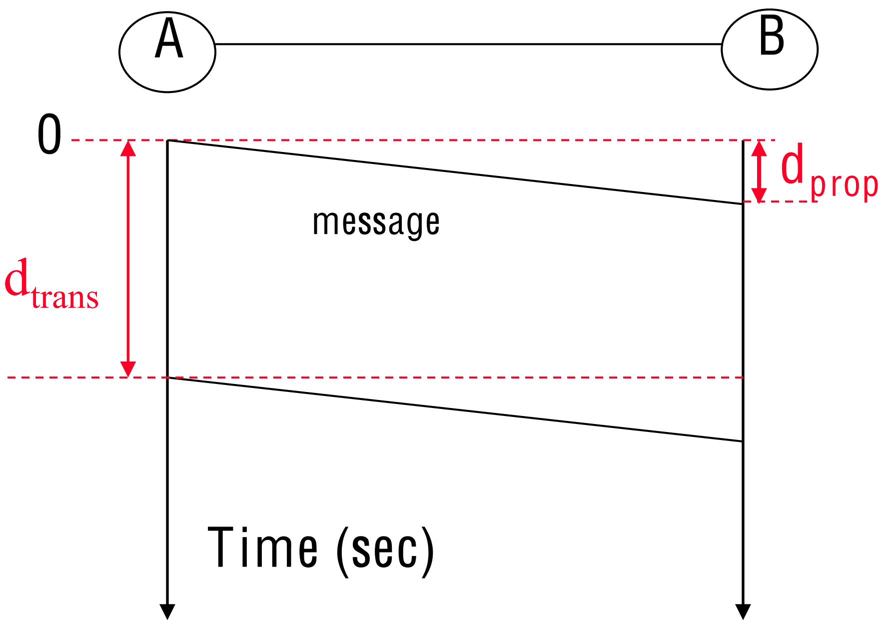
  [e.g., in WANs d_{prop} is in the order of msecs]

- Note: d_{prop} is dependent on the link length (meters)

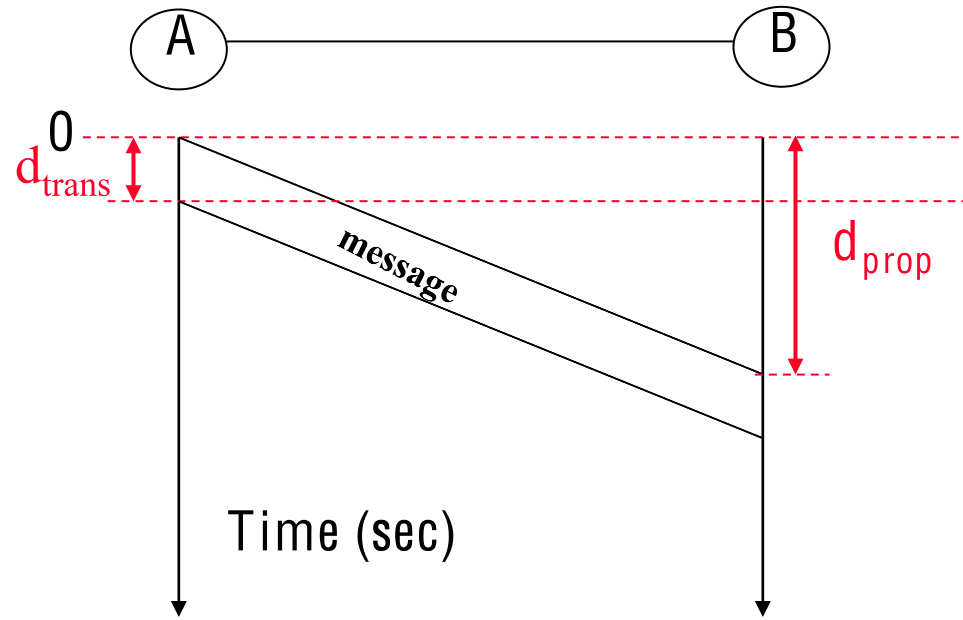


Transmission and Propagation Delay Comparison

- **Example: show a case where $d_{trans} > d_{prop}$ and another where $d_{trans} < d_{prop}$**



$d_{trans} > d_{prop}$

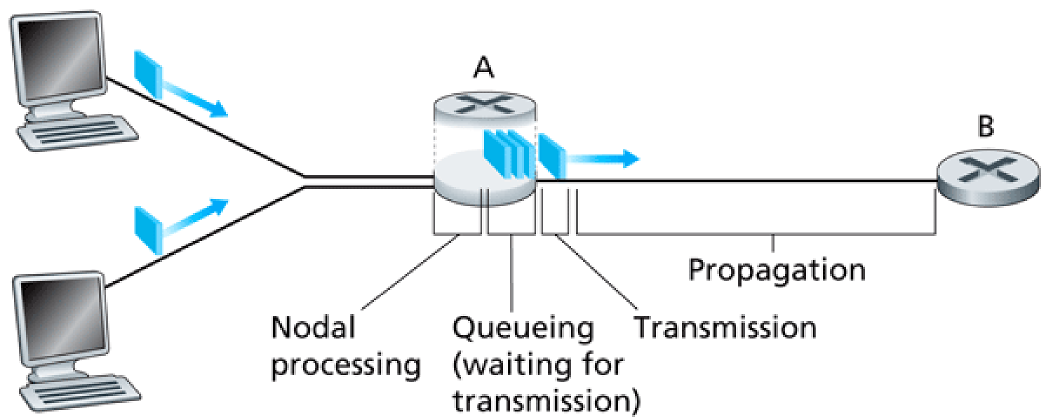


$d_{trans} < d_{prop}$

- **Note: d_{trans} and d_{prop} have no relation between each other (compare their equations)**

Queuing Delay, d_{queue}

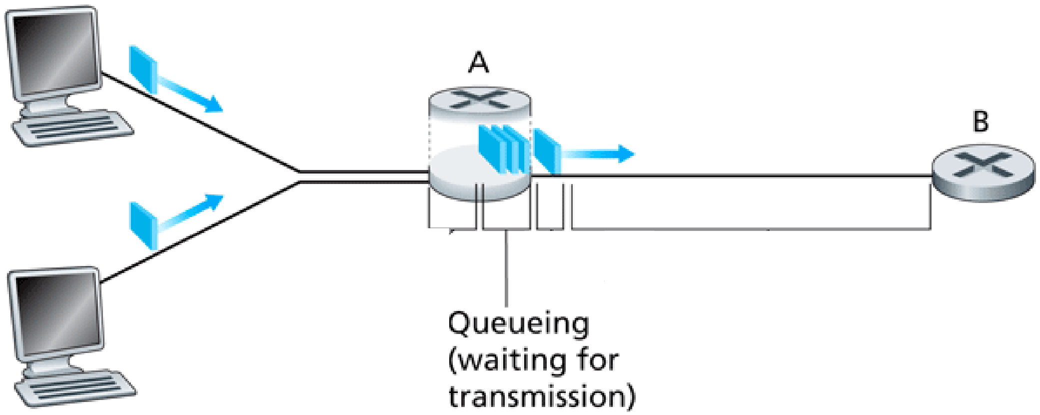
- For each attached outbound link, Router A has an output buffer (aka the output queue) which stores packets that the router is about to send into that link
- When a message (a packet) arrives at router A from an upstream node, router A examines the packet's header to determine the appropriate link to direct the packet to
- In the following example, router A chooses the link that leads to router B. A packet can be transmitted on a link only if there is no other packet currently being transmitted on the link and if there are no other packets preceding it in the queue



 (Note: Router A has a buffer (aka a queue))

d_{queue} (Cont.) -> Infinity

- The amount of time a packet needs to wait in the queue of a router
- If the queue is empty, then d_{queue} is zero, however in case the network is busy, packets experience a nonzero d_{queue}
- If network is congested the queue of router A will eventually become full, and if more packets come, the buffer will drop them typically based on the last come, first drop criteria



(Note: Router A has a buffer (aka a queue)

Processing Delay, d_{proc}

Random

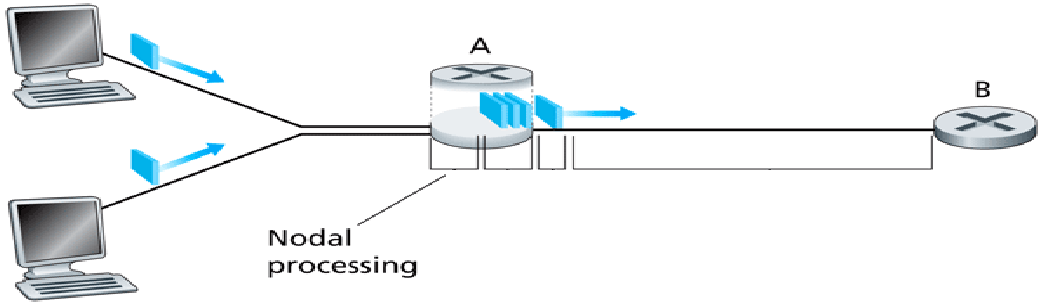
- Nodal processing delay is the time required to process the packet at the source prior to sending, at any intermediate router or switch, and at the destination prior to delivering to an application

L/R d/s

- The main components for processing delay, d_{proc} (for node A)
  - The time A needs to check for bit-level errors in the packet that may have occurred in transmitting from upstream to router A

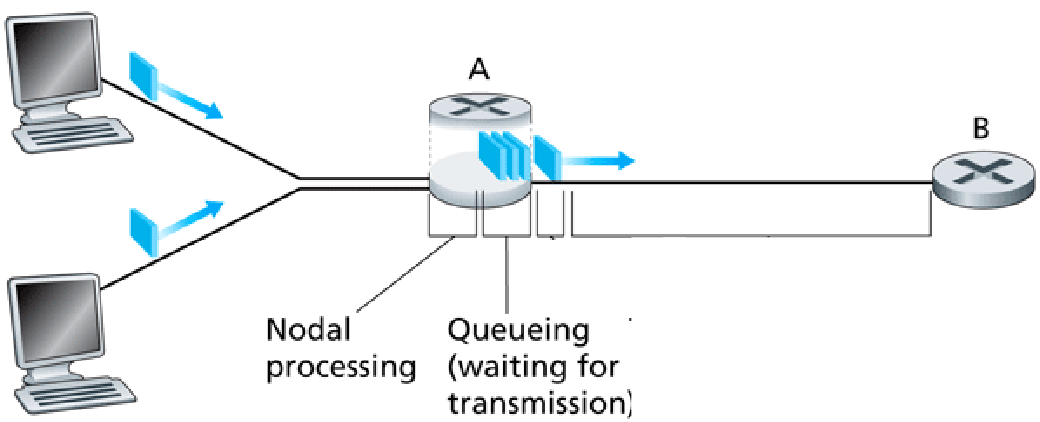
header

  - The time A needs to examine packet's header and determine where to direct the packet (routing decisions)
- Note: the router processor is very powerful to be able to handle the routing decisions with low d_{proc}



Queuing Delay and Processing Delay

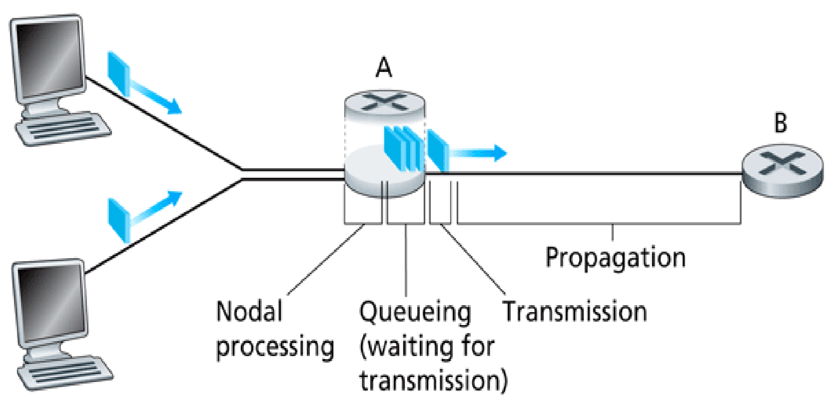
- Random variables
  - Depend on the number of packets in the queue (buffer) and how congested the network is, etc.
- These two parameters should hence be modeled probabilistically and statistically



Nodal delay (i.e., the Delay of a Single Router

- d_{proc} = processing delay
  - Typically a few µsecs or less
- d_{queue} = queuing delay
  - Depends on congestion
- d_{trans} = transmission delay = L/R,
  - Significant for low-speed links
- d_{prop} = propagation delay = d/s
  - A few µsecs to hundreds of msecs

$$\boxed{d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}}$$



Nodal delay

$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

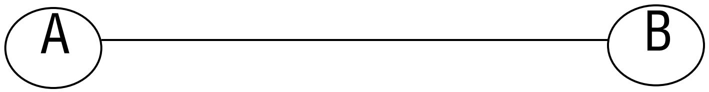
- The contribution of these delay components can vary significantly
  - d_{proc} is often negligible; however it strongly influences a router's max throughput (i.e., the maximum rate at which a router can forward packets)
  - d_{trans} is typically negligible for rates of 10Mbps and higher (e.g., LANs) but can be 100s of msecs for large Internet packets over a dialup service
  - d_{prop} can be negligible if the two routers connecting the link are close, e.g., on the same university campus, however it can be the dominant component of d_{nodal} if the routers are connected by a geostationary satellite link (Note: the altitude for geo satellites are in the order of 22000miles or 35000km)
  - d_{queue} varies from packet to packet and is the most complicated component of d_{nodal}

End-to-end delay

- For source and destination hosts directly connected together without a router

e_2_eAB = nodel _delayA

dprocA + dqueueA +



dtransA + dpropAB

- End-to-end delay vs nodal delay

Sender_receiver sender

- Note that for an uncongested network d_{queue} is assumed to be negligible

dqueue -> 0

dproc -> 0

End-to-end delay (Cont.)

- For a source and destination host connected through one router the nodal delay of that router needs to be added as well

e_2_eAB = dprocA + dqueueA + dtransA + dpropAR+

= Nodal Delay (A) + Nodal Delay (R)

- In general, there are N routers (with identical delay parameters and transmission rate) between the source and destination hosts

e_2_e = dproAB + SUM(Dtrans)

(N+1) Dprop + (N+1) dtrans

End-to-End Delay (Cont.)

- **Random variable**

  -> dequeu = 0

  -> dproc = 0

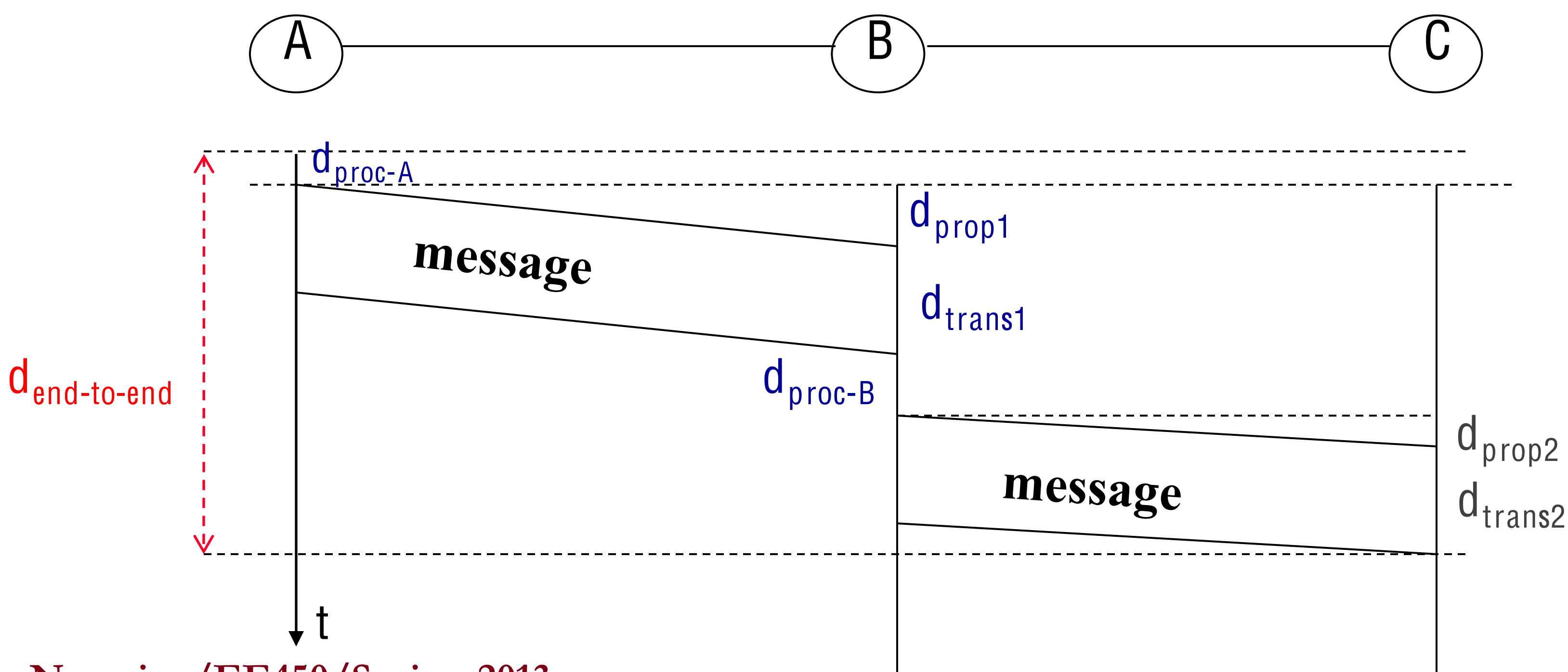
- **Example: Host H1, is downloading season 7 finale of Dexter from its peer, H2. Describe the relation between end-to-end delay and download time**

  e_2_eh2h1 = NDh2

  Uploading is usually slower than downloading

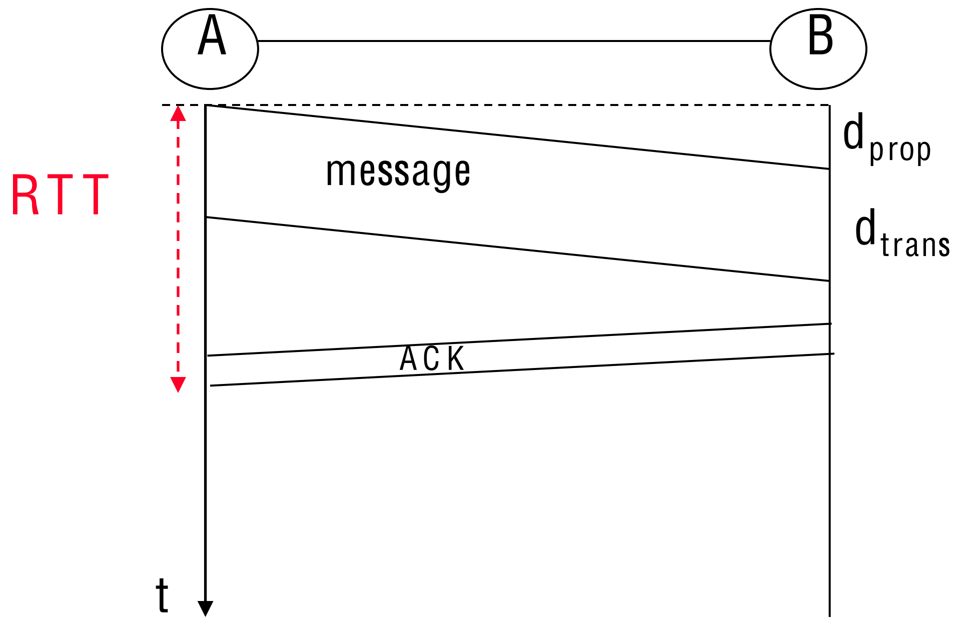
End-to-End Delay

- Note: end-to-end delay is the difference between the time the first bit of the message is processed and transmitted by the source and the time the last bit of the message is received by the final destination host
- Example:



Round Trip Time (RTT)

- RTT a.k.a. the Response Time is the time it takes for a message, say a packet, to travel from a source to a destination and then acknowledgement sent (from the destination) back to the source
  - RTT depends on link length, message length, propagation speed, nodal processing, network load (congestion), link direction (upstream, downstream,) etc.

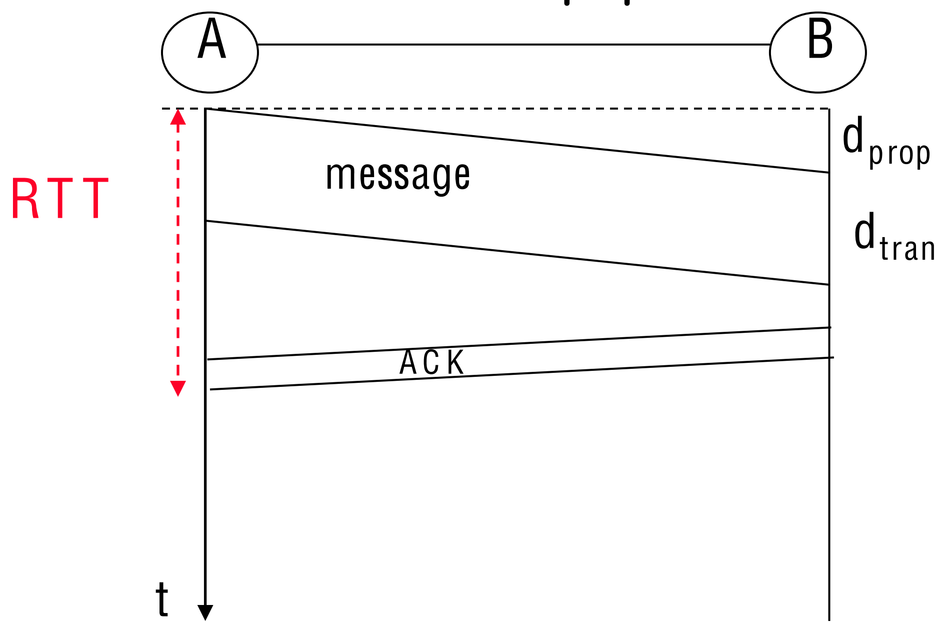


RTT (Cont.)

- Assuming the dominating components of RTT are the propagation delay as well as message transmission delay (i.e., ignoring other components) RTT becomes $2d_{prop}$

$$RTT = d_{trans\text{-}msg} + d_{prop} + d_{proc} + d_{queue} + d_{proc} + d_{trans\text{-}ACK} + d_{prop} \cong$$
$$d_{trans\text{-}msg} + 2d_{prop}$$

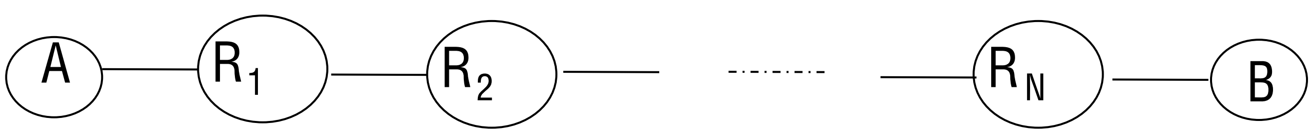
- if $d_{trans\text{-}msg}$ is also neglected, then RTT $\cong 2d_{prop}$
- Here it was assumed that both message and ACK experience the same propagation speed and link length



RTT (Cont.)

- Using the same assumption (i.e., considering d_{prop} as the only component in RTT for the case that A and B are connected through one router (and with identical link medium and length)
- RTT $\cong 4d_{prop}$
  - Look at RTT as the number of links $\times$ 2 $\times$ d_{prop}
- If A and B are connected through N routers:

$$RTT = \text{number of links} \times 2 \times d_{prop} \approx (N-1+2) \times 2 \times d_{prop}$$
$$\approx 2(N+1)\, d_{prop}$$



Throughput

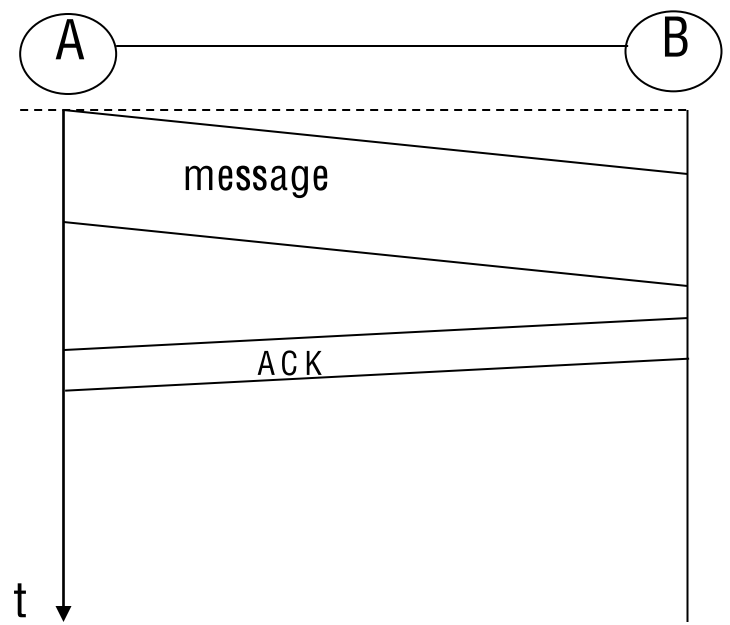
- **Throughput** (bps) is the ratio of the message length to RTT; in other words, it is the number of reliably delivered bits per unit time

Throughput = L (bits) / RTT (sec)

Bit width dAB/BWxdelay

Throughput ≅ $L/ (d_{trans-msg} + 2d_{prop})$

- Does message length include the number of bits in the ACK? *Answer*: No, ACK is not part of the message, e.g., a longer ACK would not result in a higher throughput!



* L = message length in number of bits

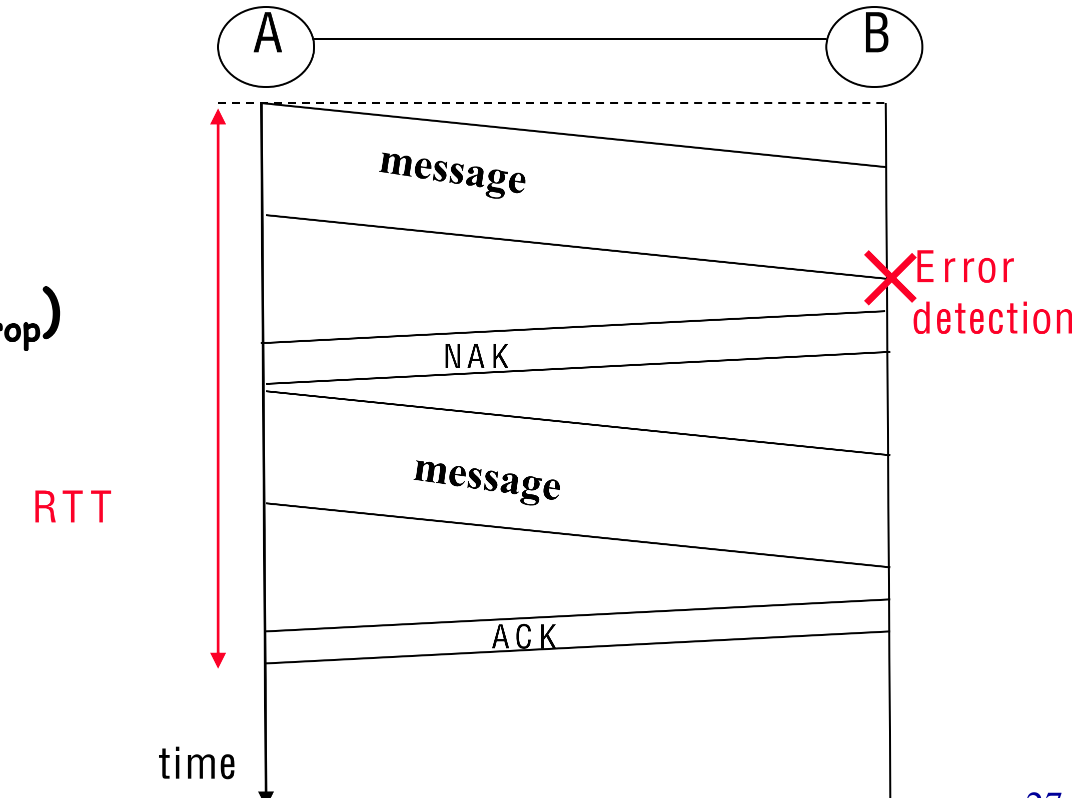
Throughput (Cont.)

- Note: the throughput definition is based on reliable data delivery, meaning if the destination detects an error, message needs to be retransmitted. Source will know about this through a handshaking process. Throughput would then be smaller, because of a higher RTT (for the same message length)

Example:

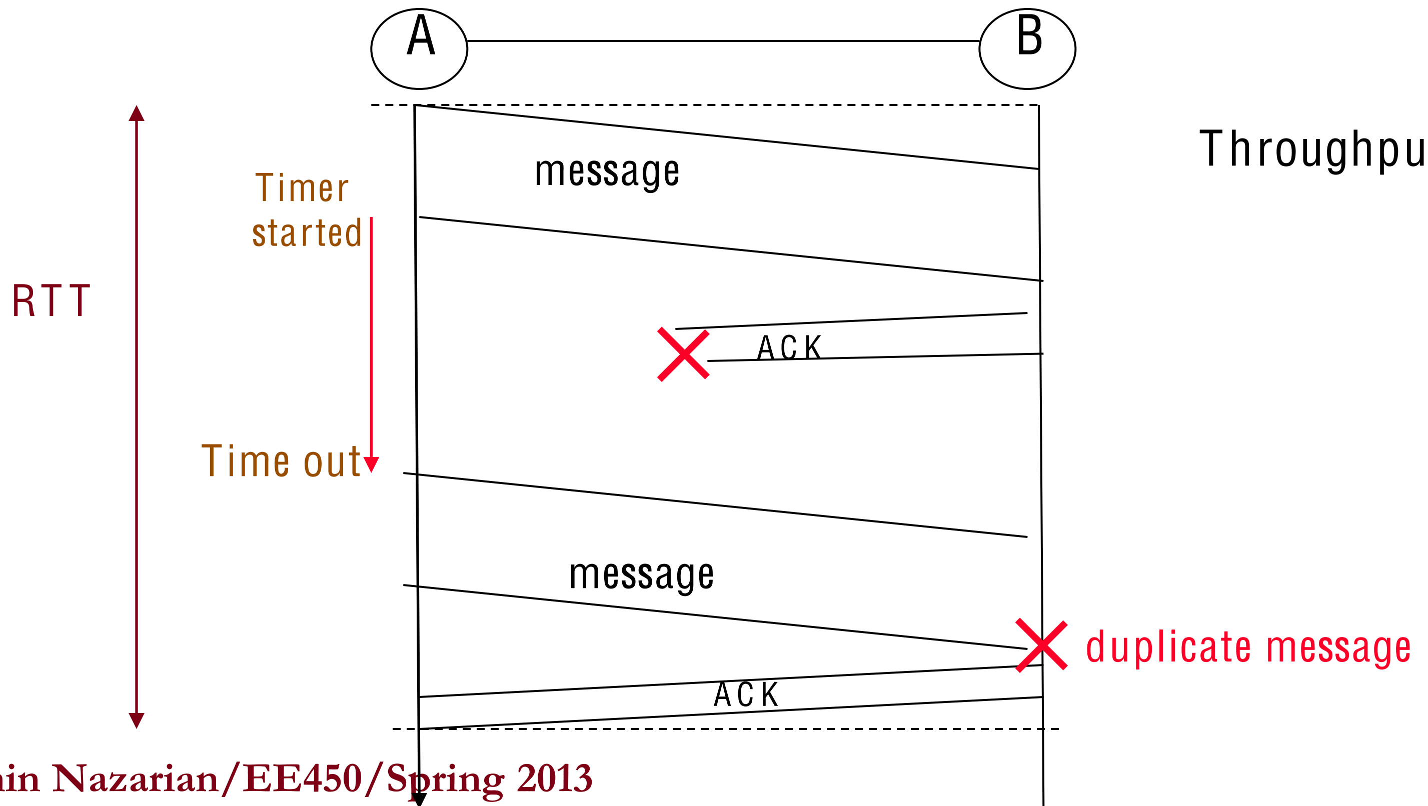
Throughput $\cong$

$$L / (2d_{trans} + 4d_{prop})$$



Throughput (Cont.)

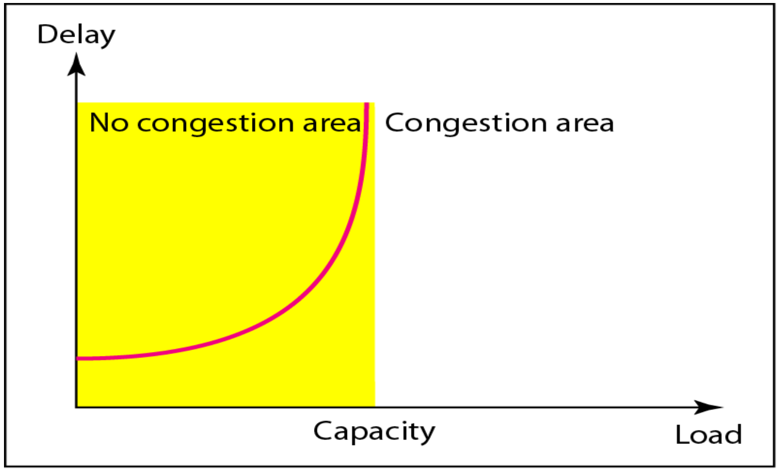
- Is it possible that ACK itself becomes erroneous? Although the message has a higher chance of becoming erroneous (because it is longer than ACK) it is still possible that ACK has error in it, or ACK message may be lost and not received by the source. Source has a timeout policy, which makes it re-send the message after not hearing from the destination within a certain amount of time



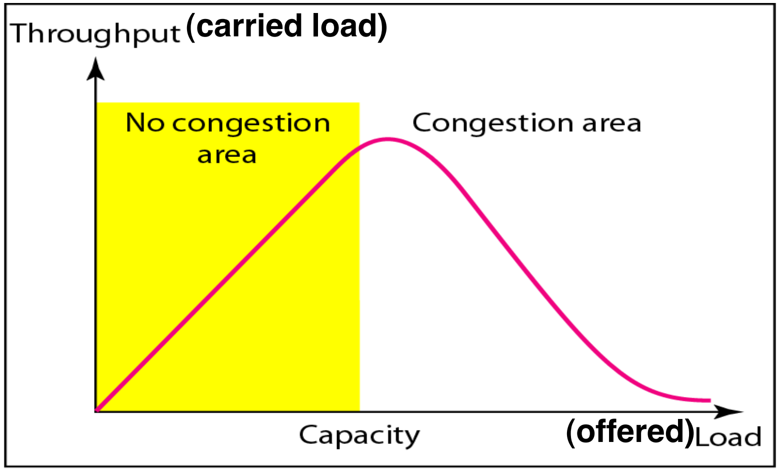
Throughput = L/ RTT

Throughput (Cont.)

- The number of bits that are sent per unit time (offered load) is not equal to the number of bits actually delivered (carried load or throughput)
- Note: **Offered load** is what is dumped into the network (considering all the users, not just one user) and **throughput** (**carried load**) is what is delivered both are in bits per second or packets per second



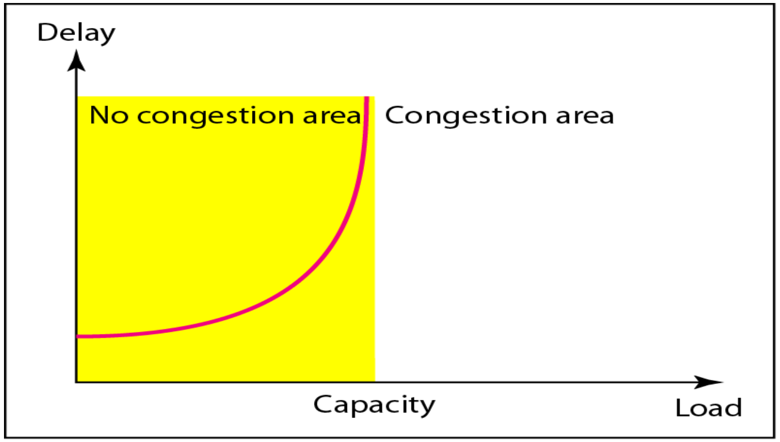
a. Delay as a function of load



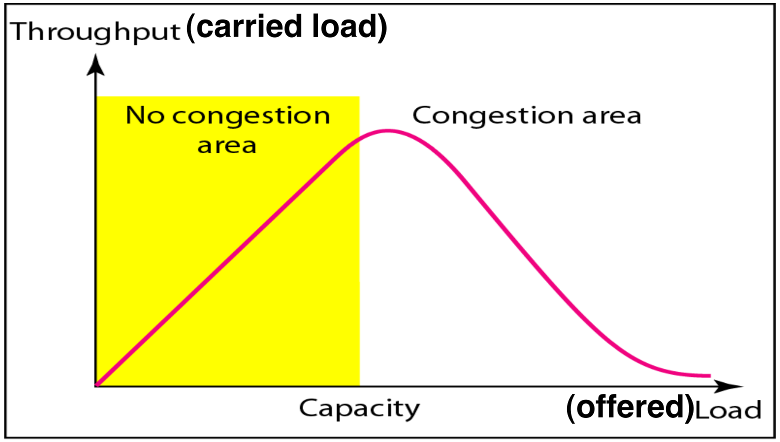
b. Throughput as a function of load

Principles of Congestion Control

- For small offered loads, they will all be delivered, because network is not congested, however network users may add to the offered load, and eventually router buffers pile up and become full, therefore they start dropping messages
- Users then attempt to retransmit their messages and this aggravates the-already-bad situation, throughput keeps going down with router buffers being full, at last network gets congested



a. Delay as a function of load



b. Throughput as a function of load

Bandwidth

- **Bandwidth** is the number bits per second that can be transmitted on a link. It's the same as the bit rate, R, we used in d_{trans} equation
- Note 1: Bandwidth is typically measured in hertz
- Note 2: Unlike throughput the definition of bandwidth does not consider the message reliability (see next two slides)
- Example: For an Ethernet with bandwidth of 10Mbps, the number of bits that can be transmitted in every second is 10 million, i.e., the bit rate of the links is 10Mbps; however when the rate is measured in practice (i.e., the throughput) it may be way lower than 2Mbps

Do not confuse bandwidth and throughput!

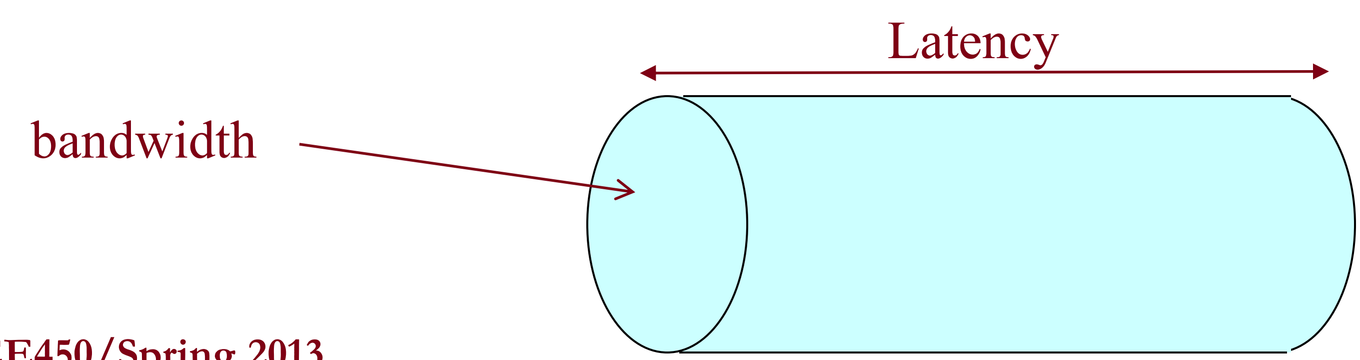
- They are two of the most confusing terms used in networking. While we try here to give precise definition of each term, you should know that people may use them interchangeably
- First of all, bandwidth is literally a measure of the width of the frequency band, e.g., a telephone line that supports a frequency ranging from 300 to 3300Hz, is said to have a bandwidth of 3300-300=3000Hz, therefore when bandwidth is measured in hertz, it refers to the range of signals that can be accommodated
- The bandwidth of a data communication link is the number of bits per second that can be transmitted on the link, e.g., the bandwidth of Ethernet might be mentioned as 10Mbps

Bandwidth and Throughput (Cont.)

- A useful distinction might be made, however between the bandwidth that is available on the link and the number of bits per second that we can actually transmit over the link in practice. We tend to use the word throughput to refer to the "measured performance" of a system
- Thus because of various inefficiencies of implementation, a pair of nodes connected by a link with a bandwidth of 10Mbps might achieve a throughput of only 2Mbps, meaning that an application on one host could send data to the other host at only 2Mbps

Bandwidth-Latency product

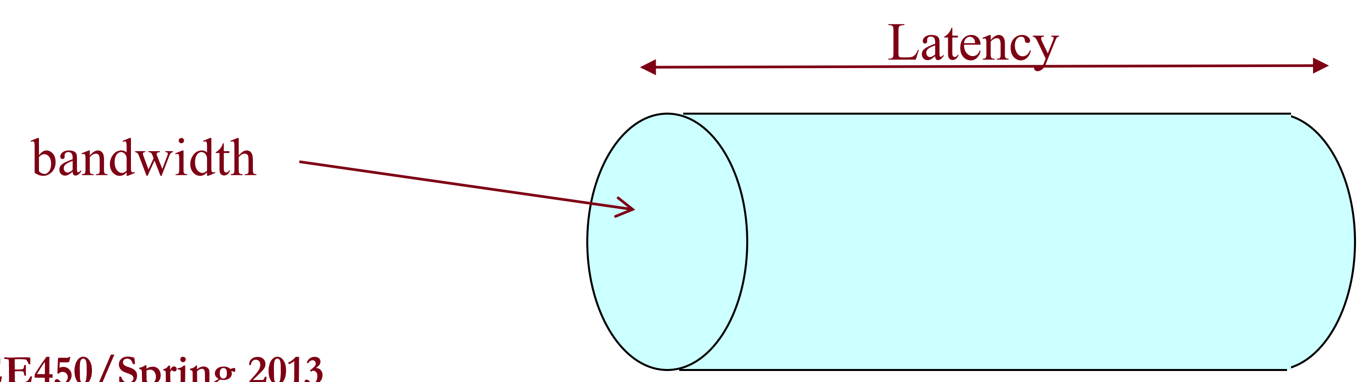
- It is useful to look at the two performance measures of network, i.e., bandwidth and latency (delay) as a product
- If we think of a channel between a pair of processes as a hollow pipe, where the latency corresponds to the length of the pipe and the bandwidth gives the cross-area of the pipe, then **Bandwidth × Latency** [measured in bits] gives the volume of the pipe, which is the maximum number of bits that can be in transit through the pipe at any given instance, i.e., how many bits fit in the pipe
- Note: latency (delay) may be considered as RTT or one way



Bandwidth × Latency product (Cont.)

- Example: A channel with one-way latency of 50ms and a bandwidth of 35Mbps is able to hold 1.75×10^6 bits:

  50×10^{-3} sec $\times 35 \times 10^6$ bits/sec $= 1.75 \times 10^6$ bits
- Bandwidth × Latency product is mainly used for performance tuning
- It is an important measure to know when constructing high performance networks because it corresponds to how many bits the sender must transmit before the first bit arrives at the receiver



Bandwidth × Latency product (Cont.)

- There are a certain number of bits that can be transmitted to the link during RTT, considering the link characteristics and the node bit rate. One cannot transmit more than this limit, and if one attempts to do so, it then needs to wait to get the ACK for those bits (the time then goes beyond that original RTT)

d/(max # bits)

Dprop = d/s