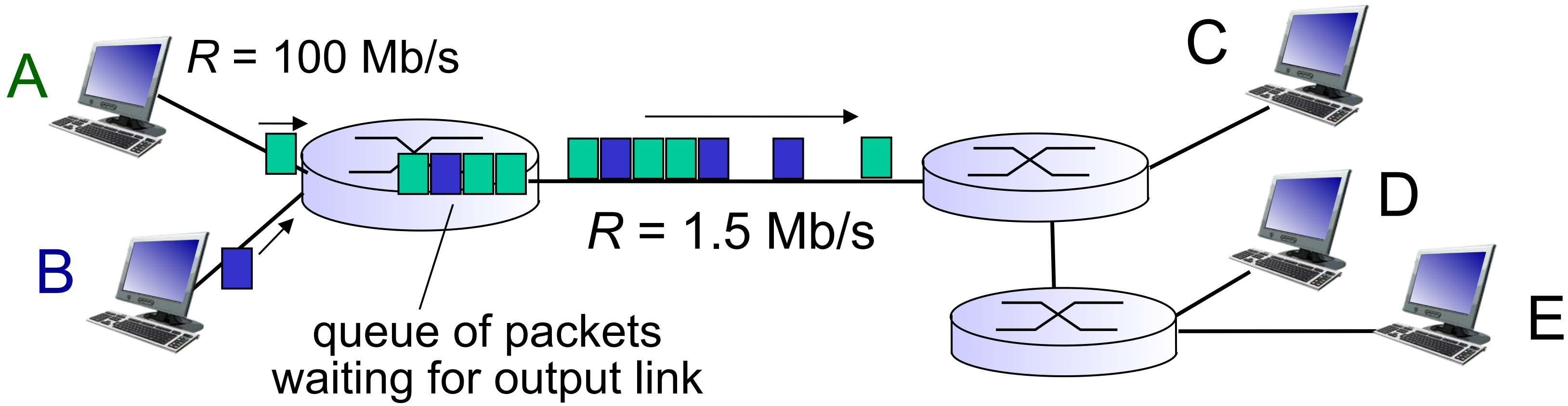
# Packet Switching: queueing delay, loss



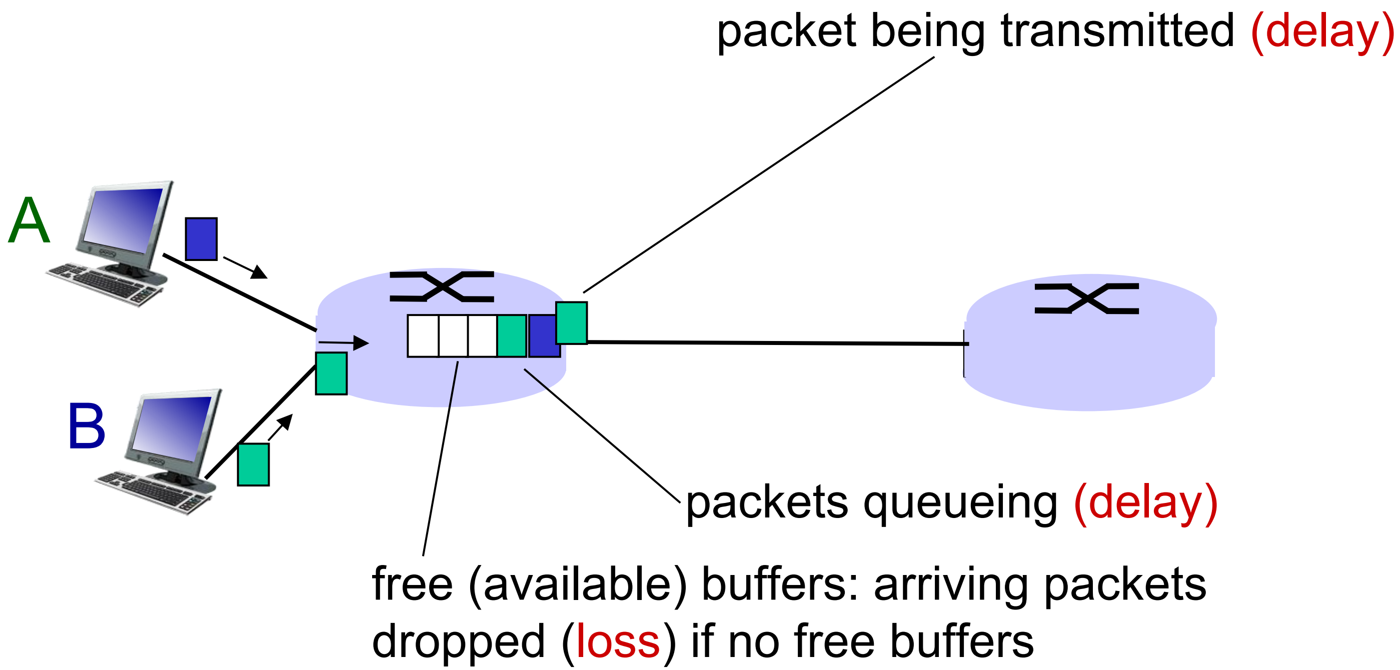

## queuing and loss:

- If arrival rate (in bits) to link exceeds transmission rate of link for a period of time:
  - packets will queue, wait to be transmitted on link
  - packets can be dropped (lost) if memory (buffer) fills up

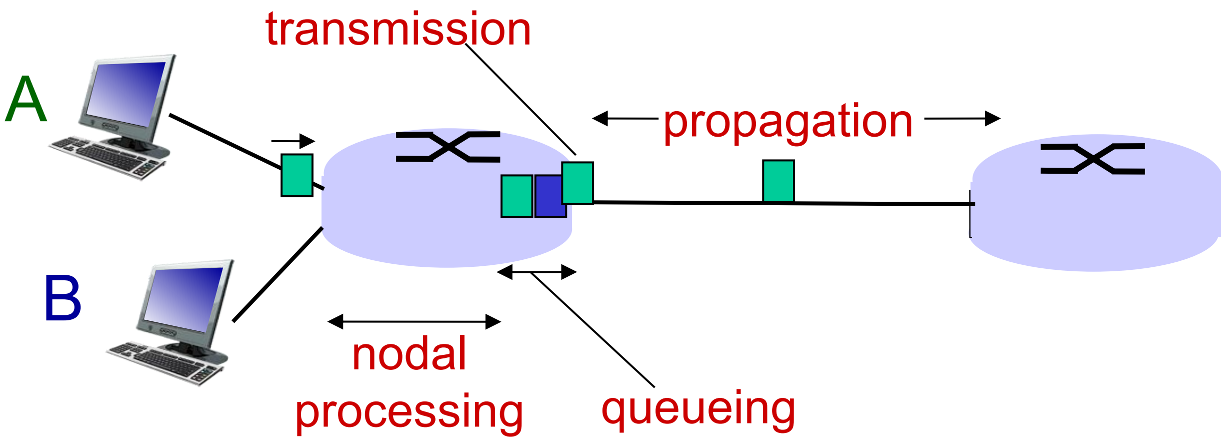
# How do loss and delay occur?

packets *queue* in router buffers

- packet arrival rate to link (temporarily) exceeds output link capacity
- packets queue, wait for turn



# Four sources of packet delay




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

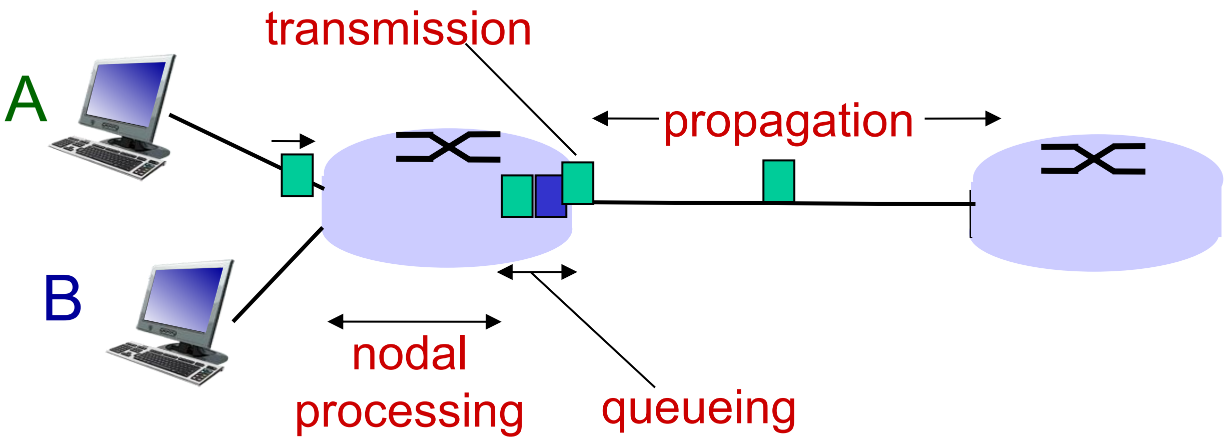
## $d_{proc}$: nodal processing

- check bit errors
- determine output link
- typically < msec

## $d_{queue}$: queueing delay

- time waiting at output link for transmission
- depends on congestion level of router

# Four sources of packet delay




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

**$d_{trans}$: transmission delay:**

- *L*: packet length (bits)
- *R*: link *bandwidth (bps)*
- $d_{trans} = L/R$

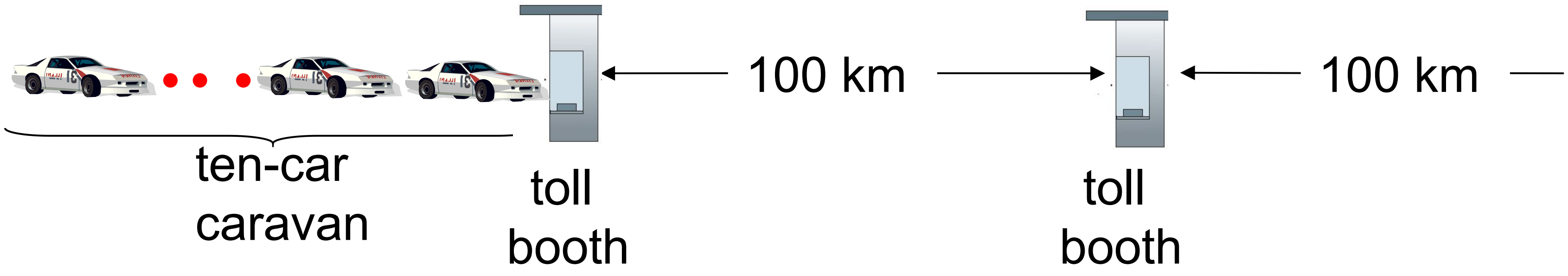
**$d_{prop}$: propagation delay:**

- *d*: length of physical link
- *s*: propagation speed in medium (~$2 \times 10^8$ m/sec)
- $d_{prop} = d/s$

$d_{trans}$ and $d_{prop}$ *very* different

* Check out the Java applet for an interactive animation on trans vs. prop delay

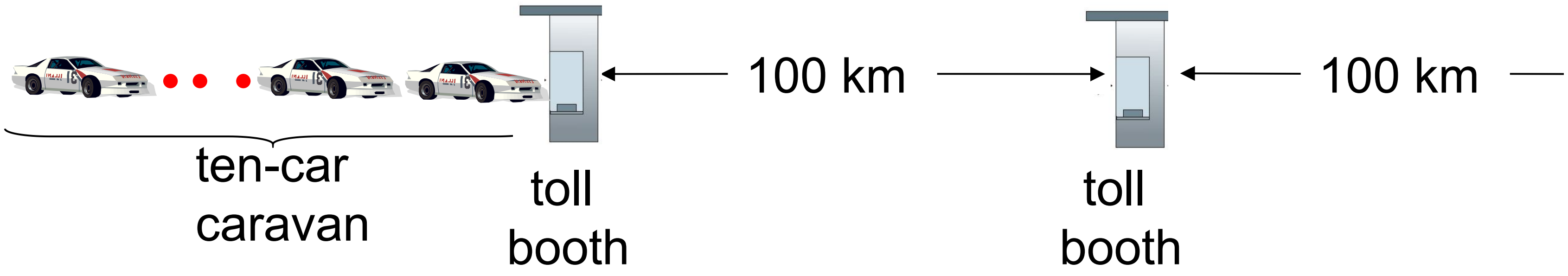
# Caravan analogy




- cars "propagate" at 100 km/hr
- toll booth takes 12 sec to service car (bit transmission time)
- car~bit; caravan ~ packet
- *Q:* How long until caravan is lined up before 2nd toll booth?

- time to "push" entire caravan through toll booth onto highway = 12*10 = 120 sec
- time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr)= 1 hr
- *A:* 62 minutes

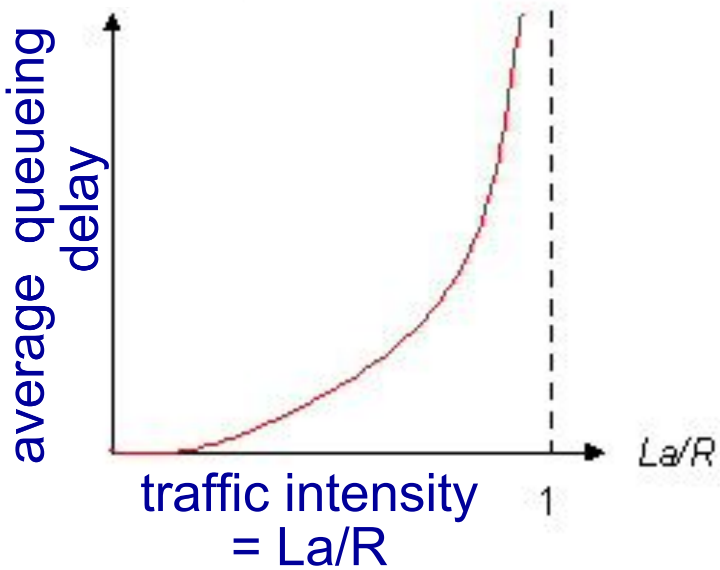
# Caravan analogy (more)




- suppose cars now “propagate” at 1000 km/hr
- and suppose toll booth now takes one min to service a car
- *Q:* Will cars arrive to 2nd booth before all cars serviced at first booth?
  - *A: Yes!* after 7 min, 1st car arrives at second booth; three cars still at 1st booth.

# Queueing delay (revisited)

- *R:* link bandwidth (bps)
- *L:* packet length (bits)
- a: average packet arrival rate




- *La/R* ~ 0: avg. queueing delay small
- *La/R* -> 1: avg. queueing delay large
- *La/R* > 1: more "work" arriving than can be serviced, average delay infinite!


La/R ~ 0


La/R -> 1

* Check out the Java applet for an interactive animation on queuing and loss

# Queueing Theory Basics

- Each ‘node’ or ’station’ or router called a queue
- Each packet called a ’job’
- A queue has a servicing/processing station and a buffer or queue where jobs wait for service
- The behaviour of a queue is determined by the queueing policy (e.g. FIFO) and the service time (e.g. proportional to packet length or fixed)
- The performance (throughput, delay etc) depends on the queue parameters and the arrival process of jobs

# Analysis

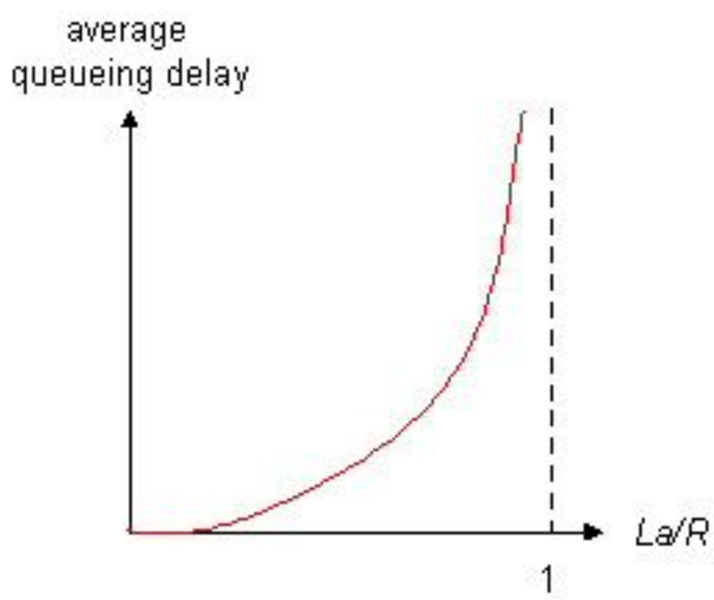
- Analysis of a single queue is difficult
- Analysis of networks of queues is even more difficult.
- The best-known results are derived with striong assumptions on all parameters.
- The standard naming scheme of queues is of the form X/Y/k/b where X = arrival process, Y = service time process, k= number of service stations, b = length of buffer
- We will only look at $M/M/1/\infty$ queues (M=markovian)
- For networks of $M/M/1/\infty$ queues, it is enough to analyze single queues. Network performance can be very easily obtained from individual queue performance.

# M/M/1/∞ queues

- The first M: Poisson arrival process. Probability of N(t) packets arriving in any interval of time t is $P(N(t)=k) = (\lambda t)^k \exp(-\lambda t)/k!$, k = 0,1,2,.....
- The second M: Exponential interarrival times Probability of job k arriving t units after job k-1 is $P(x=t) = \mu \exp(-\mu t)$ if $t>0$ and 0 otherwise. It follows that $E[x] = 1/\mu$, $\text{variance}[x] = 1/\mu^2$
- Under these assumptions, utilization = Prob(queue is non-empty) = $\rho$ where $\rho = \lambda/\mu$
- So when $\lambda$ approaches $\mu$ (cannot exceed $\mu$), utilization goes towards 100%

# M/M/1/∞ queues - contd.

- However, expected number of jobs in the queue is $= \rho/(1 - \rho)$ where $\rho = \lambda/\mu$
- So when $\lambda$ approaches $\mu$ the number of jobs in the queue approaches infinity!!
- As a result delay goes up.
- Therefore most systems cannot be driven at capacity.



# Little's Law

- One of the very few general laws:

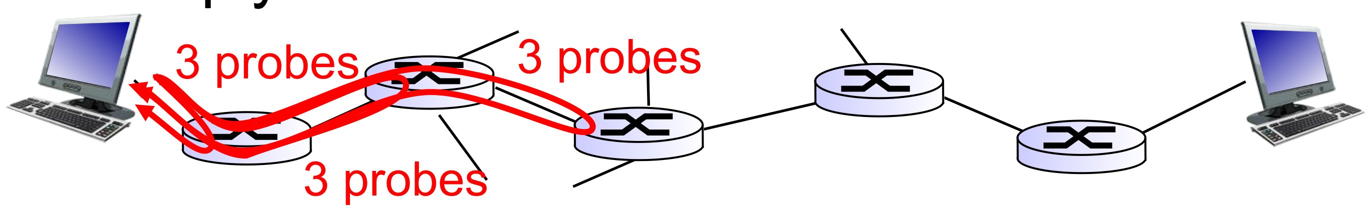
  The average number of customers in a (stable) queueing system L is equal to the long-term average effective arrival rate, $\lambda$, multiplied by the average time a customer spends in the system, W; or $L = \lambda W$.

Applies to single queues or networks

So average delay seen by a packet (from previous slide) = $\rho / [\lambda (1 - \rho)]$

# "Real" Internet delays and routes

- what do "real" Internet delay & loss look like?
- `traceroute` program: provides delay measurement from source to router along end-end Internet path towards destination. For all *i:*
  - sends three packets that will reach router *i* on path towards destination
  - router *i* will return packets to sender
  - sender times interval between transmission and reply.



# "Real" Internet delays, routes

traceroute: gaia.cs.umass.edu to www.eurecom.fr

3 delay measurements from gaia.cs.umass.edu to cs-gw.cs.umass.edu

1 cs-gw (128.119.240.254) 1 ms 1 ms 2 ms
2 border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145) 1 ms 1 ms 2 ms
3 cht-vbns.gw.umass.edu (128.119.3.130) 6 ms 5 ms 5 ms
4 jn1-at1-0-0-19.wor.vbns.net (204.147.132.129) 16 ms 11 ms 13 ms
5 jn1-so7-0-0-0.wae.vbns.net (204.147.136.136) 21 ms 18 ms 18 ms
6 abilene-vbns.abilene.ucaid.edu (198.32.11.9) 22 ms 18 ms 22 ms
7 nycm-wash.abilene.ucaid.edu (198.32.8.46) 22 ms 22 ms 22 ms
8 62.40.103.253 (62.40.103.253) 104 ms 109 ms 106 ms
9 de2-1.de1.de.geant.net (62.40.96.129) 109 ms 102 ms 104 ms
10 de.fr1.fr.geant.net (62.40.96.50) 113 ms 121 ms 114 ms
11 renater-gw.fr1.fr.geant.net (62.40.103.54) 112 ms 114 ms 112 ms
12 nio-n2.cssi.renater.fr (193.51.206.13) 111 ms 114 ms 116 ms
13 nice.cssi.renater.fr (195.220.98.102) 123 ms 125 ms 124 ms
14 r3t2-nice.cssi.renater.fr (195.220.98.110) 126 ms 126 ms 124 ms
15 eurecom-valbonne.r3t2.ft.net (193.48.50.54) 135 ms 128 ms 133 ms
16 194.214.211.25 (194.214.211.25) 126 ms 128 ms 126 ms
17 * * *
18 * * *
19 fantasia.eurecom.fr (193.55.113.142) 132 ms 128 ms 136 ms

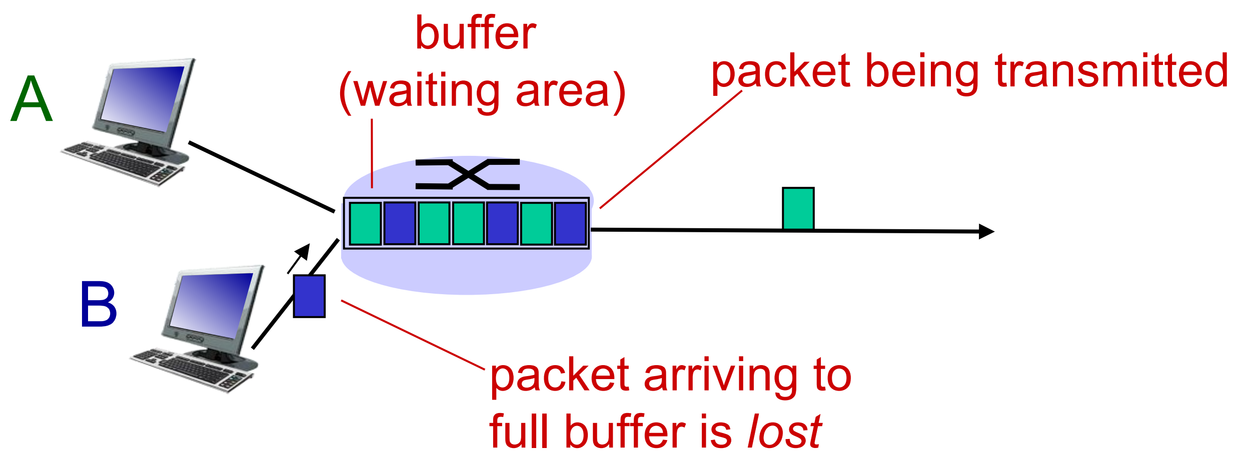
trans-oceanic link

* means no response (probe lost, router not replying)

* Do some traceroutes from exotic countries at www.traceroute.org

# Packet loss

- 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




* Check out the Java applet for an interactive animation on queuing and loss

# Throughput

- *throughput:* rate (bits/time unit) at which bits transferred between sender/receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time