CS244a: An Introduction to Computer Networks

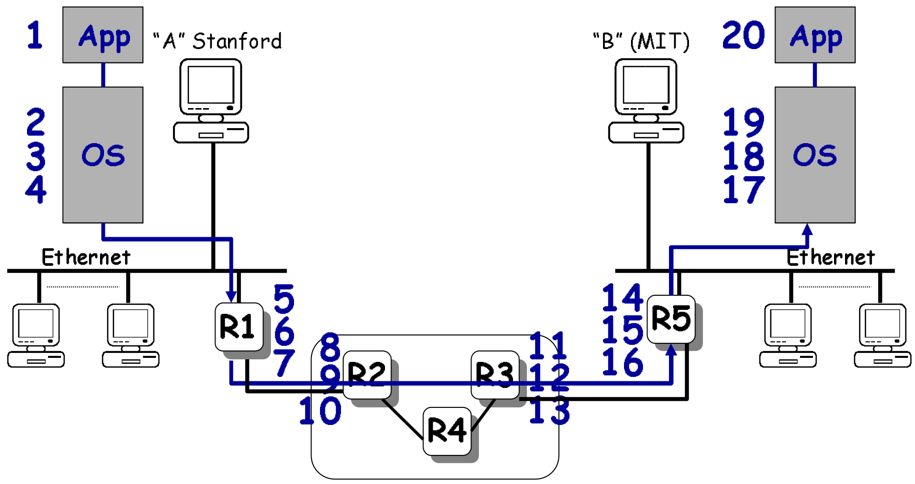
Handout 3: Foundations and Basic Concepts

Nick McKeown
Professor of Electrical Engineering and Computer Science, Stanford University
nickm@stanford.edu
http://www.stanford.edu/~nickm

Outline

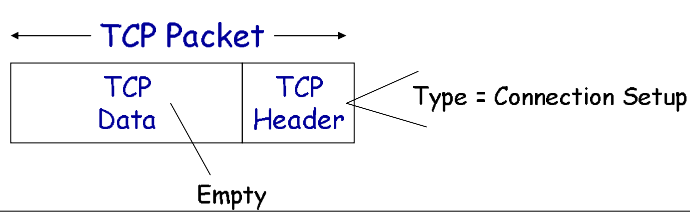
- A Detailed FTP Example
- Layering
- Packet Switching and Circuit Switching
- Some terms
  - Data rate, "Bandwidth" and "throughput"
  - Propagation delay
  - Packet, header, address
  - Bandwidth-delay product, RTT

Example: FTP over the Internet Using TCP/IP and Ethernet



In the sending host

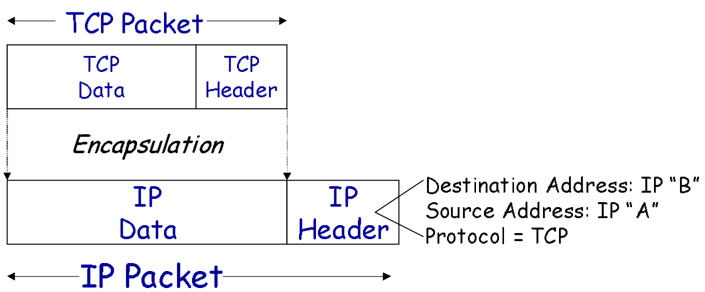
1. **Application-Programming Interface (API)**
   - Application requests TCP connection with "B"
2. **Transmission Control Protocol (TCP)**
   - Creates TCP "Connection setup" packet
   - TCP requests IP packet to be sent to "B"



In the sending host (2)

3. Internet Protocol (IP)

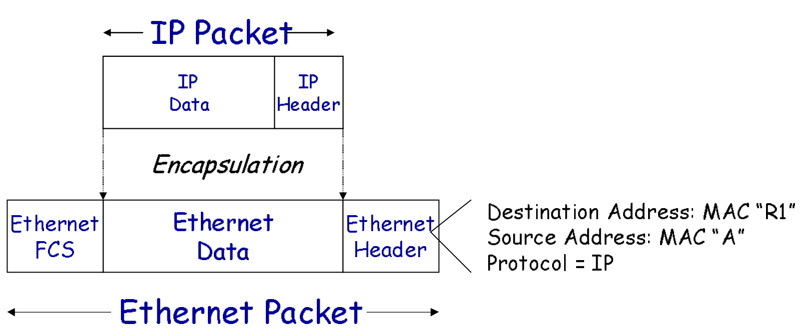
- Creates IP packet with correct addresses.
- IP requests packet to be sent to router.



In the sending host (3)

4. Link ("MAC" or Ethernet) Protocol

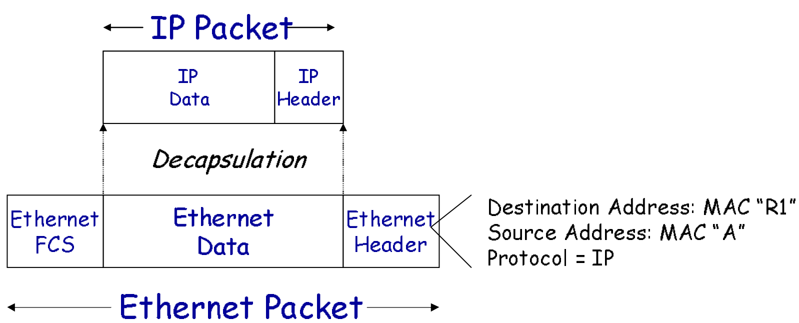
- Creates MAC frame with Frame Check Sequence (FCS).
- Wait for Access to the line.
- MAC requests PHY to send each bit of the frame.



In Router R1

5. Link ("MAC" or Ethernet) Protocol

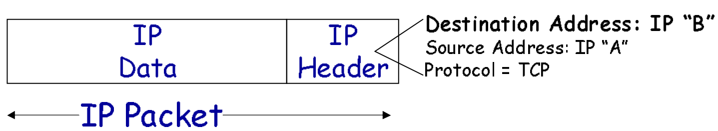
- Accept MAC frame, check address and Frame Check Sequence (FCS).
- Pass data to IP Protocol.



In Router R1

6. Internet Protocol (IP)

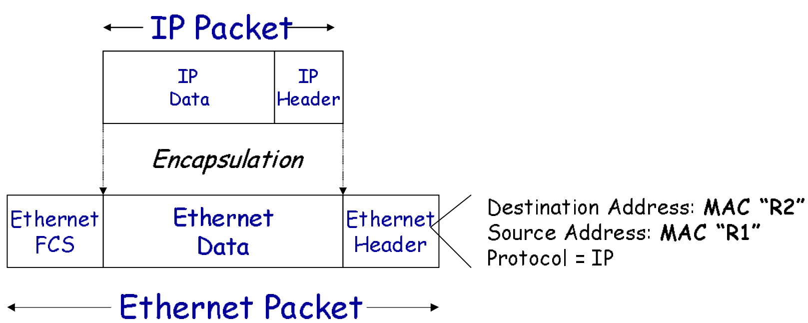
- Use IP destination address to decide where to send packet next ("next-hop routing").
- Request Link Protocol to transmit packet.



In Router R1

7. Link ("MAC" or Ethernet) Protocol

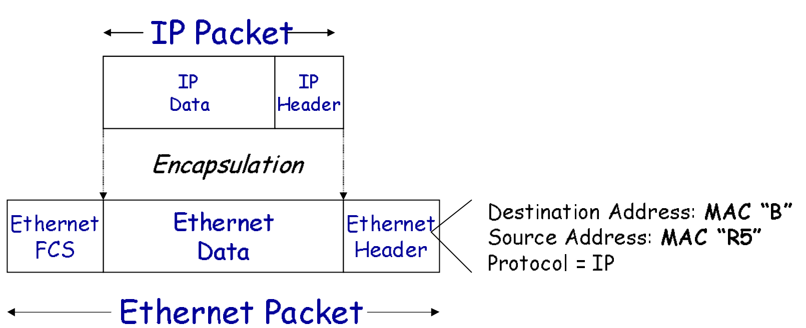
- Creates MAC frame with Frame Check Sequence (FCS).
- Wait for Access to the line.
- MAC requests PHY to send each bit of the frame.



In Router R5

16. Link ("MAC" or Ethernet) Protocol

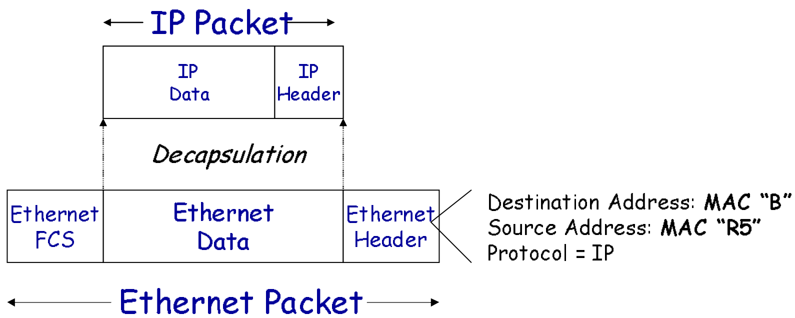
- Creates MAC frame with Frame Check Sequence (FCS).
- Wait for Access to the line.
- MAC requests PHY to send each bit of the frame.



In the receiving host

17. Link ("MAC" or Ethernet) Protocol

- Accept MAC frame, check address and Frame Check Sequence (FCS).
- Pass data to IP Protocol.

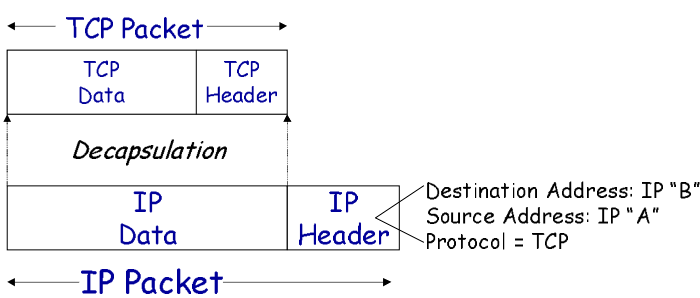


Winter 2008 CS244a Handout 3 11

In the receiving host (2)

18. Internet Protocol (IP)

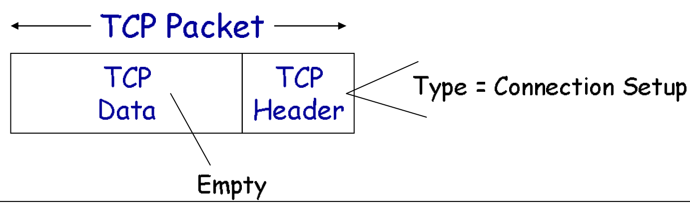
- Verify IP address.
- Extract/decapsulate TCP packet from IP packet.
- Pass TCP packet to TCP Protocol.



Winter 2008 CS244a Handout 3 12

In the receiving host (3)

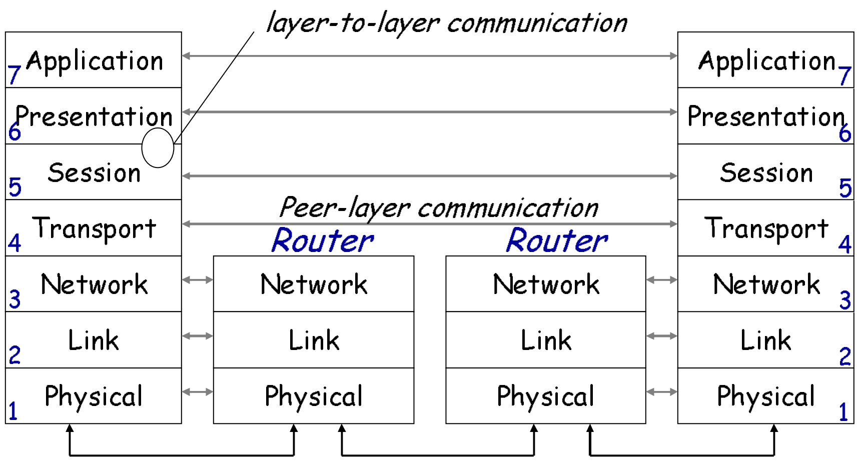
19. **Transmission Control Protocol (TCP)**
    - Accepts TCP "Connection setup" packet
    - Establishes connection by sending "Ack".
20. **Application-Programming Interface (API)**
    - Application receives request for TCP connection with "A".



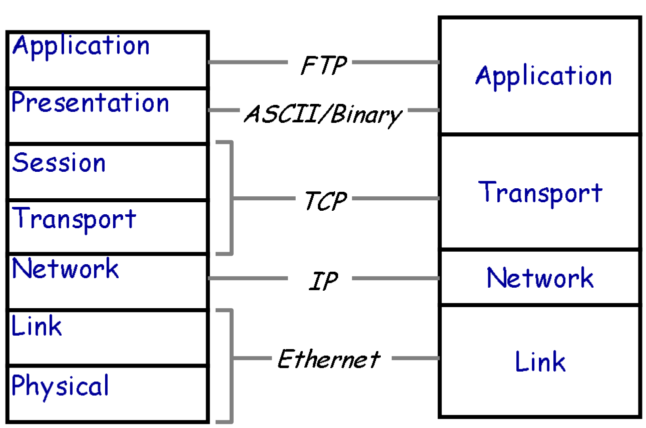
Outline

- A Detailed FTP Example
- Layering
- Packet Switching and Circuit Switching
- Some terms
    - Data rate, "Bandwidth" and "throughput"
    - Propagation delay
    - Packet, header, address
    - Bandwidth-delay product, RTT

Layering: The OSI Model



Layering: Our FTP Example



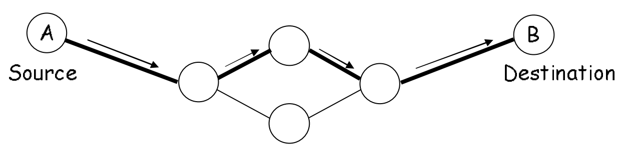
The 7-layer OSI Model

The 4-layer Internet model

Outline

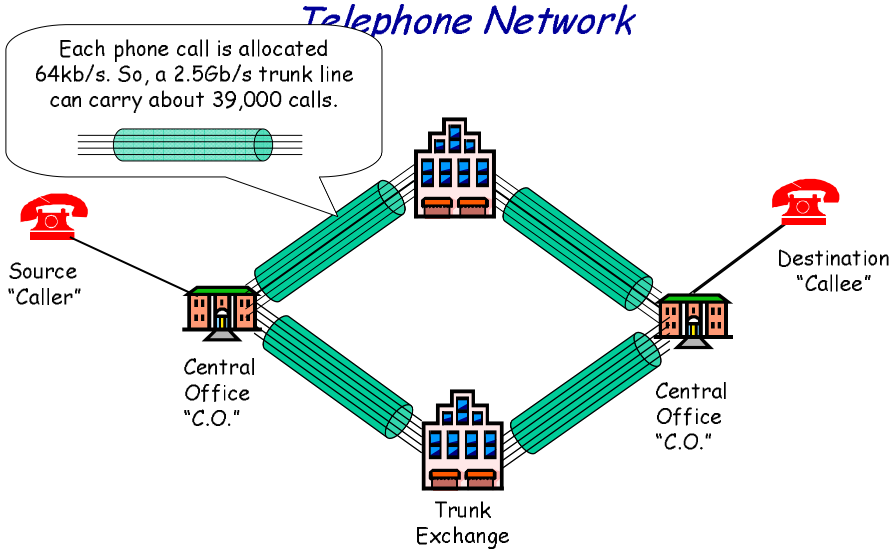
- A Detailed FTP Example
- Layering
- Packet Switching and Circuit Switching
- Some terms
  - Data rate, "Bandwidth" and "throughput"
  - Propagation delay
  - Packet, header, address
  - Bandwidth-delay product, RTT

Circuit Switching

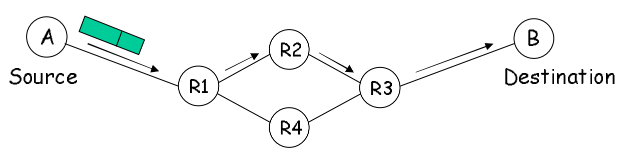


- It's the method used by the telephone network.
- A call has three phases:
  1. Establish circuit from end-to-end ("dialing"),
  2. Communicate,
  3. Close circuit ("tear down").
- Originally, a circuit was an end-to-end physical wire.
- Nowadays, a circuit is like a virtual private wire: each call has its own private, guaranteed data rate from end-to-end.

Circuit Switching
Telephone Network

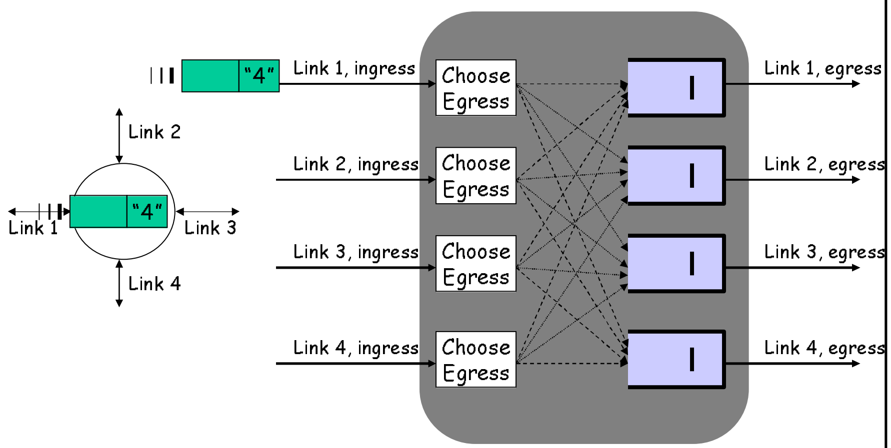


Packet Switching



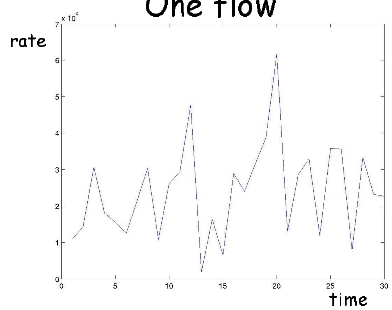
- It's the method used by the Internet.
- Each packet is individually routed packet-by-packet, using the router's local routing table.
- The routers maintain no per-flow state.
- Different packets may take different paths.
- Several packets may arrive for the same output link at the same time, therefore a packet switch has buffers.

Packet Switching
Simple router model

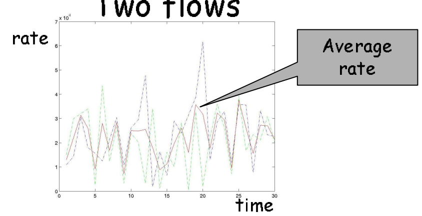


Statistical Multiplexing
Basic idea

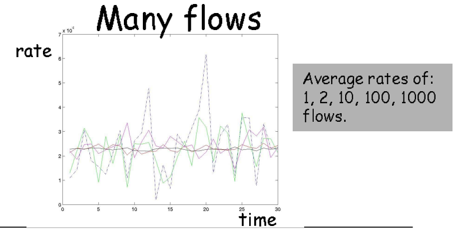
One flow



Two flows



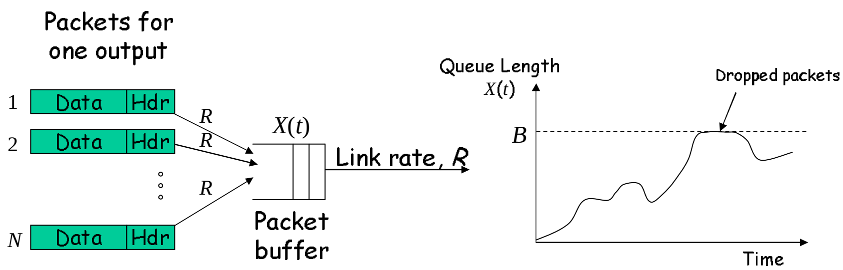
Many flows



- Network traffic is bursty. i.e. the rate changes frequently.
- Peaks from independent flows generally occur at different times.
- Conclusion: The more flows we have, the smoother the traffic.

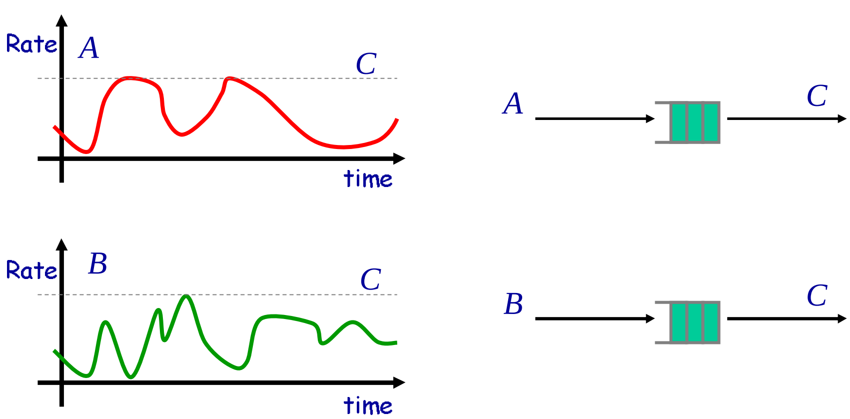
Packet Switching

Statistical Multiplexing

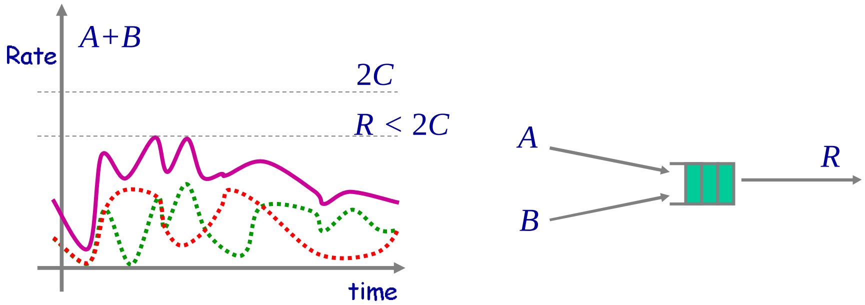


- Because the buffer absorbs temporary bursts, the egress link need not operate at rate $N.R$.
- But the buffer has finite size, B, so losses will occur.

Statistical Multiplexing



Statistical Multiplexing Gain



Statistical multiplexing gain = $2C/R$

Other definitions of SMG: The ratio of rates that give rise to a particular queue occupancy, or particular loss probability.

Why does the Internet use packet switching?

1. Efficient use of expensive links:
   - The links are assumed to be expensive and scarce.
   - Packet switching allows many, bursty flows to share the same link efficiently.
   - "Circuit switching is rarely used for data networks, ... because of very inefficient use of the links" - *Gallager*
2. Resilience to failure of links & routers:
   - "For high reliability, ... [the Internet] was to be a datagram subnet, so if some lines and [routers] were destroyed, messages could be ... rerouted" - *Tanenbaum*

Some Definitions

- Packet length, P, is the length of a packet in bits.
- Link length, L, is the length of a link in meters.
- Data rate, R, is the rate at which bits can be sent, in bits/second, or b/s.[1]
- Propagation delay, $PROP$, is the time for one bit to travel along a link of length, L.

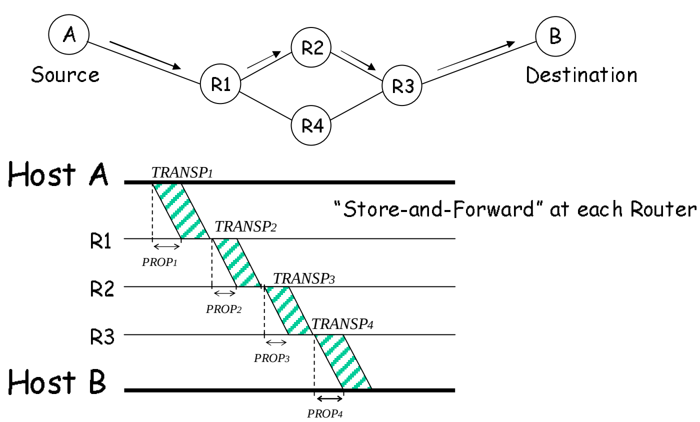
  $$PROP = L/c.$$
- Transmission time, $TRANSP$, is the time to transmit a packet of length P.

  $$TRANSP = P/R.$$
- Latency is the time from when the first bit begins transmission, until the last bit has been received. On a link:

  $$Latency = PROP + TRANSP.$$

1. Note that a kilobit/second, kb/s, is 1000 bits/second, not 1024 bits/second.

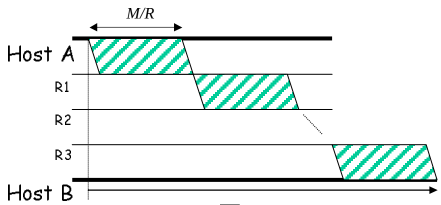
Packet Switching



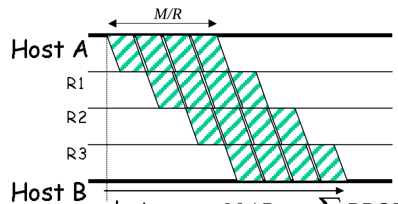
$$\text{Minimum end to end latency} = \sum_i (TRANSP_i + PROP_i)$$

Packet Switching

Why not send the entire message in one packet?



Latency $= \sum_i (PROP_i + M / R_i)$



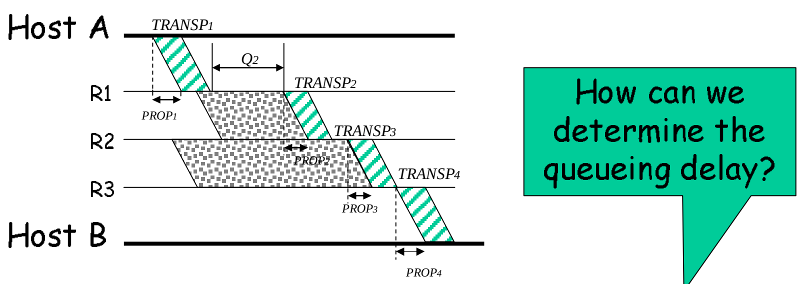
Latency $= M / R_{min} + \sum_i PROP_i$

Breaking message into packets allows parallel transmission across all links, reducing end to end latency. It also prevents a link from being "hogged" for a long time by one message.

Packet Switching

Queueing Delay

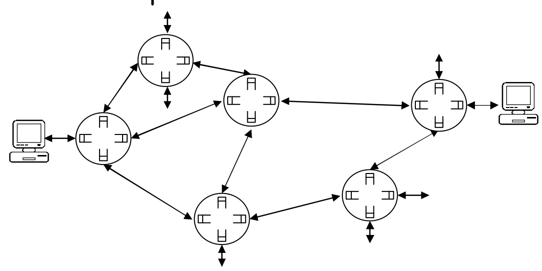
Because the egress link is not necessarily free when a packet arrives, it may be queued in a buffer. If the network is busy, packets might have to wait a long time.



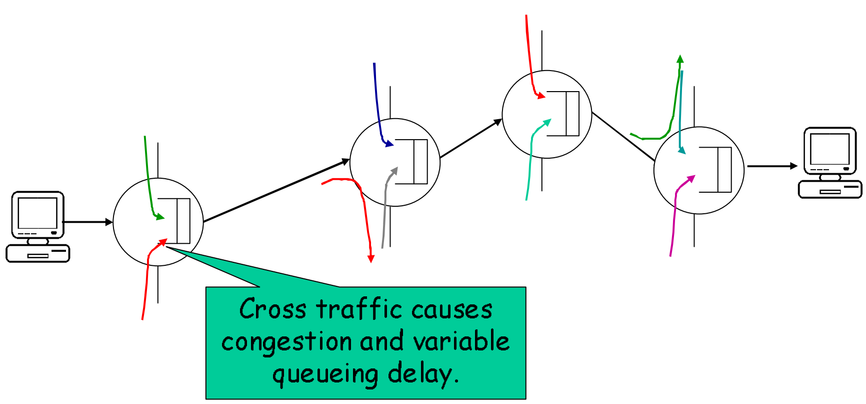
Actual end to end latency $= \sum_i (TRANSP_i + PROP_i + Q_i)$

Queues and Queueing Delay

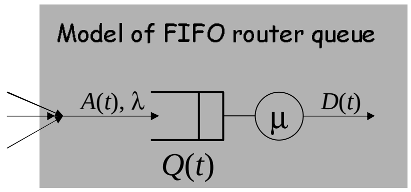
- To understand the performance of a packet switched network, we can think of it as a series of queues interconnected by links.
- For given link rates and lengths, the only variable is the queueing delay.
- If we can understand the queueing delay, we can understand how the network performs.



Queues and Queueing Delay



A router queue



$A(t)$: The arrival process. The number of arrivals in interval $[0,t]$.

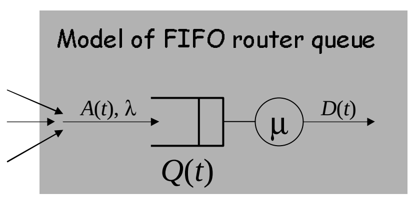
λ: The average rate of new arrivals in packets/second.

$D(t)$: The departure process. The number of departures in interval $[0,t]$.

$1/\mu$: The average time to service each packet.

$Q(t)$: The number of packets in the queue at time t.

A simple deterministic model

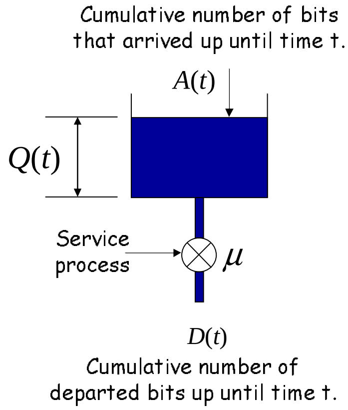


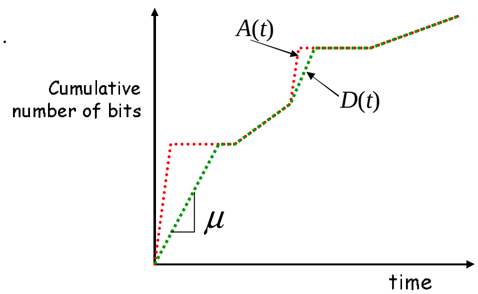
Properties of $A(t)$, $D(t)$:

- $A(t)$, $D(t)$ are non-decreasing
- $A(t) >= D(t)$

A simple deterministic model

bytes or "fluid"

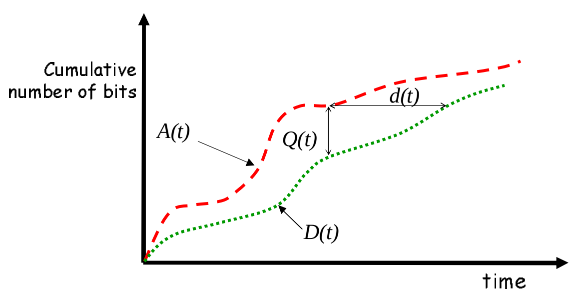




Properties of $A(t)$, $D(t)$:

- $A(t)$, $D(t)$ are non-decreasing
- $A(t) >= D(t)$

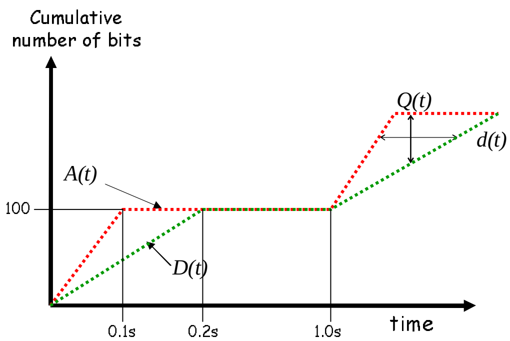
Simple deterministic model



Queue occupancy: $Q(t) = A(t) - D(t)$.

Queueing delay, $d(t)$, is the time spent in the queue by a bit that arrived at time t, and if the queue is served first-come-first-served (FCFS or FIFO)

Example



Example: Every second, a train of 100 bits arrive at rate 1000b/s. The maximum departure rate is 500b/s. What is the average queue occupancy?

Solution: During each cycle, the queue fills at rate 500b/s for 0.1s, then drains at rate 500b/s for 0.1s.The average queue occupancy when the queue is non-empty is therefore: $(\bar{Q}(t)|Q(t)>0) = 0.5\times(0.1\times500) = 25$ bits.
The queue is empty for 0.8s each cycle, and so: $\bar{Q}(t) = (0.2\times25)+(0.8\times0) = 5$ bits.
(You'll probably have to think about this for a while...).

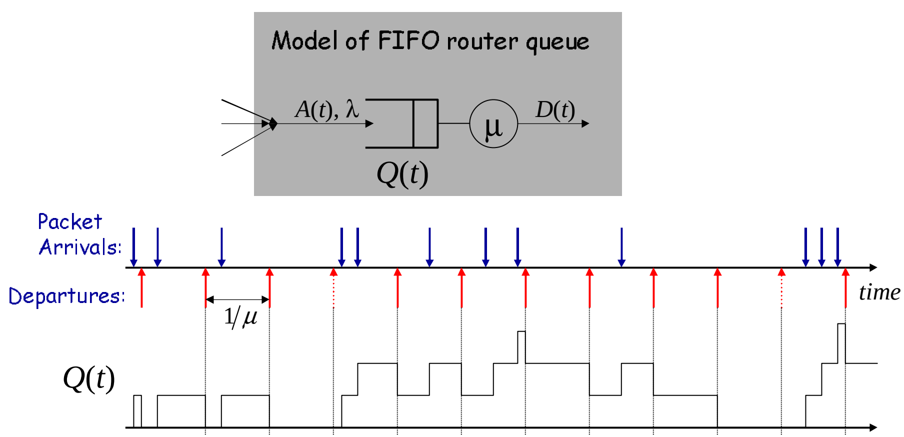
Queues with Random Arrival Processes

1. Usually, arrival processes are complicated, so we often model them as random processes.
2. The study of queues with random arrival processes is called Queueing Theory.
3. Queues with random arrival processes have some interesting properties. We'll consider some here.

Properties of queues

- Time evolution of queues.
- Examples
  - Burstiness increases delay
  - Determinism minimizes delay
- Little's Result.
- The M/M/1 queue.

Time evolution of a queue
Packets



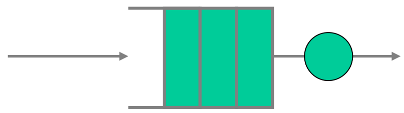
Burstiness increases delay

- Example 1: Periodic arrivals
  - 1 packet arrives every 1 second
  - 1 packet can depart every 1 second
  - Depending on when we sample the queue, it will contain 0 or 1 packets.
- Example 2:
  - N packets arrive together every N seconds (same rate)
  - 1 packet departs every second
  - Queue might contain 0,1, …, N packets.
  - Both the average queue occupancy and the variance have increased.
- In general, burstiness increases queue occupancy (which increases queueing delay).

Determinism minimizes delay

- Example 3: Random arrivals
  - Packets arrive randomly; on average, 1 packet arrives per second.
  - Exactly 1 packet can depart every 1 second.
  - Depending on when we sample the queue, it will contain 0, 1, 2, … packets depending on the distribution of the arrivals.
- In general, determinism minimizes delay. *i.e.* random arrival processes lead to larger delay than simple periodic arrival processes.

Little's Result



$$L = \lambda d$$

Where:

L is the average number of customers in the system (the number in the queue + the number in service),

λ is the arrival rate, in customers per second, and

d is the average time that a customer waits in the system (time in queue + time in service).

Result holds so long as no customers are lost/dropped.

Winter 2008 CS244a Handout 3 43

The Poisson process

Poisson process is a simple arrival process in which:

1. Probability of k arrivals in an interval of t seconds is:

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$$

2. The expected number of arrivals in interval t is: λt.
3. Successive interarrival times are independent of each other (i.e. arrivals are not bursty).

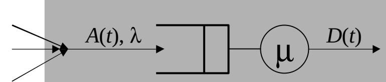
Winter 2008 CS244a Handout 3 44

The Poisson process

- Why use the Poisson process?
  - It is the continuous time equivalent of a series of coin tosses.
  - The Poisson process is known to model well systems in which a large number of independent events are aggregated together. e.g.
    - Arrival of new phone calls to a telephone switch
    - Decay of nuclear particles
    - "Shot noise" in an electrical circuit
  - It makes the math easy.
- Be warned
  - Network traffic is very bursty!
  - Packet arrivals are not Poisson.
  - But it models quite well the arrival of new flows.

An M/M/1 queue

Model of FIFO router queue



- If $A(t)$ is a Poisson process with rate λ, and the time to serve each packet is exponentially distributed with rate μ, then:

Average delay, $d = \frac{1}{\mu - \lambda}$; and so from Little's Result: $L = \lambda d = \frac{\lambda}{\mu - \lambda}$