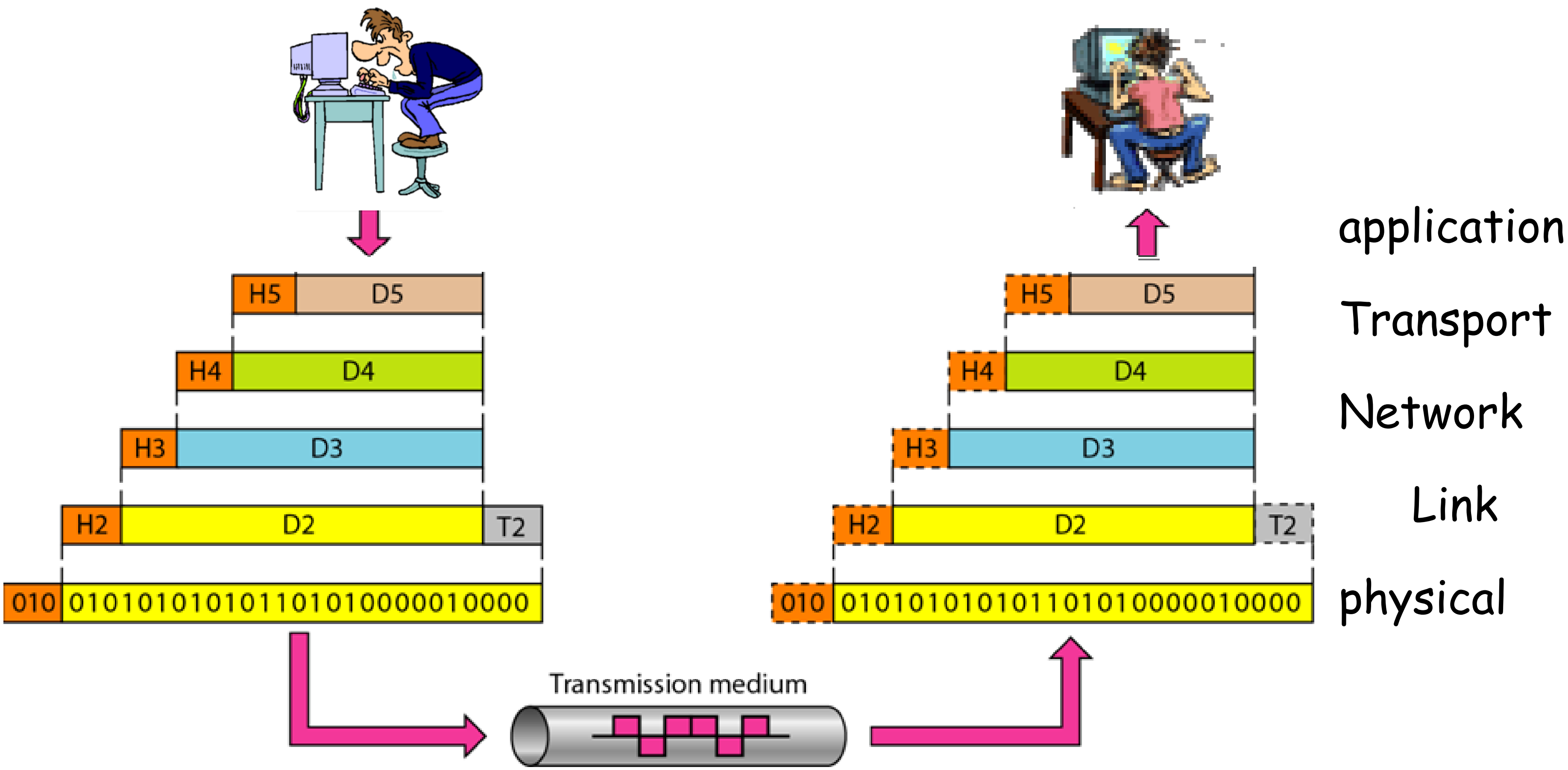
16CSCN01I: Introduction to Computer Networks

Lecture 2: Protocol Layers & Network Delays

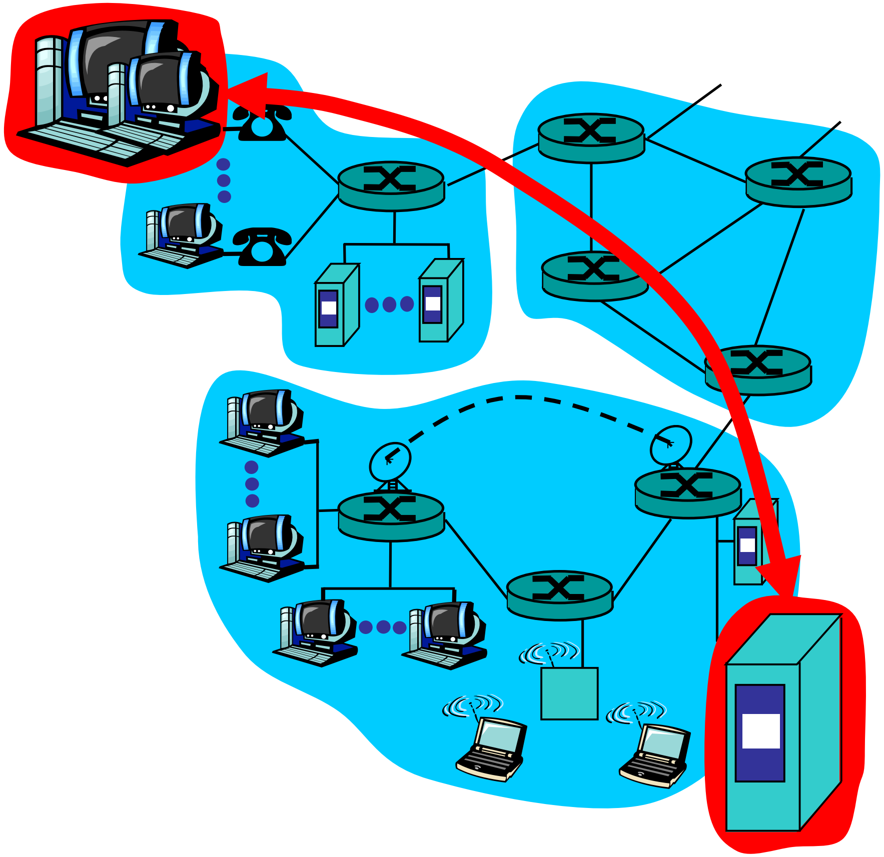
Dr. Amal ElNahas

Networking: A Top-Down Approach



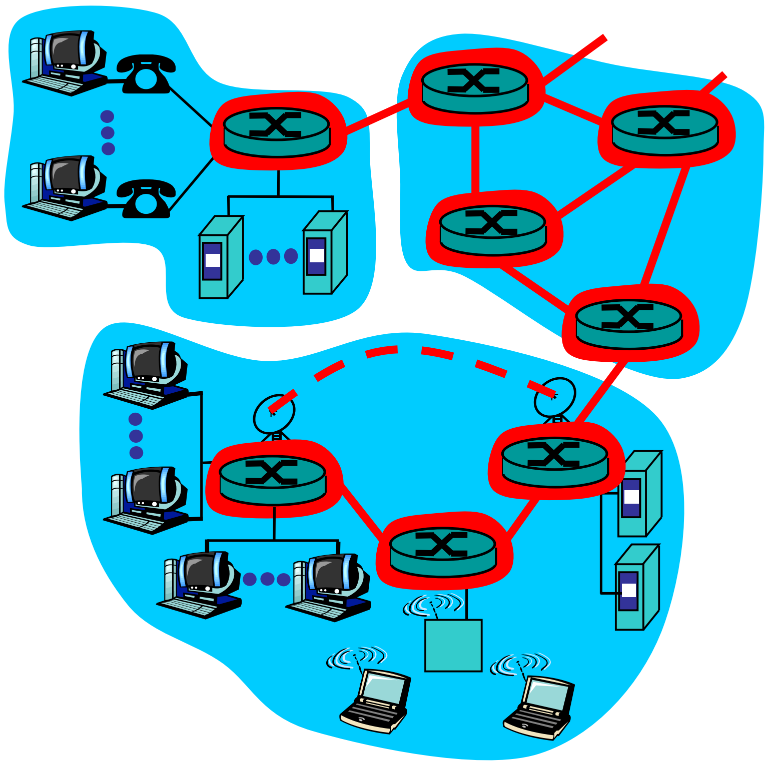
Components of the Network Edge

- End systems (hosts/servers):
  - Run application programs (Web, email) at "edge of network"
- Models of hosts communication:
  - Client/server model
  - Peer-peer model
  - Hybrid model



Network Core

- Mesh of interconnected routers sharing the infrastructure
- How to build a network core: (how data is transferred through the net?)
  - Circuit switching: dedicated circuit per call: telephone net
  - Packet-switching: data sent thru net in discrete "chunks"



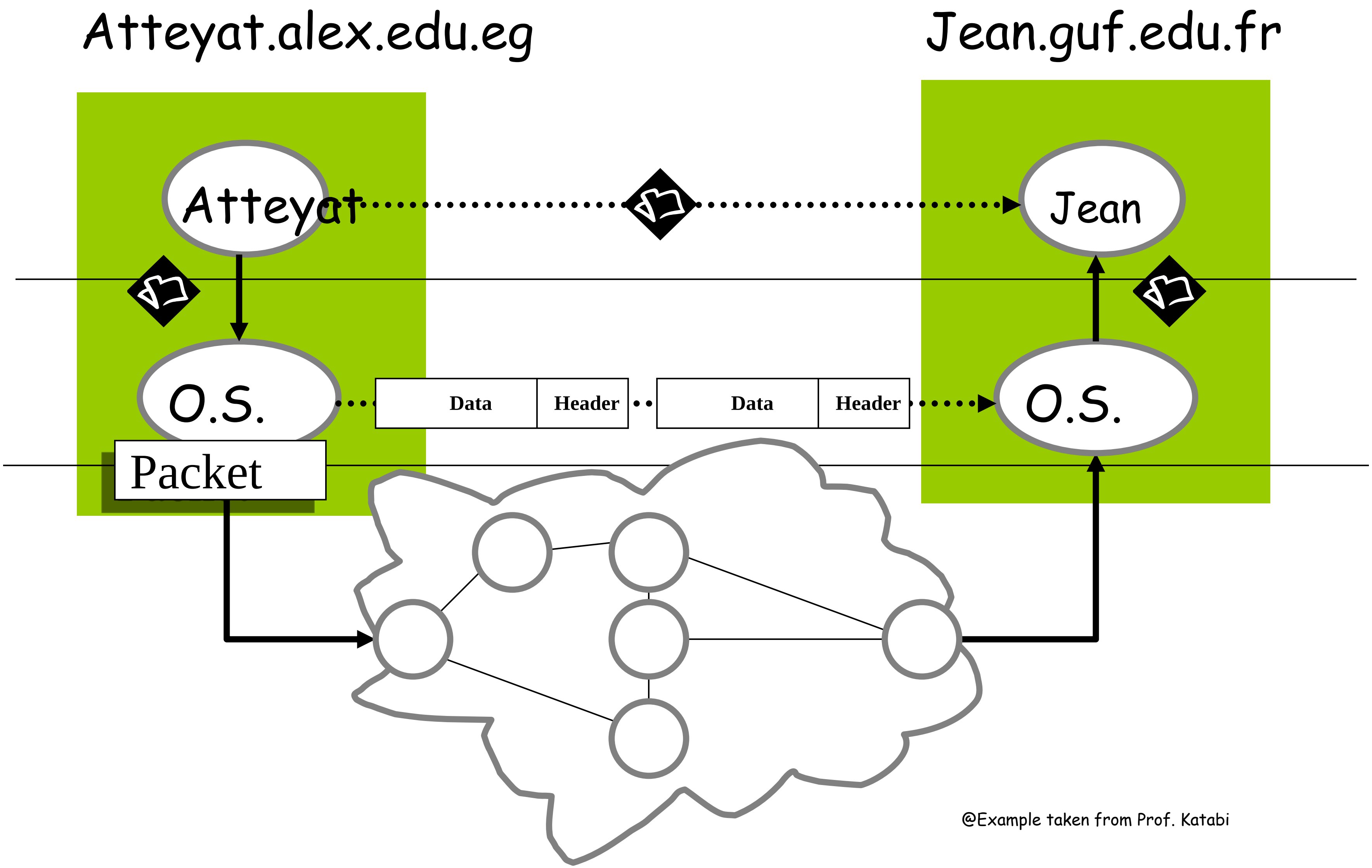
Network Core: Circuit Switching

- Circuit switching: dedicated circuit per call: telephone net
- Model: data sent continuously
- Created a session (e.g., phone call) reserves dedicated bandwidth in series of switches between caller and recipient
- Guaranteed capacity (in both directions) so long as session up

Packet Switching

- No call setup before data transfer
- Data is divided into packets that are sent independently (header contains control info, e.g., source and destination addresses)
- At each node the entire packet is received, stored, and then forwarded (store-and-forward)
- No capacity is allocated
- On demand use of resources: If you need more, you get more, If you need less, you get less

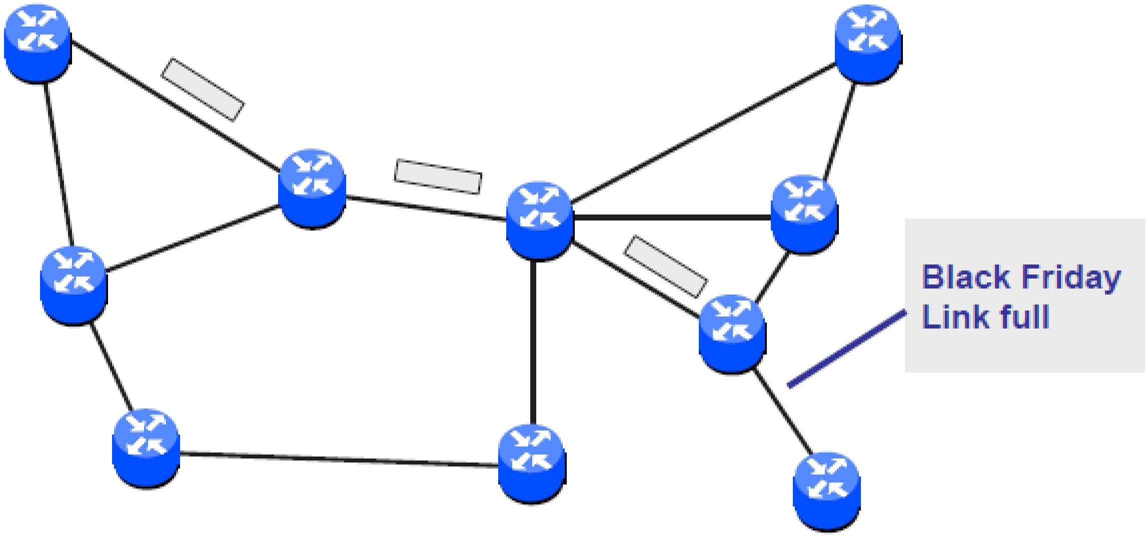
The Internet



@Example taken from Prof. Katabi

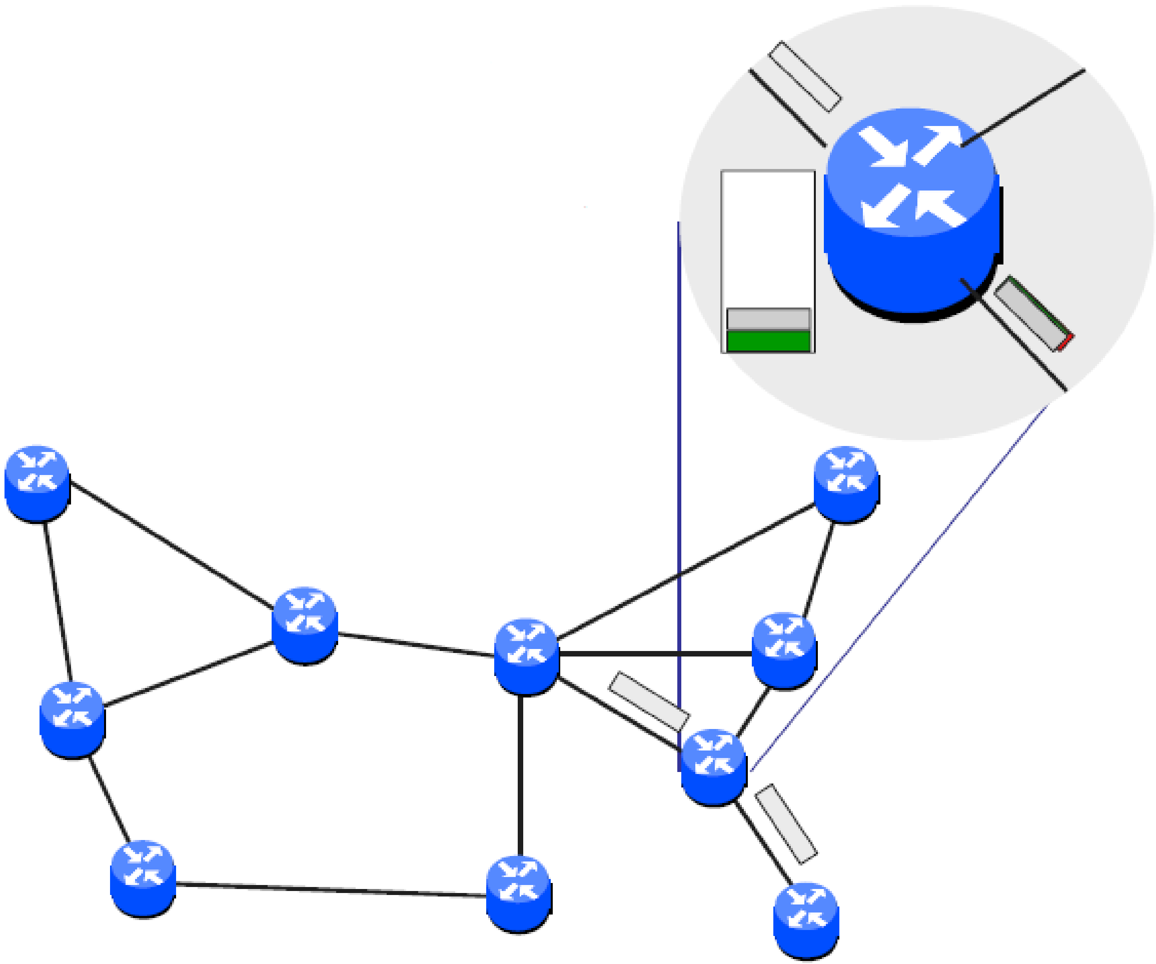
Packet Switching

- What if link is full?



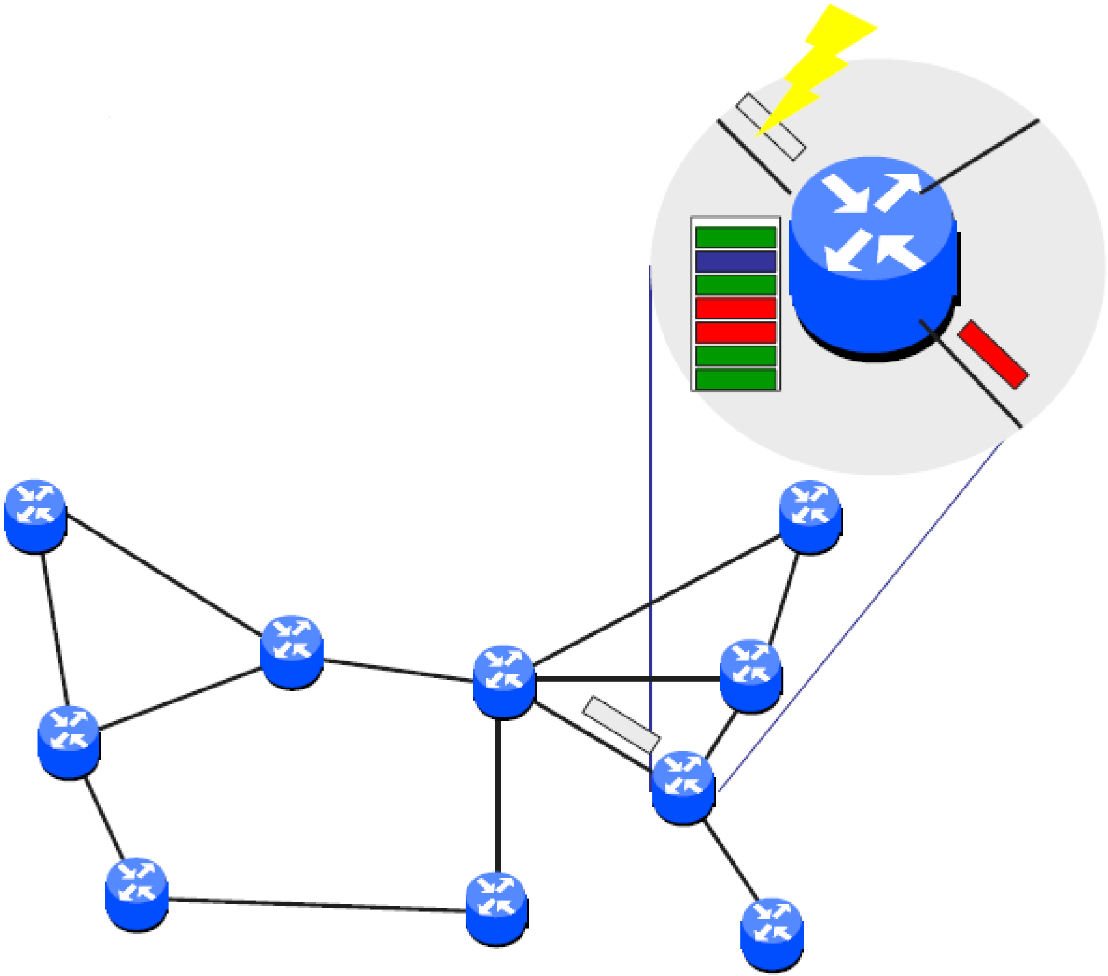
Packet Switching

- What if link is full? **Queue** the packet

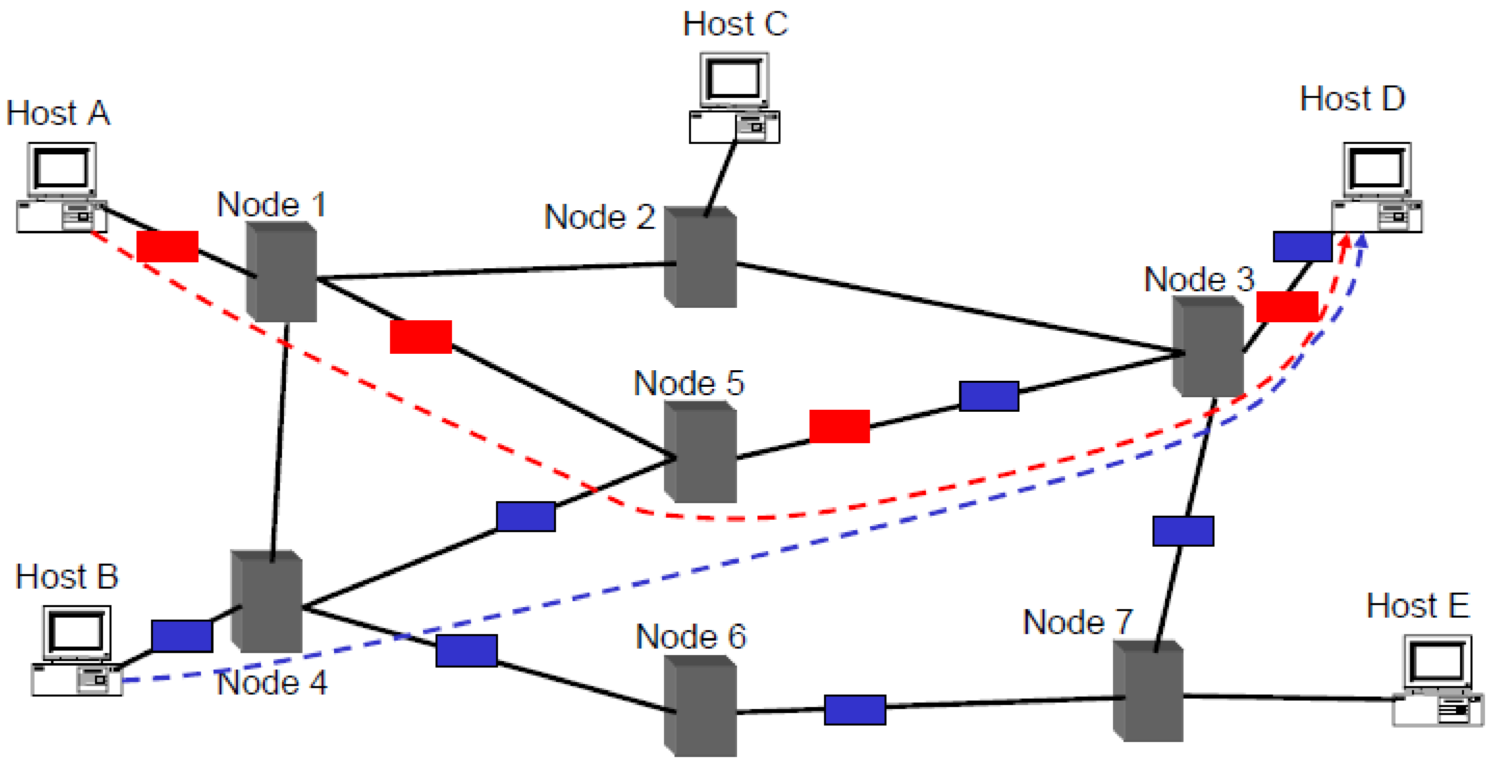


Packet Switching

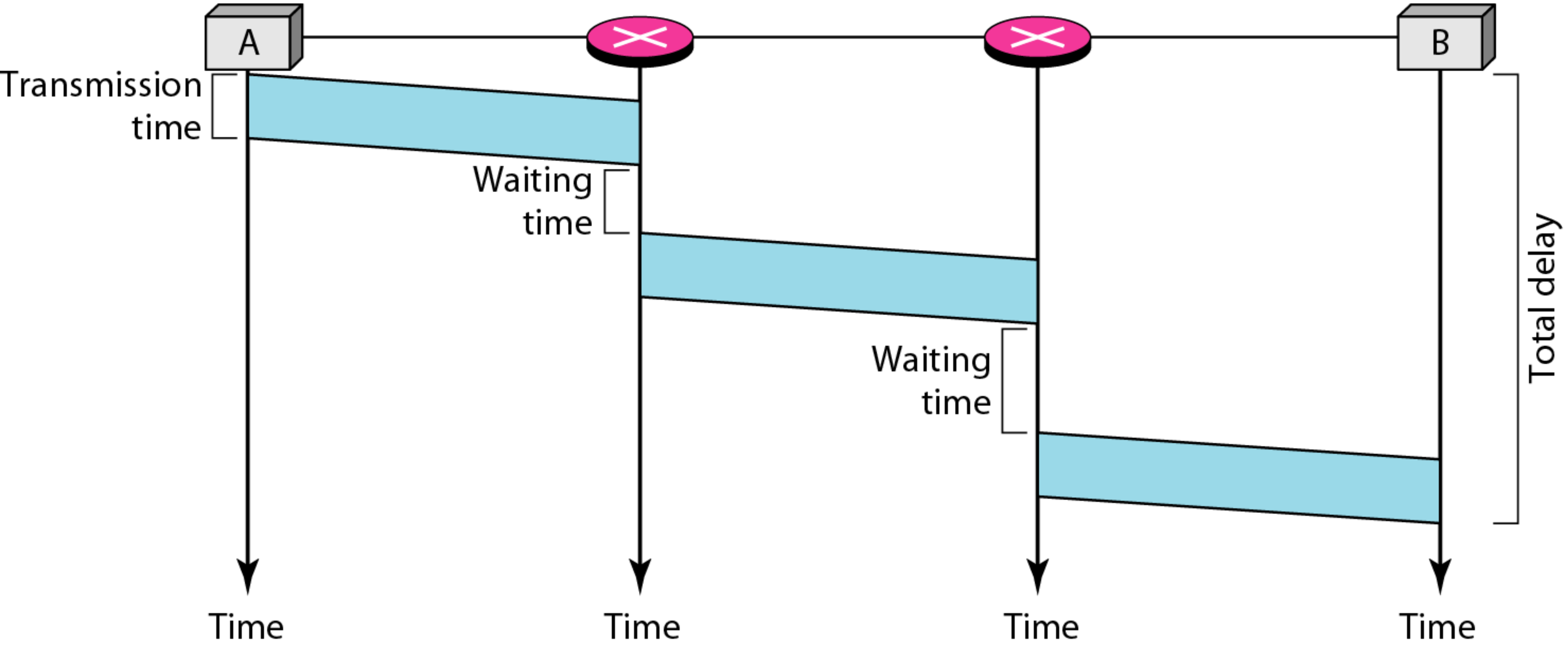
- What if queue is full? **Drop** the packet



Packet Switching



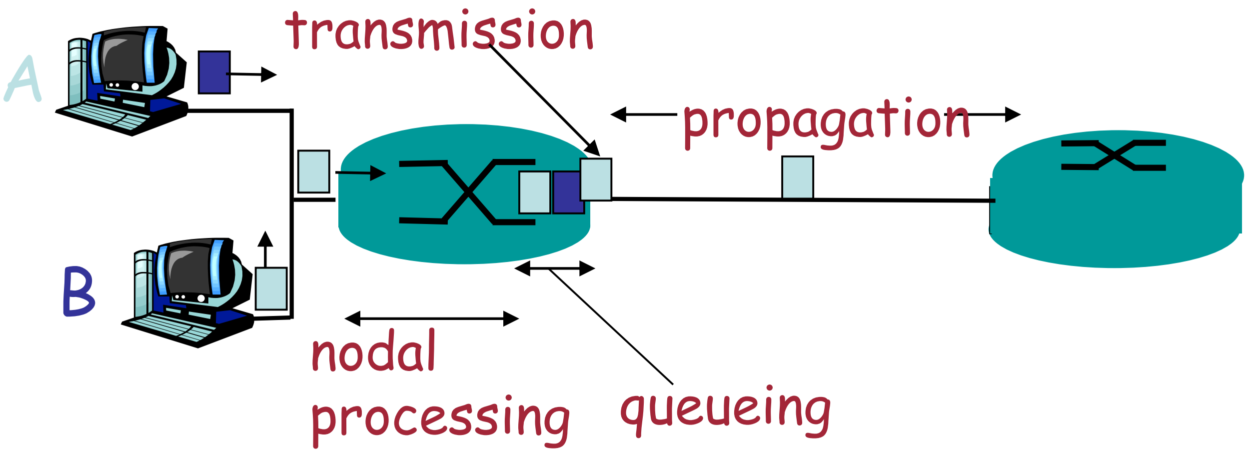
Delay in Packet-Switched Network



Four Sources of Packet Delay

1. **Nodal processing:**
   - check bit errors
   - determine output link

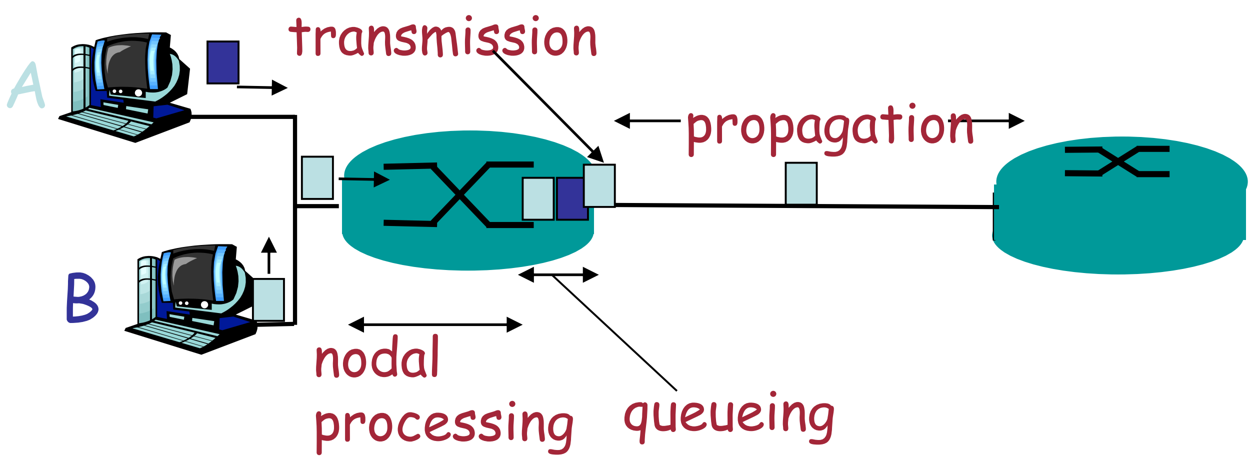
2. **Queueing**
   - time waiting at output link for transmission
   - depends on congestion level of router



Four Sources of Packet Delay

3. **Transmission delay:**
   - R=link bandwidth (bps)
   - L=packet length (bits)
   - time to send bits into link = L/R

4. **Propagation delay:**
   - d = length of physical link
   - s = propagation speed in medium (~2x108 m/sec)
   - propagation delay = d/s



Nodal delay

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

- d_{proc} = processing delay (typically a few microsecs or less)
- d_{queue} = queuing delay (depends on congestion)
- d_{trans} = transmission delay (is L/R, significant for low-speed links)
- d_{prop} = propagation delay (a few microsecs to hundreds of msecs)

Example

- Propagation delay
  - suppose the distance between A and B is 4000 km, propagation speed in medium (~2x10^5 km/sec), then one-way propagation delay is:

$$\frac{4000km}{200,000km/s} = 20ms$$

- Transmission delay
  - suppose a channel of 14 Kbps bandwidth, then the transmission delay of a packet of 1 Kbits is

$$\frac{1kbits}{14kbps} \approx 70ms$$

Packet switching versus circuit switching

- Packet switching:
  - ✓ Most suitable for bursty traffic
  - ✓ No call setup
  - ✓ Efficient use of resources

--Protocols needed for reliability and congestion control

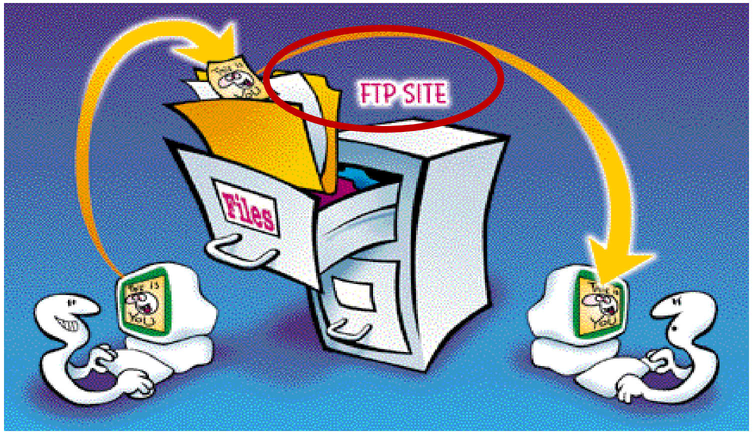
--No performance guarantee (still working on it)

Example

Consider sending a 30 Mbit MP3 file from a source host to a destination host. All links in the path between source and destination have a transmission rate of 10 Mbps. Assume that the propagation speed is 2 * 10^8 meters/sec, and the distance between source and destination is 10,000 km. If there is only one link between source and destination and the file is sent as 1 message,

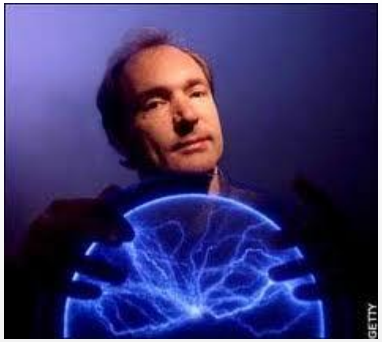
- what is the total delay?
- How many bits will the source have transmitted when the first bit arrives at the destination?

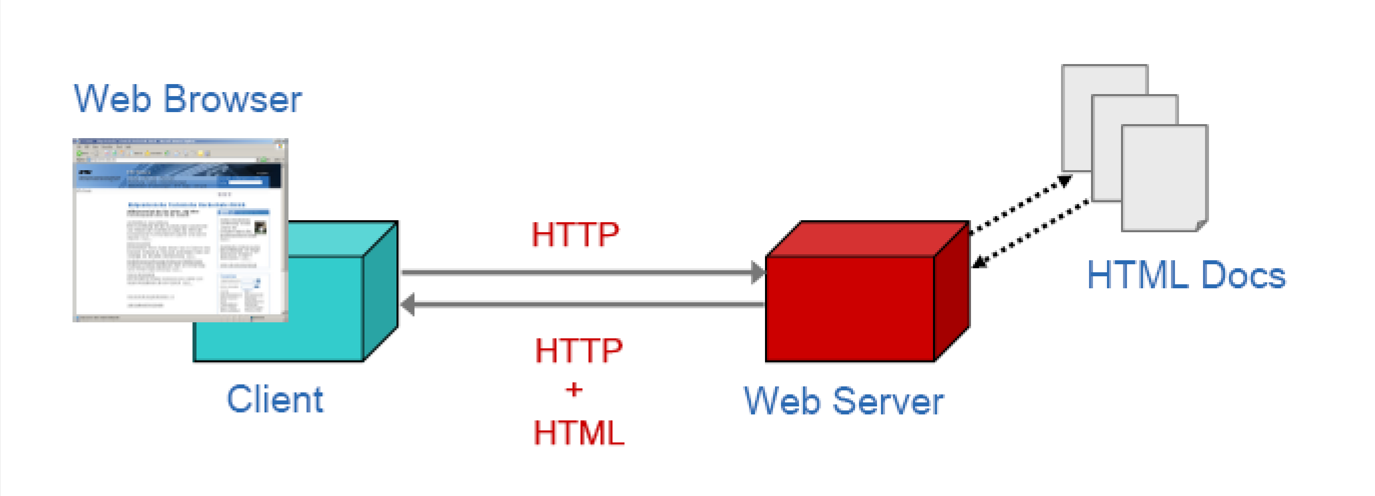
Application Layer
What Will We Study?



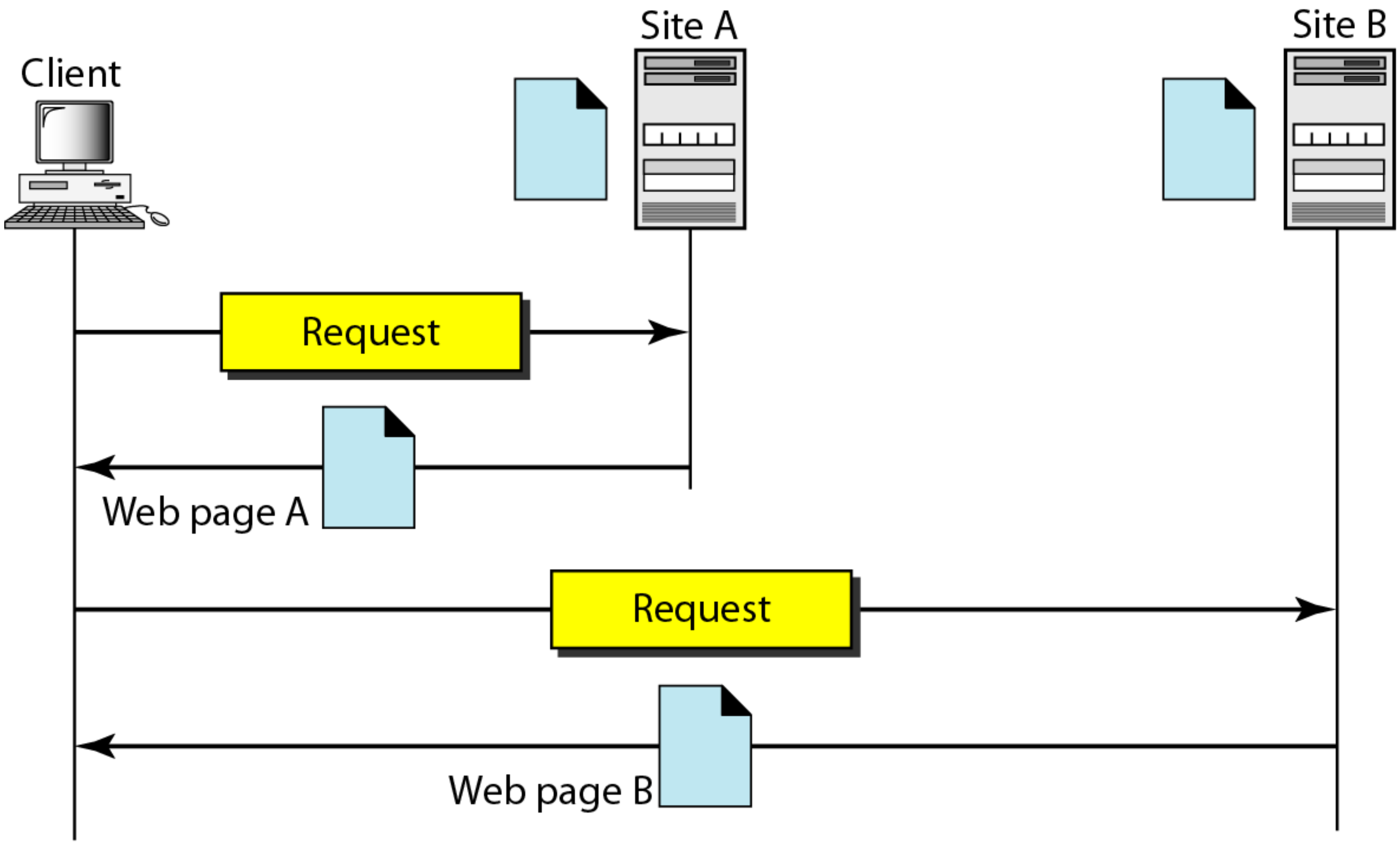
HTTP: Hyper Text Transfer Protocol

- Protocol used to access data on the web
- Defines how web clients request web pages from the server and how web servers transfer web pages to clients



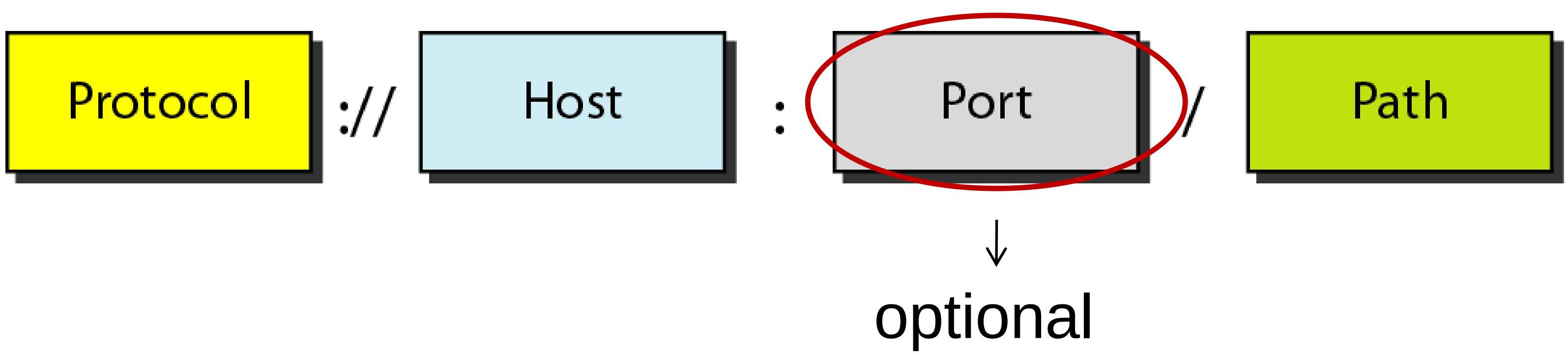


Architecture of WWW



Web Document

- A web document consists of a <u>base HTML-file</u> which includes several referenced objects (HTML file, JPEG image, Java applet, audio file,…)
- Each object is addressable by a URL. A URL is composed of host name of the server and object's path name
- Example:



HTTP: Example

You enter the following : www.uni.edu/courses/L10.index that contains text, references to 10 jpeg objects

- **1a**. http client initiates TCP connection to http server (process) at www.uni.edu

1b. http server at host www.guc.edu waiting for TCP connection at port 80. "accepts" connection, notifying client

2. http client sends http *request message* (containing URL) into TCP connection socket (wants object courses/L10.index)

3. http server receives request, forms *response msg* containing requested object (courses/L10.index), sends message into socket

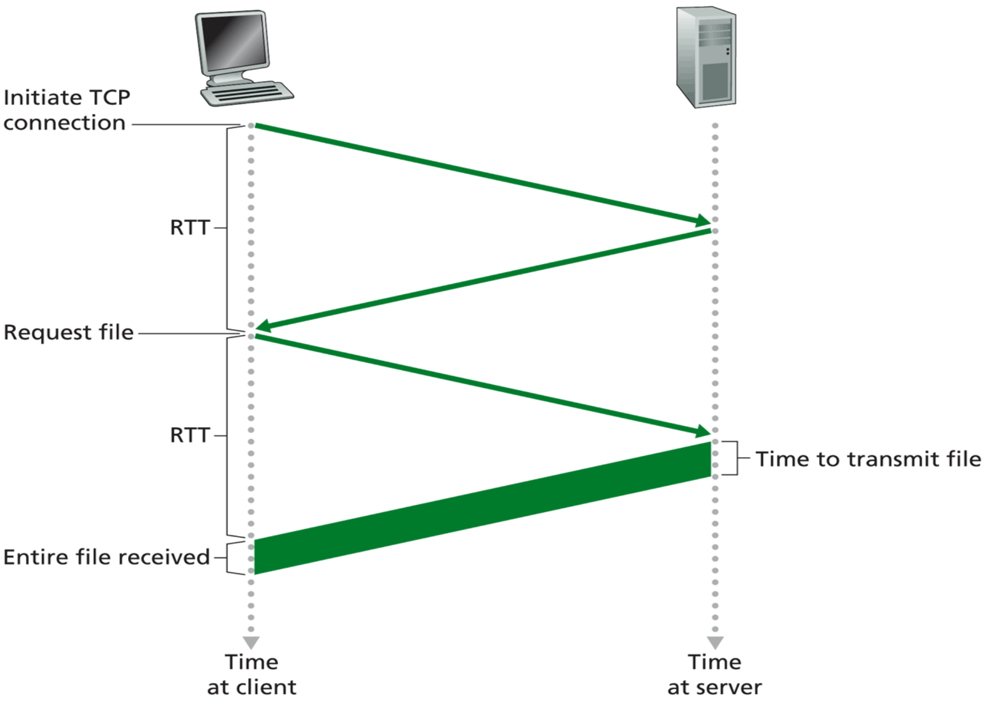
4. http server closes TCP conn.

5. http client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

6. Steps 1-4 repeated for each of 10 jpeg objects

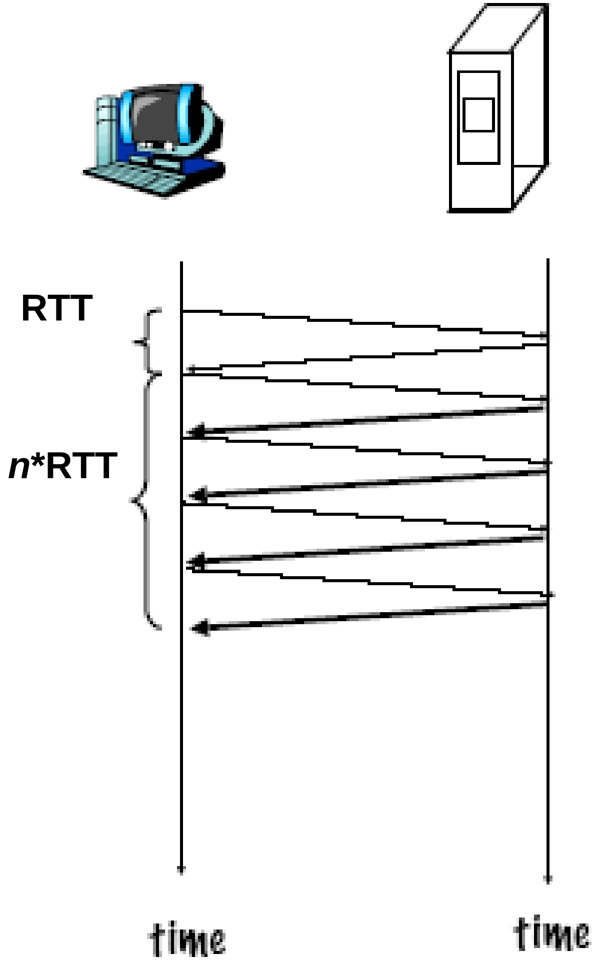
Non-Persistent HTTP Connections

- http/1.0 : server parses request, responds, closes connection (1 object per connection)
- Request time for a single object = 2*RTT + file transmission time
- Request time for n object = 2*n*RTT + files transmission time
- Browsers often open parallel connections



Persistent HTTP Connections

- Default for http/1.1
- TCP connection is initiated only once, All objects (files) are transmitted), TCP connection is closed
- Request time for n objects = (n+1)*RTT + transmission time



HTTP

- **Non-persistent:**

  n **objects → 2*n**RTT**

- **Persistent:**

  n objects → n*RTT

N.B: after considering base file

To be continued….