State information
Connection modes
Switching modes
Packet & circuit
Datagram & virtual circuit

Top-left corner indicates source of slide:
? From Kurose & Ross. See copyright notice at end of this set.
? From Tanenbaum
? From Moors
Outline

State information
Connection-oriented and connectionless communication
Switching forms:
  Circuit switching
  Message switching
  Packet switching
    Datagrams and virtual circuits
Measuring delay
References

Tanenbaum Sections:
- 1.3.3: Connection-oriented & -less service
- 2.5.5: Switching
- 5.1.1, .3, .4, .5: Store-and-forward, datagrams and virtual circuit switching
State information

State information is “memory in the system used to influence future behaviour”†

e.g.:

**Application layer**: In e-commerce, when adding objects to a “shopping basket”, the state of the shopping basket (current contents) is important.

**Transport layer**: To transfer information reliably from source to destination, source needs to remember what information has been transmitted but not yet acknowledged (⇒ may need to be retransmitted)

**Network layer**: May want to determine best route from source to destination, and configure routers/switches s.t. all packets follow this route (ensuring sequence).

Connections

When entities share state, they are said to be “connected”. Connections may exist between either:

adjacent layers (through service interfaces) or peer layers (through protocols)

Connections are “established”/“setup”, used, then “released”.
Avoid the words “open”/“close”, since their meaning differs between CS (open file ⇒ usable) & EE (open circuits don’t conduct)

See T § 1.3.3
Connection management

Communication is often classified as being either:

**Connection-oriented:**
- 3 distinct phases:
  1. Connection establishment
  2. Data transfer
  3. Connection release

  Classic example: Telephone network.

  *or*

**Connectionless:** No connections; data is pushed into the network

Classic example: Postal network.
Examples of connection(less)

Connection-oriented:
- Transmission Control Protocol (TCP – transport layer)
  - Client sends SYN segment; server responds with SYN+ACK
  - Client & server exchange data
  - Client or server sends RST/FIN
- Networks that provide hard service guarantees (e.g. phone)
  - Client requests certain service
  - Provider indicates whether it can honour the request.
    Reserves capacity for this request.
  - Client uses service.
  - Client disconnects
- Sockets bind() function: associate particular local port numbers with a socket.

Connectionless:
- User Datagram Protocol (UDP – transport layer)
- IP datagrams
- Ethernet transmission
Outline
Network traffic tends to be bursty

Network traffic is often bursty

e.g. individual using a web browser:

1. Send request
2. [propagation and server processing delays] 100ms
3. Receive response
4. [Process response – e.g. read web page] 10KB

Repeat

Other examples, e.g. compressed voice/video

- 1 minute
- hour
- overnight
- over weekend

Burstiness can be apparent:

- At different time scales, e.g. seconds, minutes, ...
  → “self-similar” traffic.
- When traffic from many users is aggregated

Often inefficient to dedicate resources to bursty traffic.
The Network Core

- mesh of interconnected routers
- the fundamental question: how is data transferred through net?
  - circuit switching
  - message switching
  - packet-switching
State in various switching modes

**Circuit switching**: Highly stateful:
- Route of data through network is programmed into switches
- Resources (e.g. bandwidth) for carrying data are reserved.

**Message switching**: Little state
- Whole message is sent without any association of parts
- Client/server may remember that a message has been sent / response expected.

**Packet switching**: Moderately stateful
- Source and destination must remember what packets of message have been sent/received
- Intermediate systems that route packets through network may:
  - Be stateless, e.g. datagram transfer
  - Have state about
    - route, e.g. virtual circuits
    - route + resources, e.g. virtual circuits with service guarantees

However, the amount of state does not, itself, determine whether switching mode is circuit switching or packet switching.
3 forms of switching

Note that the actual time depends on many factors ⇒ packet switching isn’t necessarily the fastest.
Circuit Switching

End-end resources reserved for “call”
- link bandwidth, switch capacity
- dedicated resources: no sharing
- circuit-like (guaranteed) performance
- call setup required

Switches can determine source (and destination) of information by virtue of when/where/how it arrives, irrespective of what arrive.

Switches are pre-programmed (during connections setup) to switch certain inputs to certain outputs, deterministically (e.g. frequency F on port 1 to frequency G on port 2)
Circuit Switching

network resources (e.g., bandwidth) divided into “pieces”
- pieces allocated to calls
- resource piece *idle* if not used by owning call (*no sharing*)

- dividing link bandwidth into “pieces”
  - frequency division
  - time division
Circuit Switching: FDMA and TDMA

FDMA

Example:
4 users

TDMA
Speed of circuit switching

A. Signalling to establish circuit

B. Data transfer

1. Transmission delay: \[ \text{time to send bits into link} \]
   \[ R = \text{link bandwidth (b/s)} \]
   \[ L = \text{packet length (bits)} \]
   \[ \text{transmission delay} = \frac{L}{R} \]

2. Propagation delay: \[ \text{d = length of physical link} \]
   \[ s = \text{propagation speed in medium (} \sim 2 \times 10^8 \text{ m/sec)} \]
   \[ \text{propagation delay} = \frac{d}{s} \]

**Note:** \( s \) and \( R \) are very different quantities!
Time delay

http://www.educatorscorner.com/index.cgi?CONTENT_ID=570
Problems with circuit switching

- Resources on links near source are held while waiting for resources on links near destination ⇒ inefficient.
  e.g. hold AB trunk while waiting for CD trunk

- Delay before any data can be transferred (2-way propagation delay aka “Round-trip time”)
  ⇒ large overhead unless transfers are long, but much traffic consists of short bursts.

- Resources are dedicated to certain sources, irrespective of whether they are needed. ⇒ inefficient for bursty traffic.
Message switching

Rather than continuously “stream” information through and end-to-end circuit across the network, store all information into one “message”, and send that one “hop” at a time across the network ⇒ “store-and-forward.” Each router stores the incoming message, and when it is completely received, forwards it on the next hop to the next router.
Message switching

Benefits:
✓ Can handle varying link rates.
✓ Good for asynchronous communication: Source and destination need not be available simultaneously. ⇒ used for email.

Limitations:
✗ Can’t start until source has complete message
✗ Large buffers in routers: must be capable of storing largest message transferred by end-system.
✗ Large delays for some users: One message may tie up resources (e.g. a transmission link) for long periods. Even if the other user has something short & urgent to communicate.
✗ Inefficient: If an error occurs, the entire message must be retransmitted.
Store-and-forward message switching

- Takes $L/R$ seconds to transmit (push out) message of $L$ bits on to link or $R$ b/s
- delay = $3L/R$

**Example:**
- $L = 7.5$ Mbits
- $R = 1.5$ Mb/s
- delay = 15 sec