Best-Effort Networks and Layering

Lecture 22
April 28, 2008
6.02 Spring 2008

• Understanding network delays
• Little’s law
• Best-effort network: losses, corruption, reordering
• Coping using layered protocols

Queues in Packet-Switched Networks

• Queues absorb bursts of arriving packets

• What happens if queue overflows?
  • Packets are dropped (lost)
• Can we design a packet-switched network to avoid such losses?
  • Turns out we can’t (or shouldn’t); it would be a bad idea to try!
Packets May Be Dropped in a Packet-Switched Network

- Can’t usually provision link for peak rate
  - Peak-to-average very high for data traffic (provisioning for peak wastes bandwidth)
  - Moreover, may not be able to predict traffic patterns well

- What if we use memory to simply buffer all packets?
  - Bad idea: long queues don’t help throughput
  - They only increase delay (intuitive; from Little’s law)
  - Use queues only to absorb transients (hard problem because it’s hard to tell what a “transient” is in practice)

- So, drop packets when queues overflow
  - [Or, drop packets pre-emptively to signal congestion early]
  - At a higher layer, detect and recover from loss
  - Not a new symptom: noise, interference/collisions, cause uncorrectable bit errors (so packet must be dropped)

Understanding Network Delay (Latency)

- Little’s law is an important queueing theory result
  - Relates mean number of packets (jobs) to processing rate and mean delay

- Four independent causes of delay
Understanding Little’s Law

- \( E[N] = \text{rate (or throughput)} \times \text{delay (or latency)} \)

\[ N(t) = \# \text{ pkts at time } t \text{ in queue} \]

- Suppose \( T \) is large and that \( S \) packets are forwarded in that time
- Let \( W = \text{area under } N(t) \text{ curve from 0 to } T \)
- Then, rate = \( S/T \); and mean number of pkts in queue, \( E[N] = W/T \)
- How to calculate mean delay per forwarded packet?
  - \( W \) is aggregate delay weighted by each packet’s time in queue (why?)
  - So, mean delay per packet sent = \( W/S \)
- Therefore, \( E[N] = \text{rate} \times \text{(mean delay)} \)
- For a given link rate, increasing queue size increases delay

Networks Have Four Kinds of Delays

- Propagation delay
  - Time for one bit to go from sender to receiver
  - Depends on speed of light in communication media

- Transmission delay
  - Time for packet of size \( S \) to reach other end
  - Depends on rate, \( R \), of path (delay = \( S/R \))

- Processing delay
  - Time for each switch to process packet (lookup, etc.)

- Queueing delay (variable)
Packets May Be Reordered

- Routing protocol can choose different paths
- Helps recover from link/switch failures
- May help balance load across different links or paths

```
A  C  F
|
B  D  E
|
G  
```

A to G: Could pick ACFG, ABDEG, ACEG, etc.

“Best-Effort” Network

- Packets may be lost
- Packets may be corrupted
- Packet delays are variable
- Packets may be reordered
- Packets may even be duplicated

- But, it’s relatively easy to be a switch in a network with such a “contract”

- How to find good paths and deliver data in the face of these best-effort properties?
Getting Organized

- Need modularity

![Diagram of A and B with call and result](image)

- A need not know how B is implemented
- B’s implementation can change as long as B’s interface doesn’t
- That is, the abstraction provided by B is the only contract between B and A

Layering

- Layering is a particular form of abstraction
- The system is broken into a vertical stack of protocols
- The service provided by one layer is based solely on the service provided by layer below
  - This is the “up/down” interface
- Peer interfaces across the network implement communication protocols
Layering in the Internet

HTTP, FTP, SMTP, BitTorrent, …

TCP, UDP

IP

Ethernet, WiFi, …

Application

Transport

Network

Data Link

Physical

End-to-End Layer
Everything else!
Reliability, integrity,
ordering, jitter ctrl
(smoothing), congestion control, …

Forwarding & routing
(and addressing)
Framing, (limited) retry,
channel access
Coding, (De)modulation

Layering Interfaces

Host

Application

Transport

Network

Datalink

Physical medium

Host

Application

Transport

Network

Datalink

Switch (Router)

PROTOCOL:
Peer-layer communication

• Link and network layers are implemented everywhere
• The end-to-end layer (i.e., transport and application) is implemented only at hosts
An Example

How Layering Works

- Each layer adds/strips off its own header (and possibly a trailer)
- Each layer may split up higher-level data
- Each layer multiplexes multiple higher layers
- Each layer is (mostly) transparent to higher layers
The Internet “Hourglass”

**Applications**
- Web
- FTP
- Mail
- News
- Video
- Audio
- ping
- napster

**Transport protocols**
- TCP
- SCTP
- UDP
- ICMP

**Link technologies**
- Ethernet
- 802.11
- Power lines
- ATM
- Optical
- Satellite
- Bluetooth

- Many applications, transports, and link protocols
- All use IP at the network layer: universal network layer

The Link Layer

- Ethernet, WiFi, ...
- Data Link
  - Framing, (limited) retry, channel access
- Physical
  - Coding, (De)modulation
**Link Layer Framing**

- **Q:** What if SYNC appears in payload?
- **One solution:** *bit stuffing*
  - A common SYNC is **01111110**
  - At sender: If 5 ones in a row, insert 0
  - At receiver: If 111110abc, decode as 11111abc, stripping out 0 following 5 ones
  - If receiver gets 0111111, then:
    - If next bit = 0, then SYNC, else corruption
- **Drawbacks?**
  - Worst case frame size could be quite large
  - What happens if there are bit errors?

---

**Using Length to Frame**

- No need for bit-stuffing with explicit length
- But what happens when there are bit errors?
- With errors, both bit-stuffing and explicit length may cause
  - Truncated frames
  - Dropped frames
- Remedy: Protect headers with separate CRC
Error Detection

- Use checksum or cyclic redundancy check
  - In lab 5, checksum using block parity
- Checksum may fail due to noise, collisions
- Could discard, or to improve performance, could retry missing frame
  - How many times should link layer retry before giving up?
  - A small number of times is enough - because packets get lost for many reasons, the sender will eventually retransmit if it cares
  - An example of an *end-to-end argument*

Summary

- In a *best-effort* packet-switched network:
  - Packets could get lost, corrupted, reordered, duplicated
  - And experience unpredictable delays
- Four kinds of delays in networks
  - Propagation, transmission, processing, queueing
  - Little’s law: $E[N] = \text{rate} \times \text{mean delay}$
- Design network protocols in layers to cope with best-effort properties