Computer Networks

Transport Layer

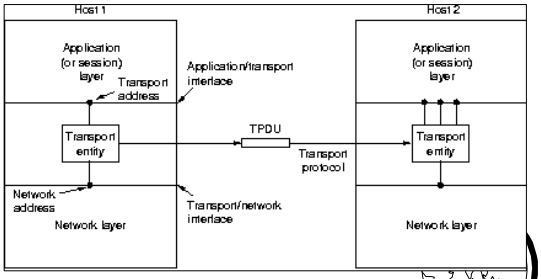
Topics

- Introduction (6.1) ←
- Connection Issues (6.2 - 6.2.3)
- TCP (6.4)

Introduction

- Efficient, reliable and cost-effective service to users (application layer)
  - despite limitations of network layer
- Features (a lot like the Network layer?)
  - *Connection oriented* vs. *Connectionless*
  - *Addressing* and *Flow Control*
- But Transport layer can make lower subnet *reliable (QoS)*, and gives *standard interface*
- Major boundary between *user* and *network*!
  - Few users write code for network layer
  - Many write code for transport layer

Transport Entity



- Logical location of transport entity
- Physical: OS, separate process, network card

Quality of Service

Connection establishment delay
Connection establishment failure probability
Throughput
Transit delay
Residual error ratio
Protection
Priority
Resilience

- Typical networks do not do all

Transport Protocol

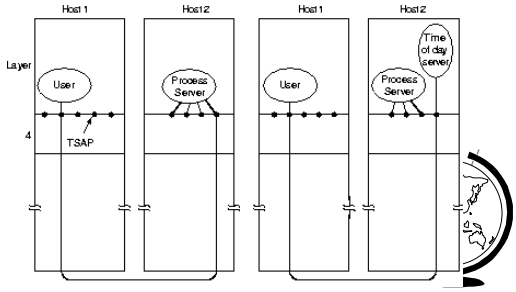
- Like Data Link layer:
  - error control, sequencing, flow control…
- But different:
  - must specify router (data link layer always same)
  - destination may be down
  - network may store packets
  - many lines and variance make buffering and flow control different

Finding a Server

- "Connect to a Server" is a Transport level service
- How do you find it?
  - *service mapper* - names to transport layer address
  - *name server*
- Analogy
  - how do you find phone number?

Finding a Server

- Standard servers wait at well-known port
  - but what if infrequently used?



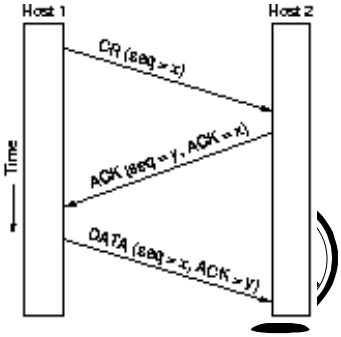
Establishing a Connection

- Subnet can delay, lose, duplicate packets
  - Connection can happen twice!
  - Use *unique sequence numbers* to avoid
- When establish connection, exchange sequence numbers
  - *three-way handshake*
  - prevents establishment of unwanted connection

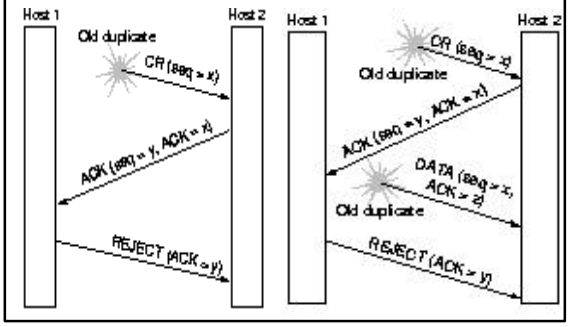
Three-Way Handshake

CR = Connection Request

ACK = Connection Accepted

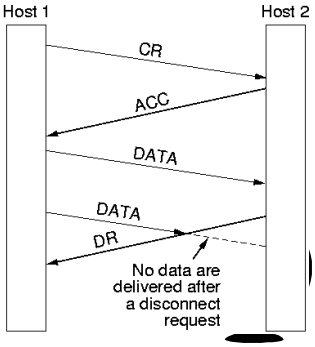


Three-Way Handshake Handles Problems

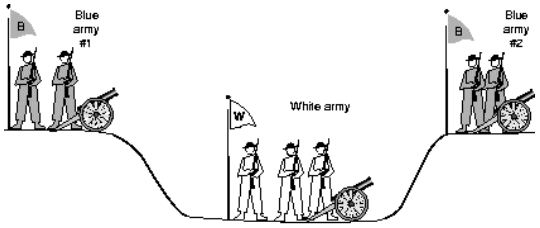


Releasing a Connection

- Asymmetric release can result in data loss
- Symmetric release easy?
  - "I'm done"
  - "Me, too"



Two-Army Problem

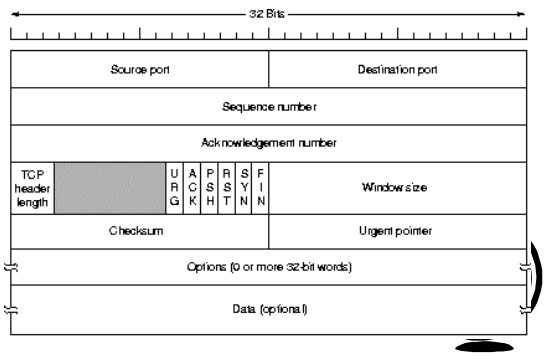


- No safe solution
- Use 3-way handshake with timers (fig 6-14)

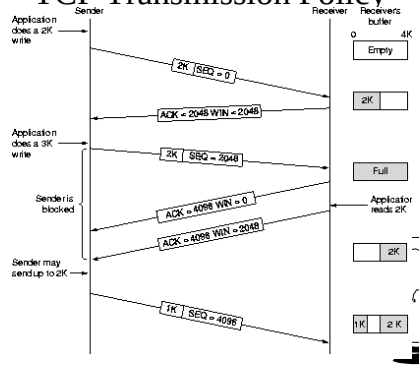
TCP

- Connection-oriented
- Reliable, end-to-end byte-stream
  - message boundaries not preserved
- Adapt to a variety of underlying networks
- Robust in the face of failures
- Break data into *segments*
  - 64 Kbytes max (often, only 1.5 Kbytes)
  - 20 byte header
- Sliding window

TCP Segment Header



TCP Transmission Policy

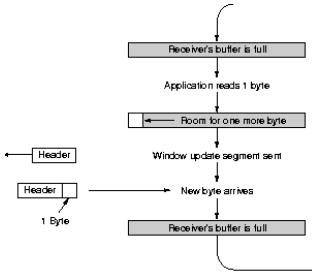


TCP Transmission Policy

- Do not have to send immediately
  - avoid many small packets
- *Nagle's Algorithm*
  - only 1 outstanding byte at a time
  - fill up, then send
  - time delay, then send
  - bad for some apps (X - with mouse movements)

Silly Window Syndrome

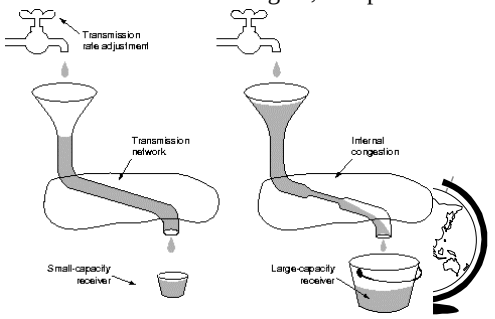
- Application reads 1 byte at a time



- Fix: only send window when 1/2 full

TCP Congestion Control

- Even if sender and receiver agree, still problems



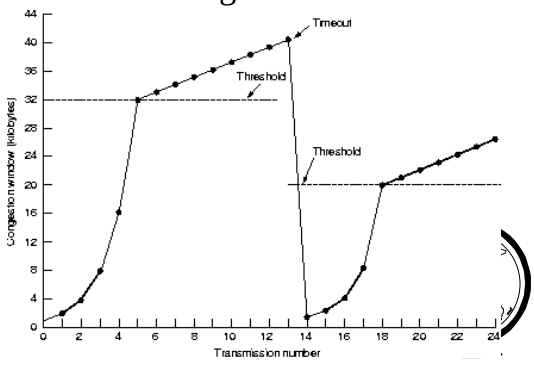
TCP Congestion Control

- "Receiver buffer" via receiver's window
- "Network buffer" via congestion window
- "Effective buffer" is minimum of receiver and network
- Ex:
  - Receiver says "8k", Network says "4k" then 8k
  - Receiver says "8k", Network says "32k" then 8k

Avoiding Congestion

- Network buffer
  - starts at 1 segment
  - increases exponentially (doubles)
  - until timeout or receiver's window reached
  - or threshold, then increases linearly
  - *slow start* (required by TCP)
- Internet congestion includes threshold
  - linear past threshold (called *congestion avoidance*)
  - when timeout, reduce threshold to half of *current window* and restart slow start
    - can go up

TCP Congestion Control



TCP Congestion Control Summary

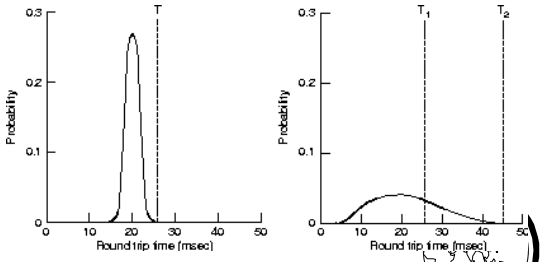
- When below threshold, grow exponentially
  - slow start
- When above threshold, grow linearly
  - congestion avoidance
- When timeout, set threshold to 1/2 current window and set window to 1
- How do you select timer values?
  - Important, since timeouts restrict throughput
  - Timer management

Timer Management

- Want to set timeout to minimal value where segment is known to be lost
  - should be just larger than current round-trip time (RTT). Why?
- So, need estimate of round-trip time (RTT)
  - how to get it?
- Why can't you just measure RTT once and fix timeout timer?

Timer Management

- Difficult when much variance



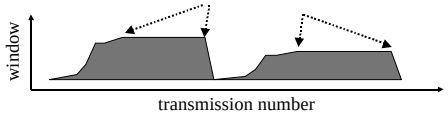
- RTT = αRTT + (1-α)M (α = 7/8, M ack time)
- + add variance, don't update on retransmits

Enhancement to TCP, or … A Trip to Nevada

- Tahoe (traditional TCP)
- Reno (most TCP implementations)
- Vegas (not yet, but may be coming)

TCP Tahoe

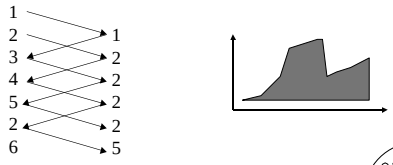
- Tahoe can have long flat periods
  - why?



- Can we avoid some of these long waits
  - Use duplicate acks

TCP Reno

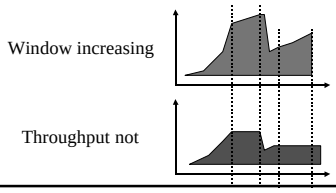
- If see three duplicate acks, then retransmit
  - *fast retransmit*



- And when 3 acks, just halve congestion window and do congestion avoidance
  - *fast recovery*

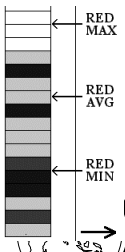
TCP Vegas

- Tahoe and Reno react to congestion
- Vegas attempts to avoid congestion
  - detect congestion before loss occurs
  - lower rate linearly
- Base detection on increasing RTT



Random Early Drop (RED)

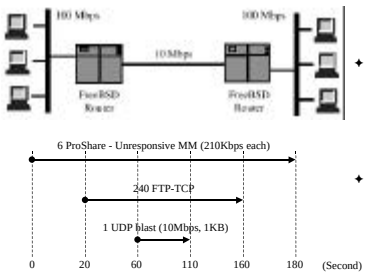
- Traditional Internet routers FIFO
- Limitations
  - FIFO cannot detect congestion until too late
- Instead, detect congestion
  - below min, nothing
  - above min, then probabilistic
  - above max, drop all
- Note, red average, not instant



Explicit Congestion Notification

- Routers use loss as a means of indicating congestion
  - FIFO can't help it
  - RED tries to tell TCP flows congestion is coming
  - *implicit*
- Instead, routers can indicate congestion with a bit
  - *explicit*
- In acks to sender, better but tough (why?)
  - so on outgoing packets

Non-Responsive Flows and Fairness



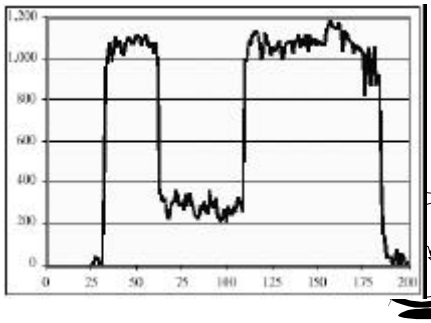
- RED Settings:

```
qsize   = 60 pkts
max-th  = 30 pkts
min-th  = 15 pkts
qweight = 0.002
max-pro = 0.1
```

- CBT Settings:

```
mm-th     = 40 pkts
udp-th    = 2  pkts
```

Aggregate TCP Throughput



Aggregate TCP Throughput with CBT