Read 7.E

Reliability & Flow Control

Prof. Dina Katabi

Some slides are from lectures by Nick McKeown, Ion Stoica, Frans Kaashoek, Hari Balakrishnan, and Sam Madden
Previous Lecture

How the link layer delivers data over a link

How the network layer performs routing and forwarding

Hierarchical Routing and Addressing
Hierarchical Routing

Internet: collection of domains/networks
Inside a domain: Route over a graph of routers
Between domains: Route over a graph of domains
Address: concatenation of “Domain Id”, “Node Id”
Hierarchical Routing

Advantage
scalable
Smaller tables
Smaller messages
Delegation
Each domain can run its own routing protocol

Disadvantage
Mobility is difficult
Address depends on geographic location
Sup-optimal paths
E.g., in the figure, the shortest path between the two machines should traverse the yellow domain. But hierarchical routing goes directly between the green and blue domains, then finds the local destination path traverses more routers.
This Lecture

Transport Layer

Reliable data transmission
Flow Control
Multiplexing
Layering

The 4-layer Internet model

Application
Transport
Network
Link

HTTP, FTP
TCP
IP
Ethernet

End-to-End Layer
Transport Layer

Network layer provides best-effort service
  Loss, delay, jitter, duplicates, reordering, ...
  Not convenient for applications
Transport layer builds on the best effort service to provide applications with a convenient environment
  Reliability:
    • at least once
    • at most once
  Performance:
    • flow and congestion control
Ordering
  Data integrity (checksum)
  Timeliness (remove jitter)
Also transport provides multiplexing between multiple applications
This Lecture

Transport Layer

- Reliable data transmission
- Flow Control
- Multiplexing
At Least Once

Sender persistently sends until it receives an ack
How long should the timeout be?

Fixed is bad. RTT changes depending on congestion
- Pick a values that’s too big and it will wait too long to retransmit a packet,
- Pick a value too small, and it will unnecessarily retransmit packets.

Adapt the estimate of RTT adaptive timeout
RTT Measurements
(collected by Caida)
Adaptive Timeout

Samples \( S_1, S_2, S_3, \ldots \)

Algorithm

\[
\text{EstimatedRTT} = T_0
\]

\[
\text{EstimatedRTT} = \alpha S + (1 - \alpha) \text{EstimatedRTT}
\]

where \( 0 \leq \alpha \leq 1 \)

What values should one pick for \( \alpha \) and \( T_0 \)?

Adaptive timeout is also hard
Different Approach: NACK

Minimize reliance on timer
Add sequence numbers to packets
Send a Nack when the receiver finds a hole in the sequence numbers

Difficulties
Reordering
Cannot eliminate acks, because we need to ack the last packet
At Most Once

Suppress duplicates
Packets must have ids to allow the receiver to distinguish a duplicate from a new packet
Receiver should keep track of which packet ids have been delivered to applications
To simplify tracking, senders pick monotonically increasing packet ids, i.e., sequence numbers
Receiver delivers packets to application in order. It keeps track of the largest id delivered so far
This Lecture

Transport Layer

- Reliable data transmission
- Flow Control
- Multiplexing
How fast should the sender sends?

Waiting for acks is too slow
Throughput is one packet/RTT
Say packet is 500B
RTT 100ms
Throughput = 40Kb/s, Awful!
Overlap pkt transmission
Send a window of packets

Assume the receiver is the bottleneck
Maybe because the receiver is a slow machine
Receiver needs to tell the sender when and how much it can send
The window advances once all previous packets are acked too slow
Sliding Window

Senders advances the window whenever it receives an ack sliding window

But what is the right value for the window?
The Right Window Size

Note that

if $W/RTT < \text{Bottleneck Capacity}$ under utilization
If $W/RTT > \text{Bottleneck Capacity}$ Large queues
This Lecture

Transport Layer
- Reliable data transmission
- Flow Control
- Multiplexing
Multiplexing by Transport

Multiple applications run on the same machine but use different ports