CS 716: Introduction to communication networks

- 3rd class; 29th July 2011

Instructor: Sridhar Iyer
IIT Bombay

Assignment 1 (To do in-class)

Q1: TCP is a transport layer protocol that provides a service to the application layer. TCP guarantees reliable, in-sequence delivery of packets between a sender and a receiver, over an unreliable network. Your goal is to design a TCP-like protocol.

1. Describe the steps/actions that would be performed by the sender and the receiver in your protocol, to ensure reliable, in-sequence delivery.
2. Explain the working of your protocol using an example, including the case when a packet is lost.
3. State the key assumptions in your protocol. For example, limit on buffer size?

Assume that the network layer provides the routing service (to TCP), i.e., given a packet and a destination, it takes care of forwarding the packet along multiple hops to the destination. Note that: (i) Different packets between a sender-receiver pair may take different paths, and (ii) Packets may get discarded (lost) at any intermediate point, without any warning/reason.

Assignment 1 (contd ...)

Q2: Have you taken a networking course before.

- If yes, at what level (UG/ PG), and in which department?

Q3: Did you attend either of the two previous classes in this course (CS 716)?

- If yes, which classes (22nd July/ 27th July/ Both)?

Q4: Did you use the CEO example to solve the TCP question (Q1)?

- If yes, how helpful was it (very helpful/ somewhat helpful/ not helpful)?

Assignment 1 (contd ...)

Q5: If you were present for the CEO example discussion:

a) How did Secretary A infer that a messenger boy has failed to deliver a section of the document?

b) How did Secretary A ensure that lost sections of the document could be re-sent?

c) How did Secretary B ensure proper delivery of the document to CEO B?

Grading: Each assignment will be 1% weightage. Best 5 out of N such assignments will be considered. Simply coming to class will ensure that you get most of these marks!

Discussion of solution to Asgt1-Q1

- Exchange answer books with your neighbour and discuss; See if you have got the below features.
- Key features for sender side:
  - Maintain application data in a buffer, Do packetization, Insert sequence numbers, Send to network layer, Start re-transmission timer and wait for ACK.
    - If timeout before ACK, re-transmit packet.
    - IF ACK before timeout, delete packet from buffer.
- Key features for receiver side:
  - Send ACK for received pkts, Maintain received pkts in a buffer, Do re-assembly of data, Send to upper layer.

Recap of previous classes

- Overview of networking
  - Analogy: Communication between CEO A and CEO B, given messenger constraints.
  - Group discussion led to discovery of some concepts:
    - Layering; Protocols; Packetization; Addressing, …
- Mapping of concepts from Analogy to Networking.
- Importance of ABCDE problem-solving process:

  (a) assume simple scenarios, (b) brainstorm for possible solutions, (c) choose one solution that satisfies constraints, (d) do the detailing, (e) examine correctness and completeness.

Today: Fast forward to TCP functioning

Application data is broken into Segments (what TCP considers the best sized units to send)

- Segment: unit of data passed from TCP to IP
- MSS: Maximum segment size

TCP sequences data by associating a sequence number with every **byte** it sends

Sending TCP maintains a timer for each segment sent

- waiting for acknowledgement (ACK)
- If ACK doesn't come in time, segment is retransmitted

TCP functioning (contd ...)

Receiving TCP

- Sends ACK: ACK number is the sequence number of the next byte expected
- Re-sequences the data
- Discards duplicates

•Congestion and Flow control

- Sending TCP regulates amount of data to avoid network congestion
- Receiving TCP prevents fast senders from swamping it

Questions (for later classes)

- How does the transport layer at S distinguish between packets coming down to it from multiple applications (ex: http and ssh)?
  - Their destination (D) may be the same or different.
- How to determine the number bits to allocate for the sequence number (unique packet id)?
  - Suppose you use 3 bits, the 1st packet and 8th packet will both have [000] as the sequence number.
- How to decide the number of packets that S could transmit before it waits for the 1st acknowledgment?
  - Suppose you transmit one packet, wait for its ack, then the next packet and so on, what is the drawback?

Reflection

- What did I learn in today's class?
- Each student to mention one point.

- Take-home questions:
  - What functions are part of the service interface of a layer? What functions are part of the protocol of a layer?