

CSE 588 Class Notes
4/23/2002 (2nd half)

Mobility

Mobility can be supported if address is fixed
IP mobility – Additional aggregation will break if IP is broadcasted.

Name routing -

Why do we need name routing?

- Applications can handle IP changes

Anycast -

- Select the closest server to the client, client will decide.
- Router can decide for the client

We will discuss about this paper in detail when we talk about peer-peer networks.

TCP

Basics of TCP were discussed in this class.

How to build a reliable transport ?

IP gives some reliability but it is not sufficient. TCP is designed to provide reliability.
TCP is an end to end transport protocol.

APIs for TCP – Sockets

Communication happens through two way byte stream

1. Listener (server)
2. Connect (client) – client can send arbitrary amount of data

A portion of the TCP packet header looks like this

| |
|---------------|
| To <port #> |
| From <sender> |
| Sequence # |
| Acknowledge # |
| Length |
| |

Sequence numbers are randomly picked.

Acknowledgment number is the next byte expected.

The following are the sequence of steps that occurs during a TCP connection.

1. Client sends SYN and Sequence number to the server
2. The server sends back the SYN+ACK and Sequence number (a different seq number). $ACK = \text{previous Seq number} + 1$
3. Client sends Ack for server's response with $ACK = \text{server's seq number} + 1$
4. From here onwards, client starts sending data and server responds with an acknowledgement.

How does HTTP connection work ?

- Listen on port 80.
- Wait in the kernel for connections to port 80
- If 2 clients are sending from same machine, the "from" port will be different. The port is picked randomly by kernel.

Sliding Window

If the client sends a packet and waits for acknowledgement from the server for each packet before sending the next packet, it is going to be slower.

If round trip time is more, sliding window sends more packets at once. Different queues are maintained at client end –

1. packets sent and received ack.
2. packets sent and not received ack.
3. packets not yet sent.

The corresponding queues on the server side are,

1. received packets and sent ack.
2. received but in out of order

To improve performance, Selective ACK (SACK) be done which is ACKing for range of bytes received.