CS 456 Hao Pan
Salahuddin, Mohammad
Spring 2018

Chapter 2.7 Socket Programming With UDP and TCP

07/06/2018 [Th]

2.7.1 Socket Programming

- The **goal** is to learn how to build client-server applications that communicate using sockets.
- **Sockets** are doors between application processes and end-end-transport protocols.
- There are two socket types for transport services:
  - **UDP**, an unreliable datagram.
  - **TCP**, which is reliable and byte stream-oriented.

2.7.2 Socket Programming with UDP

- There is *no connection* between client and server. That means that there is no handshaking before sending data, the sender attaches IP destinaton address and port number to each packet, and the receiver extracts that info from the received packet.
- Note that data transmitted through UDP may be *lost* or received *out-of-order*.
- In the *application viewpoint*, UDP provides an unreliable transfer of groups of bytes between client and server.
- See example of UDP client-server socket interaction on slides 2-102 to 2-104.

2.7.3 Socket Programming with TCP

- The client must *contact the server*, which must be running at the time of contact and have already created sockets which welcome client contact.
- Clients contact server by creating TCP sockets and specifying IP address and port number of server processes. When the client creates a socket, the client TCP establishes a connection with the server TCP.
- Once contacted by a client, the server TCP creates a new socket for the server process to communicate with that client. Servers can talk with multiple clients, where source port numbers are used to distinguish clients.
- In the *application viewpoint*, TCP provides reliable, in-order byte-system transfers (”pipe”) between client and server.
- See example of TCP client-server socket interaction on slides 2-106 to 2-108.