

- Name one task that can be duplicated at various layers?
- Write down the names of all internet protocol layers.
- What is an application layer message?
- Which layers are processed by link layer switches?
- What is the difference between a virus, a worm and a Trojan horse?
- What's a botnet?
- How can you read one line of text at a time from a file called "data.in"?
- What are the functions of the ServerSocket() and accept() methods?
- When would you prefer to have a concurrent server?

1

Chapter 2: Application layer

- 2.1 Client/ Server network applications
- 2.2 Web and HTTP
- **2.3 FTP**
- 2.4 Electronic Mail
 SMTP, POP3, IMAP
 2.5 DNS

2.6 P2P applications

- 2.7 Socket programming with TCP
- 2.8 Socket programming with UDP
- 2.9 Building a Web server

Connectionless Servers

The examples shown so far are based on connection-oriented transfers

- Provides reliable, bidirectional, point-to-point stream based connections between two hosts
- * Based on the TCP protocol
- Java also supports connectionless transfers using datagrams
 - Based on the UDP protocol
 - In situations faster communication is needed, and not care if few packets are lost

Java and UDP

- Java implements datagrams on top of the UDP protocol using two classes
 - The DatagramPacket class acts as the data container
 - The DatagramSocket class is the means for sending and receiving datagrams
- Datagram packets can be created using one of the two constructors with prototypes:

1. DatagramPacket (byte buf[], int size);

2. DatagramPacket(byte buf[], int size, inetAddr addr, int port)

The first form is used to receive data and the second form is used to send data

Connectionless Server Example

Both the client and server are combined in the same program

- The server program must be invoked as:
 - java DatagramDemo server
- The client program must be invoked as:
 - java DatagramDemo client

Pure P2P architecture

- □ *no* always-on server
- arbitrary end systems directly communicate peer-peer
- peers are intermittently connected and change IP addresses

Highly scalable but difficult to manage



<u>App-layer protocol defines</u>

- Types of messages exchanged,
 - e.g., request, response
- Message syntax:
 - what fields in messages & how fields are delineated
- Message semantics
 - meaning of information in fields
- Rules for when and how processes send & respond to messages

Public-domain protocols:
defined in RFCs
allows for interoperability
e.g., HTTP, SMTP
Proprietary protocols:
e.g., Skype