

1 Chap0 8Fallacies

1.Network Reliable. |PowerSupply, Hard-Disk, NodeFailures, Configurations, Bugs| **Effect:**applicationHangs, crashes |**Countermeasures:** Redundancy HW&SW systems,middleware &aplication; CatchExceptions, CheckCodes, React;Retry ConnectingUponTimeouts

2.LatencyZero. |Latency:timeForData Transfer(speedOfLight)& Bandwidth:howMuchData transferred

3.Bandwidth is infinite.

4.The network is secure.

5.Topology doesn't change.

6.There is one administrator.

7.Transport cost is zero.

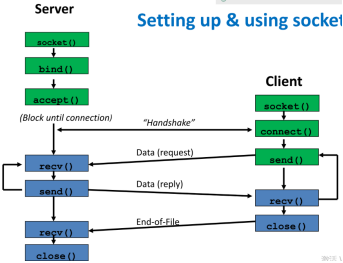
8.The network is homogeneous

2 Chap1-CommunicationBasics NETWORKING BASICS

ISO OSI: Open Systems Interconnection model, Basis for standards development on systems interconnection.

SOCKET

Move data/message/(invoke operation/-service and return result/failure) from Application I on Host A to Application K on Host B. **Client:**Issues requests to server(send & receive). **Server:**Starts up and listens for connections, requests, and sends/receives. **Client/Server examples:** telnet/telnetd, ftp/ftpd (sftp/sftpd), Fire-fox/Apache. **Socket:** network programming abstraction for communicating among processes (applications) based on (Unix) file descriptors. **File descriptor:**an integer representing an open file managed by the OS \In Unix any I/O is done by reading/writing from/to file descriptors. **Socket types:** Stream socket:java.net.ServerSocket, TCP based, Ordering guaranteed, Error-free \Datagram socket:java.net.DatagramSocket, UDP based \IPv4 & IPv6



NIO(Nonblocking sockets)

Synchronous: Single thread reading data from clients(stream) and blocked until ready(no multiple read) **Asynchronous:** Single thread reading data from clients: Thread → Channel: read data into buffer, Channel → Buffer: fill data into buffer, Thread → Buffer: check data in buffer

(main thread not blocked) **Synchronous vs. Asynchronous:** S: A thread enters into action and waits until I/O is completed \Limited scalability, one thread per I/O connection(Overhead:context switching → time between diff. tasks) A: Passes the request immediatly to the OS-kernel and then do other tasks → worker thread **while (true)** { only do computation, never blocked, no context switch **Java NIO Channels:** All IO operations can be done with channels(File, TCP, UDP) \Multiple types of channels(FileChannel (File on disk),DatagramChannel (UDP), SocketChannel (TCP, support concurrent read/write), ServerSocketChannel (TCP)) \Responsibilities(Read, write buffer)

UI Finite state machines that describe a **communication session between a client and a server**. The first FSM represents the server and the second FSM represents the client. Both parties (client and server) keep the communication session open and exchange messages until one of them decides to close

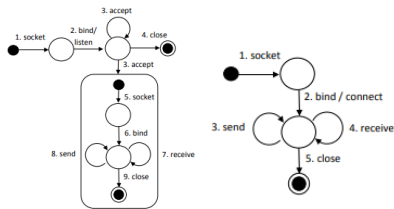
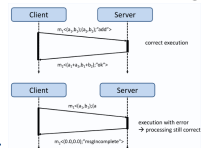


Figure 1.1: FSM for server.

Figure 1.2: FSM for client.

it **accept** is **while** loop, detail in rectangle: create a new socket (and therad) for commu. with client **Simple protocol design** complex number as string: $c_i = (a, b)$, $op \in \{add, sub, mul, div\}$, C to S message format: $m_1 < c_1; c_2; op >$, Status: $st \in \{OK, msgIncomplete, \dots\}$, S to C message

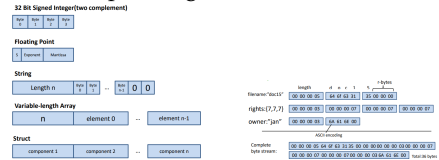


format: $m_2 < C_r; st >$

3 C2 EXTERNAL DATA REPRESENTATION, Presentation Layer

Heterogeneity HW: Diff. HW architectures store bytes:Big, Small Endian ProgrammingLanguage:Diff. PL store data types differently:AB, 0AB **Transformation between representations:** Transformation between local and remote representations|Information may be lost **Two realizations:** 1.Pairwise transformation between n local representations(vollständigGraph, $\#n^2 - n$, Either sender or receiver has to transform) 2.**Transformation to and from canonical representation**(a single canonical

representation|No local information about communication partner needed| $\#2 * (n - 2)$, -2 if canonical is one of n) **XDR** encodes only data items, not information about their types |exactly 32 bit integer is stored according to **big endian** |Data is encoded into blocks of multiples of 4: n-bytes contain data; r-bytes are used for padding with $n + r \bmod 4 = 0$



class file:String filename;
int rights<>;String owner;
ASN.1:Abstract description of data types |Enables exchange in hetero-

geneous systems



Java object serialization, JOS Stream-based transmission of serialized objects(Via TCP or UDP sockets), Receiver of object needs implementation of class, Serialization does not require class specific code(Java reflection), Classes implements java.io.Serializable interface **serialize:obj2bit**Socket s = **new** Socket("localhost", 8022);ObjectOutputStream oos = **new** ObjectOutputStream(s.getOutputStream()); oos.writeObject(obj); **deserialize:bit2obj**ServerSocket ss = **new** ServerSocket(8022);Socket s = serverSocket.accept();ObjectInputStream ois = **new** ObjectInputStream(s.getInputStream());obj=(Obj)ois.readObject();