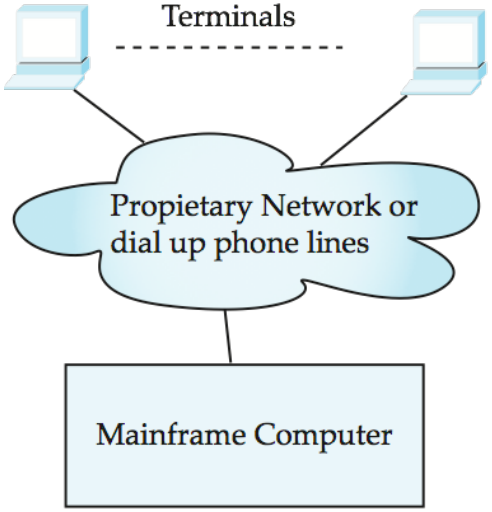
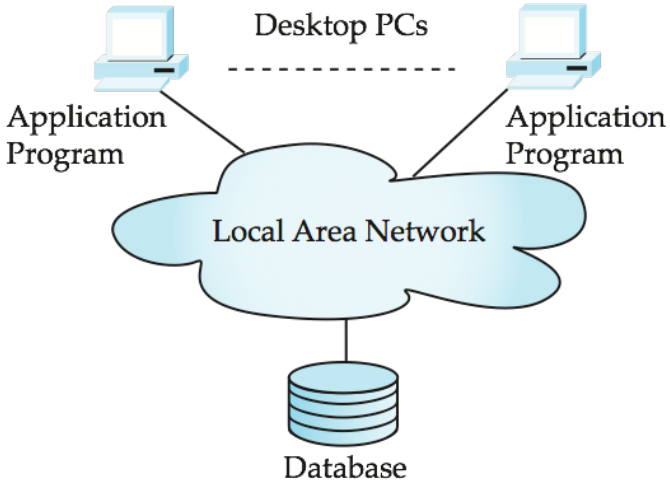
Application Architecture
Networking
Programming Assignment #1

Application Architecture Evolution

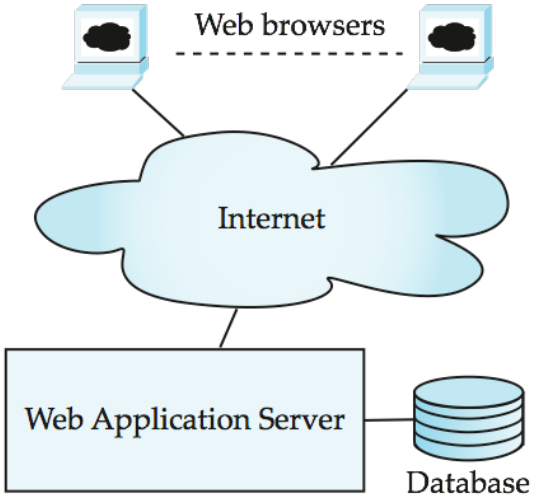
- Three distinct era's of application architecture
  - mainframe (1960's and 70's)
  - personal computer era (1980's)
  - C/S era (1990's onwards)
  - Cluod (today …)



(a) Mainframe Era

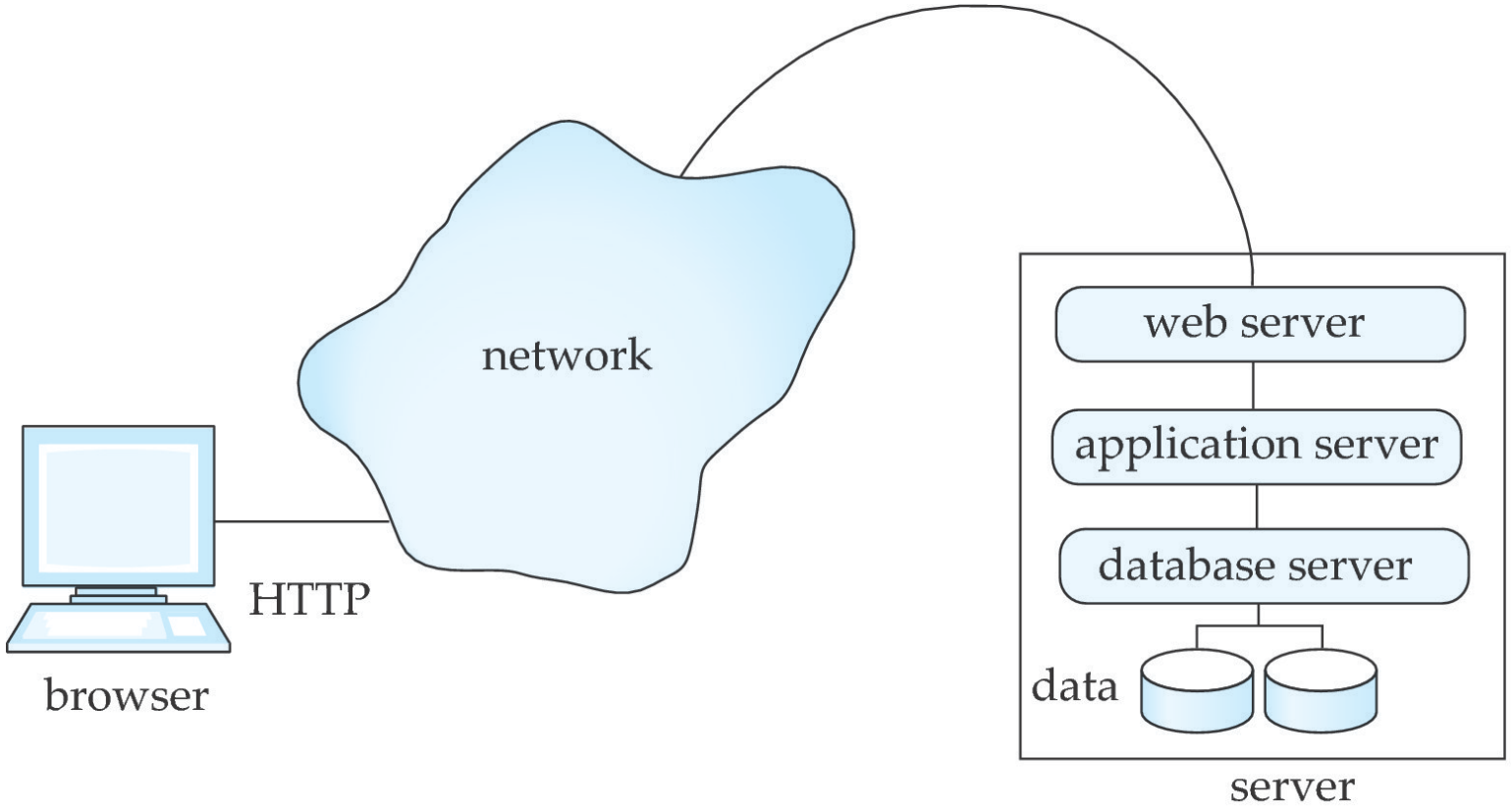


(b) Personal Computer Era



(c) Web era

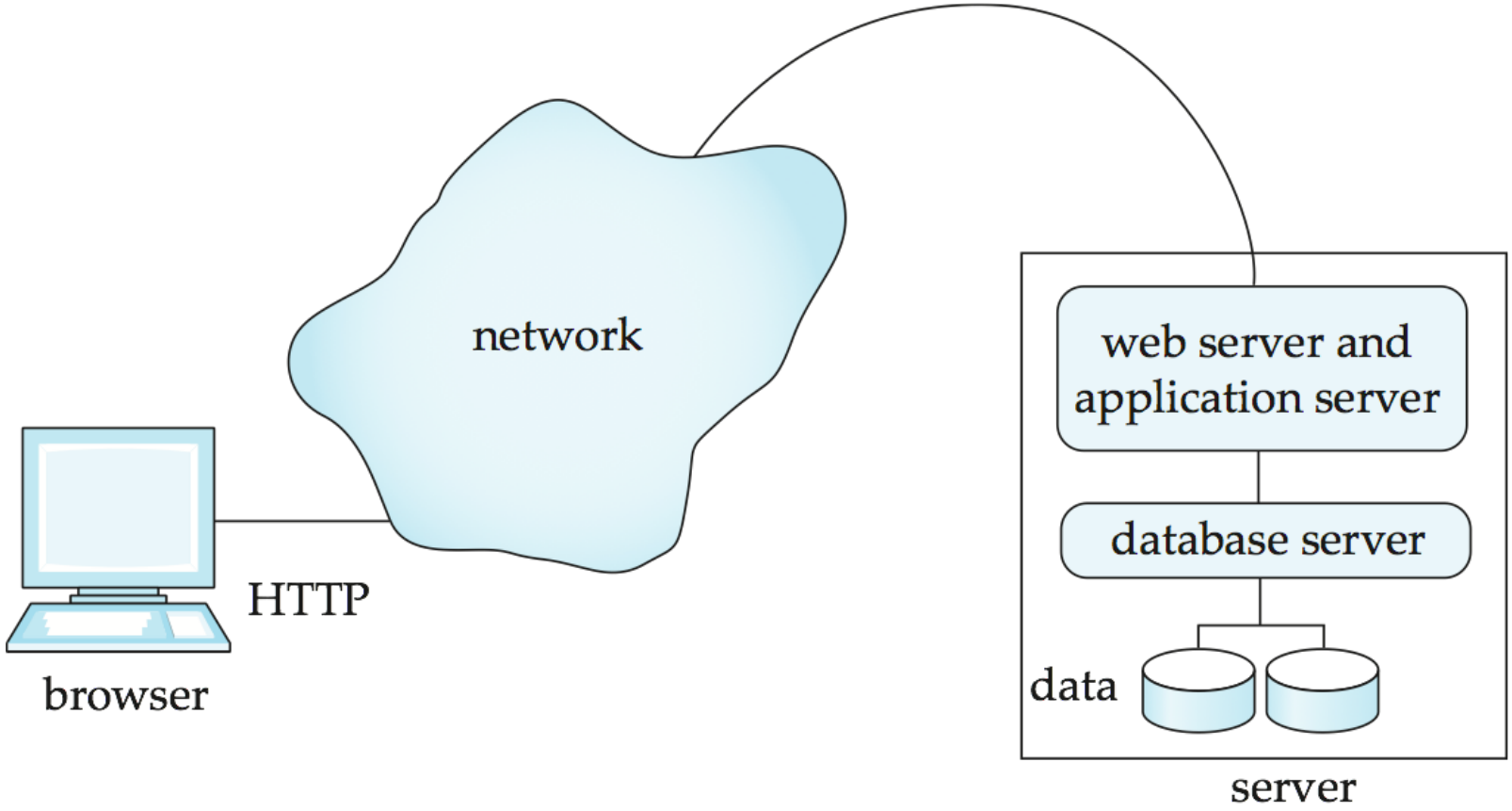
Three-Layer Web Architecture



Two-Layer Web Architecture

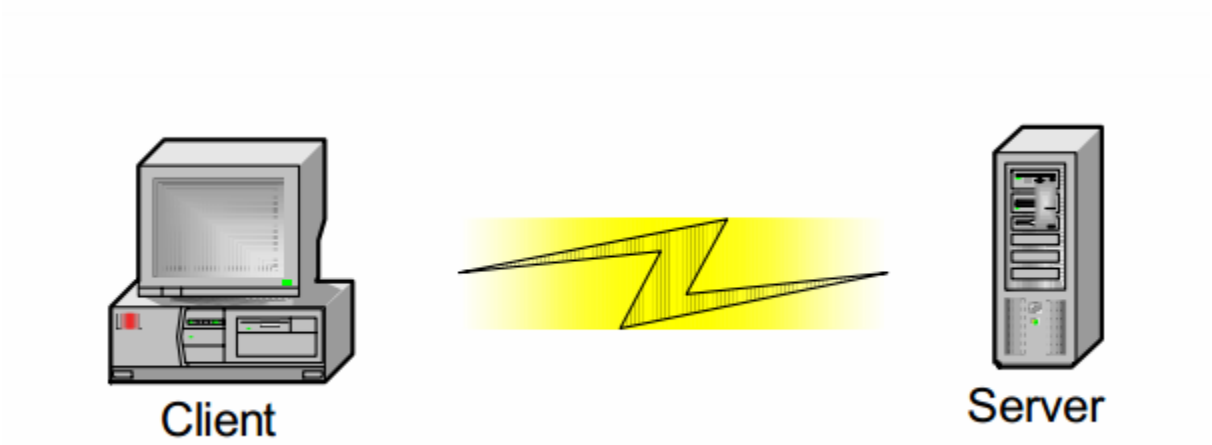
- Multiple levels of indirection have overheads

  Alternative: two-layer architecture



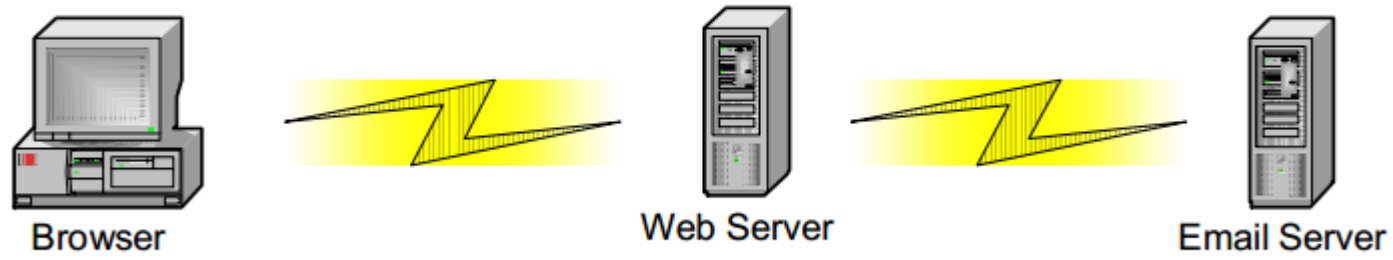
n-Tier Enterprise Applications

- N = 2
- client tier
  - presents data to user, gathers data from user
- server tier
  - hosts application logic, databases, and data services



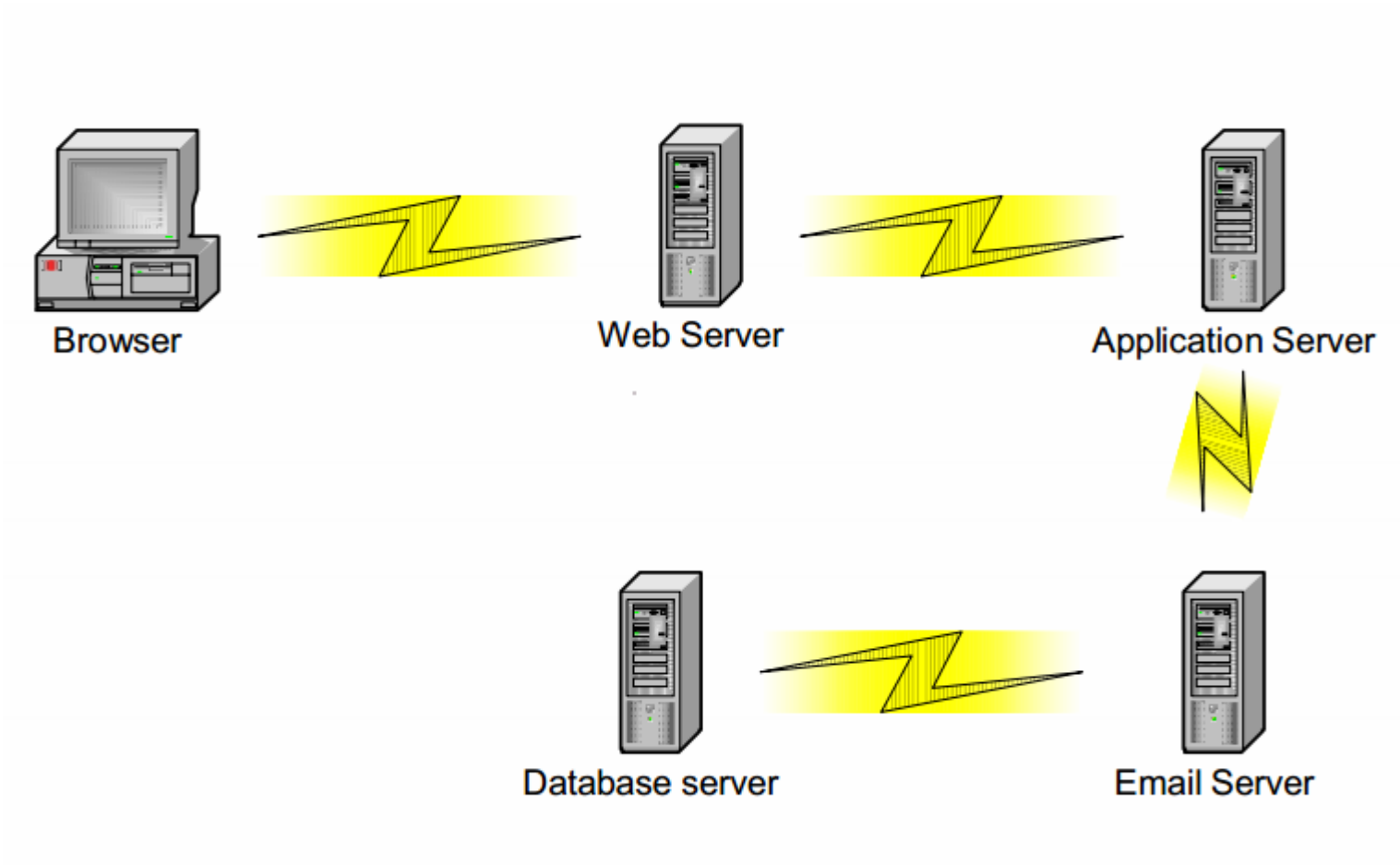
n-Tier Enterprise Applications

- N = 3
- client tier
  - presents data to user, gathers data from user
- Middle Tier
  - hosts application logic
- Back-End Tier
  - hosts databases and data services



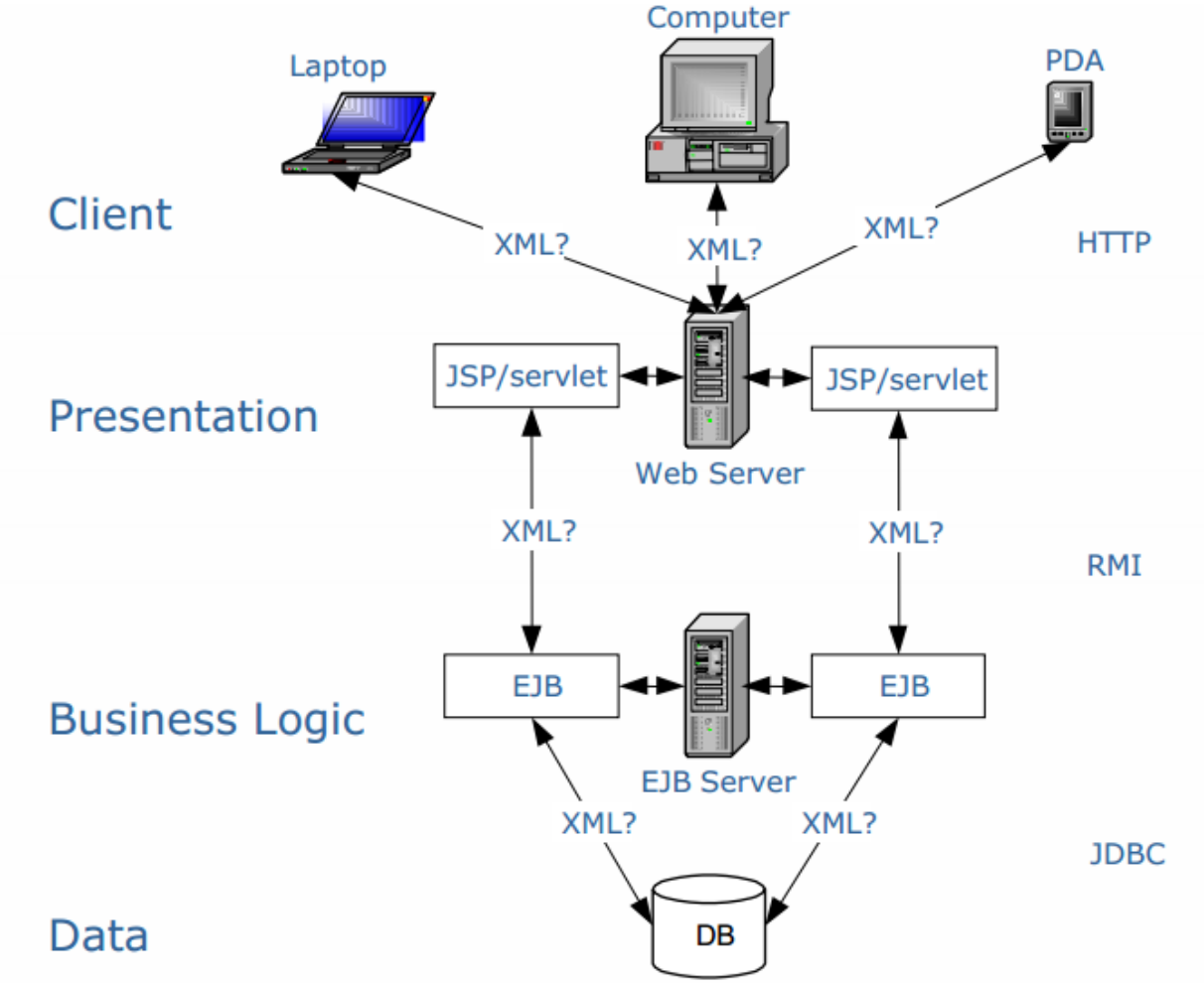
n-Tier Enterprise Applications

- N = 5



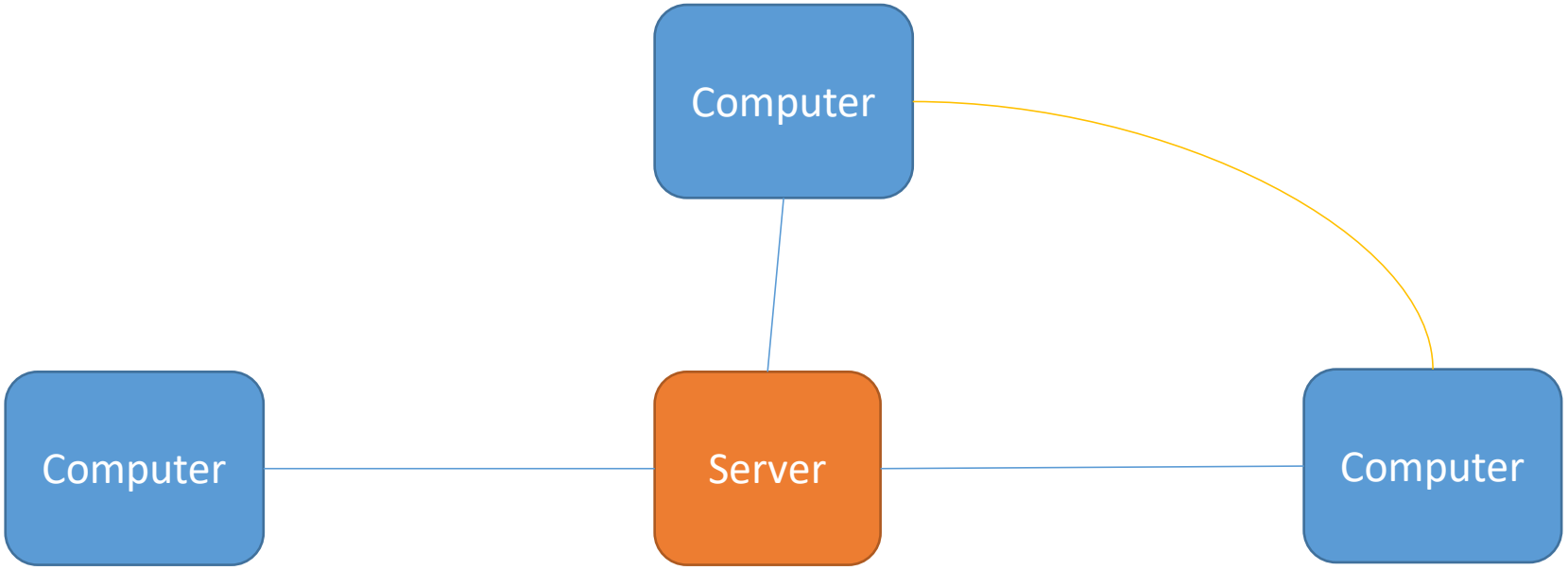
n-Tier Enterprise Applications

- Typical J2EE Architecture



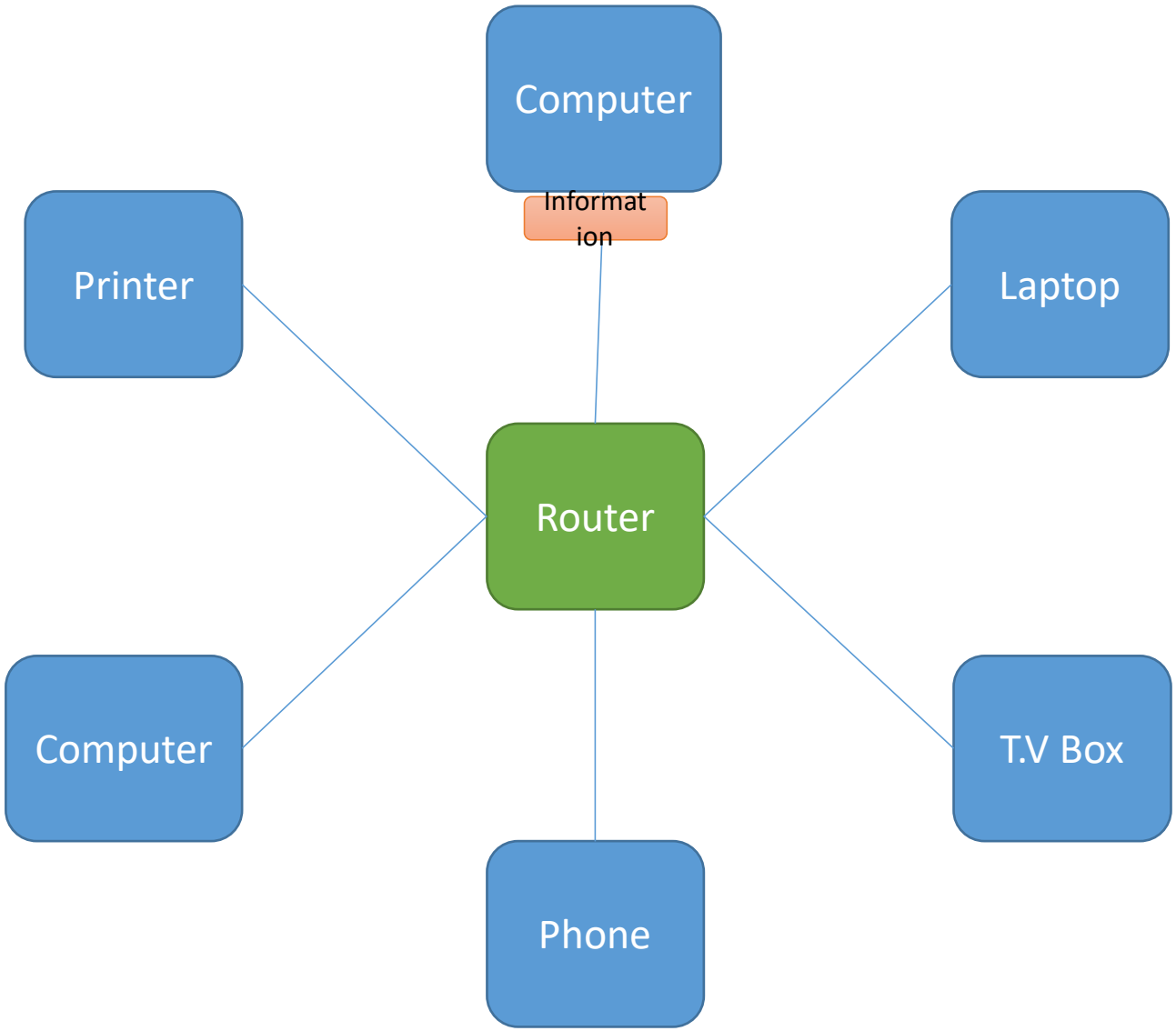
THE WHOLE PICTURE

The internet

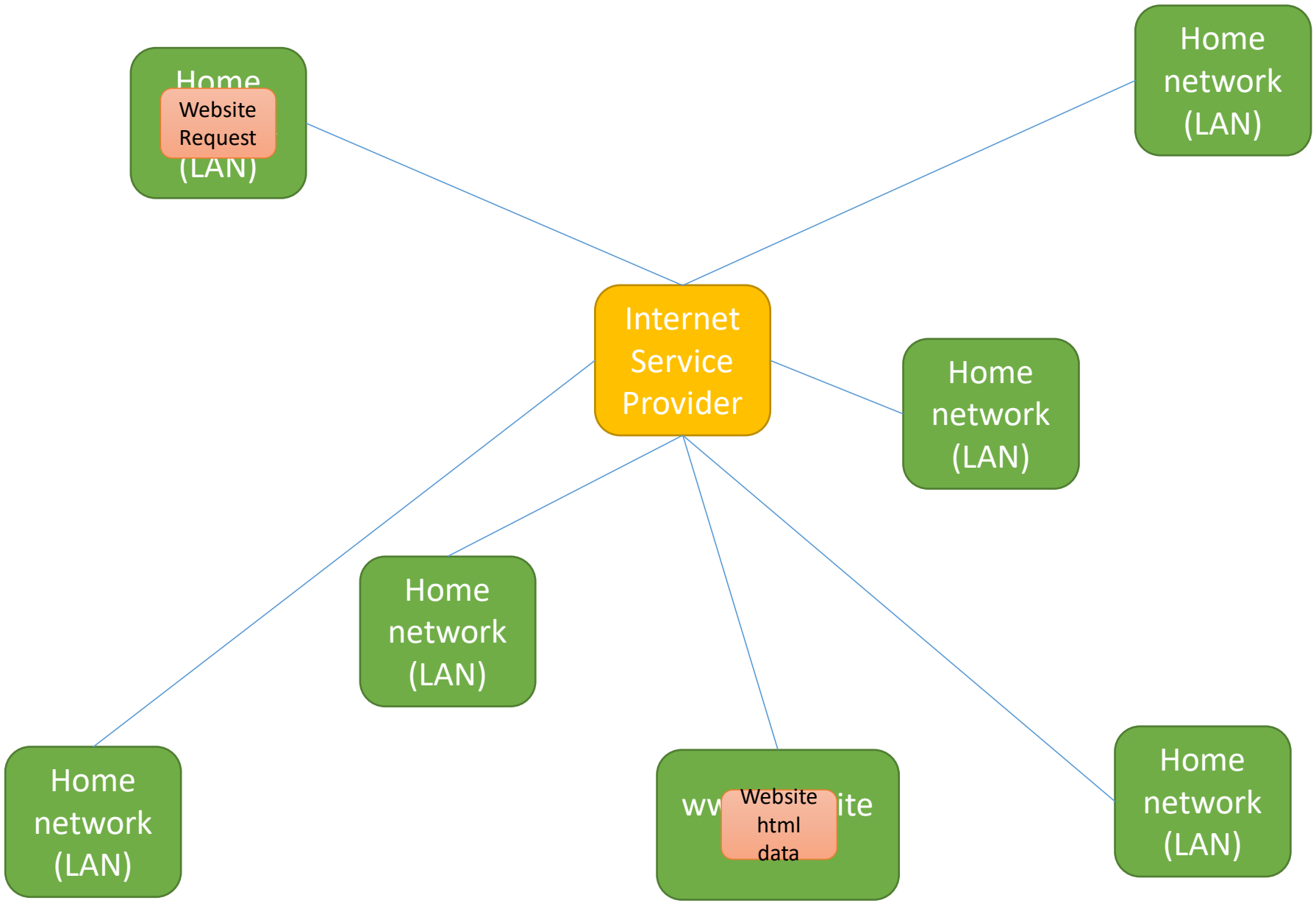


This is a computer
A network is a combination of computers and servers. They are all connected to allowing them to share information.
Most of the time, computers are connected to one another through a server.
But sometimes computers are directly connected to each other.

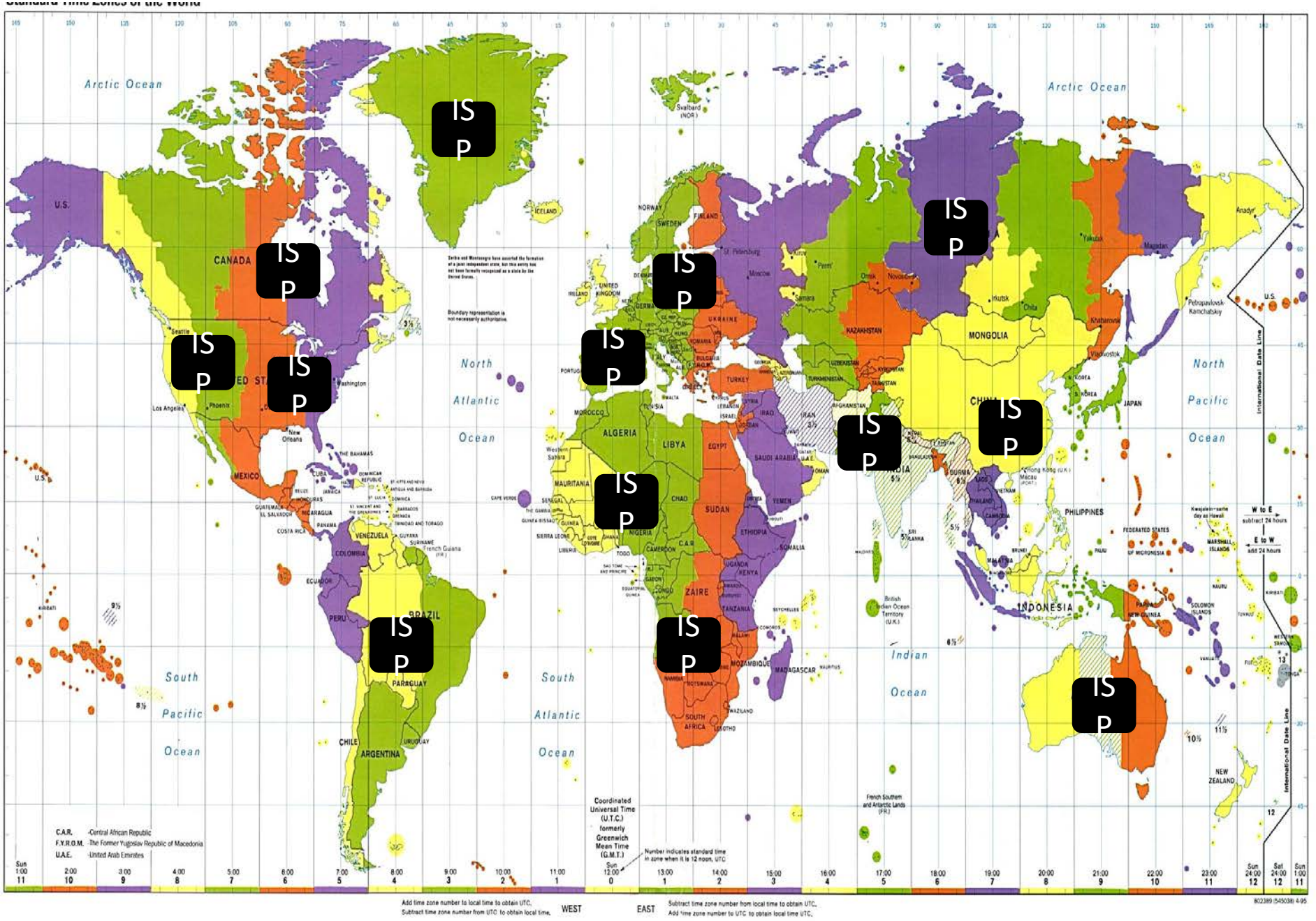
Your home


Computer
Information
Printer
Laptop
Router
Computer
T.V Box
Phone

Your ISP

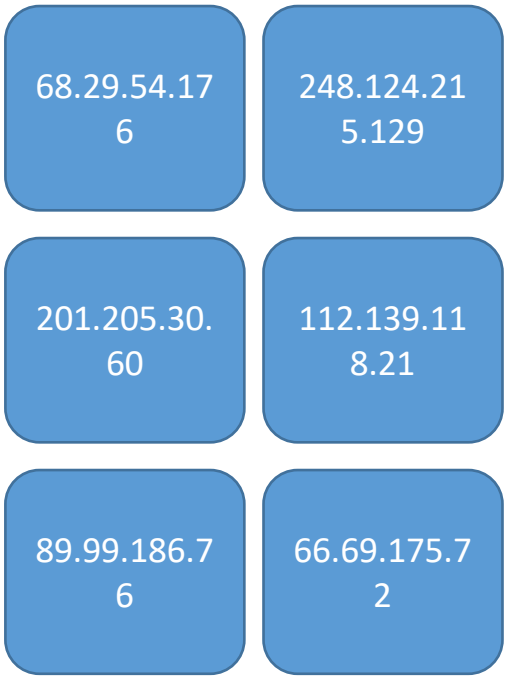

Home
Website Request
(LAN)
Home network (LAN)
Internet Service Provider
Home network (LAN)
Home network (LAN)
Home network (LAN)
ww
Website html data
ite
Home network (LAN)

The Internet.



IP Address

Bad!



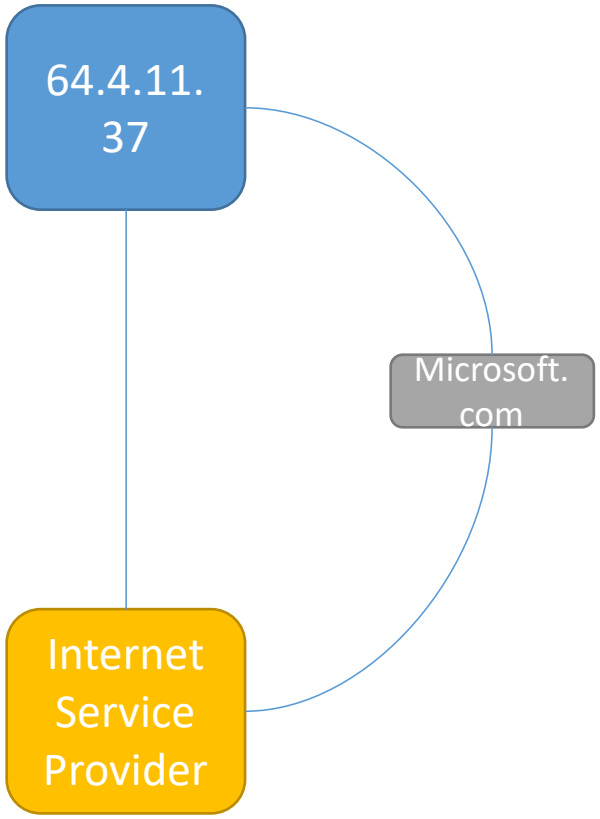
Very good!

All computers have names. Computers names aren't as unique as Alex, Jason, Susan, Jamie, etc.
Computers like numbers.

IPv4 – X.X.X.X uses 32-bit address scheme

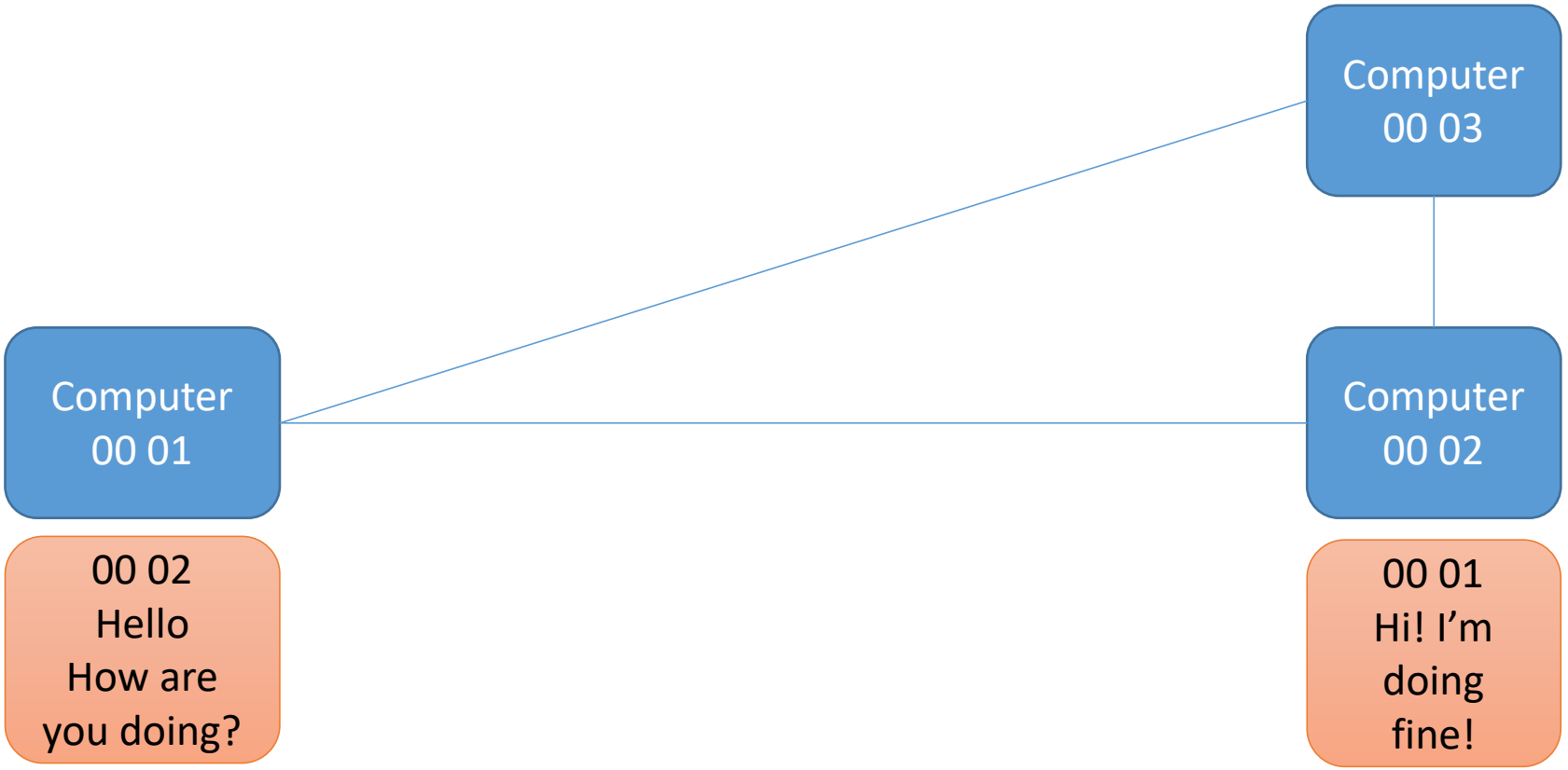
IPv6 – X:X:X:X:X:X uses 128-bit address scheme (uses Hex digits)

Websites?



http://64.4.11.37/

Information Sending – Packets – UDP – Sending Packets



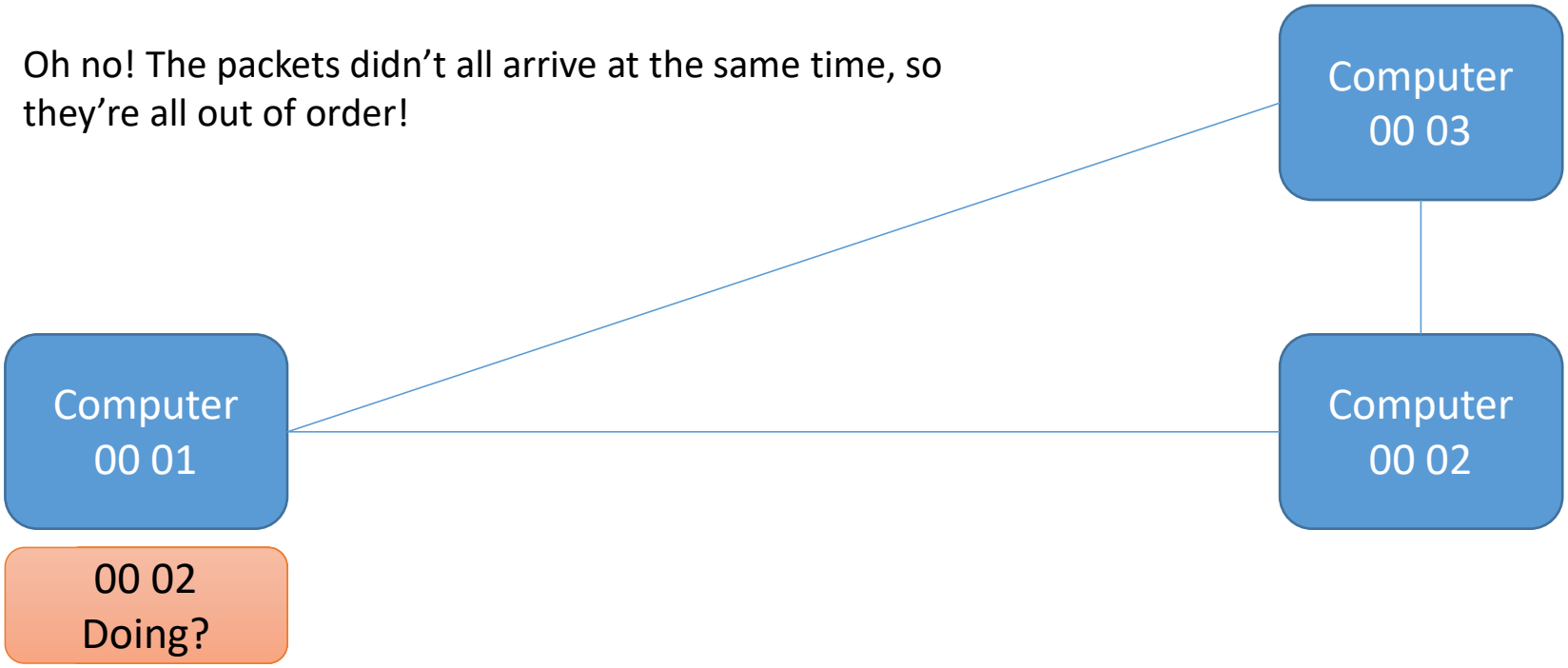
Basic Packet Protocol

Who are you sending the information to?

What is the information you want to send?

Information Sending – Packets – UDP – Out of Order!

Oh no! The packets didn't all arrive at the same time, so they're all out of order!

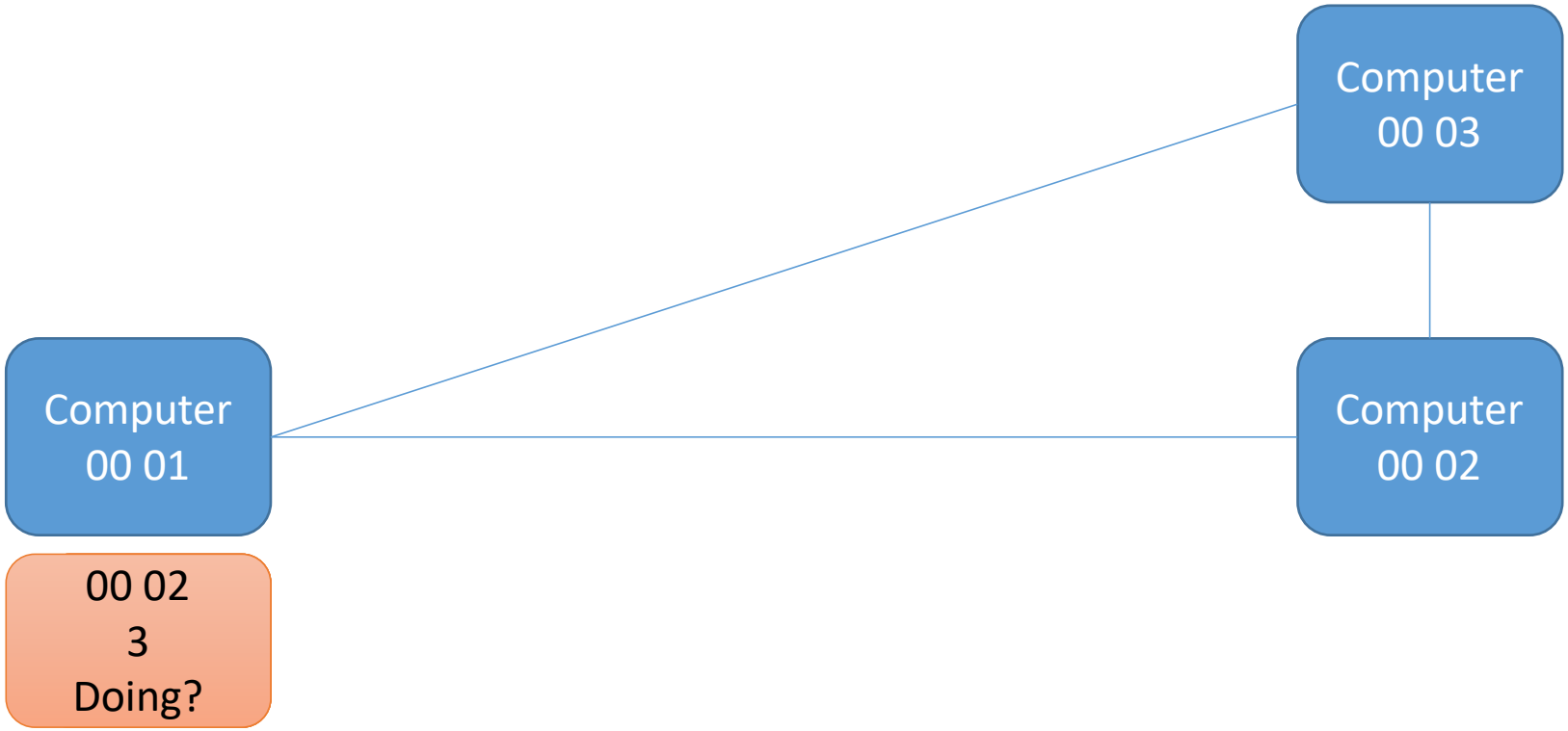


Basic Packet Protocol

Who are you sending the information to?

What is the information you want to send?

Information Sending – Packets – UDP – Reorder those packets.

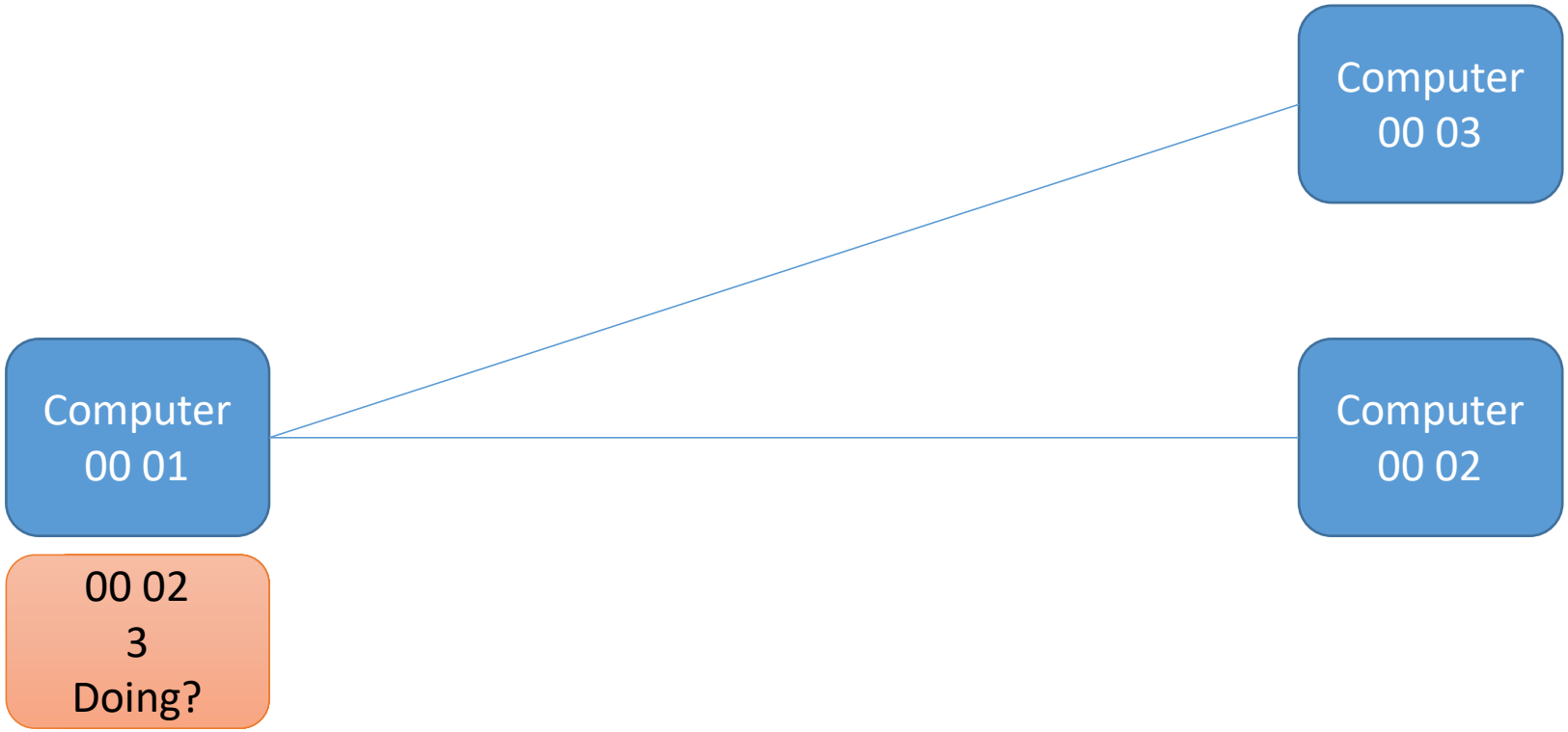


Basic Packet Protocol

Who are you sending the information to?

What is the information you want to send?

Information Sending – Packets – UDP – Broken Connections

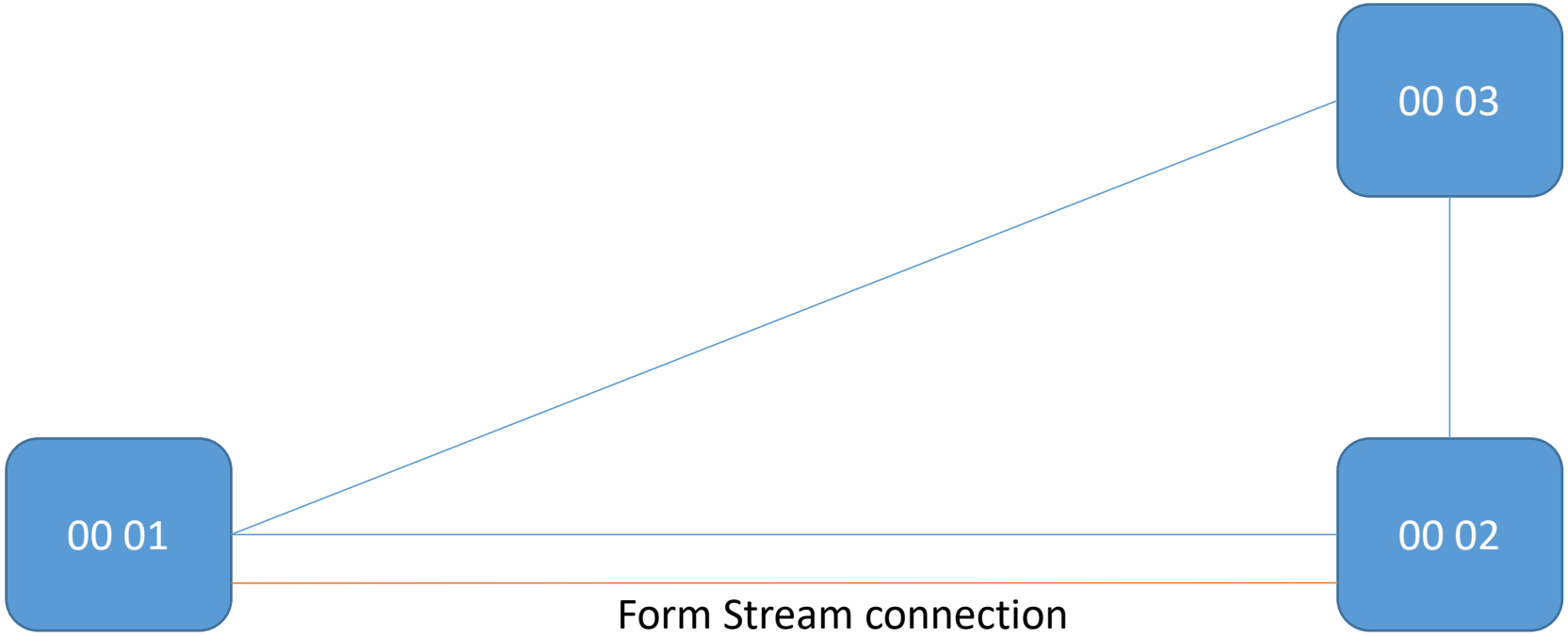


Basic Packet Protocol

Who are you sending the information to?

What is the information you want to send?

Information Sending – Streams – TCP – Sending Packets



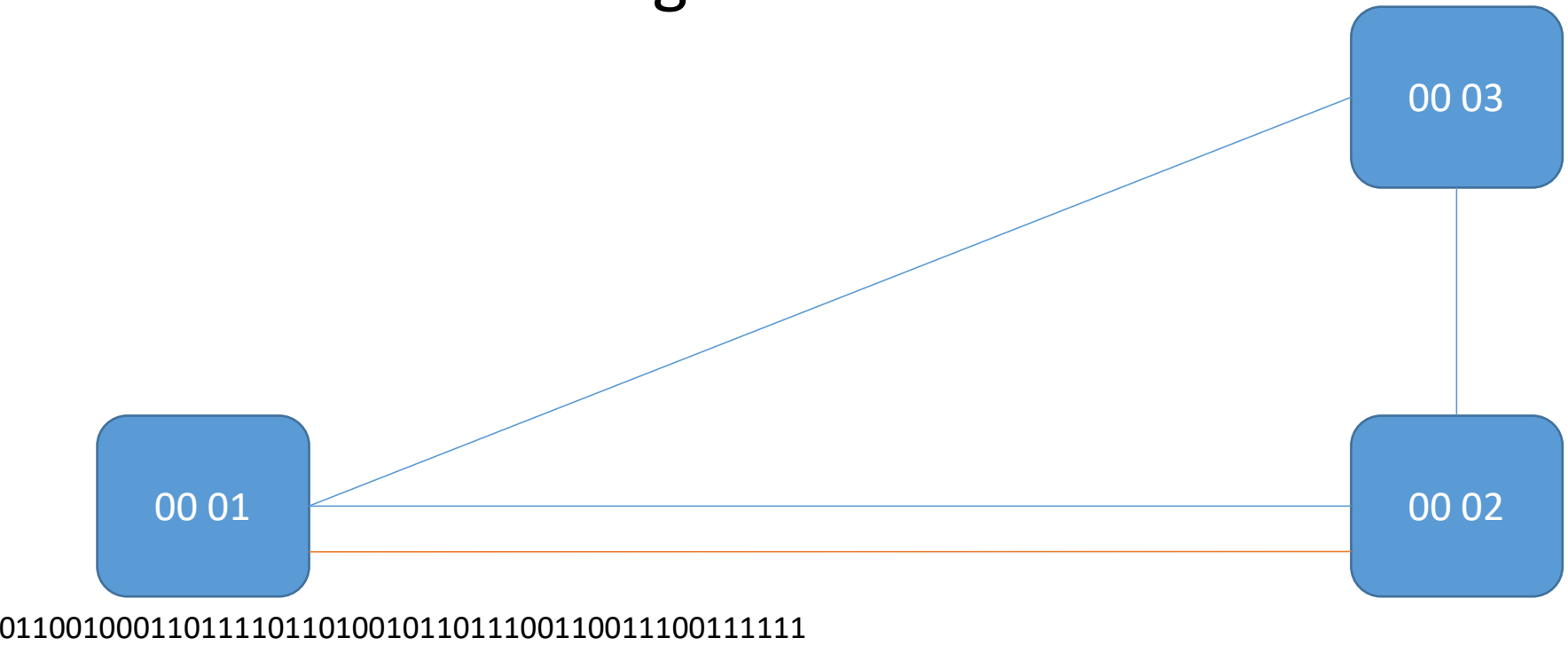
Hello how are you doing?

Stream protocol

Form connection with end computer

Send information.

Information Sending – Streams – TCP – Stream of Info



Stream protocol

Form connection with end computer

Send information.

Two-way communication

- TCP/IP Transmission Control Protocol
  - Reliable connection that uses the internet client/server model

1. Client opens connection to the server
2. Client sends a request to the server
3. Server sends a response to the client
4. Client closes the connection to the server

- (*steps 2 and 3 may be repeated*)

java.net package

- Low-level API
  - Networking concepts that uses **sockets** to establish connections (send and responses)
    - A socket is one end-point of the two-way connection *(client will have a socket and the server will have a socket)*
  - When you have multiple clients connecting to the same server, they'll use the same **port number**.
    - **Socket** class for the client socket
    - **ServerSocket** class for the server's socket
- Establishing a connection between client and server is a process that's classed **handshaking**
- High-level API

Java Network ASSIGNMENT

java.net.* package

Assignment #1

PROGRAMMING ASSIGNMENT #1

- Using java.net.* package implement the following:

1. a server class that is capable of handling clients requests as follows:
   A. clients send a messages to the server the server respond that It got the message (515OK)
   B. clients messages are request to a simple math calculation for example if a client sends 3*5 the server responds with 3*5=15. supported operations are (+,-,/,//,% and *)
   C. Server needs to keep count of number of clients connected to it.
   D. if client send "COUNT" message the server responds with the total number of clients connected to it.
2. a Client class that simulates clients and support the above functionality.