

Announcements

- **Handouts**
 - class syllabus
 - programming assignment #1 (includes computer account)
- **Enrollment**
 - there are 44 people in the class, and 23 on the wait list
 - due to the size projects enrollment will not be increased
 - priority to fill drops will be given to **senior CS undergrads**
 - this class will be offered again in the spring
- **Required Background**
 - must have 311 and 330 (412 or 430 would be helpful)
 - if you have not passed 311 & 330 you will be dropped
 - strong working knowledge of C or C++ (take your pick)

Announcements (cont.)

- **Required Work**

- will require about the same amount of effort as 412
 - 412 a (slightly) harder project to debug
 - 417 project is (by design) more ambiguous
- will need to write project proposals plus the code

- **Materials**

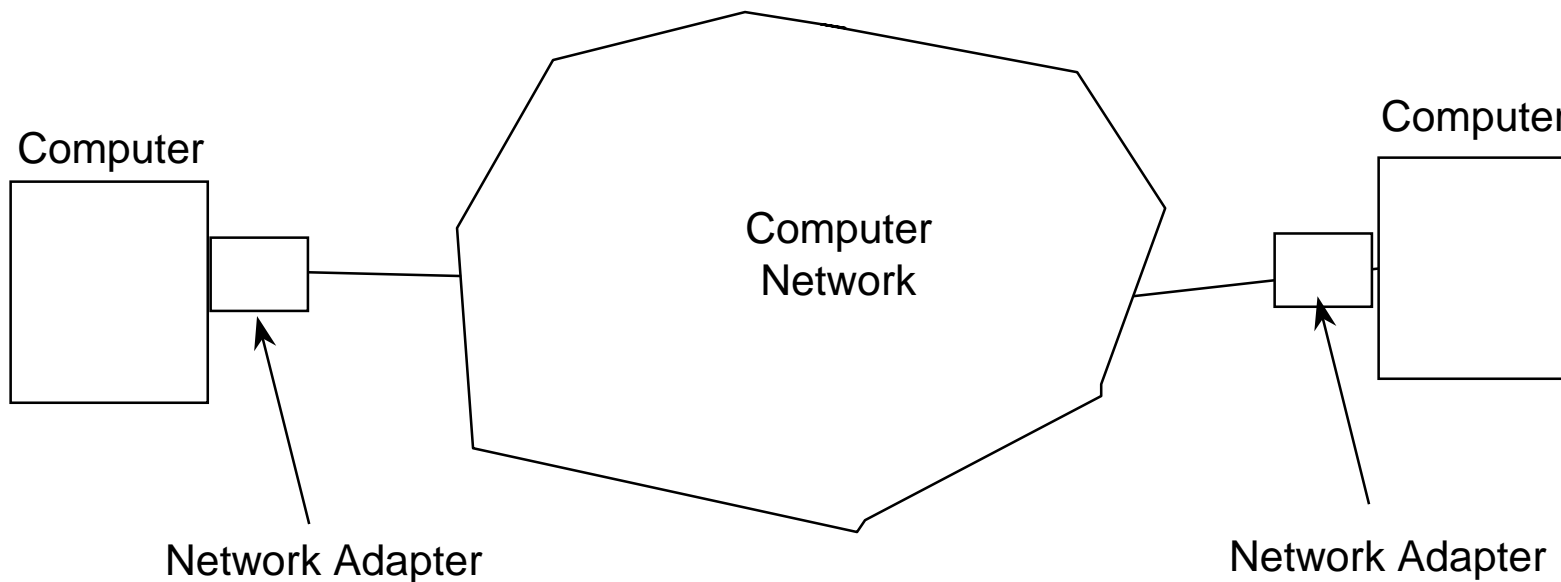
- Tanenbaum, “Computer Networks”, **3rd Edition**
- Nichols, Buttlar, and Farrell, “Pthreads Programming”
- Handouts from Web page

- **Reading (for this week)**

- Chapter 1

Networks

- Communication between semi-autonomous computers
- Attached to host system by an adapter



Many Types of Networks

- Physical Media

- copper wires (Ethernet, RS232-C, V.32, etc.)
- fiber optics (ATM, FDDI)
- air (IR, Radio, micro-wave)

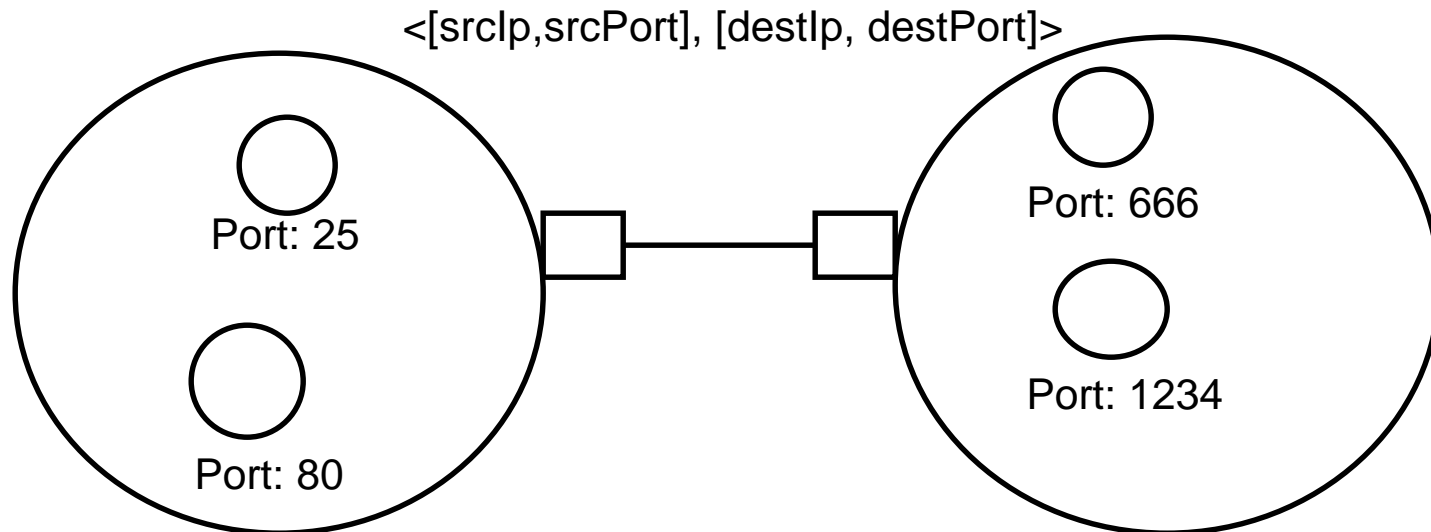
- Speeds (link not aggregate)

- low
 - modems (few k bits/sec)
 - pagers
- medium
 - Ethernet (10-1000 Mbps)
 - Token Ring (16 Mbps)
- high
 - ATM (155-655+ Mbps)
 - Myrinet (600-1200 Mbps)
 - SONET (OC-192 - 119424 Mbps)

Project #!

- Ports

- End-points for communication
- How to identify a processes rather than a machine



Debugging

learn to use the debugger (ladebug)

check that what you send is what you think you send

print data just before it is sent

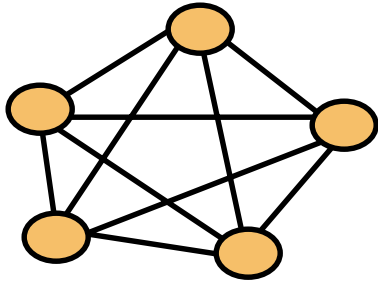
Project #1 Notes

- Small bug in the sample code (PDF handout)
 - on pg. 12, line 48: `memset((void *) &server, sizeof server)`
 - should have a second argument of zero
- Use of netstat
 - don't forget it is installed in `/usr/sbin/netstat`
 - the “-f inet” option is useful for restricting output to IP only
- might want to look at man page for `sendto/recvfrom`

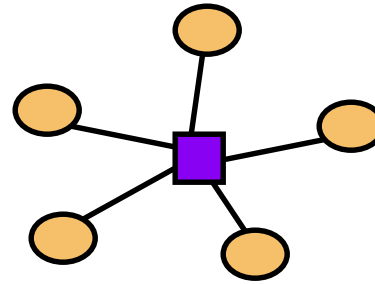
Network Topologies

- How are the communicating objects connected
- Fully connected - link between all sites
- Partially connected
 - links between subset of sites
 - can be an arbitrary graph
- Hierarchical networks
 - network topology looks like a tree
 - internal nodes route messages between different sub-trees
 - if an internal node fails, children can not communicate with each other
 - star network - hierarchical network with single internal node

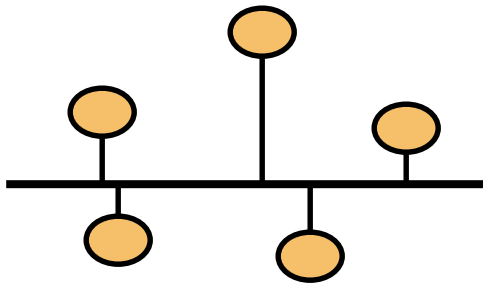
Many Ways to Connect Wires



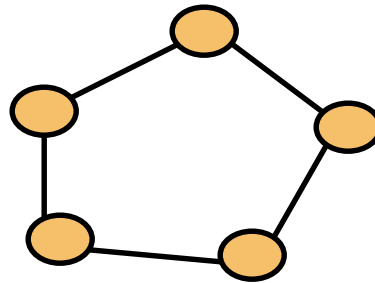
Fully Connected



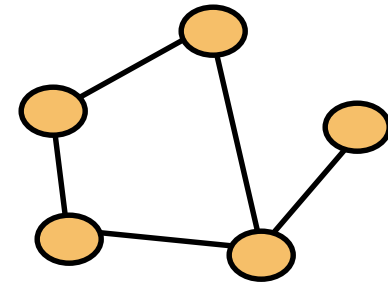
Star



Bus



Ring



Arbitrary

A Network is not an Island

- Reason for networks is to share information
 - must be able to communicate in a common language
 - called protocols
 - The nice thing about protocols is that there are so many of them!
- Protocols
 - must be unambiguous and followed exactly
 - rule of thumb for good protocol implementations
 - be rigorous in what you generate
 - be liberal in what you accept
 - there are many different aspects to protocols
 - electrical through web services

Layering

- Layers provide information hiding
 - doesn't matter what lower level layers use as long as higher layers speak the same protocol.

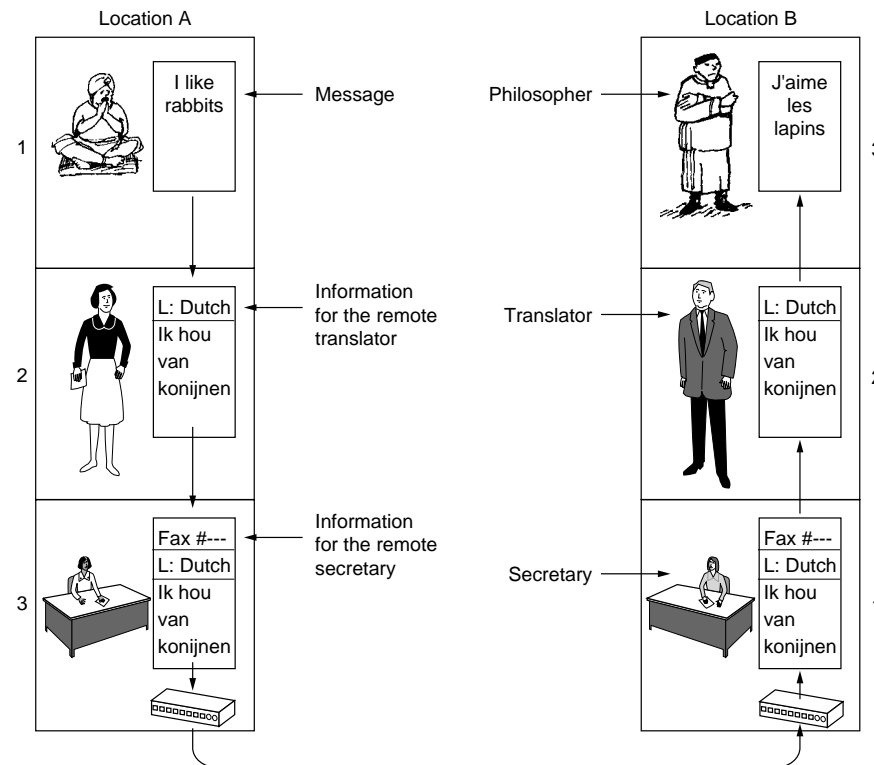


figure copyright , 1996, Andrew S. Tanenbaum