CS323 Operating Systems
Multi-Computers/Distributed Systems

Yuanyuan Zhou

Lecture 32

4/16/2003

Content of this lecture

- Reminder:
  - Midterm 4/21, no class in the morning
  - Conflict exam signup: 4/16 noon
    - Exam data: Wed 4/23 5-6pm, room TBA
- Network
  - Motivation
  - Network Categories
  - Packet Switching
  - Datagrams vs Virtual Circuits
  - Topology

Motivation

- Share: workstation, PC, Cray, database, radio telescope, work
- resource sharing
- computation speed up
- reliability
- communication

Network Category

- Resource Sharing Networks.
  - Communication is typically between a user process on one host and a resource manager process on another host.
  - Examples:
    - Access remote files
    - Transfer files between hosts
    - Database distributed among hosts
    - Access peripheral device (e.g., printer) on remote host
- Distributed Computation Networks.
  - A group of processes cooperating in one activity are distributed over several hosts throughout a network.
  - Examples:
    - Large database systems
    - Real time process-control systems

Network Categories Continued

- Remote Communication Networks.
  - Typically a batch system with most facilities in one or a few central locations, accessed from many remote locations.
  - Examples:
    - Bank ATMs

Requirements

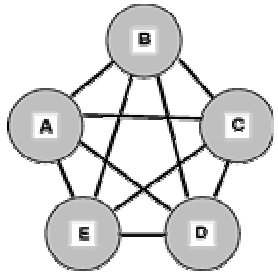
- Sharing of link costs
- Time versus space division multiplexing
- Reliability and Robustness
- Latency
- Scalability

Network Topology: criteria

- Basic cost
  - Expense.
- Communication cost
  - Latency.
- Reliability
  - What happens when a link or node fail.

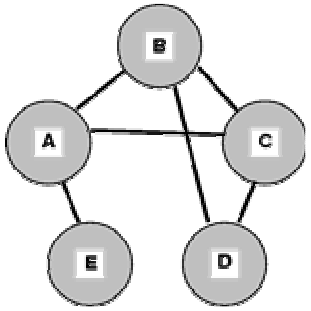
Fully Connected

- Basic cost:
  - high $\frac{n \times (n-1)}{2}$ links
- Communication cost:
  - low (single hop)
- Reliable
  - not likely to partition



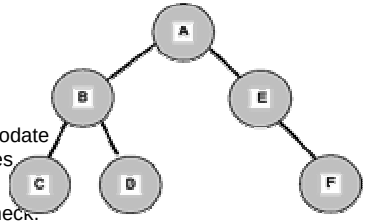
Partially Connected

- Basic cost:
  - Lower
- Communication cost:
  - Higher (multiple hops)
- Reliable
  - single failure may partition



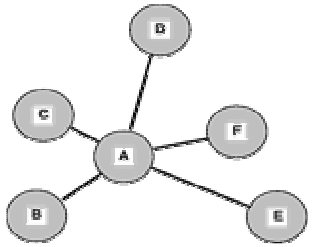
Hierarchical

- Basic cost:
  - Low, n-1 links
- Communication cost:
  - multiple hops
  - but organized to accommodate likely traffic flows (subtrees represent localities)
  - Root may become bottleneck.
- Reliable
  - single failure likely to partition



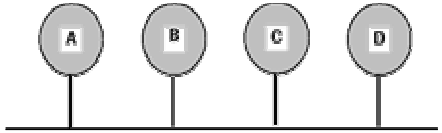
Star-Connected Network

- Basic cost:
  - ?
- Communication cost:
  - ?
- Reliable
  - ?



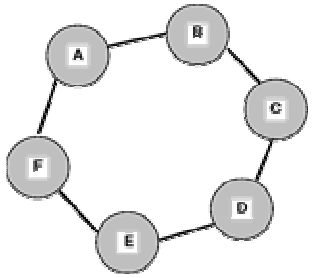
Bus-Connected Network

- Basic cost:
  - ?
- Communication cost:
  - ?
- Reliable
  - ?



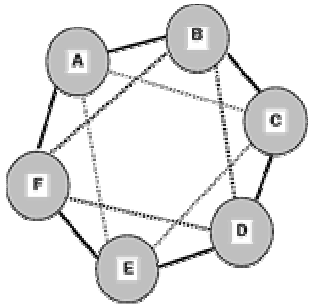
Ring-Connected Network

- Basic cost:
  - ?
- Communication cost:
  - ?
- Reliable
  - ?



Double Connected Network

- Basic cost:
  - ?
- Communication cost:
  - ?
- Reliable
  - ?



Grid Connected Network

- All nodes connected via a mesh
  - a two-dimensional design. It is highly regular and easy to scale up to large sizes.
- It has a diameter
  - the longest path between any two nodes
  - increases only as the square root of the number of nodes.
- A variant of grid is the double torus
  - a grid with the edges connected
  - more fault tolerant than grid and the diameter is also less.
- Further enhancement of the regular grid structure is cube and hypercube topology being used in parallel computers.

Packet Switching

- Messages are divided into *packets*, variable-length blocks of characters.
- Each packet is check-summed.
- Has a packet identification.
- Routing information.
- Origin and destination.

Packet Switching-- Advantages

- Advantages
  - Packet switching allows many users to share the same communication line.
  - Data communications are bursty in nature. Packets allow bandwidth to be shared for bursty traffic.
  - Small and large amounts of information are easy to transmit.
  - Latency is delay to send packet, not to establish communication circuit.
- Example Services:
  - 1967 ARPAnet
  - 1975 Telenet

Datagrams vs Virtual Circuits

- **Virtual Circuit:** The network provides a channel which the user can assume is perfect: transmission will be completely error free, and packets will be received in the order sent. If a packet arrives out of order, it is held until its predecessors arrive.
  - Virtual circuit follows the pattern of circuit switching scheme.
  - Does more of the work of transmission; less burden on hosts.
  - In some circumstances, order is less important than speed (order may even be unimportant in some applications).
- **Datagram:** Packets are transported to their destinations as isolated units (no ordering). A message distributed over several packets must be assembled by the host. The network does no error checking; that is done by the hosts.
  - Datagrams use store-and forward packet switching scheme.
  - Messages may arrive at their destination faster, since they will not be held up if out of order.
  - Requires more work from the host computers.

Network Hardware for Single Machine

- Network Interface Card
  - includes some RAM to hold incoming and outgoing packets
- Before a packet is transmitted to a network switch, the packet must be copied from the main RAM to interface RAM.
  - The reason is that interconnection networks are synchronous, so that once the packet transmission has started, the bits must continue flowing at constant rate. The same applies to incoming packets.
- Network Interface cards also have one or more DMA channels or even a complete CPU.
  - The DMA channels can copy packets between the interface board and the main RAM at high speeds.
  - If network board has a CPU, the main CPU can offload some work to the network board, such as handling reliable transmission, multicasting, and protection.

Network Hardware for Distributed Systems

- Local-Area Network (LAN)
  - Low latency for short messages
  - High bandwidth for long messages
  - Bus, Crossbar, ring, or star
  - 10, 100 Mbps(Ethernet), 100 Mbps (FDDI), 155 622 Mbps (ATM), 622Mbps 1 Gbps (HiPPI)
  - Client server configurations
  - Broadcast easy
  - Connects hosts, bridges, routers
- Metropolitan-Area Network (MAN)
  - 100 Mbps FDDI, 155, 622 Mbps (ATM)
  - Low latency, high bandwidth, scalable
- Wide-Area Network (WAN)
  - Point to point (T1 (1.2Mbps), T3 (45Mbps), packet switched (ATM (155Mbps-10Gbps))
  - Broadcast may use repeated message sends
  - High latency (speed of light and buffer sizes), medium/high-bandwidth.
  - Connects Routers

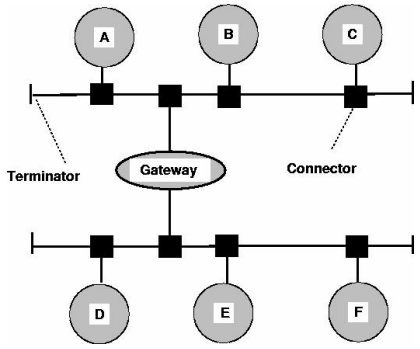
Specialized Network Nodes

- Gateways
  - Route between one kind of network and another or between networks with different administration. Example, interconnecting IP networks over T1 trunks and Ethernets or entry way to UIUC.
- Routers
  - Switch packets between subnets.
- Bridges
  - Overcome difficulties of joining Ethernets/physical networks together.
- Intelligent Hubs
  - On a star Ethernet, filters out traffic not destined for host down link.
- Passive Hubs
  - On a star Ethernet, broadcasts every packet down each link.
- Firewall
  - Filter on link that removes/transfers selected packets based on IP numbers, applications, or port addresses.

Requirements of Types of Networks

- Wide Area Networks
  - Widely dispersed sites.
  - High latency: milli-secs apart
  - Communication processors
  - Links like telephone, radio, fiber-- reliability.
- Local Area Networks
  - On one site or location.
  - Low latency: micro-secs apart
  - Ethernet, FDDI, over broadband coax, twisted pair, fiber
  - Links like telephone, radio, fiber.
  - Usually, each LAN is limited in size by physics.
  - Large networks build by piecing LANS together with gateways.

Gateway



Low-Level Communication Software

- One big problem for high-performance communication in networked systems is excess copying of packets.
- If copies to and from RAM dominate the performance, the extra copies to and from the kernel may double the end-to-end delay and cut the bandwidth in half.
- To avoid this performance hit, some multicomputers map the interface board directly into the user space (no involvement of kernel), and allow the user process to put packets directly onto the board. However this approach has two problems:
  - What if several processes are running on the node and need network access to send packets? Which one gets the interface board in its address space?
  - Solution: Map the interface board into all processes that need it, but avoid race conditions, which needs synchronization mechanisms. Difficult!!! Hence, use this approach only if only one process needs the network board at the node.
  - Kernel might need access to the network itself (e.g., to access file system). Having the kernel share the interface board with user is not a good idea. Solution: have two network interface boards, one mapped to user space for application traffic, one mapped to kernel space for OS traffic only.

User-Level Communication Software

- Message passing paradigm is used : **send and receive methods**
  - *send(dest, &mptr)* - sends message, pointed to by *mptr*, to destination *dest*. The sender blocks until the mesage is sent.
  - *receive(addr, &mptr)* - receives message, pointed to by *mptr*. The receiver listens at the address *addr* (consisting of CPU number and a process or port number) for incoming messages. The receiver blocks until the message arrives.
- **Two types of calls**
  - **blocking**, also called synchronous calls. It means that while the message is being sent, the sending process is blocked (ie suspended). Similarly, a call to receive does not return control until a message has actually been received and put in the message buffer pointed by the parameter.
  - **nonblocking calls**, also called asynchronous calls. It means if send is nonblocking, it returns control to the caller immediately, before the message is sent.
- There are couple of choices on the sending side:
  - Blocking send (CPU idle during message transmission)
  - Nonblocking send with copy (CPU time wasted for the extra copy)
  - Nonblocking send with interrupt (programming difficult)
  - Copy on write (extra copy probable needed eventually)