Distributed Systems — Last Time

- Communication
- Performance
- Reliability
- Security
- Programming models
- Transparency

$\Longrightarrow$ Processes in a distributed system all communicate via a message exchange.

Networks

Networks are usually concerned with providing efficient, correct, and robust message passing between two separate nodes.

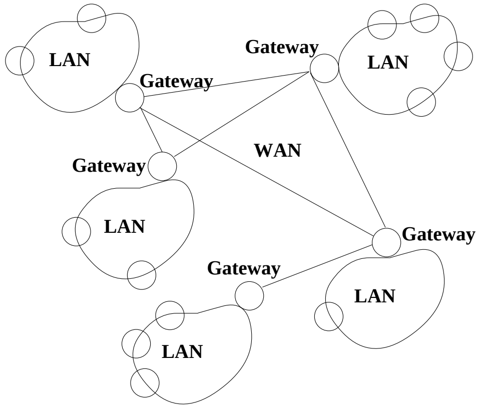
A Local Area Network (LAN) usually connects nodes in a single building and needs to be fast and reliable (for example, Ethernet).

Media: twisted-pair, coaxial cable, fiber optics

Typical bandwidth: roughly 1 order of magnitude slower than disk

A Wide Area Network (WAN) connects nodes across the state, country, or planet. WANs are typically slower and less reliable than LAN (for example, Internet).

Media: telephone lines (T1 service), microwave links, satellite channels



Network Costs

- Overhead - CPU time to put a packet on the wire.
- Latency - time for one byte to go from one place to another.
- Throughput - maximum byets per second

	Throughput (Mbits/sec)	Overhead (ms)	100 bytes (ms)	4KB (ms)	Remote 4KB Read (ms)
TCP/IP Wireless	10	0.5-1	.5 +.08	.5 +3.0	4.00
TCP/IP Ethernet	100	0.5-1	.5 +.008	.5 +.3	1.30
TCP/IP Gigabet Ethernet	1000	0.5-1	.5 +.0008	.5 +.03	1.03
AM/ Myrinet	1200	0.007	0.007 +.001	0.007 +.03	0.04

Moral: bandwidth is not the whole story

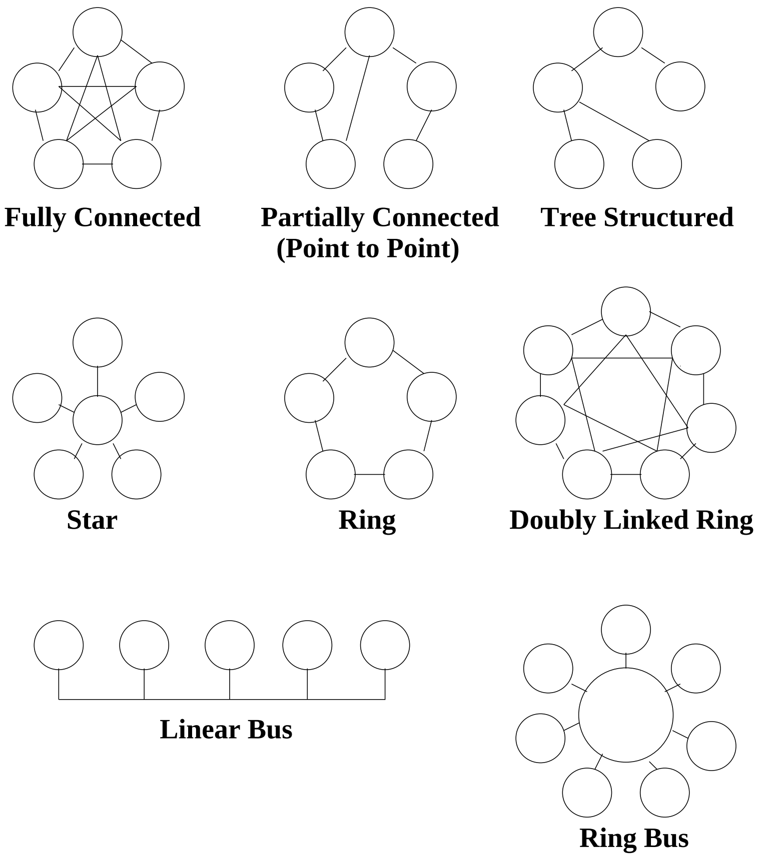
Latencies across the country:

- 3000 miles $*$ 5000 ft/mile $=$ 15ms
- A remote 4KB read is dominated by network latency

Key to good performance

- LAN – minimize overhead
- WAN – keep pipeline full

Network Topologies



Principles of Network Communication

- Data sent into the network is chopped into “packets”, the network's basic transmission unit.
- Packets are sent through the network.
- Computers at the switching points control the packet flow.
- **Analogy:** cars/road/police - packets/network/computer
- Shared resources can lead to contention (traffic jams).
- **Analogy:**

  Shared node - Downtown Austin

  Shared link - I35 and Mopac

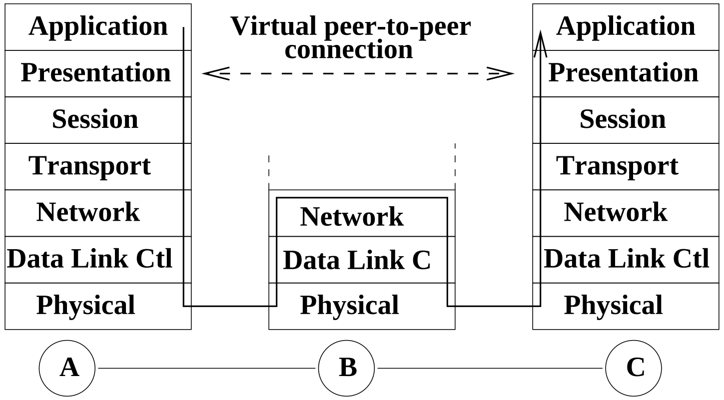
Sending messages

- **Protocols** to encode information
- **Naming** the machine to communicate with
- **Routing** information from a source to a destination

Communication Protocols

International Standards Organization/ Open Systems Interconnect (ISO/OSI)

In this example, Node A is communicating with Node C. Node B is on the path from A to C.



Each layer N, provides a service to layer N+1, by using its own layer N procedures and the interface to the N-1 layer.

Communication in Distributed Systems

Application layer: applications that use the net, e.g., mail, netscape, X-services, ftp, telnet, provide a UI

Presentation layer: data format conversion, e.g., big/little endian integer format)

Session layer: implements the communication strategy, such as RPC. Provided by libraries.

Transport layer: reliable end-to-end communication between any set of nodes. Provided by OS.

Network layer: routing and congestion control. Usually implemented in OS.

Data Link Control layer: reliable point-to-point communication of packets over an unreliable channel. Sometimes implemented in hardware, sometimes in software (PPP).

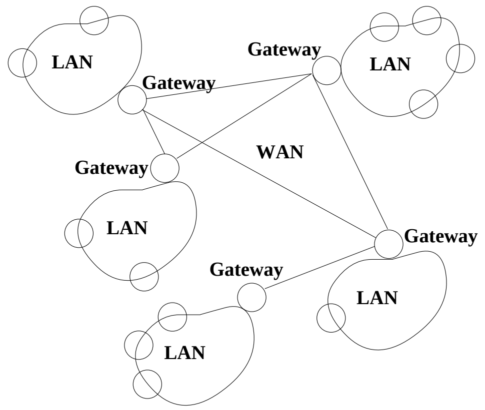
Physical layer: electrical/optical signaling across a "wire". Deals with timing issues. Implemented in hardware.

- ISO/OSI is a common model for networks to follow, but, in fact, a straightforward implementation would be inefficient.
- TCP/IP combines a bunch of layers

Network Communication

- **Protocols** to encode information
- **Naming** the machine to communicate with
- **Routing** information from a source to a destination

Application sends a message to a name, routing software turns name into an Internet address and sends the message on its way.



Naming

1. Every host node has a unique string name, and a 32-bit Internet number called its host-id.
2. Host-ids are numbers (128.119.42.5); one for each level of an address (charlotte.cs.utexas.edu). Symbolic names have most significant field first; numeric names have least significant field first.
3. The OS needs to map logical addresses: (*machine name, application*) e.g., (charlotte.cs.utexas.edu, telnet) to the host-id and port for the application.
4. Each machine has a set of local names, and the "Gateway" to which it sends messages directly.
5. The LAN node has a forwards the message to the "Gateway" node which stores the routing tables containing a cache of Internet addresses.
6. The Gateway node interfaces the LAN to the WAN.
7. The Gateway caches addresses
8. Caching Internet addresses and aging them eliminates much of this overhead.
9. On Unix, /etc/hosts contains a partial DNS table. For example, /etc/hosts on charlotte has a long list of hosts (720 of them) in the cs.utexas.edu domain. These names can all be resolved locally.

Routing

The OS *router* software decides how to move a message from A to B and typically uses tables of possible routes that includes response time information.

Fixed routing: The path from A to B is set in advance and does not change unless there is a hardware failure. To minimize communication costs, the shortest path on the least expensive media is chosen.

Advantages: Set up cost paid only once per node pair, messages arrive in order.

Disadvantages: Cannot react to changing load.

Virtual circuits: The path from A to B is session dependent. When a session starts (e.g., an ftp connection), the router sets up a path and uses it for the duration of the session.

Advantages: Set up cost paid once per session, messages arrive in order, reacts to changing load between sessions.

Disadvantages: Cannot react dynamically to load.

Dynamic routing: For each message from A to B, the router selects a path.

Advantages: Can react to changes in network load.

Disadvantages: Every message has set up overhead; out-of-order message arrival due to different paths.

Routing

Mixed routing strategies are common. Many UNIX systems use static routing from all nodes in a LAN to the gateway node, and dynamic routing from gateway to WAN.

Connection Strategies: Connecting two communicating processes over the network:

Circuit switching:

- Router establishes a permanent physical link between communicating processes that no one else can use (a fixed or virtual route).
- Example: telephone network uses circuit switching.
- Wastes network bandwidth.

Message switching:

- Router establishes a temporary physical link between the communicating processes for each message (a fixed or virtual route). All packets use this route.
- Analogy: airplane travel. Increases set up time, but keeps messages in order.

Packet switching:

- Router sends each packet separately (a fixed, virtual, or dynamic route).
- Most flexible, but messages can arrive out of order and must be reassembled in order. Most common switching technology for data networks.

Contention

How to tell if you can use the network or not.

CSMA/CD: Ethernet protocol.

- Node listens first. If no messages are on the bus, it sends.
- While it is sending, it listens for *collisions* with other messages.
- If a collision occurs, it wait a random amount of time and tries again.
- For what type of workload does this work well?

Token passing: IBM HP/Apollo use this protocol.

- Node can only send a message if you have the token.
- If you aren't sending, you must forward the token to someone else.
- If the token gets lost, you must regenerate it.
- For what type of workload does this work well?

Message slots:

- A number of fixed-length message slots continuously circulate in the system (usually a ring).
- Each slot holds a fixed-size message.
- Nodes wait for an empty slot to arrive and then inserts its message into the slot.
- Node inspects each slot as it goes by, taking out its messages.

Summary

Physical reality: packets	Abstraction: messages
limited size	arbitrary size
unordered (sometimes)	ordered
unreliable	reliable
machine to machine	process to process (routing)
asynchronous	synchronous
insecure	secure

- Unless otherwise stated, assume a reliable end-to-end message delivery.
- Virtually all computer systems contain distributed components
- Networks hook them together
- Networks make tradeoffs between speed, reliability, and expense