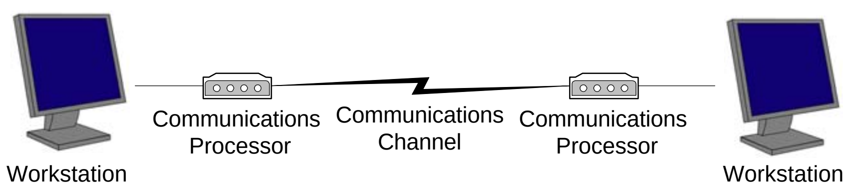
Outline: Connecting Many Computers

- Last lecture:
  - sending data between two computers
- This lecture:
  - link-level network protocols (from last lecture)
  - sending data among many computers

1

Review: A simple point-to-point network



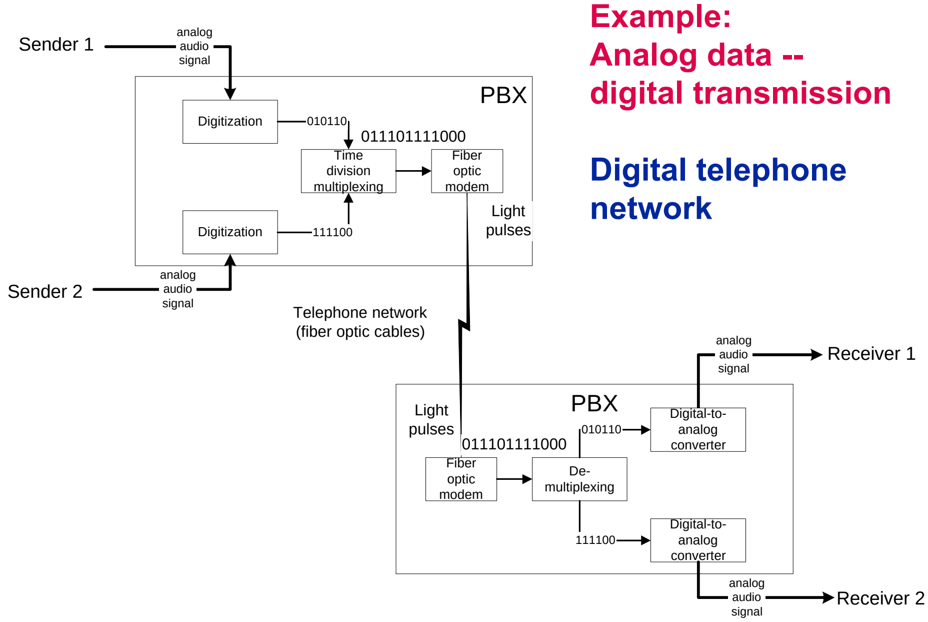
- processors convert data into signals
  - Concepts: digital vs. analog data, digital vs. analog transmission, modems, codecs
- signals are transported through channels
  - Concepts: bandwidth, data rate, multiplexing
- channels utilize one or more connection media

2

Chrysanthos Dellarocas.

Example:
Analog data -- digital transmission

Digital telephone network



3

Connection Media

- Twisted pair copper cables
  - 300BPS - 10MBPS
  - Why twisted?
- Coaxial cable
  - Higher bandwidth, up to 200MBPS
  - More expensive and difficult to wire
- Fiber optic cables
  - 10GBPS and higher data rates possible

4

Fiber Optic Cables

Connection Media (Cont'd)

- Wireless
  - Broadcast signals through the air
    - up to 100MBPS (~ like coaxial cable)
    - Scarce resource; use regulated by FCC
    - Can easily add another wire, but can't easily add more wireless capacity

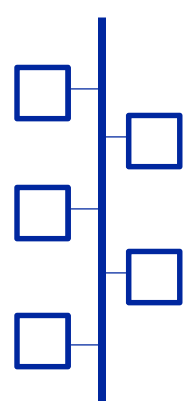
Connecting many computers

- Two basic approaches:
  - use a shared broadcast connection medium
  - use many point-to-point connection

7

Idea 1: Use broadcast medium

- All computers are connected to the same connection medium
  - e.g. a coaxial cable
- Date sent by each computer can be “heard” by all computers in the network
  - How does each computer select only relevant data?
- All computers compete for access to the common medium
  - A little bit like computer buses



8

Chrysanthos Dellarocas.

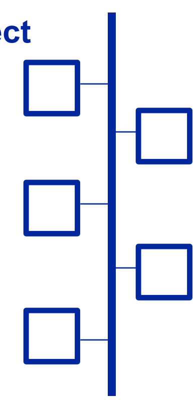
Sharing Connection Media

- Two general techniques for distributed control of shared resources
  - Prevent conflicts (locks, reservations, scheduling)
  - Recover from conflicts
- Here the shared resource is a communications channel
  - Several computers transmitting on single wire
  - Or broadcasting in the same location

9

Method 1: Systematic Conflict Recovery

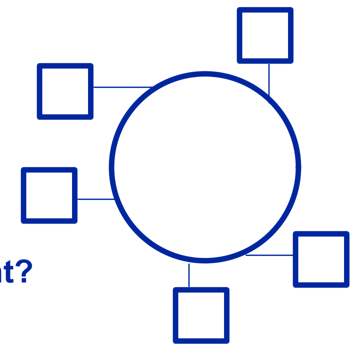
- Example: CSMA/CD
- When multiple devices share the same medium (wire)
- Carrier Sense Multiple Access/ Collision Detect
  - If medium is idle, transmit
  - If medium is busy, listen until idle, then transmit
  - If a collision is detected
    - Send jamming signal to make sure all stations know
    - Cease transmission
    - Wait a random amount of time, then try again
- What happens if load is heavy? light?
- What happens if one node fails?
- Used in Ethernet LANs



10

Method 2: Temporary Reservations

- Example: Token Ring
- "Token" is a data signal in a particular format
- When no node is transmitting, token just circles the ring
- To transmit
  - Wait for token
  - Remove token
  - Send data packet
  - When data makes it back around, put new token on ring
- What happens if load is heavy? light?
- What happens if one node fails?
- Used in IBM LANs, FDDI



11

Broadcast Network Technologies

- Ethernet
  - cheaper and simpler to install
  - unpredictable delays when load is heavy
- Token Ring
  - more expensive and complex
  - bounded delays even under heavy load

12

Limitations of Broadcast Networks

- OK for small numbers of computers
- Cannot handle many computers
  - too many conflicts
- Difficult to implement for wide-area networks
  - only wireless networks would work over long distances

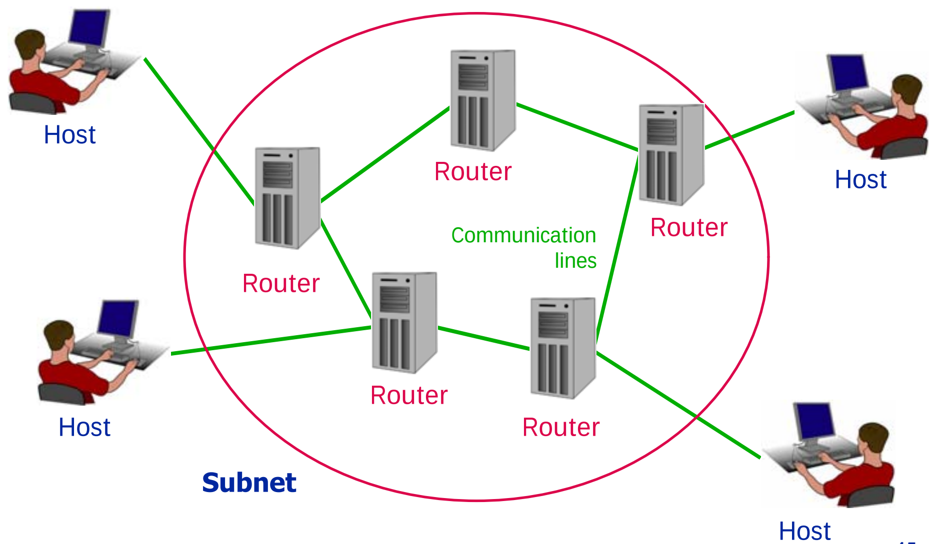
13

Idea 2: Use many point-to-point connections

- New concepts:
  - Routing
    - Which path do I follow to get from S to R?
  - Switching
    - How is bandwidth allocated for the transmission of a message?

14

Basic WAN structure



15

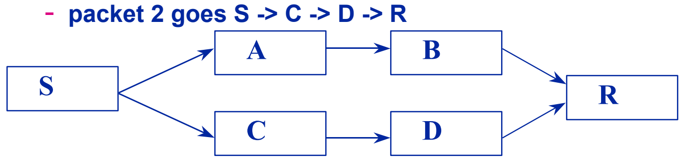
Switching

- Circuit switching
  - Set up dedicated end-to-end channel for duration of connection
  - Used for phone network
- Message switching
  - For sending data messages (e.g., email)
    - Each intermediate node stores and forwards the message
  - No wasted channels as with circuit switching
- Packet switching
  - Divide data messages into small packets
  - Each packet is "message switched"
    - Packets can take different routes
    - If one is lost, don't resend whole message

16

Why Packets?

- If part of message is lost or garbled, resend only the affected packet(s)
- Speed
  - Store-and-forward delay is minimized (pipelining)
    - Each intermediate node has to receive and store a message, then forward it
    - A can send packet 1 to B while receiving packet 2 from S.
    - Not possible if whole message sent at once
  - Packets can take different routes (parallelism)
    - packet 1 goes S -> A -> B -> R
    - packet 2 goes S -> C -> D -> R

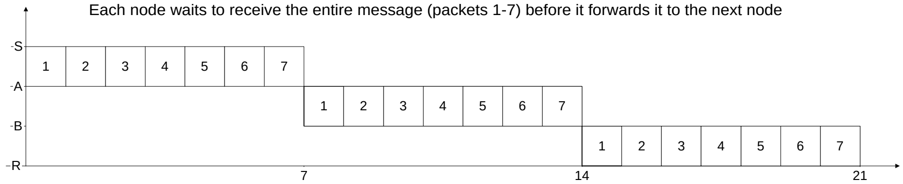


17

Illustration of Pipelining

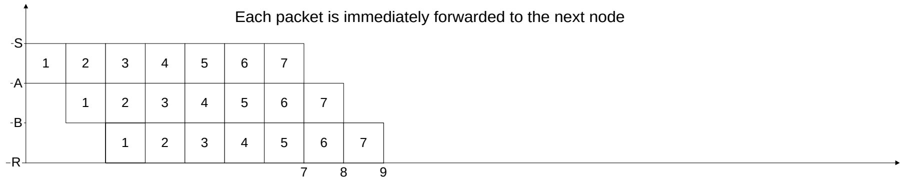
(a) Message switching

Each node waits to receive the entire message (packets 1-7) before it forwards it to the next node



(b) Packet switching

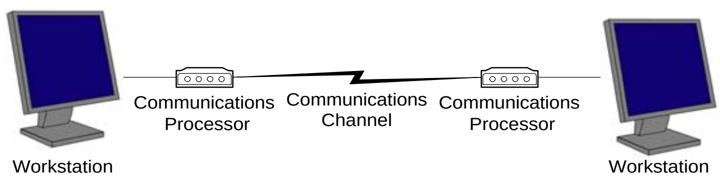
Each packet is immediately forwarded to the next node



18

LAN (point-to-point) network protocols

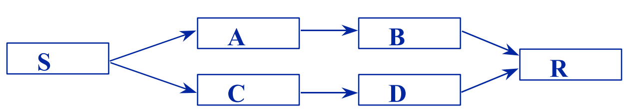
- "Data Link" protocols
  - provide point-to-point error-free transmission
  - break data into frames
  - attach error-detecting info into data
  - wait for acknowledgments and retransmit, if necessary
  - handle collisions (in broadcast networks)



19

Protocols for Packet Switching

- Assume data link protocols provide error-free point-to-point transmission
- Sender must break messages into packets
  - attach sequence number
  - attach destination address, other admin info to packets
- Receiver must reassemble message from packets
  - use sequence numbers in case packets arrive out of order
  - request retransmission of lost, garbled packets
- Intermediary nodes must route packet
  - find best next node in path for each packet
  - route packets to next node



20

Chrysanthos Dellarocas.

Flow Control and Routing

- Flow control
  - Nodes keep buffers of undelivered packets
  - Buffers can fill
  - Don't send until network and receiver are ready
- Routing
  - Need several hops
    - Determined at session creation
    - Or dynamically for each packet
  - Put complete routing information into "packet header"
    - Or store some of routing information at the intermediate nodes
      - Just put final destination into packet header

21

Acknowledgments

- Packets may arrive out of order
- Packets may be missing
- Stop and wait acknowledgement handling
  - Send just one packet
  - Wait for ACK before sending another
- Sliding window acknowledgement handling
  - Send several packets (numbered)
  - Wait for ACK of first one before sending (n+1)st
    - ACKs must be numbered as well

22

Error and Failure Detection and Handling

- If packets out of order, reorder them!
  - That's why they're numbered
- If packet missing from sequence, or unrecoverably garbled, send NACK

23