ECE390
Computer Engineering II
Lecture 22

Dr. Zbigniew Kalbarczyk

University of Illinois at Urbana- Champaign

Outline

- Networking
- Broadcast vs. unicast
- Reliable broadcast
- TCP (Transmission Control Protocol)

What is Networking?

- The means by which multiple computers communicate
- Primary example : the Internet
- Others
  - Telephone network
  - ECE390 lab
  - Home Automation

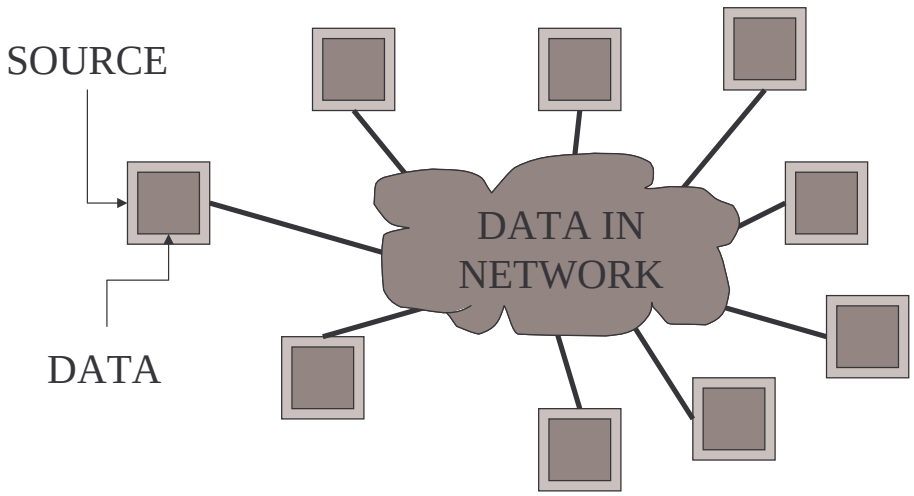
A Simple View

- Send
  - A program takes a piece of data and information about its destination and inserts them in the network
- Transmit
  - The network relays the program's data to the specified destination(s)
- Receive
  - An application removes the data from the network (and uses it)

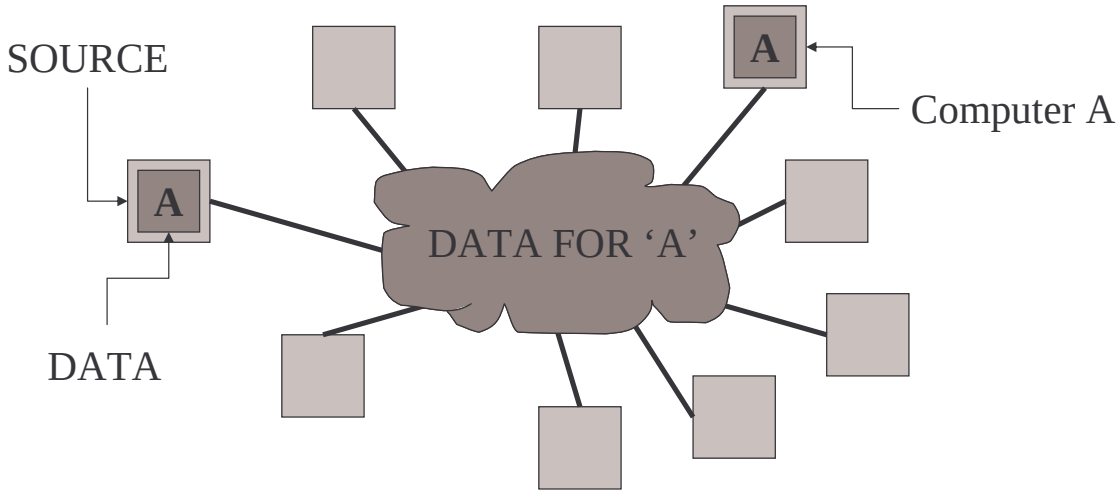
How is Data Transported?

- Data travels by
  - Wire
  - Optics
  - Air (wireless)

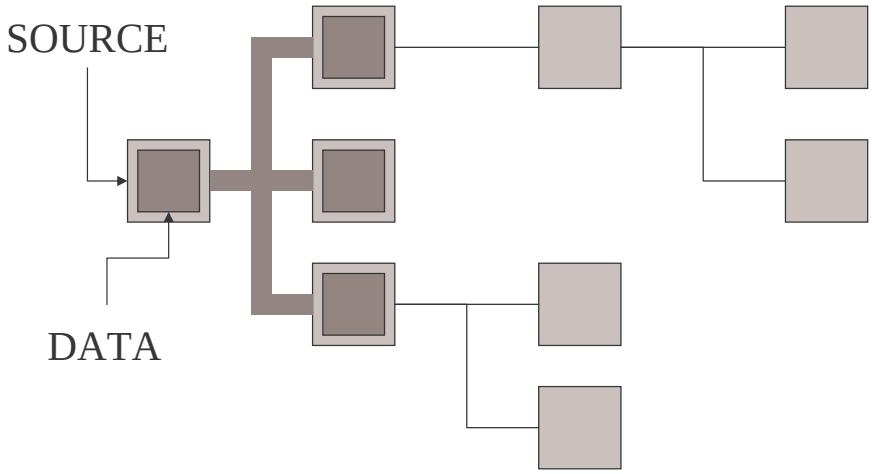
Delivery: Broadcast



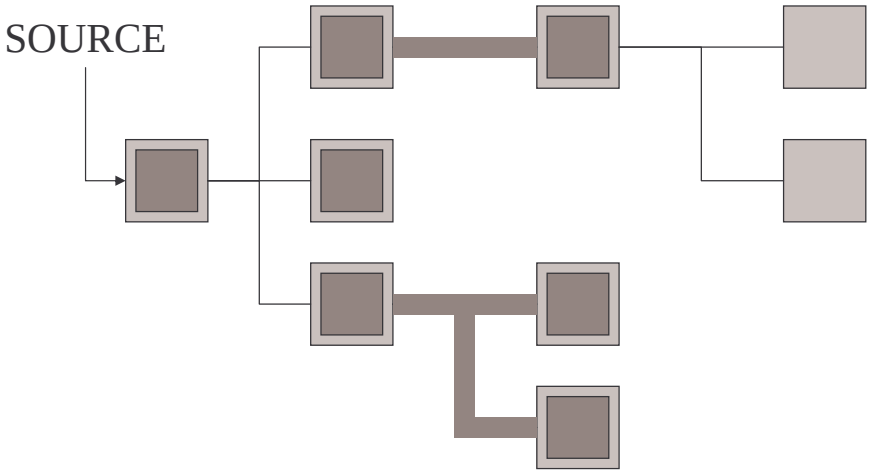
Delivery: Unicast



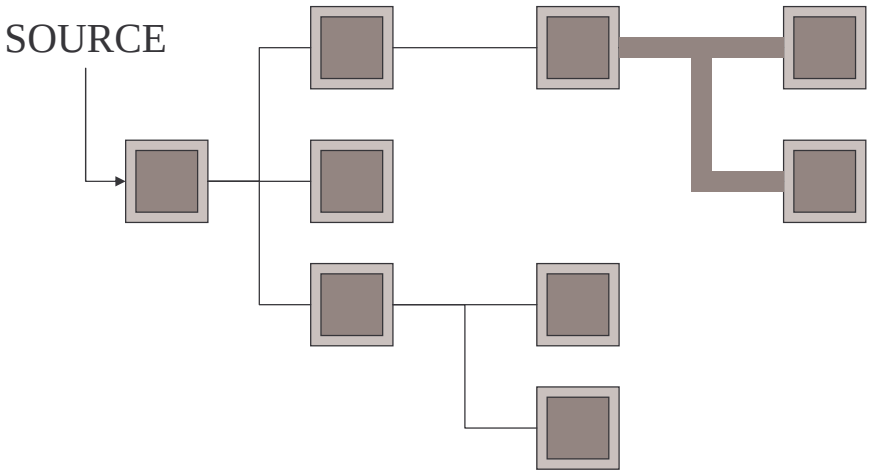
Delivery: Broadcast Routing



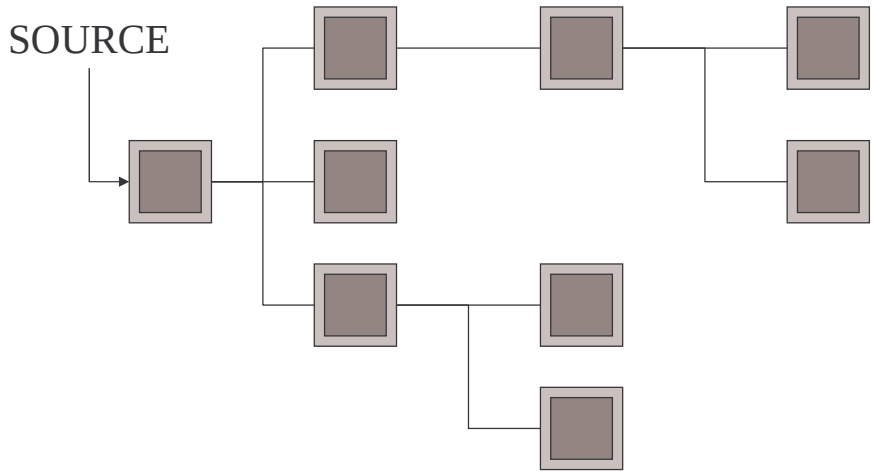
Delivery: Broadcast Routing



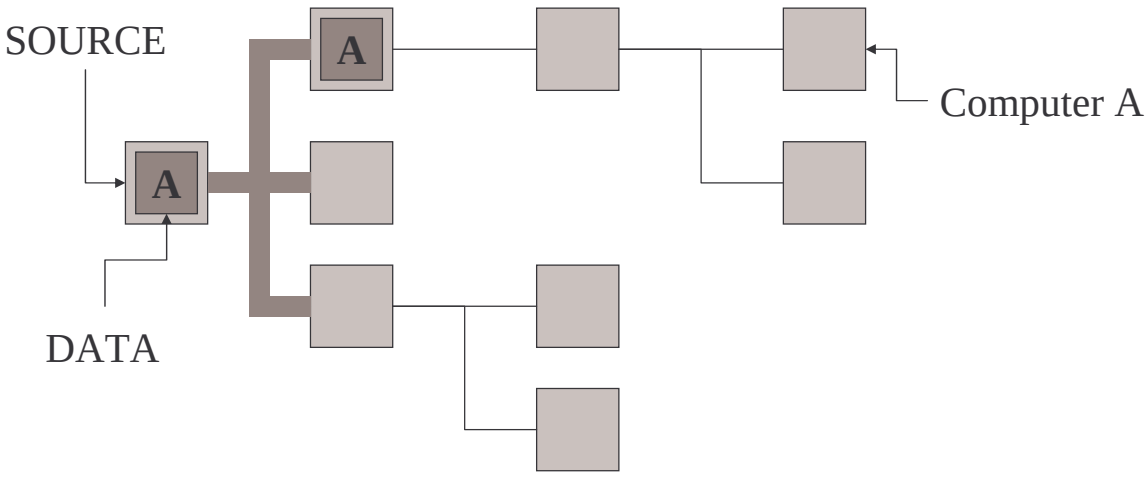
Delivery: Broadcast Routing



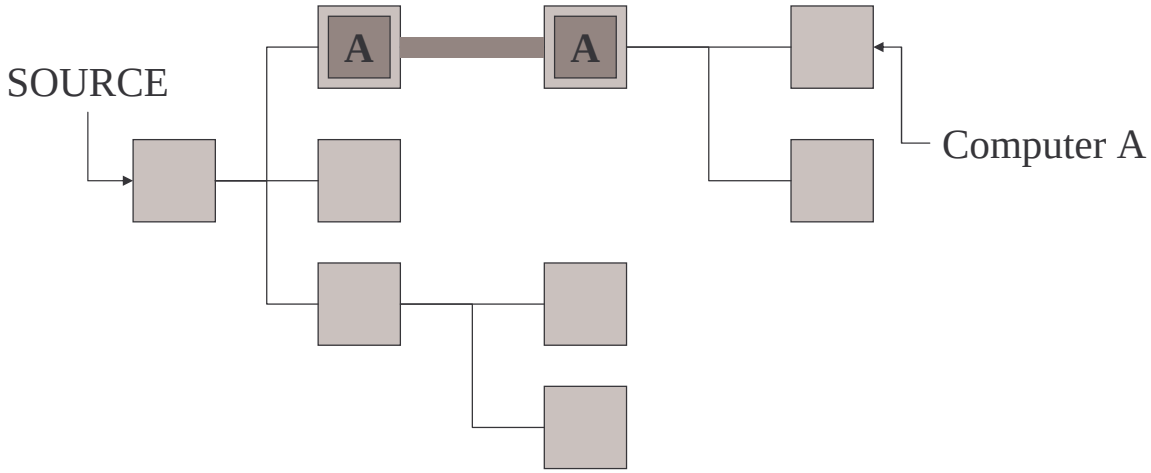
Delivery: Broadcast Routing



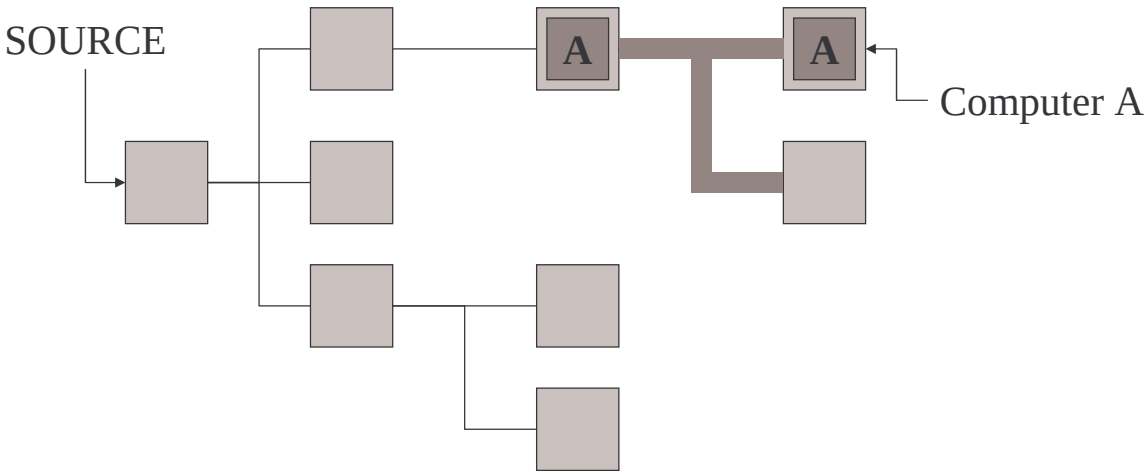
Delivery: Unicast Routing



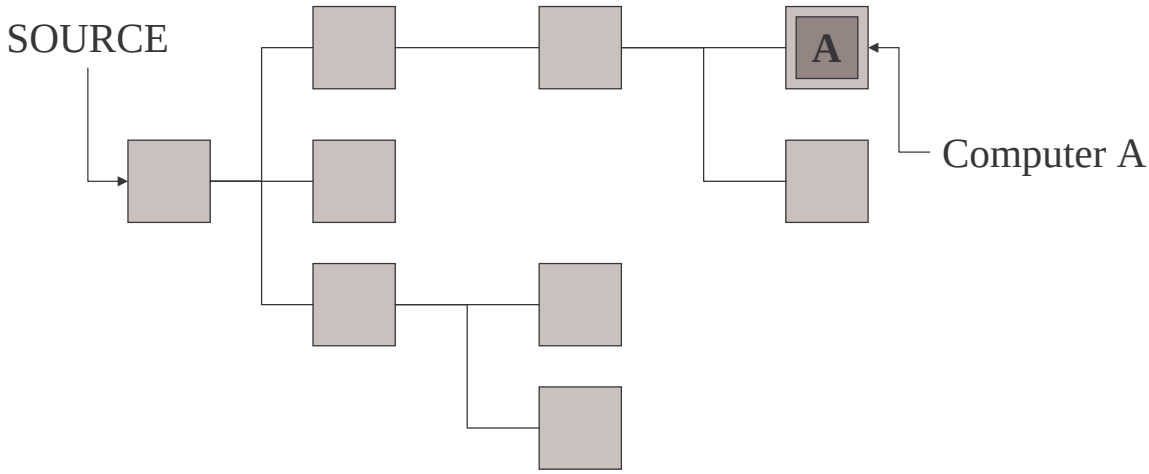
Delivery: Unicast Routing



Delivery: Unicast Routing



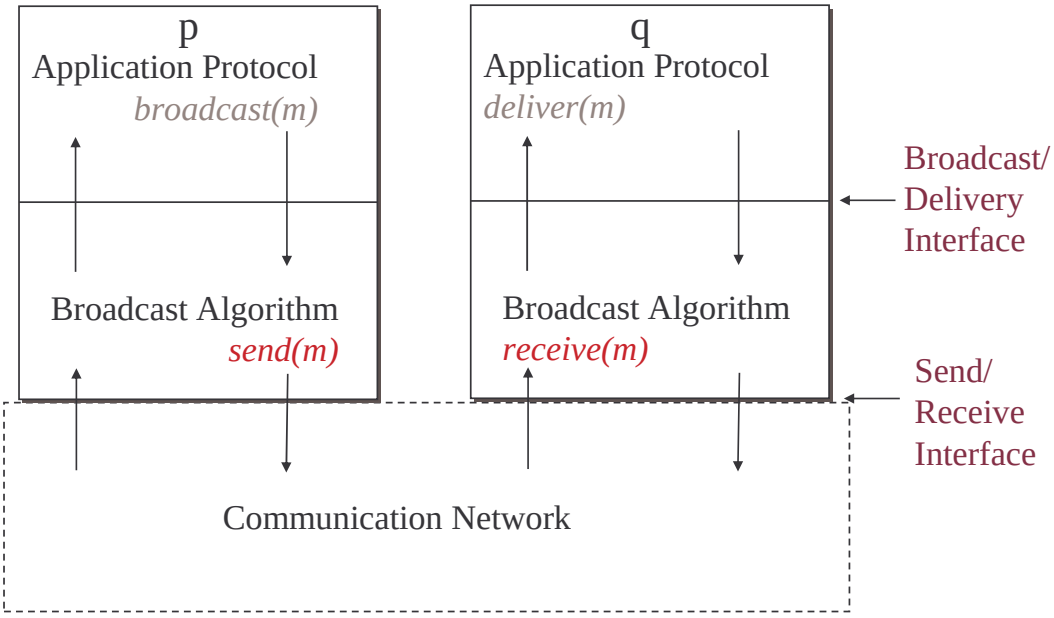
Delivery: Unicast Routing



Reliable Communications Transport Protocols

- Networking is a service provided to applications. What guarantees does this service provide?
- Low-level message-passing technologies
  - Datagram (UDP) : Few guarantees
    - Packets may be reordered, or lost!
  - Remote procedure call (RPC)
  - Connection-oriented (TCP)
    - Reliable connection between two endpoints
    - Lost packets are retransmitted
- To provide strong guarantees on message delivery we need high level protocols, such as reliable broadcast

Application/Broadcast Layering



Broadcast Protocols

- Cooperating processes in networked /distributed systems often communicate via broadcast
- A failure during a broadcast can lead to inconsistency and can compromise the integrity of the distributed system
- Need for supporting reliable broadcast protocols that provide strong guarantee on message delivery
- Example protocols include
  - reliable broadcast
  - FIFO broadcast
  - casual broadcast
  - atomic broadcast

Reliable Broadcast

- **Reliable broadcast** guarantees the following properties:
  - ***Validity:*** if a correct process broadcasts a message *m*, then all correct processes eventually deliver *m* (all correct processes agree on the set of messages they deliver)
  - ***Agreement:*** if a correct process delivers a message *m*, then all correct processes eventually deliver *m* (all messages broadcast by correct processes are delivered),
  - ***Integrity:*** for any message *m*, every correct process delivers *m* at most once and only if *m* was previously broadcast by a sender (no spurious messages are ever delivered)
- Reliable broadcast imposes no restrictions on the order of messages delivery

Programming Models

- Client-Server Model
  - Server is centralized resource
  - Clients are short-lived
  - Unicast messages between server and each client
- Peer Model
  - Resources are distributed and shared between all programs
  - Programs communicate through broadcast or multicast

PC Networking

- Network is Ethernet
  - Invented in 1973 by Bob Metcalf
  - Metcalf was at Xerox PARC, which also gave birth to the mouse and the graphical user interface
  - A broadcast medium
- Connects to the PC through a adapter card in the I/O bus (PCI)

PC Networking (2)

- To *send* a message, provide necessary data to adapter through software network driver
- When adapter *receives* a message, an interrupt is triggered which launches the network driver
- Driver executes user program's procedure (a callback)

TCP (Transmission Control Protocol)

- Reliable end-to-end byte stream over an unreliable network
- TCP entity accepts user data stream from local processes, break them up into pieces not exceeding 64K bytes and send each piece separately
- Receiving end reconstructs the original byte stream
- TCP ensures retransmission of data if needed and make sure that data are reassembled in a proper order
- To obtain TCP service, the sender and receiver must create end communication points, called sockets

Socket Primitives

- Socket – create a new socket,
- Bind – attach the local address to a socket),
- Listen – announce willingness to accept connections
- Accept – block the caller until the connection attempt arrives
- Connect – actively attempt to establish the connection
- Send – send some data over the connection
- Receive – receive some data from the connection
- Close – release the connection

NetBIOS

- Networking service for DOS/Win
- For a Local Area Network (LAN)
  - Provides global name space
  - Broadcast, multicast, unicast, plus reliable connection
- Use it through interrupt 05Ch
- Network Control Block (NCB)
  - Basic NetBIOS data structure

NetBIOS (2)

- How to receive messages
  - Blocking function: waits for an event
  - Callback function: NetBIOS calls a registered procedure, acts like an interrupt
- Names
  - 16 character identifiers
  - Local name: unique across the LAN
  - Group name: shared by multiple machines (multicast)

NetBIOS: pseudo examples

- Broad/Multicast Datagrams
  - Sender & receiver ADD NAME
  - *LOOP*
    - Sender SENDS broadcast datagram
    - Receiver RECEIVES datagram
  - Sender and receiver REMOVE NAME

- Unicast Connection
  - Sender & receiver ADD NAME
  - Receiver LISTENS
  - Sender CALLS (connection est'd)
  - *LOOP*
    - Sender SENDS
    - Receiver RECEIVES
  - Sender or receiver HANGUP (connection terminated)
  - Sender & receiver REMOVE NAME