The University Of Alabama in Huntsville Computer Science

cs570

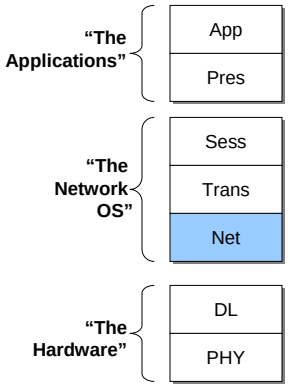
L3: Network Layer

The University Of Alabama in Huntsville Computer Science

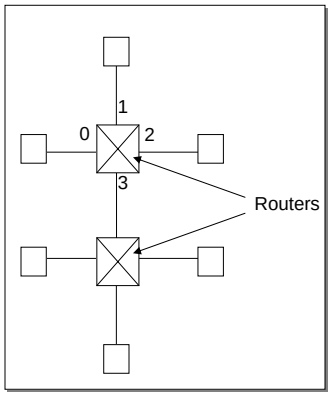
Network Layer

cs570

- In the DL layer, we can combine "similar" networks to make larger ones
- What about networks that aren't similar?
  - Different connection models?
  - Different addressing schemes?
- Network layer forms a uniform "shell" around interconnected networks of all types, provides uniform address space



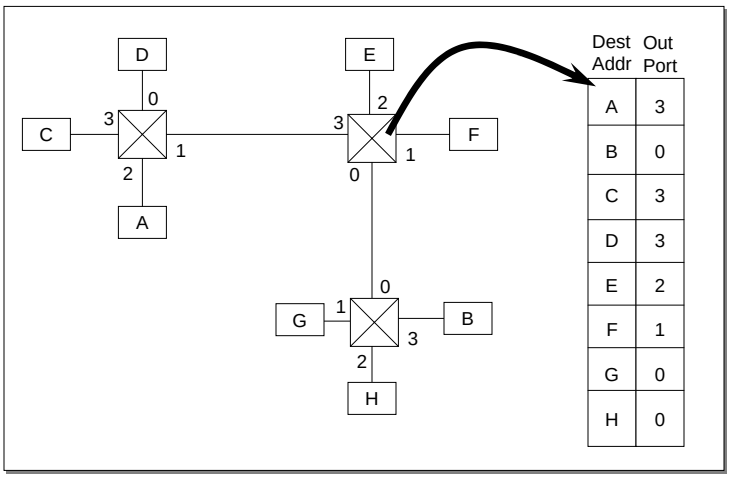
Routers



Routers:

- Connect networks at the Network Layer
- Both packet and circuit-switching varieties
- Have two primary jobs:
  - Routing: Maintaining tables that specify the router output to use for a particular destination address
  - Switching: Moving data from router inputs to outputs based on the routing table
- May also support other functions:
  - Congestion control
  - QoS

Routing tables

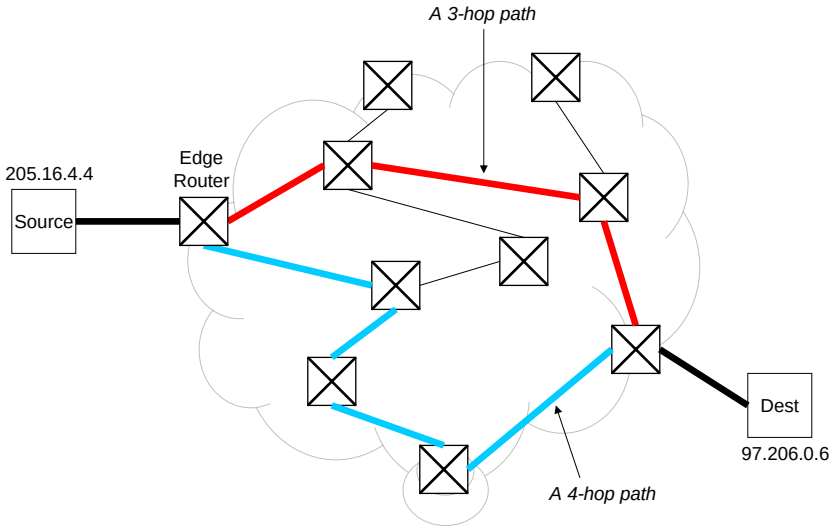


Dest Addr	Out Port
A	3
B	0
C	3
D	3
E	2
F	1
G	0
H	0

Review: The communication models

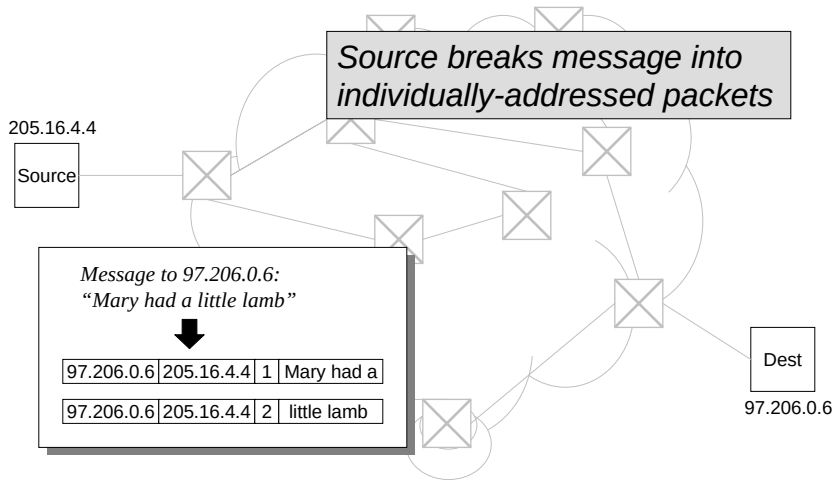
- Connectionless, AKA *packet switching*
  - The "postal service" model
  - Messages are broken into small packets. Each packet is individually forwarded through the network to the destination.
- Connection-oriented, AKA *circuit switching*
  - The "phone-company" model
  - Communicating hosts set up a dedicated channel across the network. After the channel is set up, they can exchange data with only a small amount of protocol overhead.

The idea of Shortest or "Least-Cost" paths

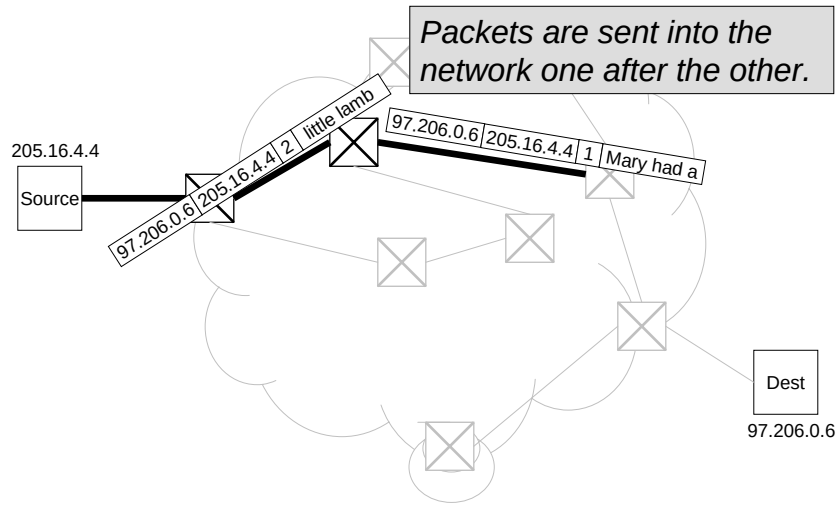


Packet Switching

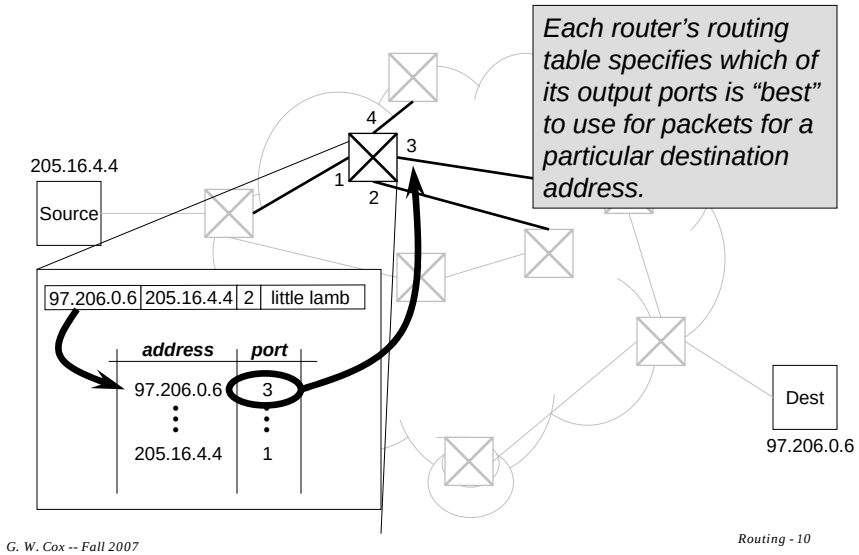
Connectionless (1)



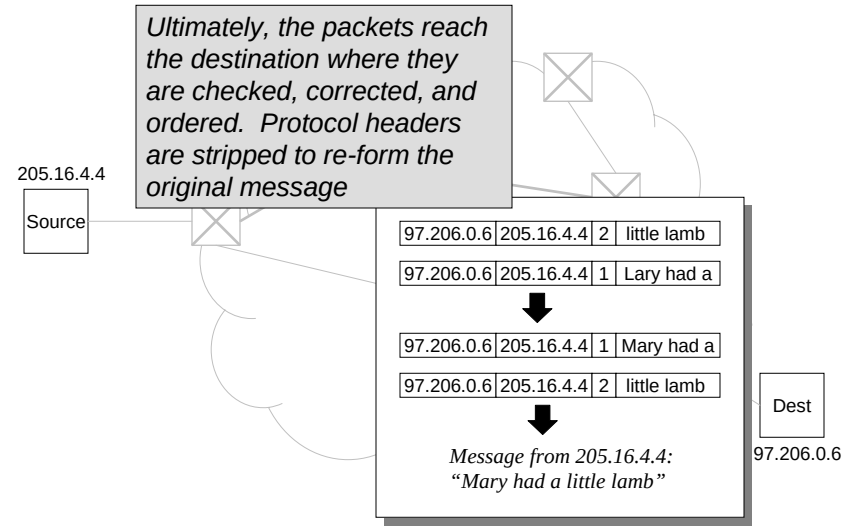
Connectionless (2)



Connectionless (3)

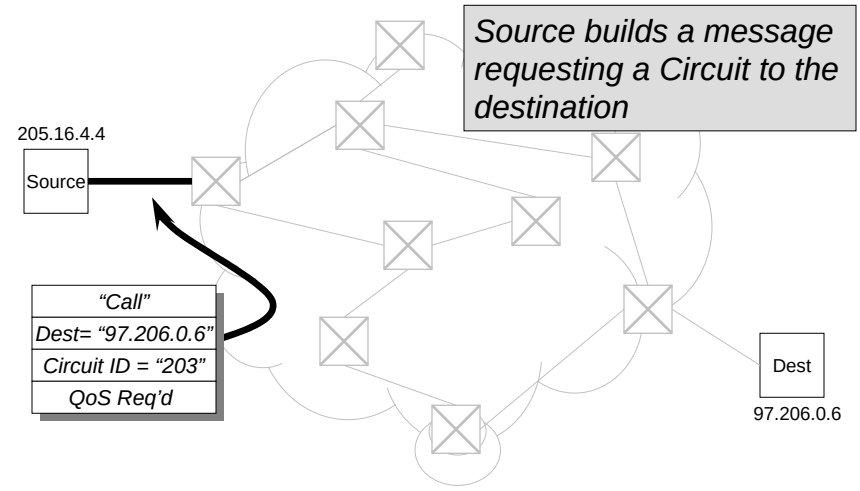


Connectionless (4)

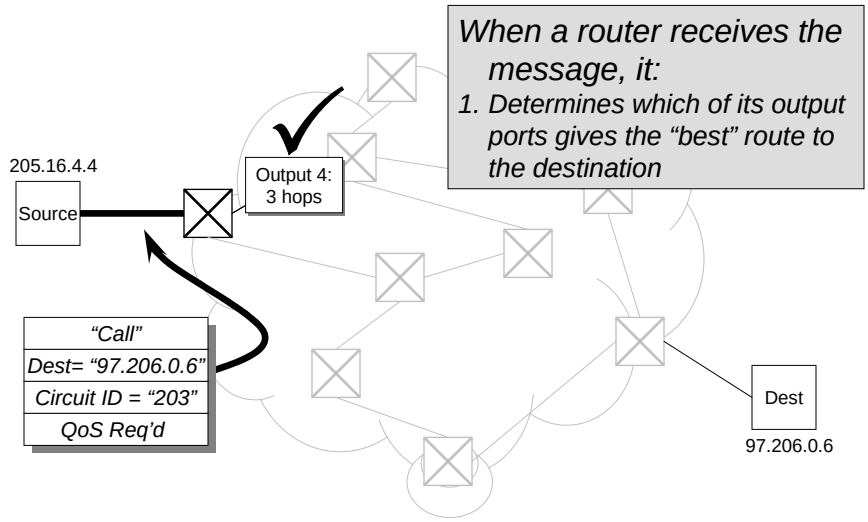


Connection-Oriented

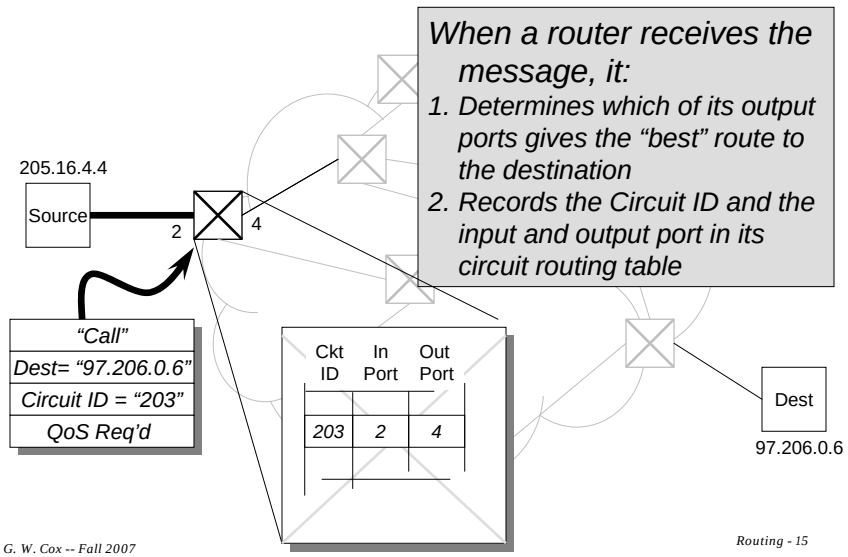
Connection-oriented: The history of a circuit (1)



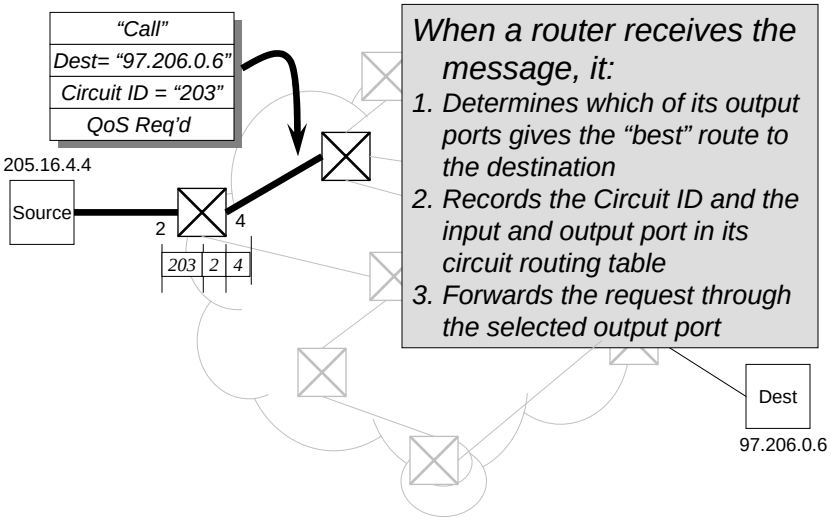
Connection-oriented: The history of a circuit (2)



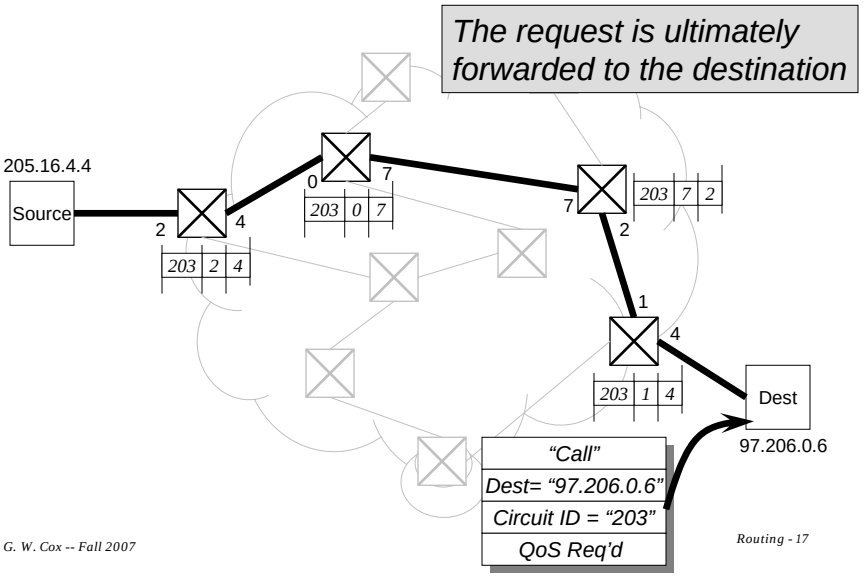
Connection-oriented: The history of a circuit (3)



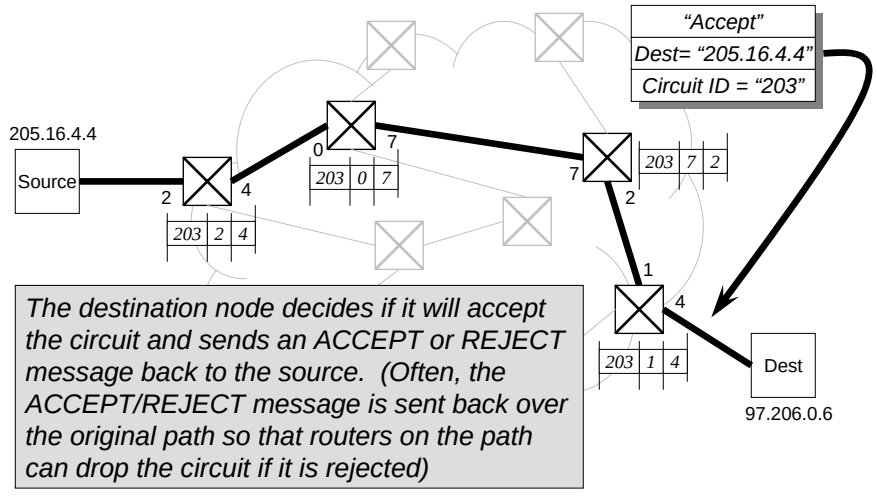
Connection-oriented: The history of a circuit (4)



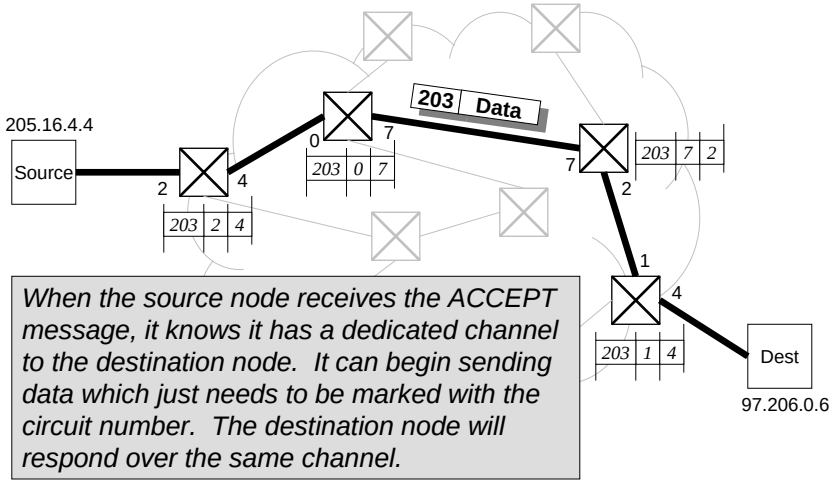
Connection-oriented: The history of a circuit (5)



Connection-oriented: The history of a circuit (6)

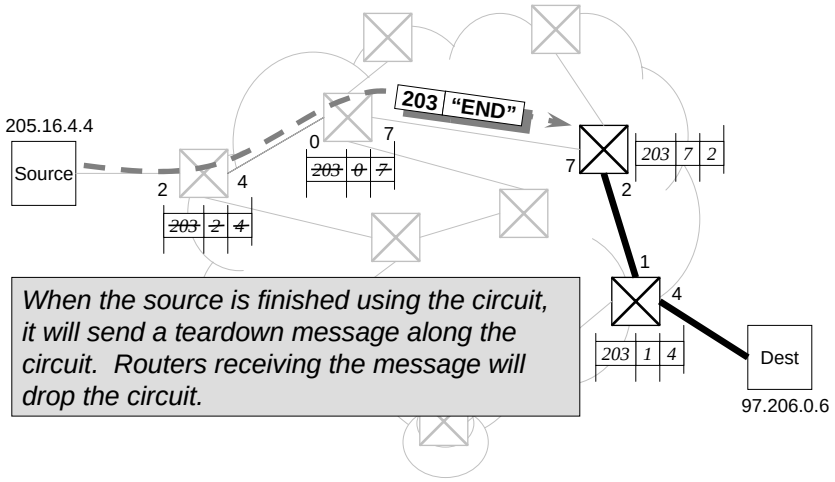


Connection-oriented: The history of a circuit (7)



When the source node receives the ACCEPT message, it knows it has a dedicated channel to the destination node. It can begin sending data which just needs to be marked with the circuit number. The destination node will respond over the same channel.

Connection-oriented: The history of a circuit (8)



When the source is finished using the circuit, it will send a teardown message along the circuit. Routers receiving the message will drop the circuit.

Thinking about connection-oriented

- From the programmer's standpoint:
  - The dedicated end-to-end "pipe" paradigm is attractive for some applications:
    - After initial setup, there's very little work needed to communicate
    - If we have a long-lived, steady stream of data (e.g, streaming video), low overhead per data item makes up for high setup overhead
- From the network designer's standpoint:
  - Some good points:
    - Since channel is dedicated, easier to set up service guarantees (QoS)
    - Don't have to worry about some packet-related problems (e.g, out-of-order packets)
  - Some bad:
    - If the traffic over a circuit is relatively sporadic (e.g, typical request/reply traffic), the utilization of the circuit will be low and we will waste network bandwidth.
    - If the lifetime of a connection is short, we may not gain enough in the low overhead per data item to make up for the high setup overhead.
      - Could send a packet and receive an ACK in the time it takes to merely set up a circuit.
    - If a device on the connection fails, do we have to re-build the whole path?

Thinking about packet switching

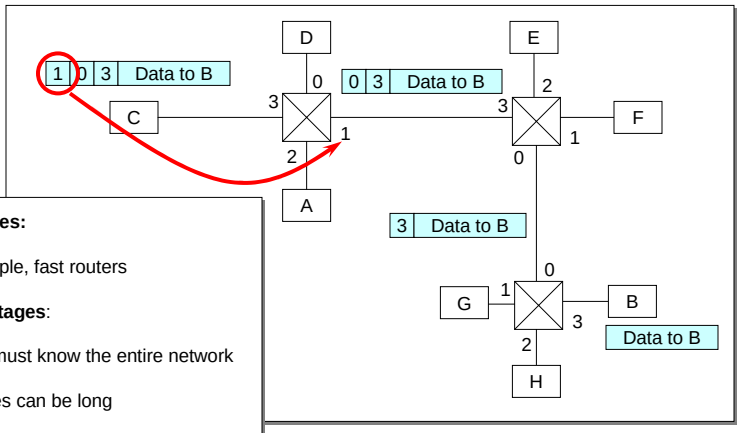
- From the programmer's standpoint:
  - For short, quick messaging, the idea of just packing the data up and firing it off seems right
- From the network designer's standpoint:
  - With packet switching, multiplexing is, in a sense, automatic – packets from multiple channels are mixed together on the links, sharing the total bandwidth
  - Since every packet is individually routed, network topology changes and failures can be handled (hard for circuit switching)
  - Packet switching routers only have one job to do – generally produces simpler, faster designs

A best-of-both-worlds approach (TCP/IP)

- Give the programmer both connection-oriented (TCP) and connectionless (UDP) paradigms
- For network efficiency, run everything over a connectionless infrastructure (IP)

A third way to route: Source routing

The route is pre-calculated by the sender and sent as part of the message.



Advantages:

- Very simple, fast routers

Disadvantages:

- Sender must know the entire network
- Messages can be long
- Practical only for small, stable networks