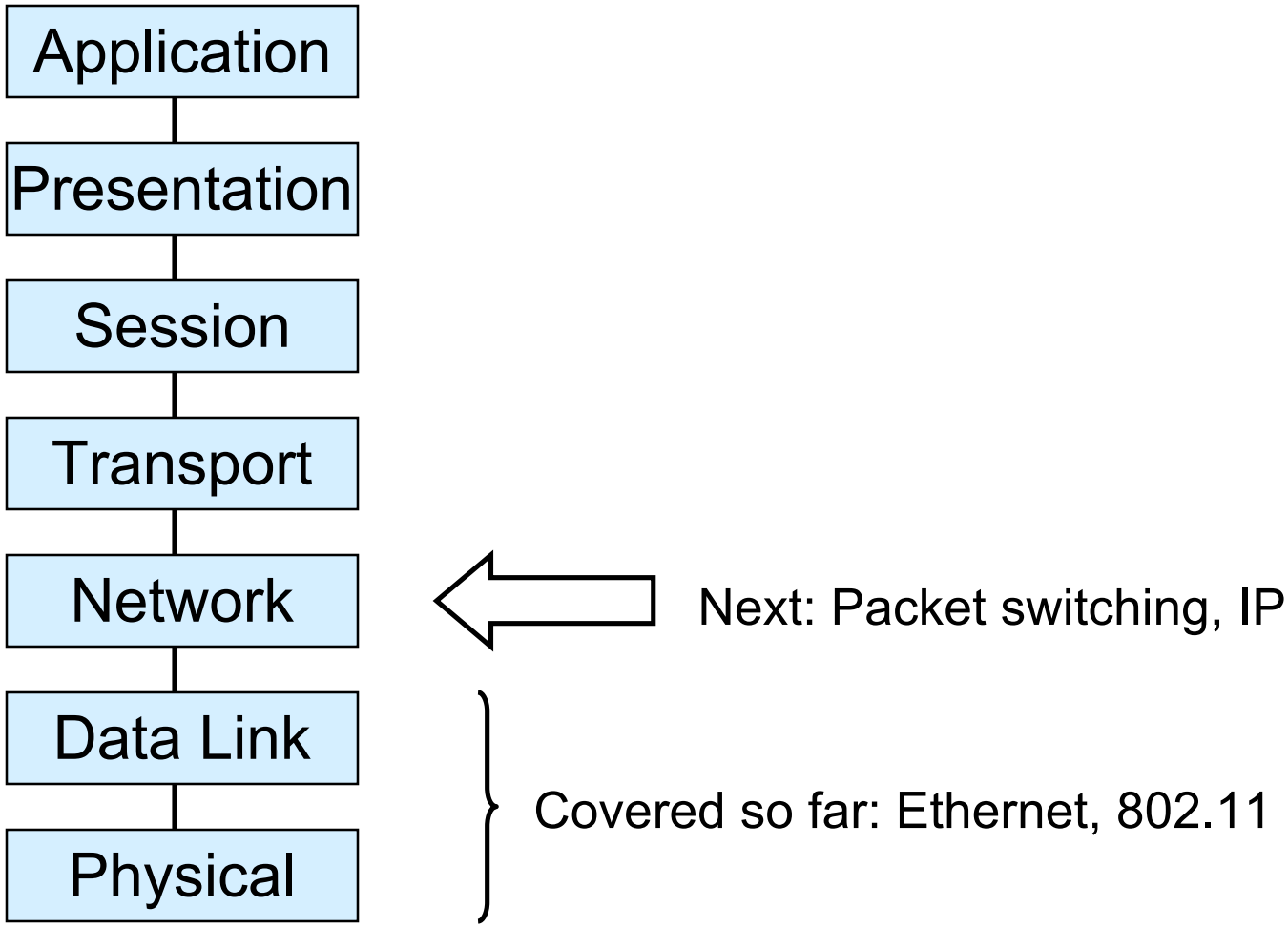
# CSE331: Introduction to Networks and Security

Lecture 7

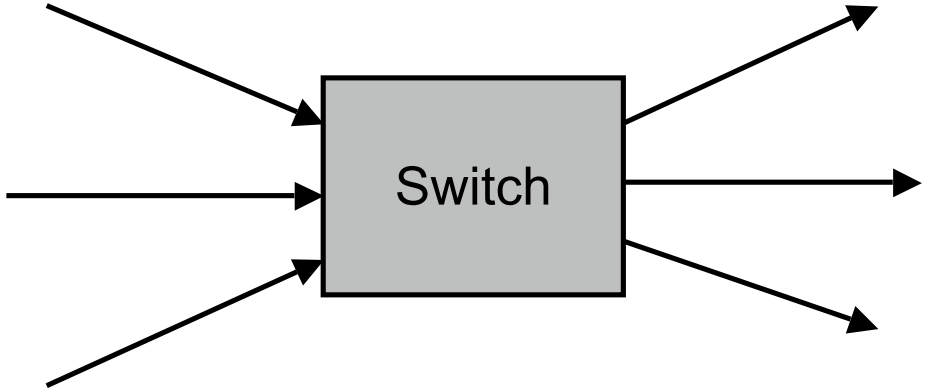
Fall 2006

# OSI Reference Model



# Packet Switching

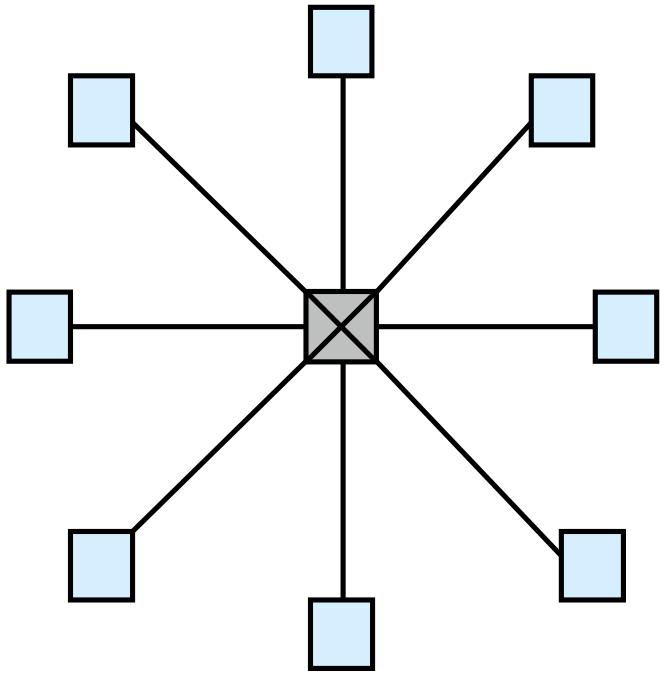
- A *switch*
  - Has many inputs and many outputs
  - Takes packets that arrive on an input and forwards them to the right output




- Key problem: finite output bandwidth

# Star Topology

- Scalability
  - Large networks can be built by interconnecting switches.
  - Can connect via high bandwidth point-to-point links = large distances.
  - Adding a new host to a switch doesn't necessarily degrade performance.



# Switching Issues

- *Contention*
  - Arrival rate of packets going to the same output exceeds output capacity
  - Switch buffers packets
- *Congestion*
  - Switch runs out of buffer space
  - Forces packets to be dropped

# Forwarding Decision

- How does the switch know where to forward a packet?
  - Looks at the packet header to make the decision
- Common approaches
  - Datagram (or *connectionless*)
    e.g. IP
  - Virtual Circuit (or *connection-oriented*)
    e.g. Frame Relay, ATM
  - (Less common) Source routing

# Datagram approach

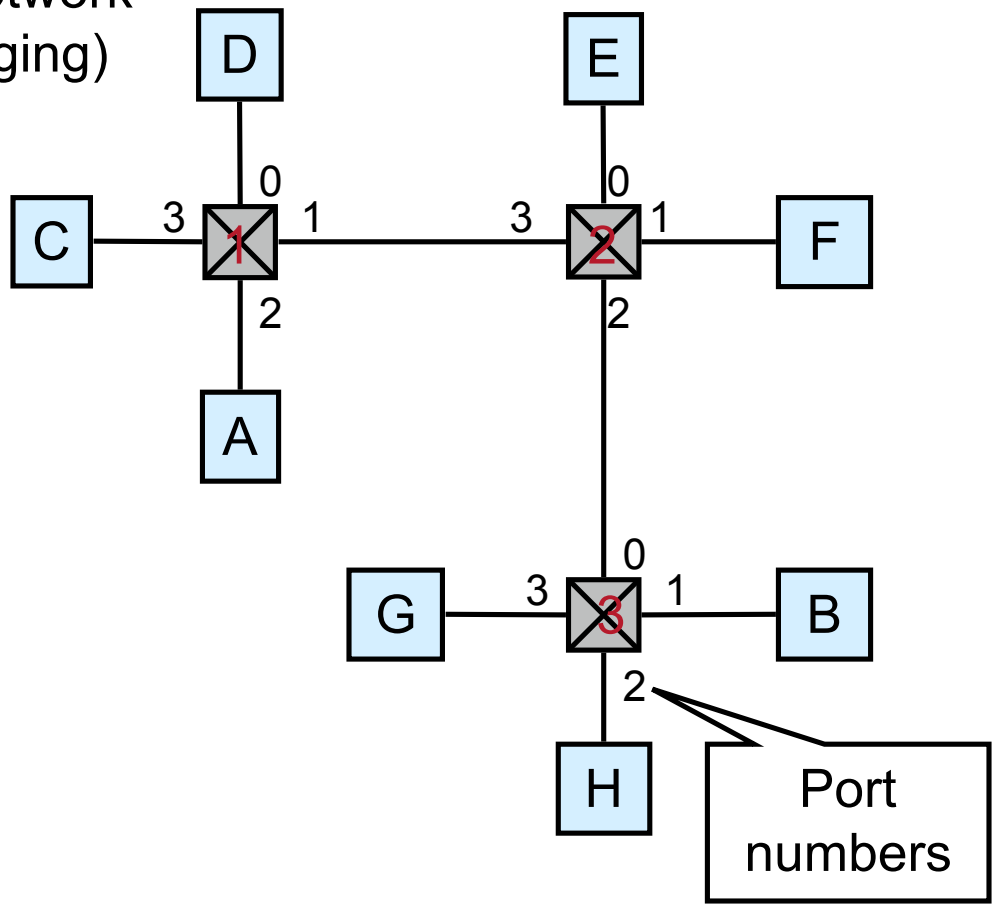
- Every packet contains a complete destination address
  - Enough information so that any switch can decide where the packet goes.
- Features of datagram approach
  - Packets can be sent at anywhere at any time
  - Sender doesn't know if network can deliver the packet (or if destination host is available)
  - Each packet is forwarded independently (two packets may take different routes)
  - Possible to route around switch or link failures

# Forwarding Tables

- Provide route information.
- Easy to determine if network is known (and unchanging)

Forwarding table for switch 2.

| Dest. | Port |
|---|---|
| A | 3 |
| B | 2 |
| C | 3 |
| D | 3 |
| E | 0 |
| F | 1 |
| G | 2 |
| H | 2 |

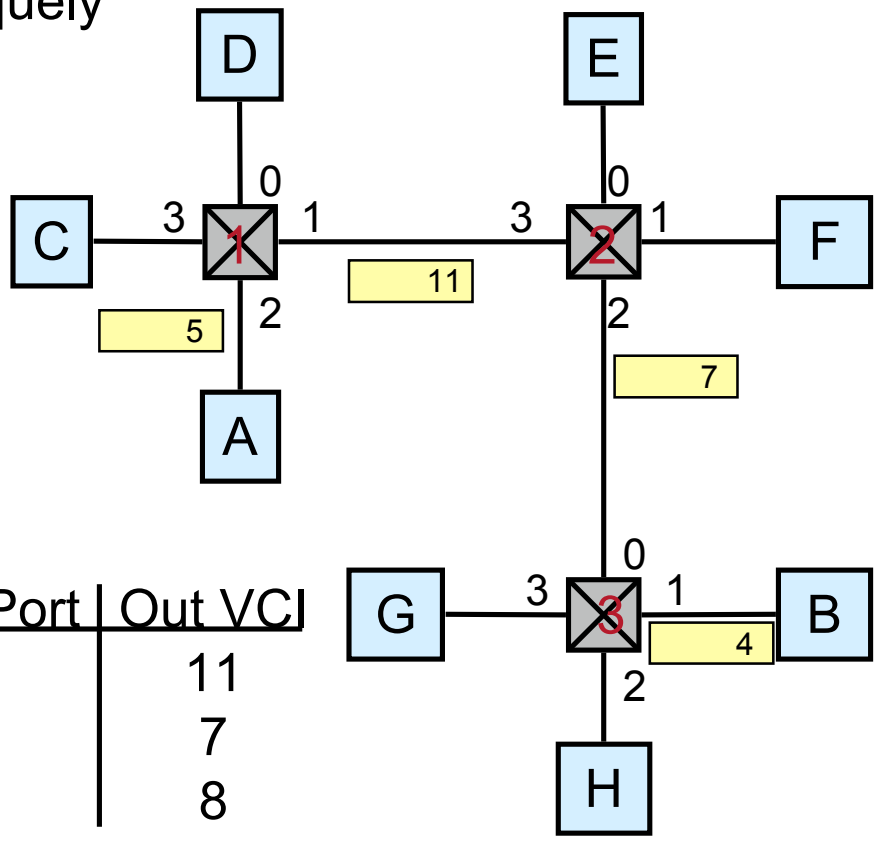


# Virtual circuit approach

- Set up the connection before data transfer
  - Allocate resources on circuits
  - Set up forwarding tables
- Benefits of virtual circuit approach
  - Performance: per-packet switching cost is low
  - Reliability: predictable latency and throughput
- Drawbacks
  - Setup time is long
  - Fault tolerance

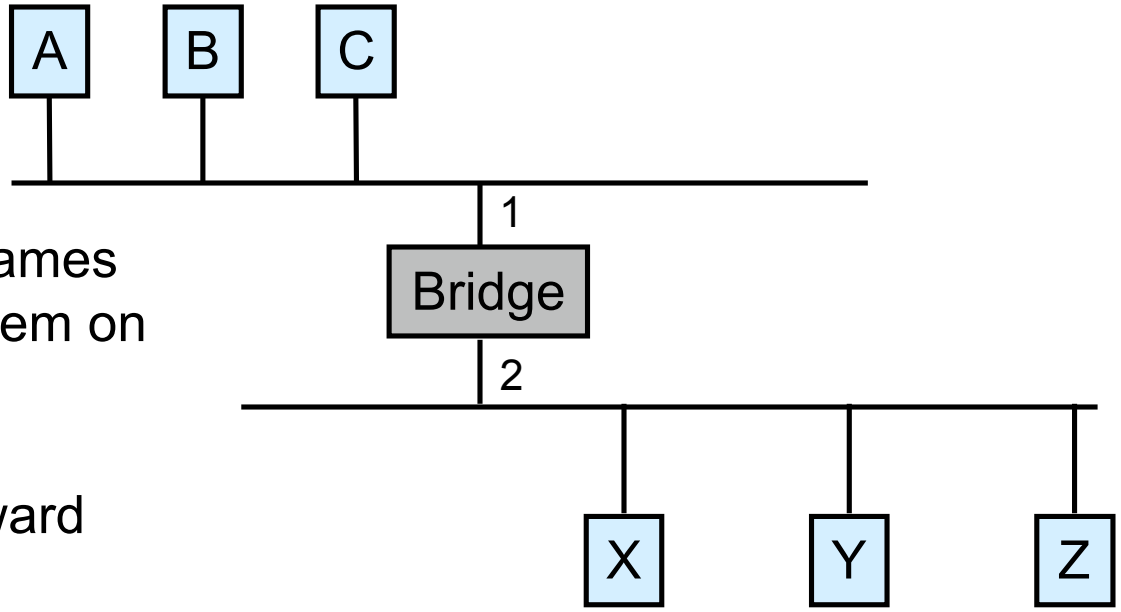
# Virtual Circuit Switching

- VCI = Virtual Circuit Identifier
- Incoming port + VCI uniquely identify virtual circuit
- Setup phase constructs circuit table entries at each switch




| Switch | In Port | In VCI | Out Port | Out VCI |
|---|---|---|---|---|
| 1 | 2 | 5 | 1 | 11 |
| 2 | 3 | 11 | 2 | 7 |
| 3 | 0 | 7 | 1 | 8 |

# Bridges and LAN Switches




- Bridge accepts LAN frames on one port, outputs them on another.
- Optimization: only forward appropriate frames
- Learning bridges
  - watch incoming *source* address S at port number P
  - add entry to forward address S to port P
  - if no entry, broadcast to all ports

# Problem: Cycles (Loops)

- Frame gets rebroadcast forever
- Could avoid by construction, BUT:
  - Hard, especially management
  - Often want redundancy
- Solution:
  - Restrict active ports to a *Spanning Tree*
  - Basic design by Radia Perlman of Digital
  - 802.1 specification of LAN Bridges is based on this algorithm

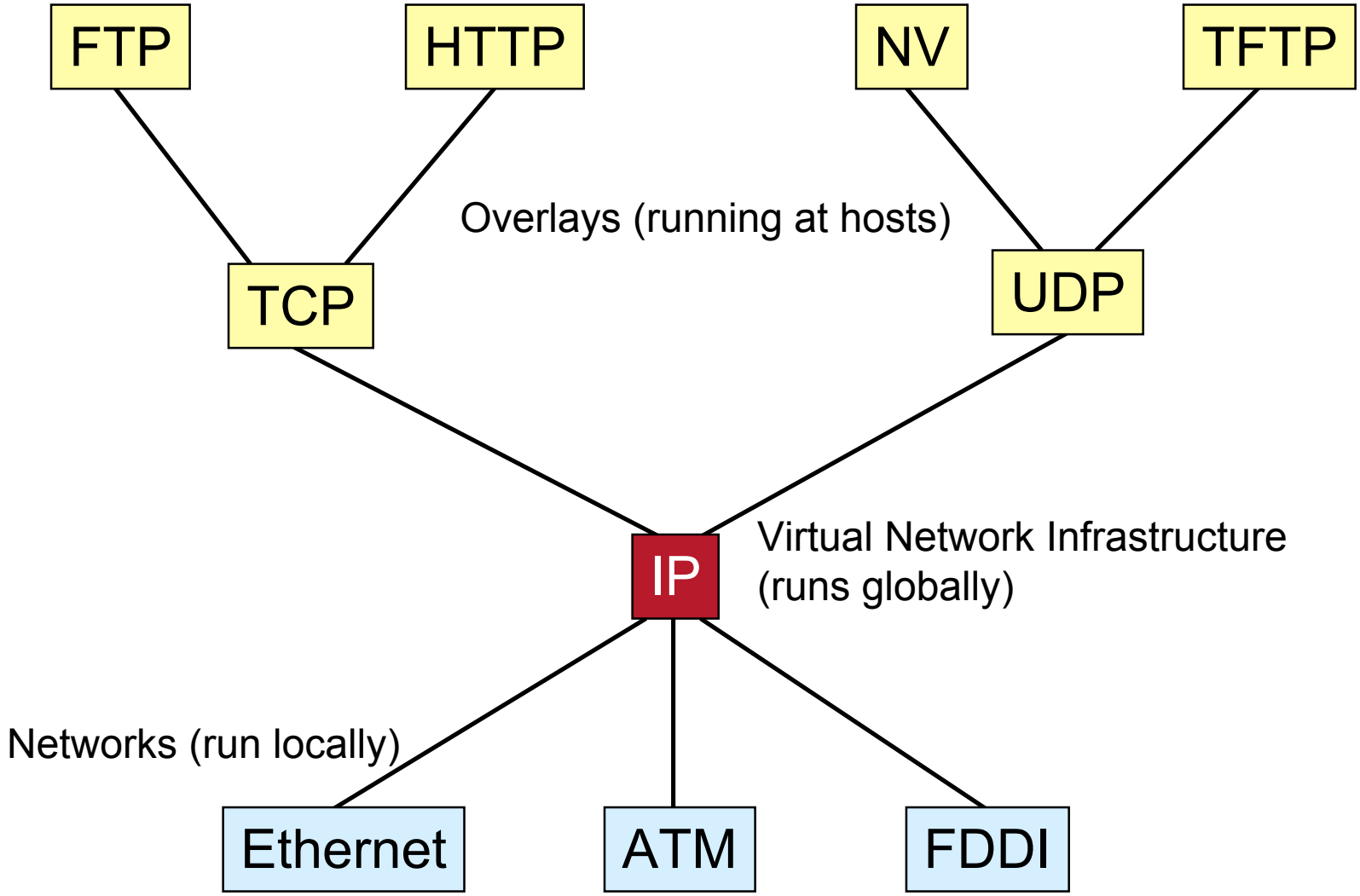
# Limitations of Bridges

- Scaling
  - Connections on order of dozens
  - Spanning tree algorithm scales linearly
  - Transparency incomplete
    - Congestion can be visible to higher protocol layers
    - Latency can be larger and more variable
- Heterogeneity
  - Limited to compatible (similarly addressed) link layers

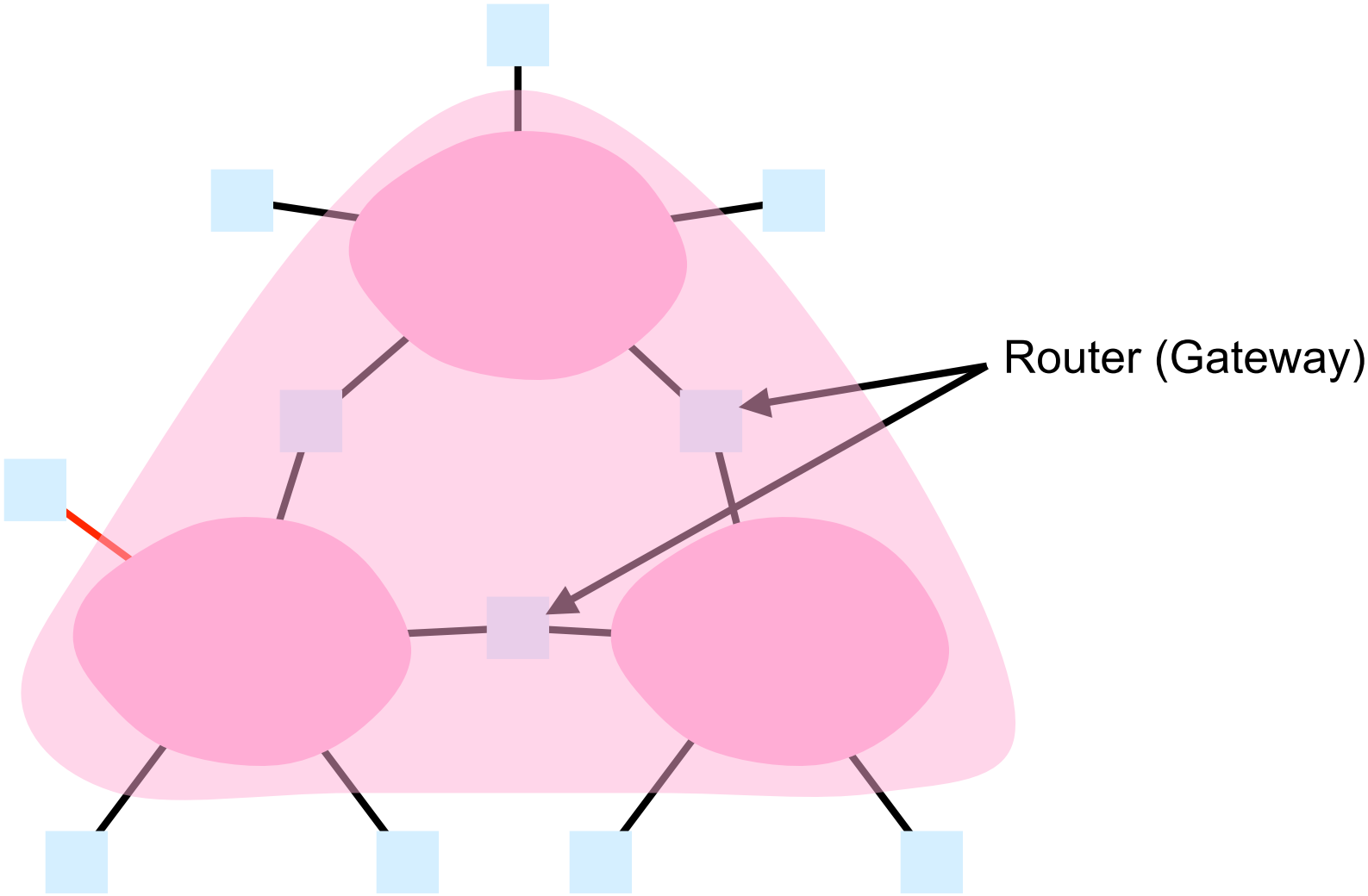
# Internet Protocol (IP)

- Terminology
- Service model
- Addresses
- Forwarding
- ARP
- ICMP

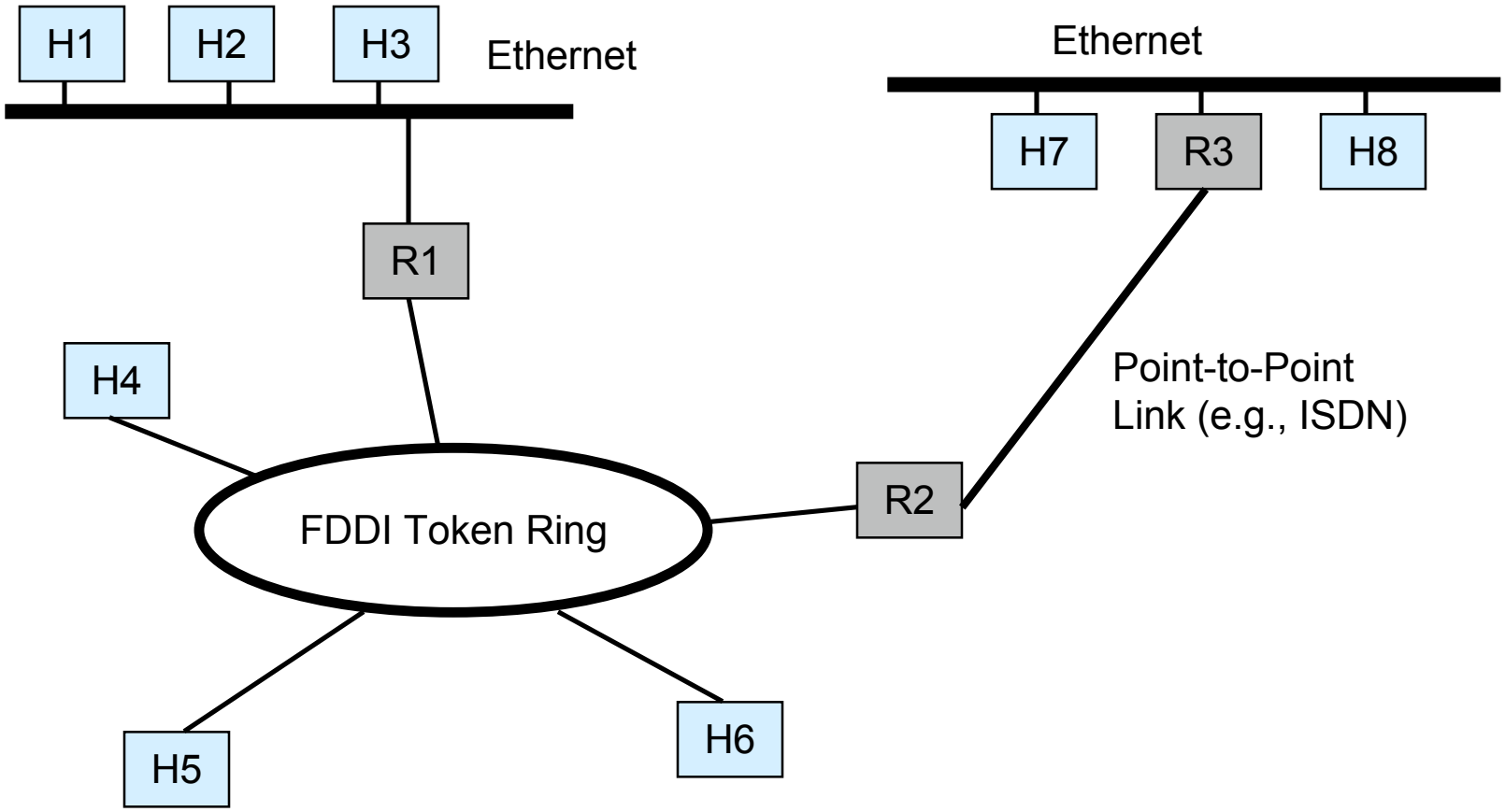
# Internet Protocol Interoperability



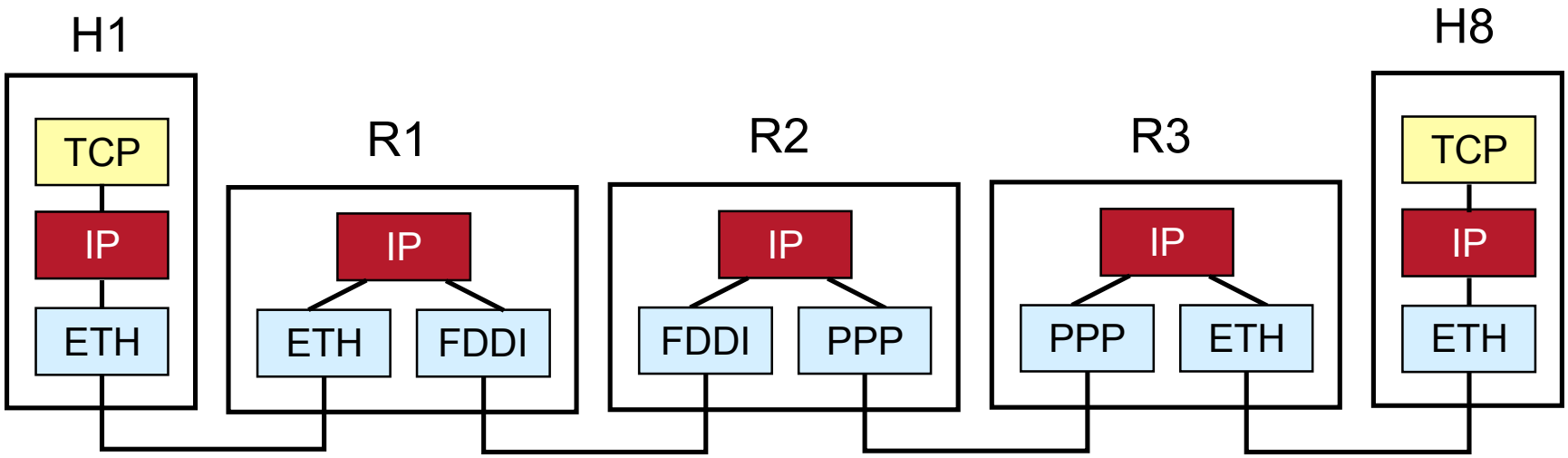
# Internetworks



# Internetworks



# IP Encapsulation

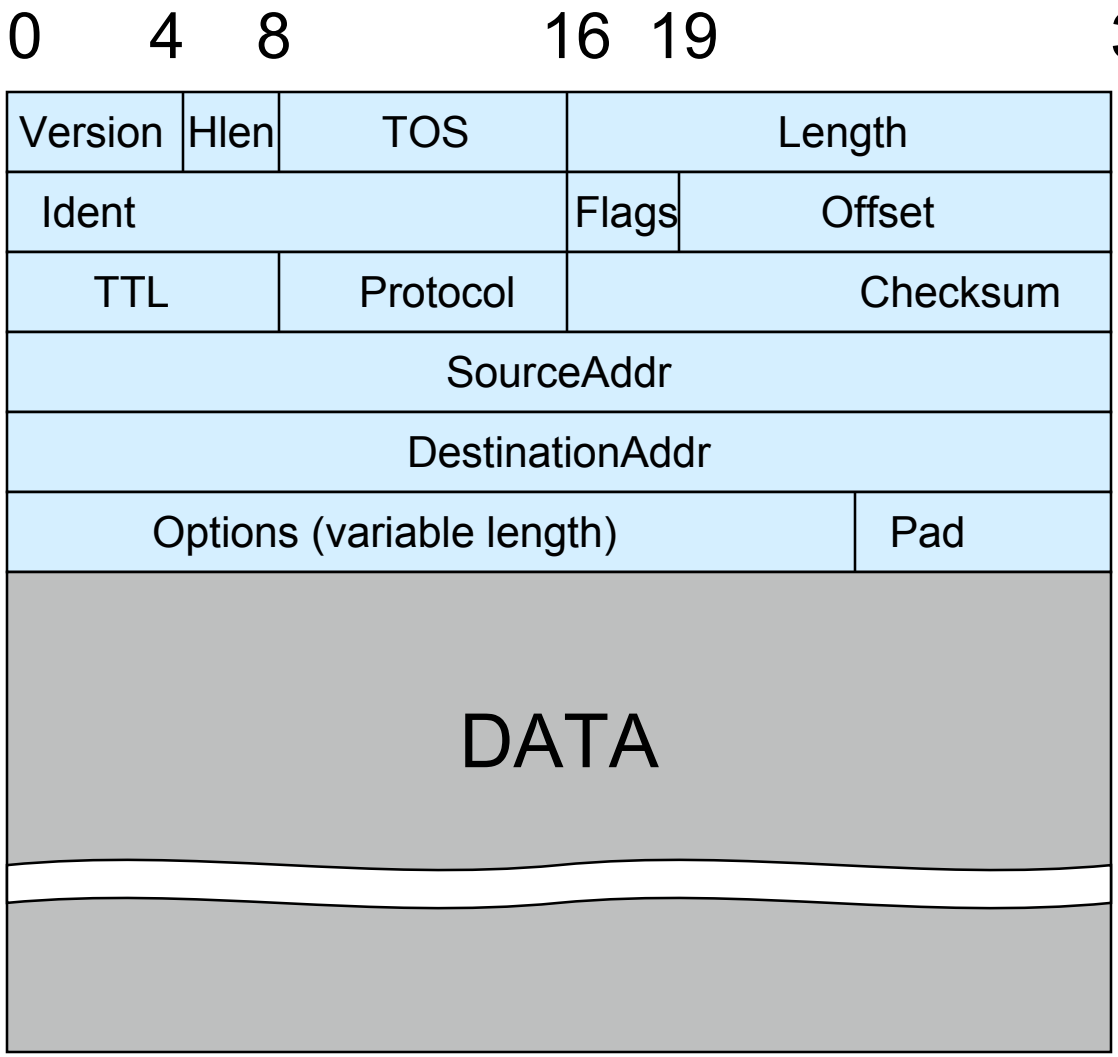



Example of protocol layers used to transmit from H1 to H8 in network shown on previous slide.

# Service Model

- Choose minimal service model
  - All nets can implement
  - "Tin cans and a string" extremum
- Features:
  - Standard packet format
  - Best-effort datagram delivery (unreliable)
  - *"Run over anything"*

# IPv4 Packet Format



# Fields of IPv4 Header

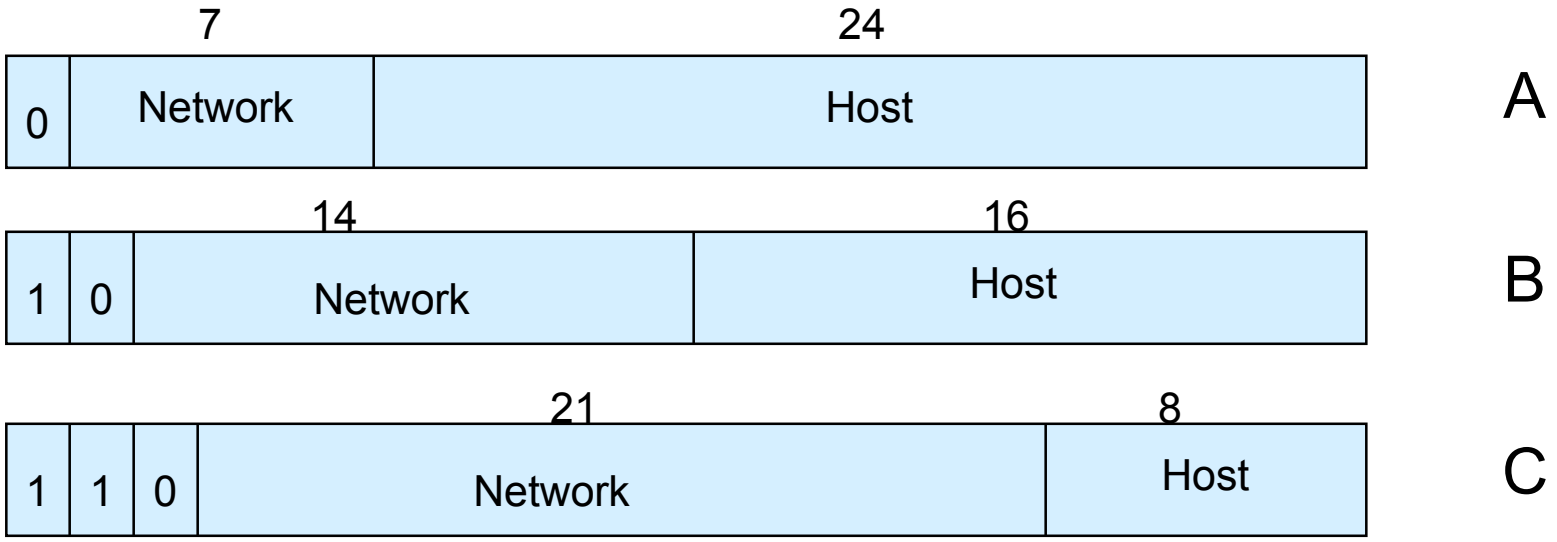
- Version
  - Version of IP, example header is IPv4
  - First field so easy to implement case statement
- Hlen
  - Header length, in 32-bit *words*
- TOS
  - Type of Service (rarely used)
  - Priorities, delay, throughput, reliability
- Length
  - Length of datagram, in *bytes*
  - 16 bits, hence max. of 65,536 bytes
- Fields for *fragmentation* and *reassembly*
  - Identifier
  - Flags
  - Offset

# Header fields, continued

- TTL
  - Time to live (in reality, hop count)
  - 64 is the current default (128 also used)
- Protocol
  - e.g., TCP (6), UDP(17), etc.
- Checksum
  - Checksum of header (not CRC)
  - If header fails checksum, discard the whole packet
- SourceAddr, DestinationAddr
  - 32 bit IP addresses - global, IP-defined
- Options
  - length can be computed using Hlen

# IP addresses

- Hierarchical, not flat as in Ethernet




- Written as four decimal numbers separated by dots: 158.130.14.2

# Network Classes

| *Class* | *# of nets* | *# of hosts per net* |
|---|---|---|
| ***A*** | 126 | ~16 million |
| ***B*** | 8192 | 65534 |
| ***C*** | ~2 million | 254 |

# IP addresses and networks

- Every network device has an IP address
- Every IP packet (datagram) contains the destination IP address
- The network part of the IP address uniquely identifies a single physical network (that is part of the larger Internet).
- Routers are connected to multiple network interfaces
  - A router has multiple network adapters
  - Routers can exchange packets on any network they're attached to.

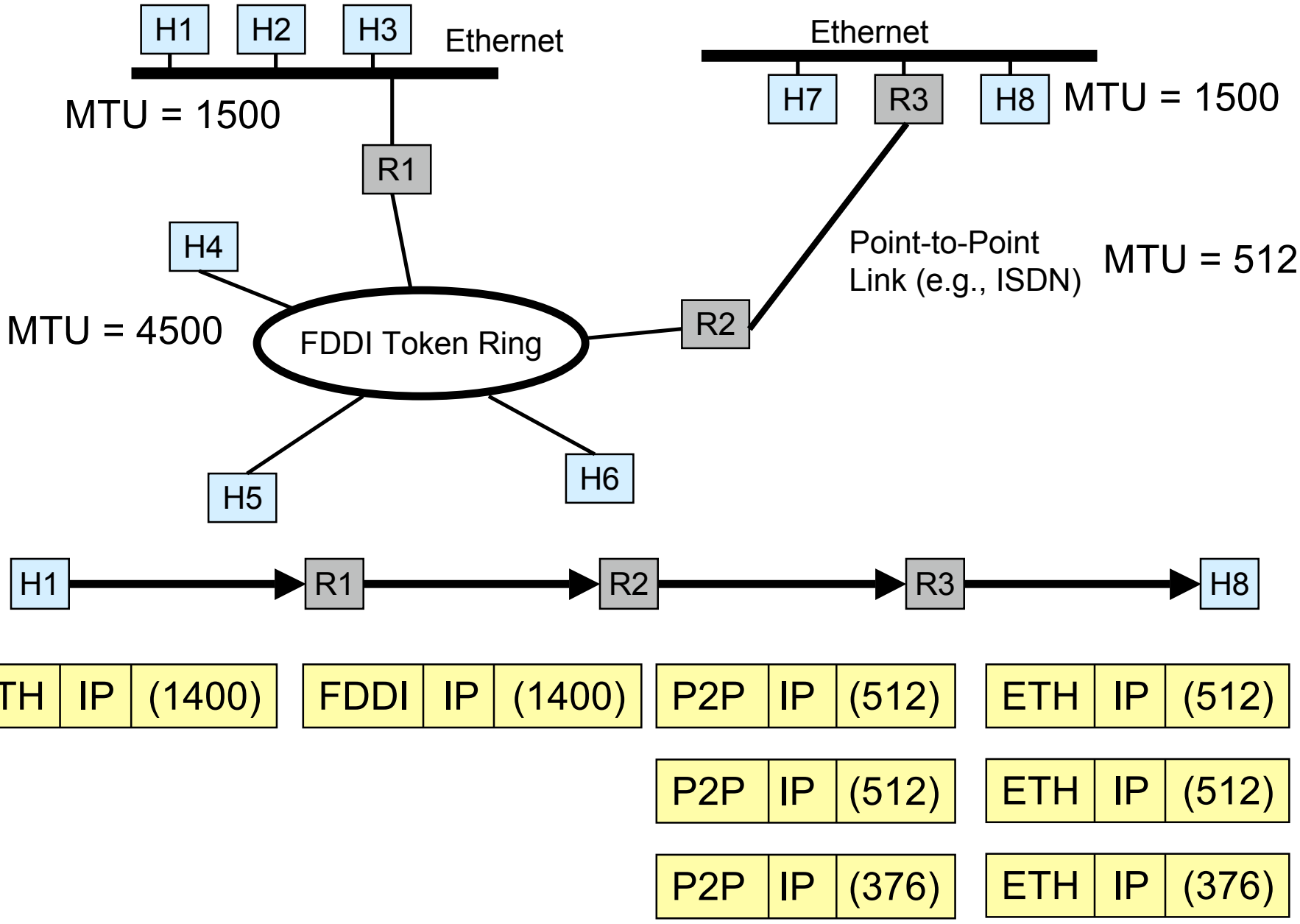
# IP Forwarding algorithm

- If I'm on the same network with destination: deliver packet to destination
- else: look up the *forwading table*:
  - if the destination network is in forwarding table: deliver packet to NextHop router
- else: deliver packet to *default router*

- Forwarding tables
  - Contain (Network #, NextHop) pairs
  - Additional information
  - Built by routing protocol

# Fragmentation and Reassembly

- Why?
  - Networks differ on maximum packet size
- How?
  - Fragment packets into pieces
  - Each fragment is itself a complete packet
  - Receiving host reassembles them
- *Maximum Transmission Unit* (MTU)
  - Path MTU is min MTU for path
  - Sender typically sends at MTU of first hop


H1
H2
H3
Ethernet
MTU = 1500
R1
H4
MTU = 4500
FDDI Token Ring
H5
H6
Ethernet
H7
R3
H8
MTU = 1500
Point-to-Point
Link (e.g., ISDN)
MTU = 512
R2
H1
R1
R2
R3
H8
ETH IP (1400)
FDDI IP (1400)
P2P IP (512)
ETH IP (512)
P2P IP (512)
ETH IP (512)
P2P IP (376)
ETH IP (376)

# Packet Fragmentation