

ELEC / COMP 177 – Fall 2013

# Computer Networking

## → Internet Protocol (IP)

Some slides from Kurose and Ross, *Computer Networking*, 5<sup>th</sup> Edition

# Upcoming Schedule

- **Presentation 2 – Privacy/Security**
  - Discuss requirements...
  - Topic Approval – **Tuesday, Nov 5<sup>th</sup>**
  - Presentations – **Nov 12<sup>th</sup> and Nov 14<sup>th</sup>**
    - Upload slides to Sakai by midnight on Nov 11<sup>th</sup>

# Upcoming Schedule

- **Project 3 – Network Tester, Part One**
  - Work day: Tuesday Oct 29<sup>th</sup>
  - **Due Thursday, October 31<sup>st</sup> by 11:55pm**

# Network Layer – IP

# Why not just use Ethernet?

- Most computer systems use Ethernet networking
- Ethernet provides facilities to
  - Locate computers
  - Forward packets directly
  - Prevent loops
  - ...
- What are the drawbacks of Ethernet for global communication?

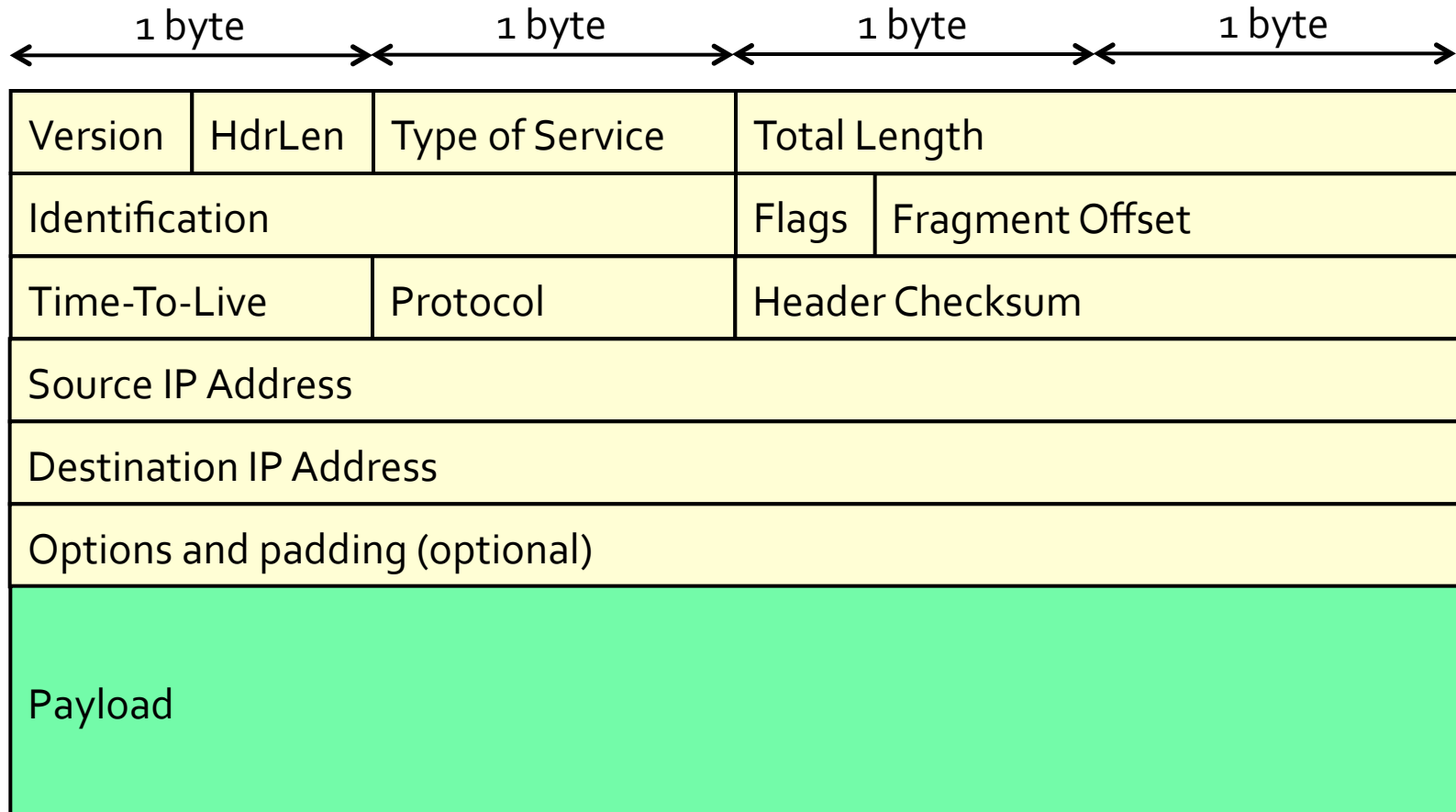
# Ethernet Drawbacks

- Locating computers
  - Do we really want to broadcast across the Internet?
- Preventing loops
  - Do we really want to rebuild an Internet-wide spanning tree whenever the topology changes?
  - Do we really want packets to live forever if loops remain?
- Unreachable computers
  - What happens if the destination is unreachable?
  - I.e., it doesn't exist, is turned off, is broken, ...

# The Internet Protocol

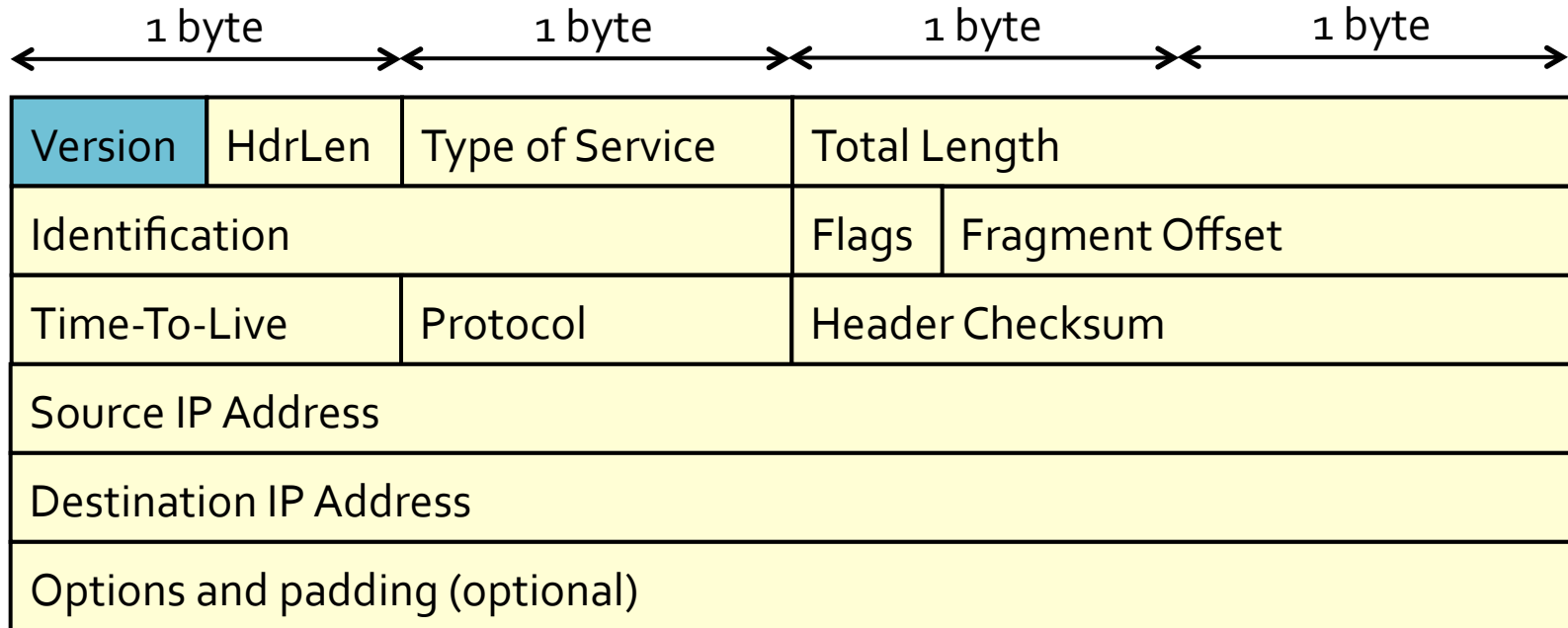
- Datagram
  - Each packet is individually routed
  - Packets may be fragmented or duplicated
    - Due to underlying networks
- Connectionless
  - No guarantee of delivery in sequence
- Unreliable
  - No guarantee of delivery
  - No guarantee of integrity of data
- Best effort
  - Only drop packets when necessary
  - No time guarantee for delivery

# An IP Datagram



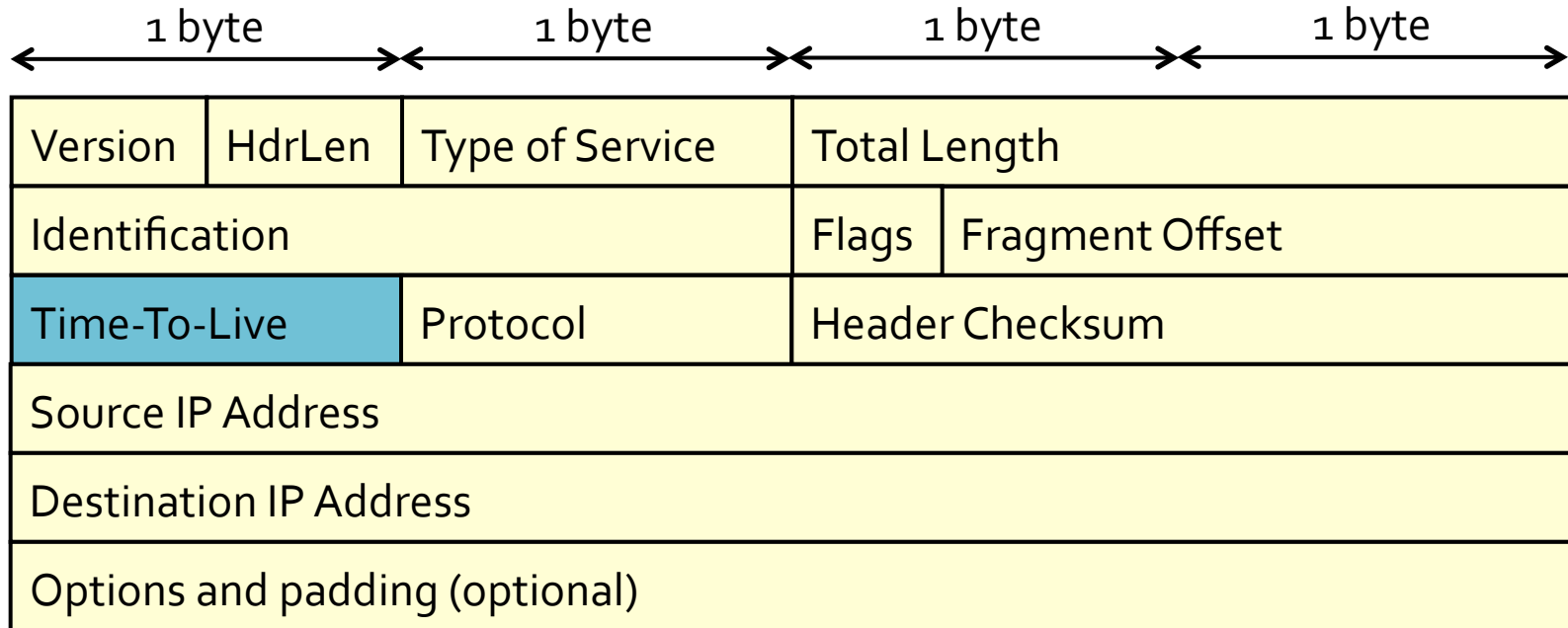


# IP Version



- IPv4 or IPv6
  - Also other, uncommon, options

# Time-To-Live

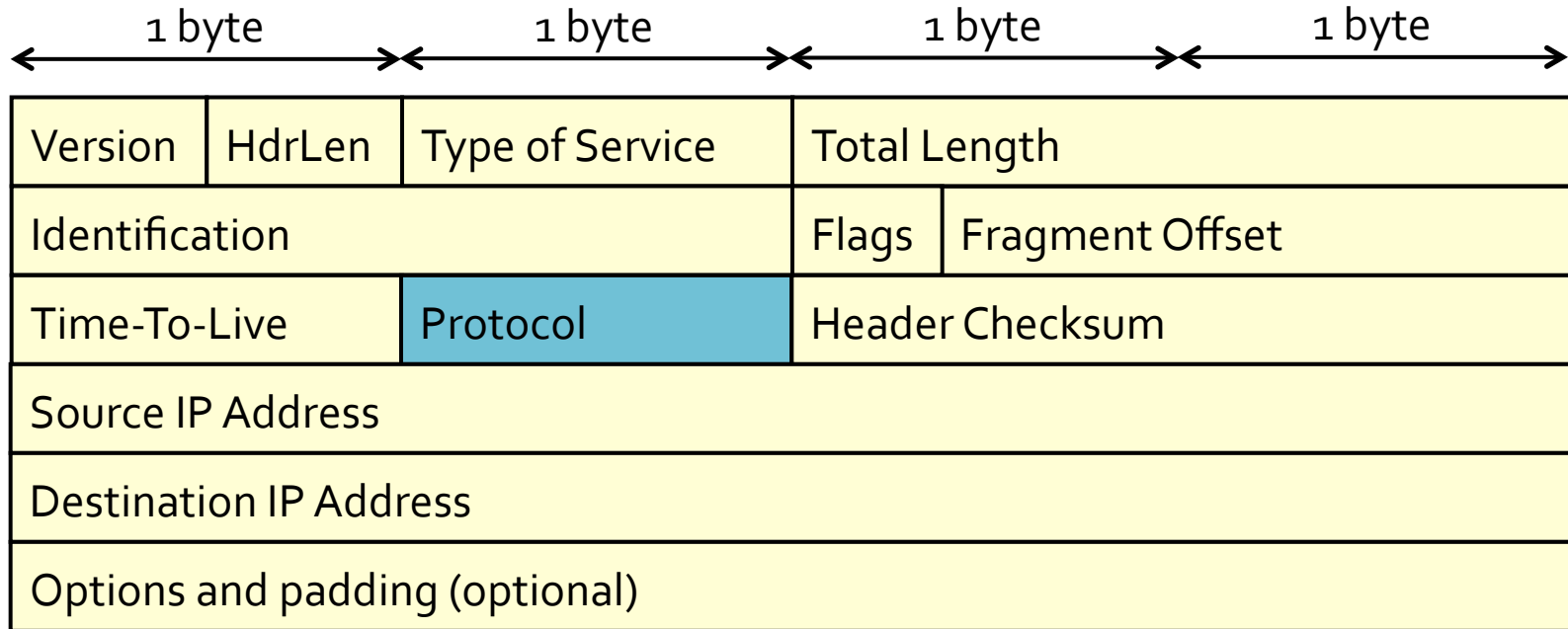


- “Hop count” – decrement each hop
- Discard datagrams with 0 TTL

# IP: Time-to-Live

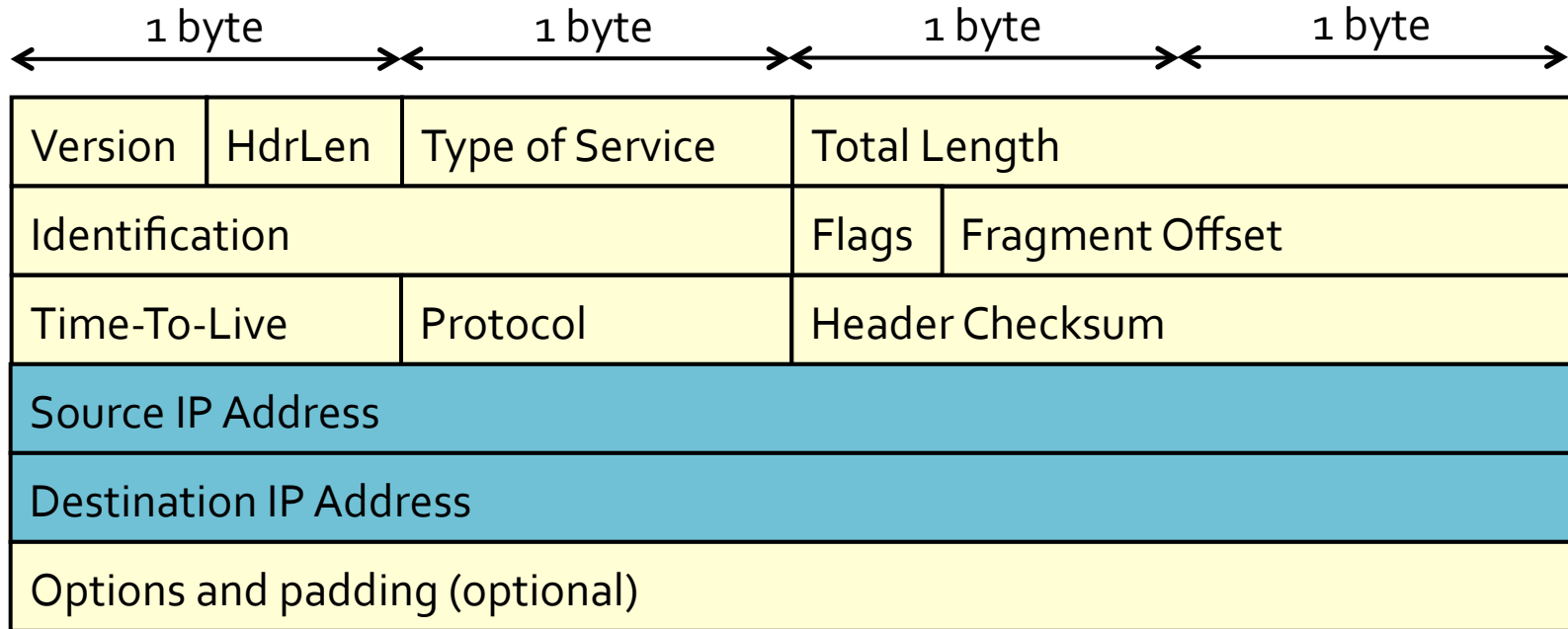
- Sender sets a TTL value for each datagram
- Each router decrements the TTL
- When the TTL reaches 0
  - The router drops the datagram
  - The router sends an ICMP error (more later) to the sender
- Effectively a “maximum hop count”
- **Why is this useful / necessary?**

# Protocol



- What is encapsulated in this IP datagram?
  - 1 = ICMP, 6 = TCP, 17 = UDP, etc...

# IP Addresses

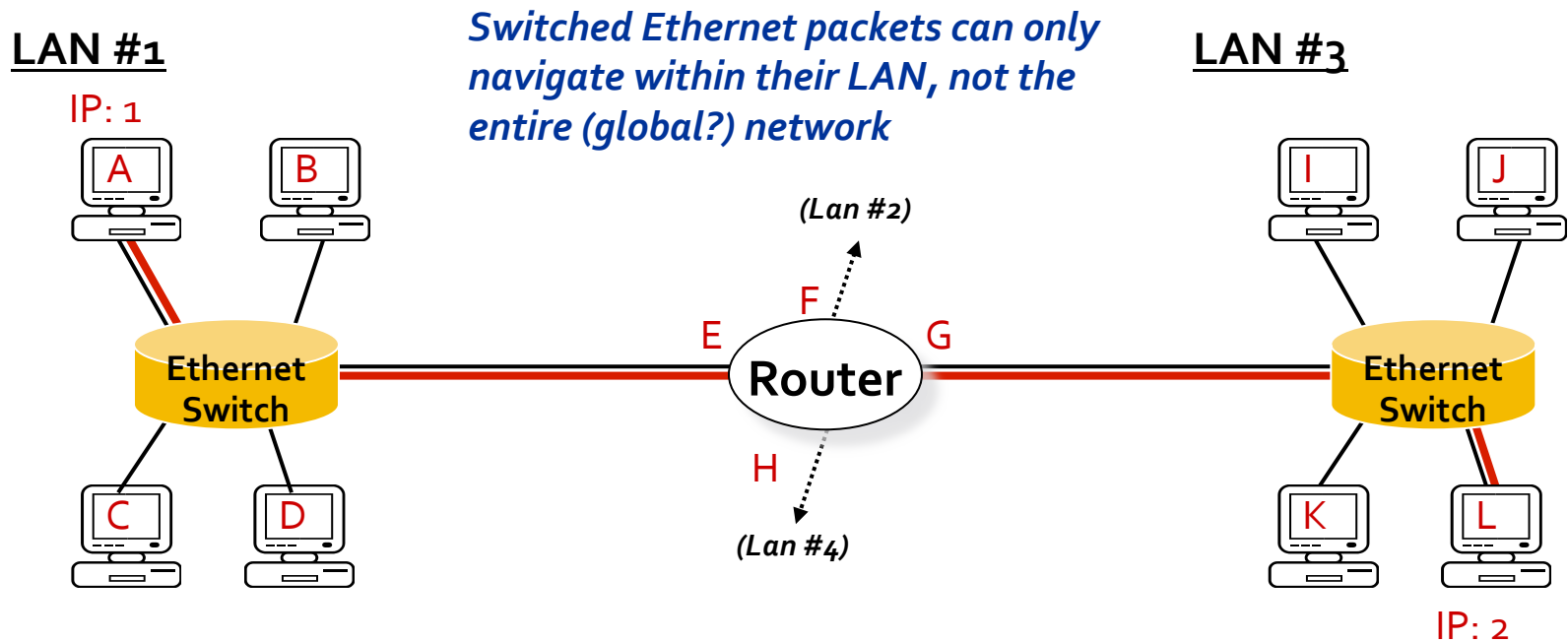


- IP address of source and destination

# IP encapsulated in Ethernet

Destination MAC Address				
Destination MAC Address		Source MAC Address		
Source MAC Address				
Type (0x0800)		Version	HdrLen	Type of Service
Total Length		Identification		
Flags	Fragment Offset		Time-To-Live	Protocol
Header Checksum		Source IP Address		
Source IP Address		Destination IP Address		
Destination IP Address		Options and Padding		
Options and Padding		Payload		
Payload				
Ethernet CRC				

# Routing Between LANs



(1) A (1) transmits to L (2) using IP.  
Ethernet frame destination is router

(3) Router uses IP protocol to forward data.  
Eth: update src/dst/crc    IP: update TTL/checksum

Frame:

EDA (E)	ESA (A)	0x0800	IPDA (2)	IPSA (1)
---------	---------	--------	----------	----------

Frame:

EDA (L)	ESA (G)	0x0800	IPDA (2)	IPSA (1)
---------	---------	--------	----------	----------

(2) Switch forwards frame to router

(4) Switch forwards frame to destination

# IP Address Format



# IP Addresses

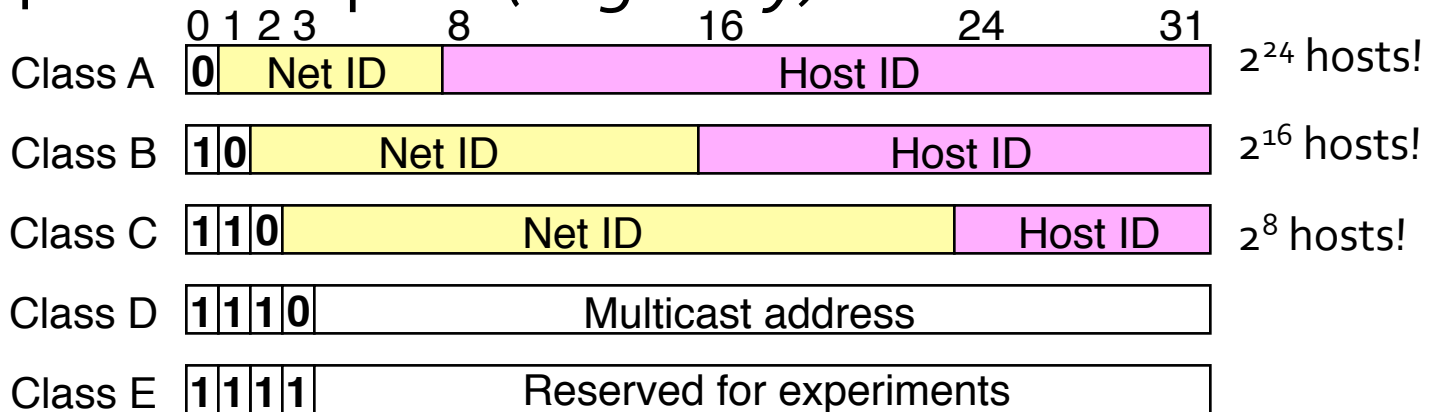
- IP version 4 addresses are 32-bits
  - Version 6 address are 128 bits
- Every network interface has at least one IP address
  - A computer might have 2 or more IP addresses
  - A router has many IP addresses
  - These addresses can be assigned statically or dynamically
- IP addresses are always in big-endian byte order (network byte order)
  - True in general for any integer transferred in a packet header from one machine to another
    - E.g., the port number used to identify a TCP connection

# IP Address Format

- IPv4 addresses are usually displayed in dotted decimal notation
  - Each byte represented by decimal value
  - Bytes are separated by a period
  - IP address  $0x8002C2F2 = 128.2.194.242$
- IP addresses are hierarchical
  - Address is composed of a network ID and a host ID
  - `www.pacific.edu`: `138.9.110.12`

# IP Address Structure

- IPv4 Address space (*originally*) divided into classes:



- Special IP addresses
  - Loop-back address: 127.0.0.1
  - Unrouted (private) IP addresses:
    - 10.0.0.0 – 10.255.255.255
    - 172.16.0.0 – 172.31.255.255
    - 192.168.0.0 – 192.168.255.255

# A Joke



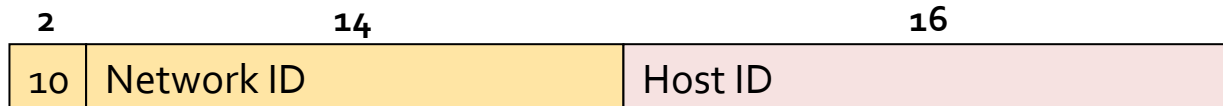
<http://xkcd.com/742/>

# Subnetting

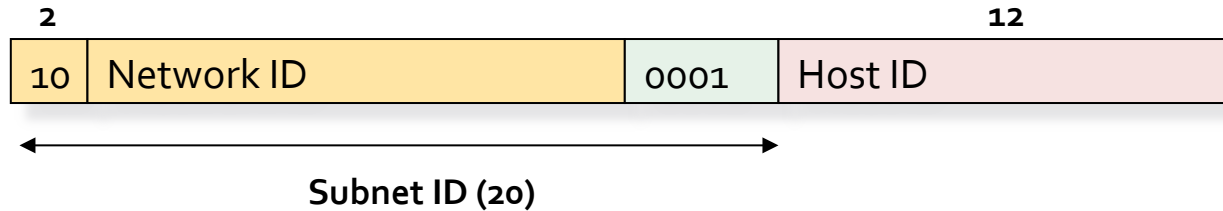
- Divide the network within an organization
  - Basically consider one Class B network to be a collection of many smaller networks
  - Size of smaller networks can be selected by the organization (don't have to be Class C sized networks)
- Internet routers don't need to know about subnetting within an organization
  - Just route their traffic to the organization

# Subnetting

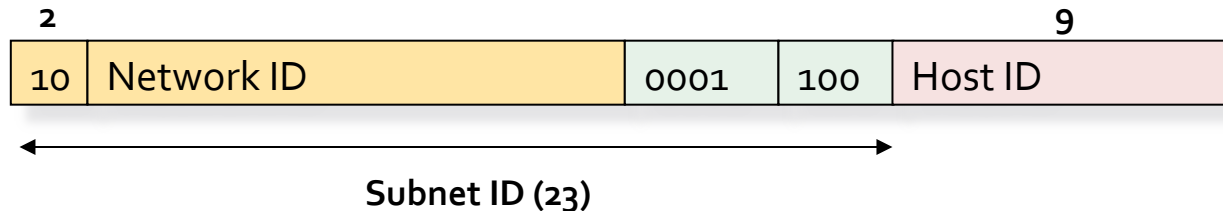
Company's Class B Network:



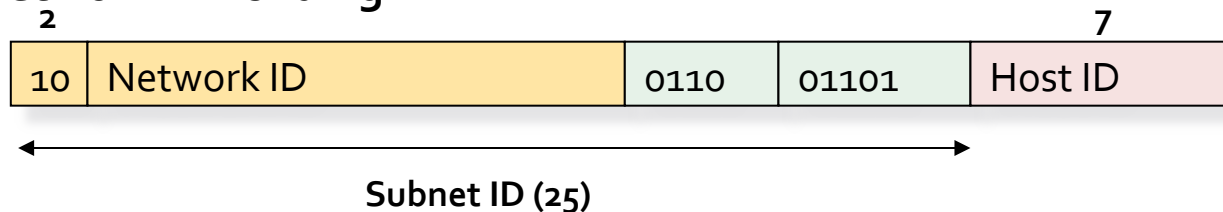
Building 1 Network:



Department 4 network in Building 1:



Floor 13 network in Building 6:



# Subnetting

- Can recursively subnet addresses down to as fine a granularity as you want
  - Almost...
  - Minimum-sized subnet has 4 addresses
    - Address 00 names the subnet
    - Address 01 and 10 names hosts
    - Address 11 is the broadcast IP address
- Subnet sizes don't have to be the same
  - One building divided by department, one by floor
  - Department/floor subnets not the same size

# Subnet Notation

- A.B.C.D/X
  - IP address of the subnet (with 0's in all host ID bits)
  - X = number of bits in the subnet network address
- Examples:
  - 17.0.0.0/8 – Apple's entire class A address space
  - 17.2.3.0/24 – A class C sized subnet in Apple's network
- Alternatively represented by subnet IP and a bit mask (netmask)
  - 17.0.0.0/255.0.0.0
  - 17.2.3.0/255.255.255.0



# Subnet Meaning

- Subnets don't have to have physical meaning
  - Although easier to keep track of if they do...
- Good subnet assignment simplifies routing for internal routers
  - All traffic for "building 1" goes through this port
  - All traffic for "department 3" goes through that port
  - ...

# Problems

- Address classes were too “rigid”
  - Class C is too small and Class B is too big in many situations
  - Inefficient use of address space
  - Leads to a shortage of addresses
- Small organizations wanted Class B networks
  - In case they grew to more than 255 hosts
  - But there are only about 16,000 Class B network IDs
- Larger organizations wanted many Class C networks
  - Separate network ID for each router link
- Every router in the Internet had to know about every network ID in every organization
  - Leads to large address tables in every router

# Classless InterDomain Routing

- CIDR introduced in 1993
  - Meant to provide more flexible routing
  - Eliminate dependences on “class” networks in routing
- “Supernetting”
  - Combine multiple contiguous networks into one larger network
  - Effectively reduces the number of entries needed in each routing table
  - Inverse of subnetting which takes one larger network and breaks it into multiple contiguous smaller networks

# CIDR Idea

- Break up IP address space into *prefixes*
  - Same idea as subnets (138.9/16)
- Each prefix has its own routing entry
  - All traffic to Pacific (138.9/16) within the Internet should be routed the same way, regardless of how Pacific subnets its address space

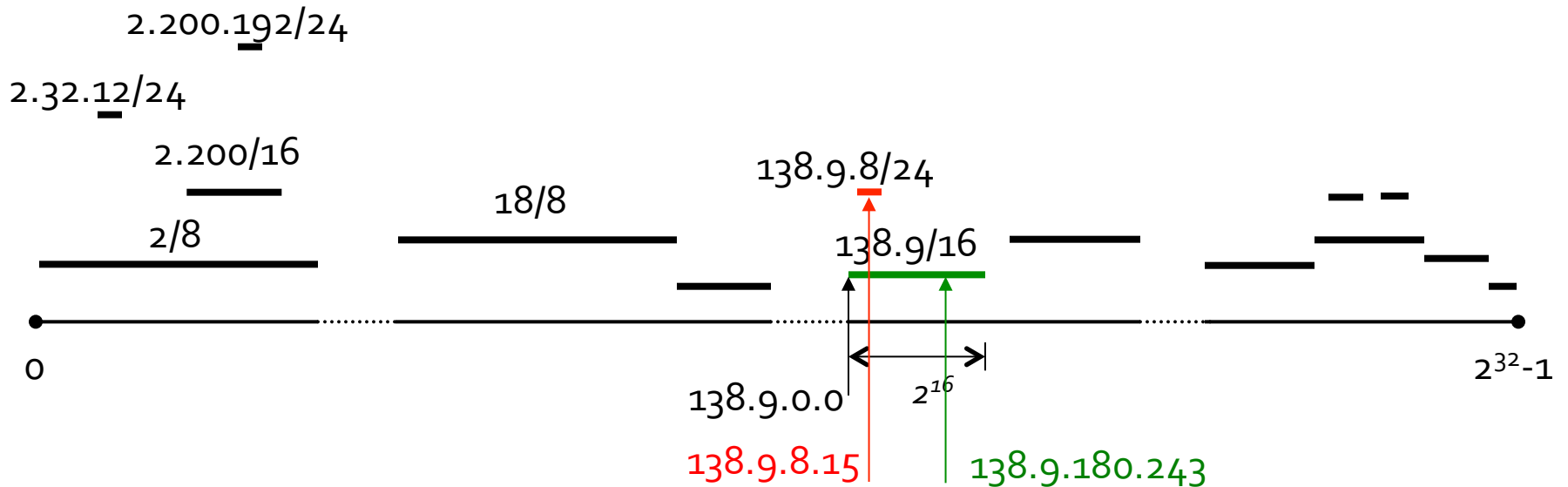
# Route Aggregation

- Example: One ISP handles traffic for two corporate networks (129.32/16 and 129.33/16)
- Aggregate route to 129.32/15 for both networks
  - External routers don't care how the ISP breaks up the network addresses internally!
- Only break them apart when necessary for the last (few) hop(s)

# What if there are holes?

- Pacific builds a 4<sup>th</sup> campus
  - 138.9.16 needs to be routed to Pacific
  - 138.9.8/24 needs to be routed to our satellite campus in Hawaii...
- Do we need to break routes up?
  - 138.9.0/20 (.0-.7)
  - 138.9.8/24 (.8)
  - 138.9.9/24 (.9), 138.9.10/24 (.10), 138.9.11/24 (.11)
  - 138.9.12/22 (.12-.15)
  - 138.9.16/20 (.16-.31)
  - 138.9.32/19 (.32-.63)
  - 138.9.64/18 (.64-.127)
  - 138.9.128/17 (.128-.255)

# IP Prefixes



- IP address space can be viewed as a number line
  - Each segment represents an aggregated route
  - Segments can overlap
- Look for smallest segment that matches the destination address : **Longest Prefix Match**

# Longest Prefix Match

- Allow more specific entries to supersede more general ones
  - 138.9.8/24
    - Route this traffic to Hawaii
  - 138.9/16
    - Route this traffic to Stockton
    - Except for addresses that match a route with a longer prefix (i.e., 138.9.8/24)
- Allows significantly more route aggregation
- Simplifies things if companies move (physically or to another ISP) their block of IP addresses



# IP Address Classes

- CIDR makes address classes less important
- With CIDR, routing is based on arbitrary subdivisions of the address space
  - Aggregate routes into largest possible group
  - Use longer prefixes to deal with exceptions
- Routing
  - Routers use longest prefix matching to determine routes
  - No longer deal with exact matches on class network IDs

## ARIN WHOIS Database Search

Relevant Links: [ARIN Home Page](#) [ARIN Site Map](#) Training: [Querying ARIN's WHOIS](#)

Search ARIN WHOIS for: 138.9.1.21

Submit

OrgName: University of the Pacific  
OrgID: **UNIVER-95**  
Address: 3601 Pacific Ave.  
City: Stockton  
StateProv: CA  
PostalCode: 95211  
Country: US

NetRange: **138.9.0.0 - 138.9.255.255**  
CIDR: 138.9.0.0/16  
NetName: **UOP**  
NetHandle: **NET-138-9-0-0-1**  
Parent: **NET-138-0-0-0-0**  
NetType: Direct Assignment  
NameServer: NS1.PACIFIC.EDU  
NameServer: NS2.PACIFIC.EDU  
Comment:  
RegDate: 1990-01-17  
Updated: 2007-09-07

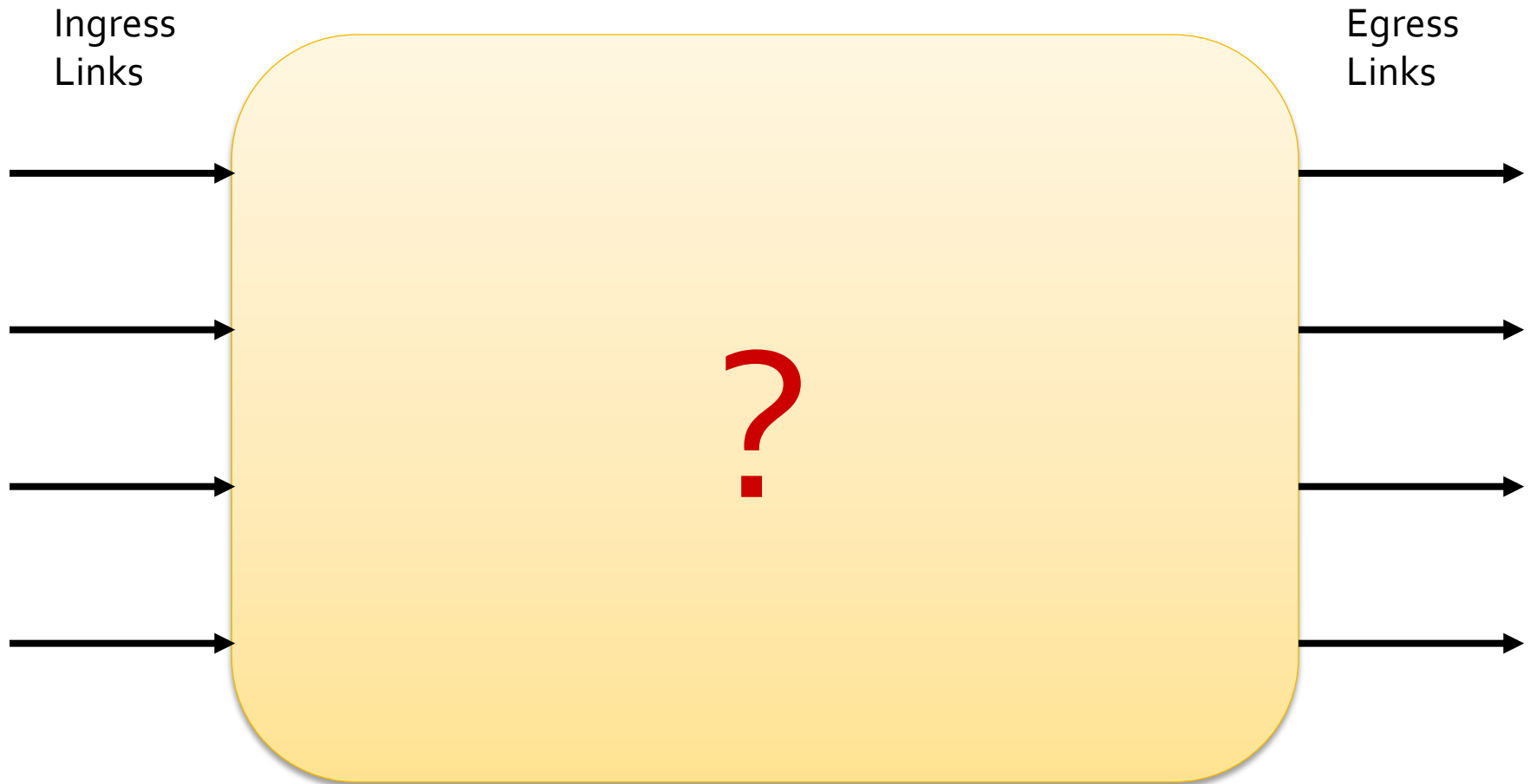
RAbuseHandle: **DAVEA-ARIN**  
RAbuseName: Lundy, Dave A.  
RAbusePhone: +1-209-946-3951  
RAbuseEmail: dlundy@pacific.edu

RTechHandle: **EES7-ARIN**  
RTechName: Escalante, Edgar  
RTechPhone: +1-209-946-3190  
RTechEmail: eescalante@pacific.edu

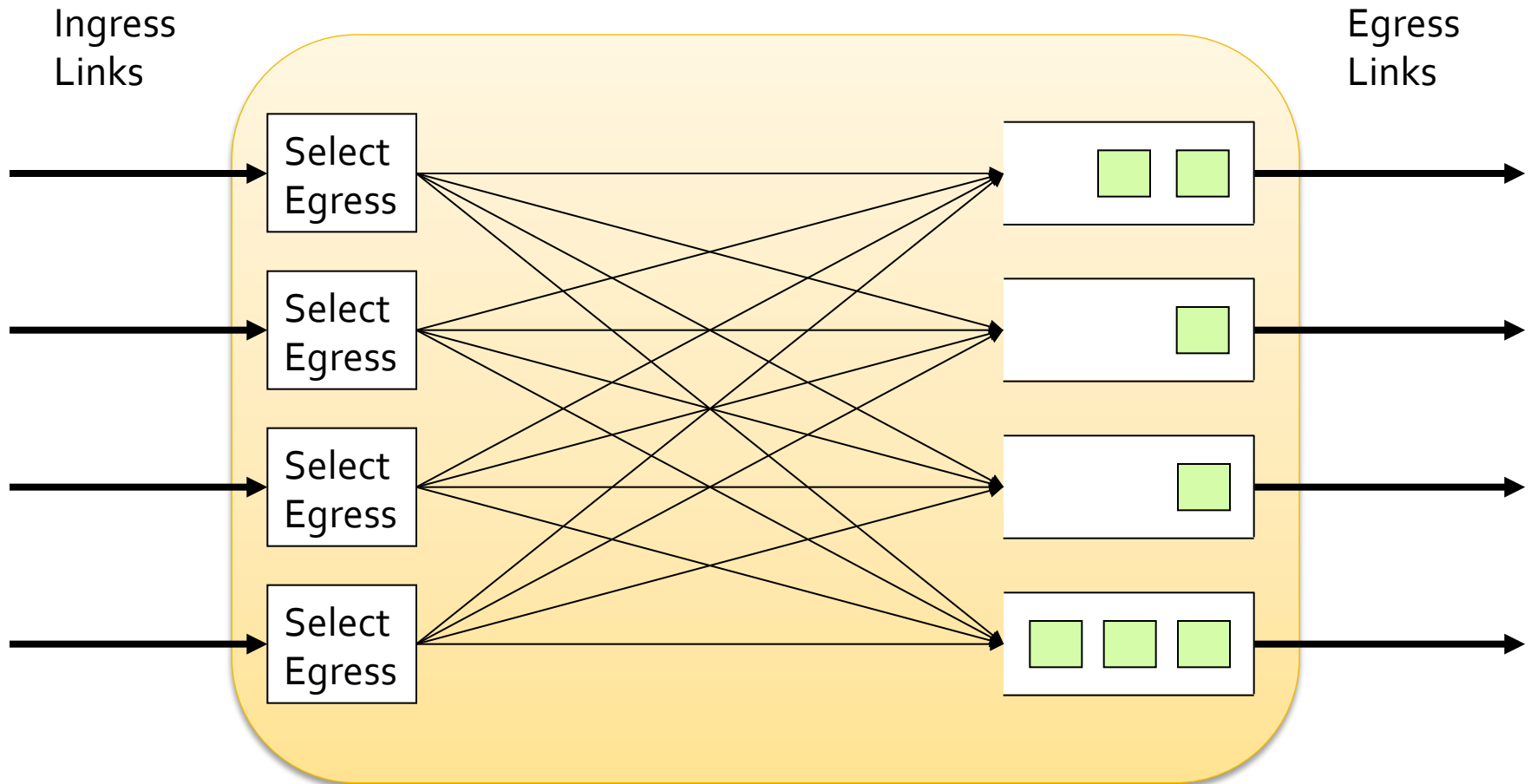
OrgTechHandle: **DAVEA-ARIN**  
OrgTechName: Lundy, Dave A.  
OrgTechPhone: +1-209-946-3951  
OrgTechEmail: dlundy@pacific.edu

# Router Operation

# What's inside a router?

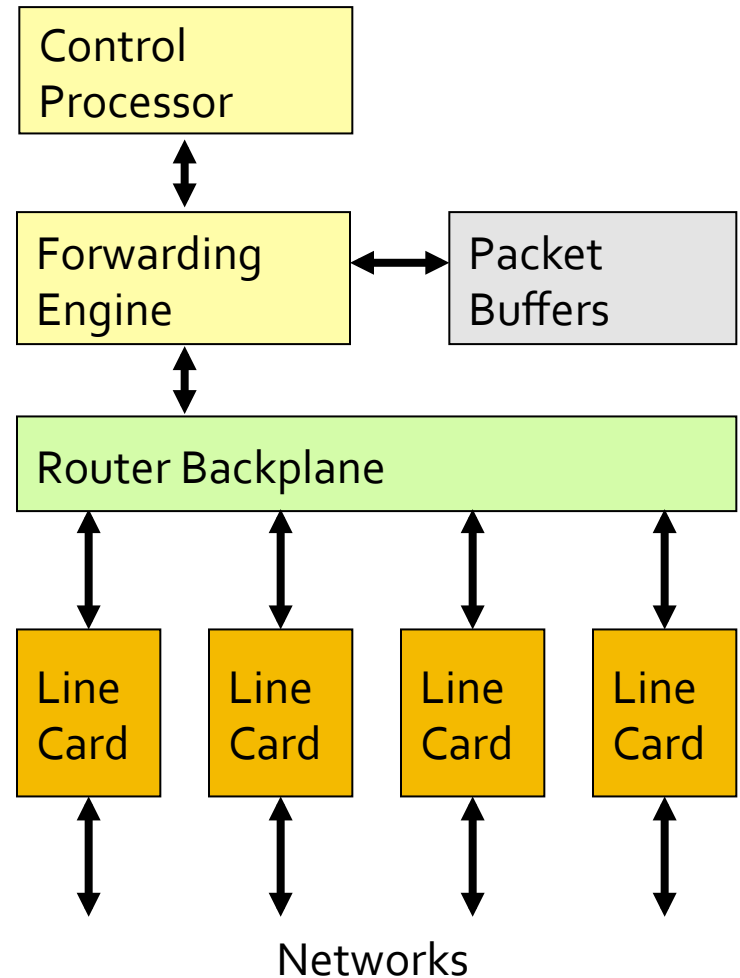


# Simplified model of a router



# Basic Router Components

- Key Modules
  - Network Interface
  - Packet processing
  - Packet buffering
  - Packet switching
- Processing and buffering can be centralized or decentralized



# Packet Processing

- What does a router need to do?
- Driven by protocols
  - Ethernet
  - IP
  - ARP
  - ICMP
  - Transport: TCP, UDP, etc.

# On packet arrival...

- Processing
  - Buffer packet?
  - Determine protocol (e.g., IP vs. ARP)
  - Verify checksum, validate the packet, etc.
  - Collect statistics?
- What's next in the "common" (valid IP packet) case?
  - Select egress link



# Selecting an Egress Link

- Forwarding table lookup
  - Longest prefix match
  - Determine next hop IP address and egress link
- **What if no match?**
- **Is this sufficient to route the packet to an output queue?**

Prefix	Next Hop	Port
63/8	128.34.12.1	3
128.42/16	128.34.12.1	3
156.3/16	128.36.21.1	2
156.3.224/19	128.36.129.1	1
128.42.96/20	128.37.37.1	4
128.42.128/24	128.36.129.1	1
128.42.160/24	128.36.21.1	2

# Updating the Destination Address

- ARP table lookup
  - Exact match on next hop IP address
  - Determine next hop MAC address
- **What if no match?**

IP	MAC
128.34.12.1	0C:FF:63:82:44:01
128.36.21.1	04:32:11:44:82:60
128.36.21.18	10:44:82:82:44:07
128.37.37.37	08:82:82:44:16:32
128.34.12.14	20:33:71:28:15:70
128.36.21.42	14:93:29:22:15:28

# Generating ARP Requests

- Broadcast on output port
  - Ask for MAC address of next hop IP address
- Wait for reply
  - **What do you do with the packet?**
  - **How long should you wait? (tradeoffs?)**
- Receive reply
  - Update ARP table
  - Packet continues along forwarding path

# Receiving ARP Requests

- Does the IP address match the IP address of the interface that received the ARP request?
  - Another system is trying to determine your MAC address
  - Respond with the appropriate ARP reply on the same interface
- **Should ARP requests be forwarded if they aren't for the router?**

# Updating Packets

- Select egress link
- Update MAC address
- **Is it now OK to forward packet to output queue?**
  
- IP packet header must be modified
  - **What needs to be modified?**
  - **When should it be modified?**

# Buffering

- Why do packets need to be buffered?
  - Waiting for access to a resource (lookup table, switch, etc.)
  - Waiting for an ARP reply
  - ...
- What happens when buffers get full?
  - Packets have to be dropped
- How large do buffers need to be?
  - Statistical multiplexing

# Error Handling

- ICMP Messages
  - Notify sender of errors
- Common error types
  - Host/network unreachable
    - No ARP response
  - Time exceeded
    - TTL decremented to zero
  - No route to host
    - No entry in routing table

# Routing Algorithms



# Two Key Network-Layer Functions

## ■ Forwarding

- Move packets from router's input to appropriate router output
- *Forwarding table*

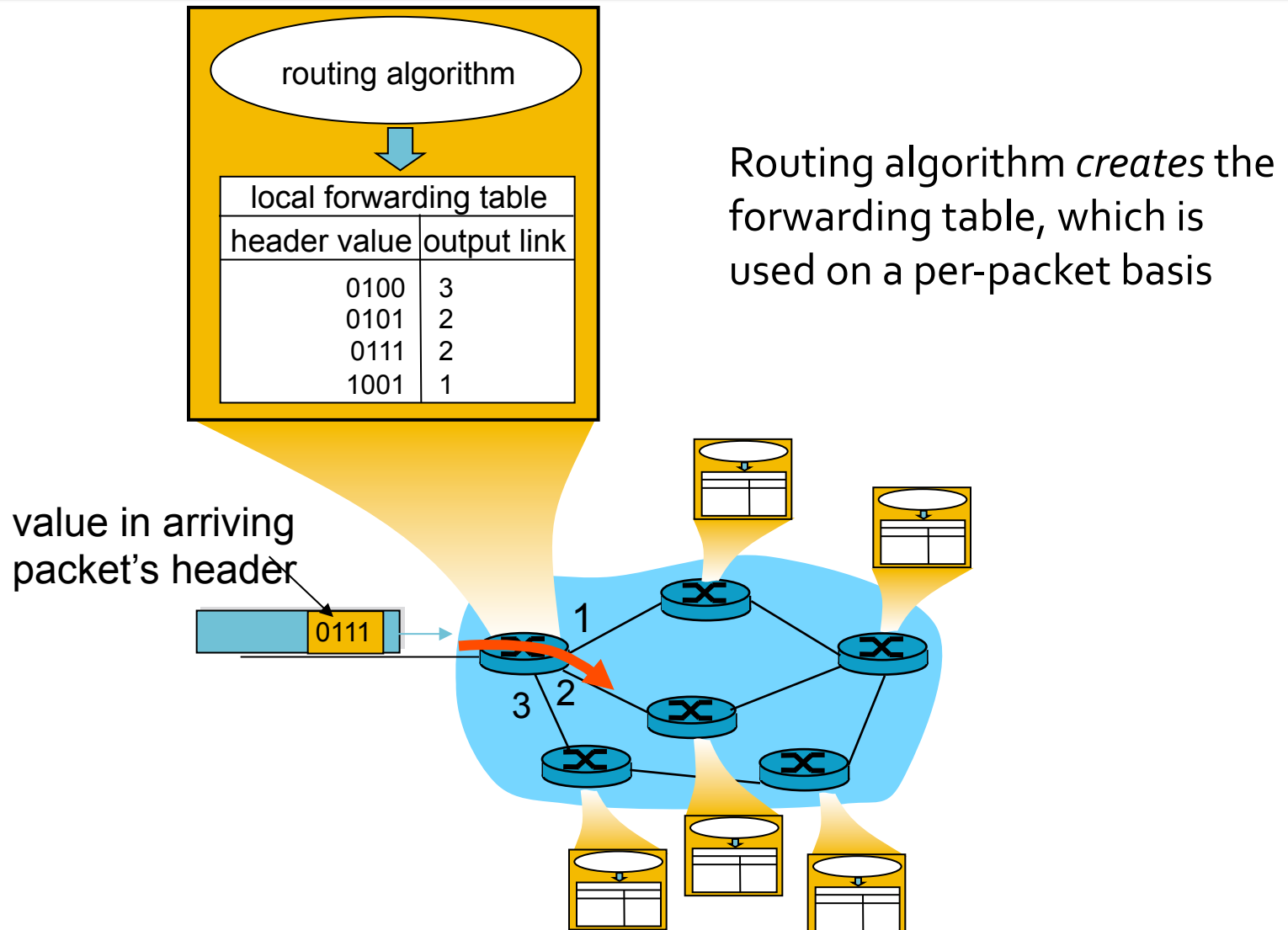
## ■ Routing

- Determine path (route) taken by packets from source to destination
- *Routing algorithms*

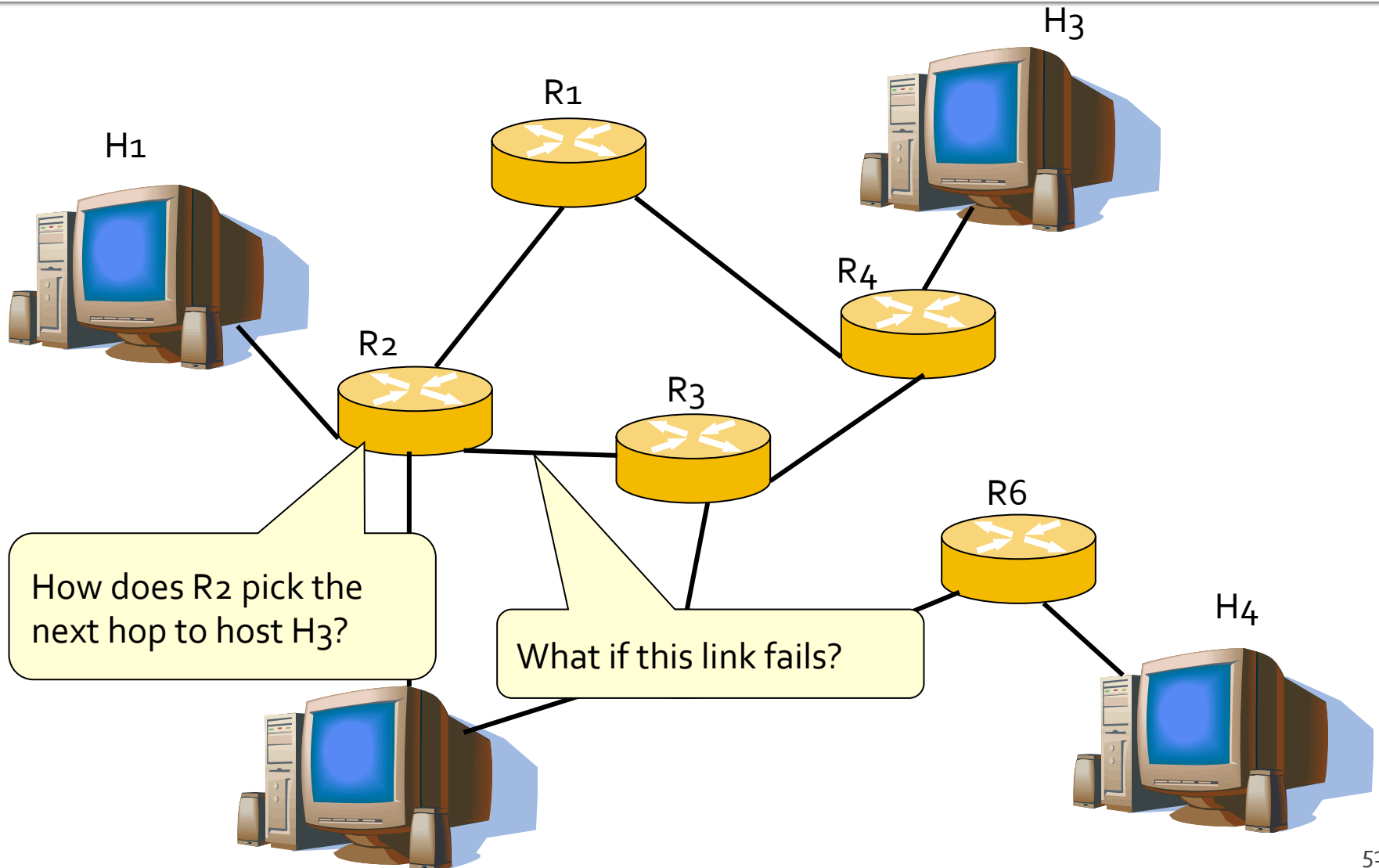
## ■ Road trip analogy:

- **Forwarding:** process of getting through single interchange
- **Routing:** process of planning trip from source to destination

# Routing versus Forwarding



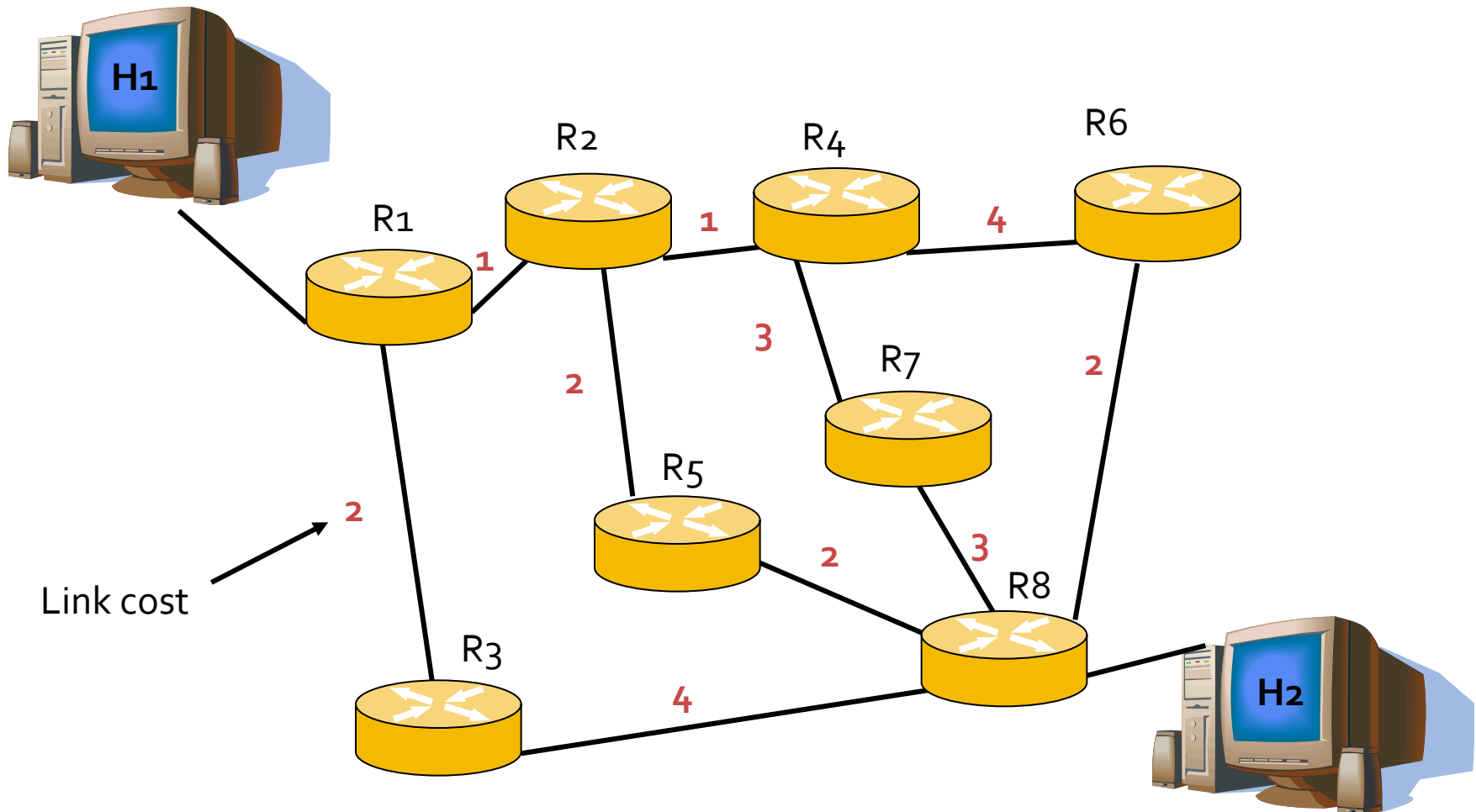
# Forwarding Table Entries



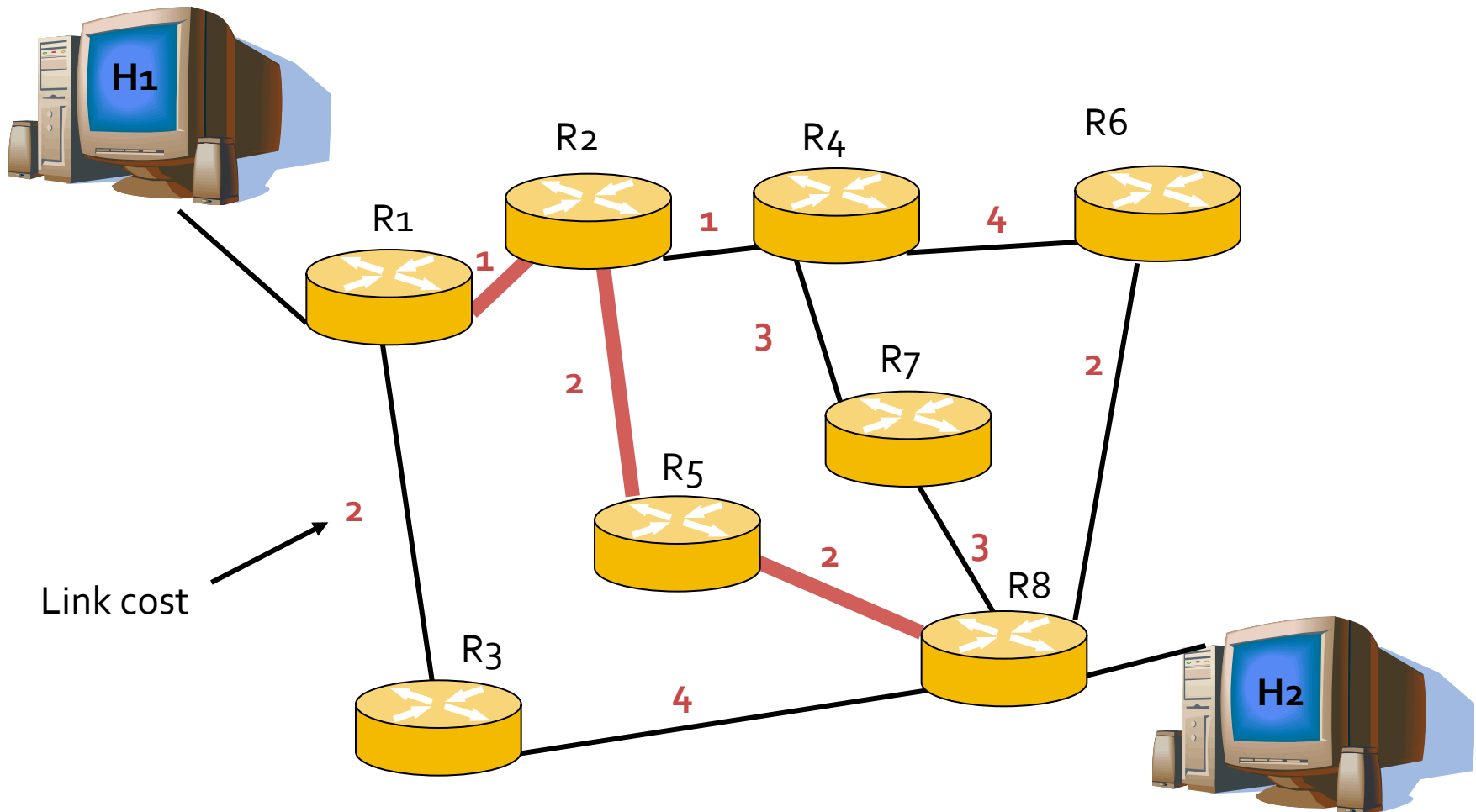
# Generating/Updating Routes

- So far, we have assumed forwarding tables are populated statically by an administrator
- In reality, they are dynamically updated
  - Faster reaction to changing network conditions
- **What makes a good route?**
  - Low delay
  - High bandwidth
  - Low link utilization
  - High link stability
  - Low cost
    - (cheaper to use ISP A than ISP B)

# Example Network



# "Best" Path



# Real Networks Are Complicated

- The Internet in 2003
  - <http://www.opte.org/maps/>

