Network Programming
Network Layer

Minchen Yu
SDS@CUHK-SZ
Spring 2026

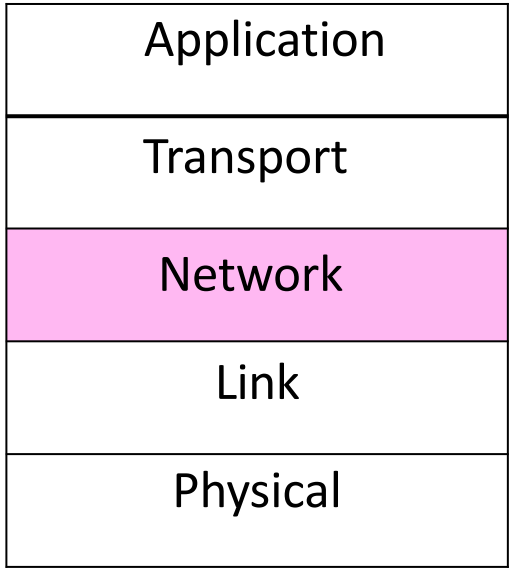
香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen
SCHOOL OF DATA SCIENCE
數據科學學院

Recall the protocol stack

5	Application	Programs that use network service
4	Transport	Provides end-to-end data delivery
3	Network	Send packets over multiple networks
2	Link	Send frames over one or more links
1	Physical	Send bits using signals

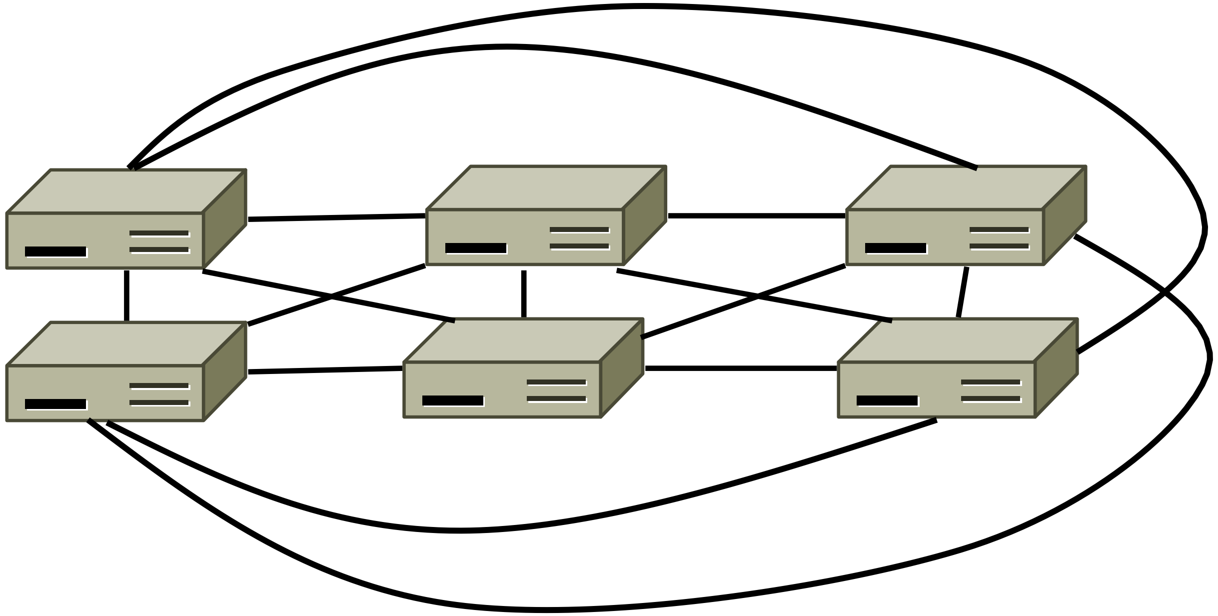
Network Layer

Goal: Get packets from source to destination, which may be separated by many hops



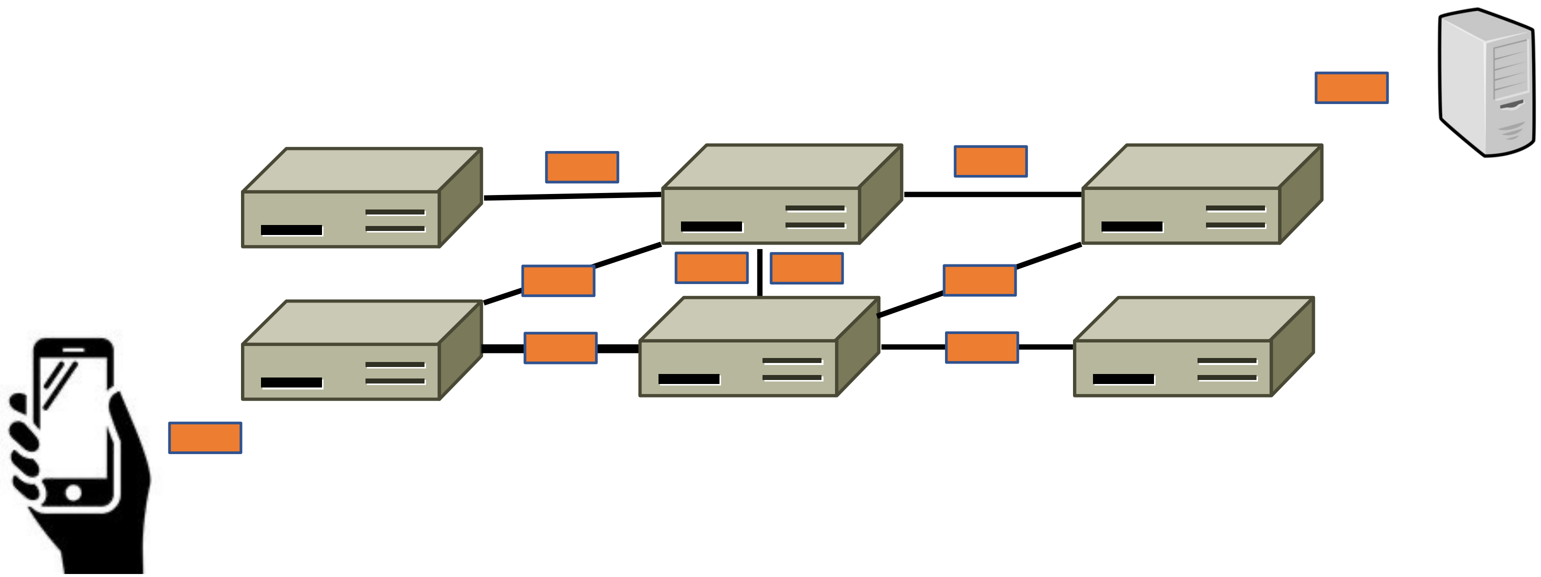
Why do we need a Network layer?

- Cannot afford to directly connect everyone
  - Cost and link layer diversity



Why do we need a Network layer?

- Cannot broadcast all packets at global scale



Why do we need a Network layer?

- Internetworking
  - Need to connect different link layer networks
- Addressing
  - Need a globally unique way to “address” hosts
- Routing and forwarding
  - Need to find and traverse paths between hosts

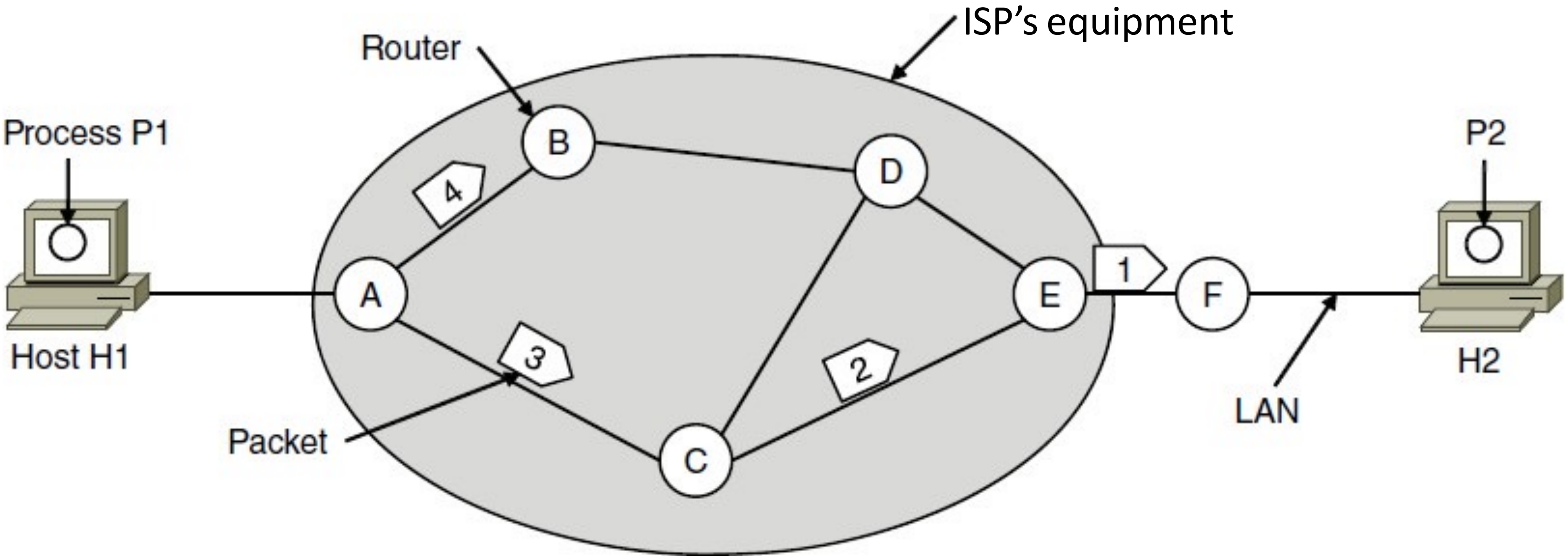
Two Network Service Models

- Datagrams, or connectionless service
  - Like postal letters
  - (IP as an example)

- Virtual circuits, or connection-oriented service
  - Like a telephone call

Datagram Model

- Packets contain a destination address; each router uses it to forward packets, maybe on different paths



Datagram Model

- Each router has a forwarding table keyed by address
  - Gives next hop for each destination address; may change

A's table (initially)

A	
B	B
C	C
D	B
E	C
F	C

Dest. Line

A's table (later)

A	
B	B
C	C
D	B
E	B
F	B

C's Table

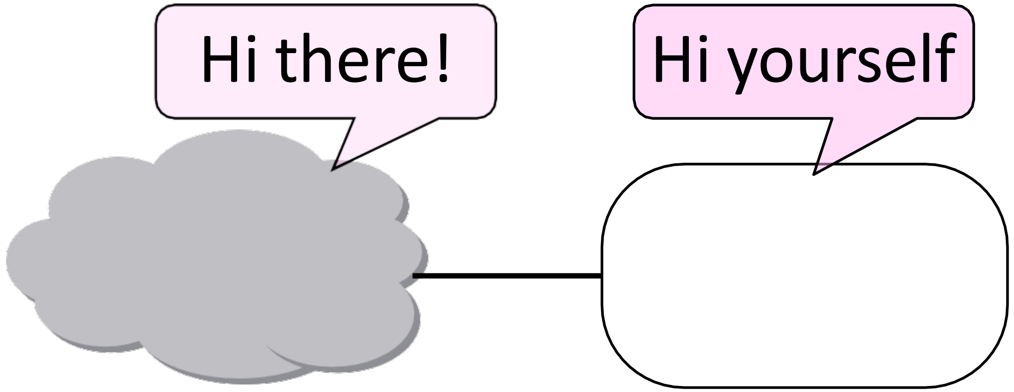
A	A
B	A
C	
D	E
E	E
F	E

E's Table

A	C
B	D
C	C
D	D
E	
F	F

Internetworking

- How do we connect different networks together?
  - This is called internetworking
  - We'll look at how IP does it



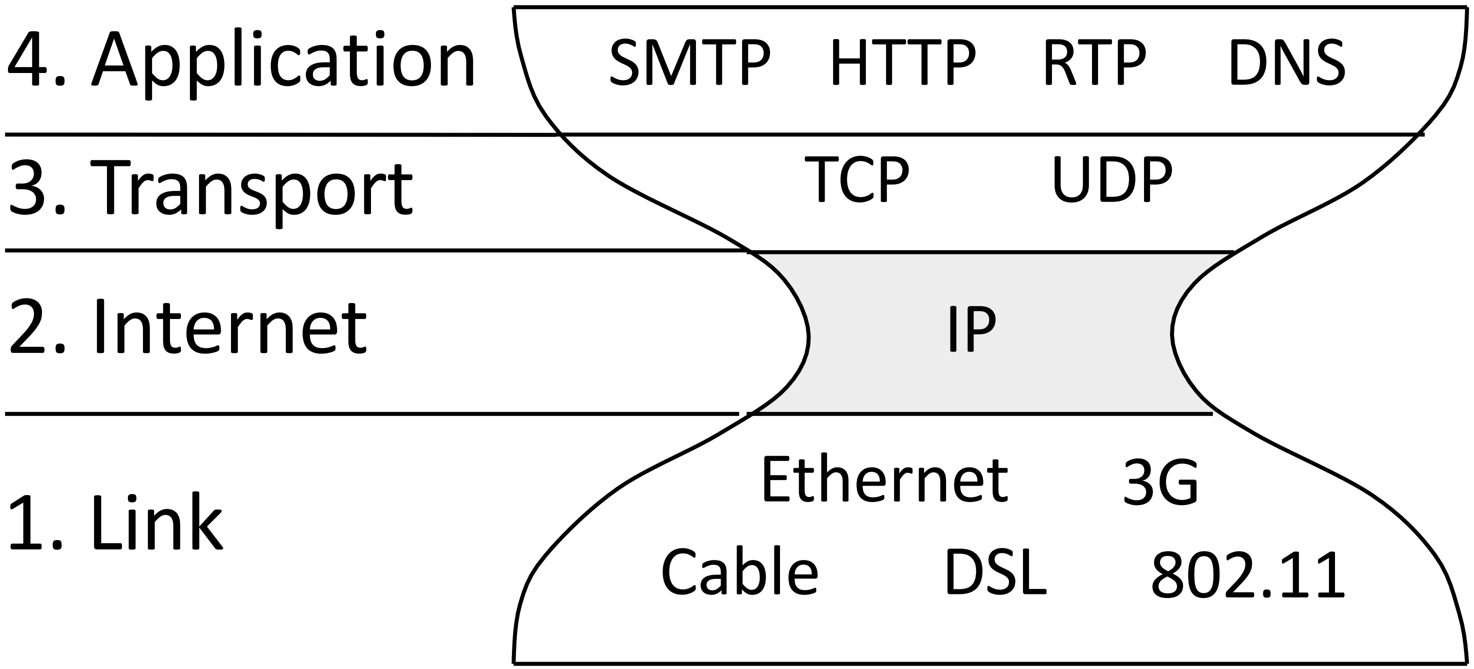
How Networks May Differ

- Lot of ways:
  - Service model (datagrams, VCs)
  - Addressing (what kind)
  - QOS (priorities, no priorities)
  - Packet sizes
  - Security (whether encrypted)

- Internetworking hides the differences with a common protocol

Internet Reference Model

- Internet Protocol (IP) is the "narrow waist"
  - Supports many different links below and apps above

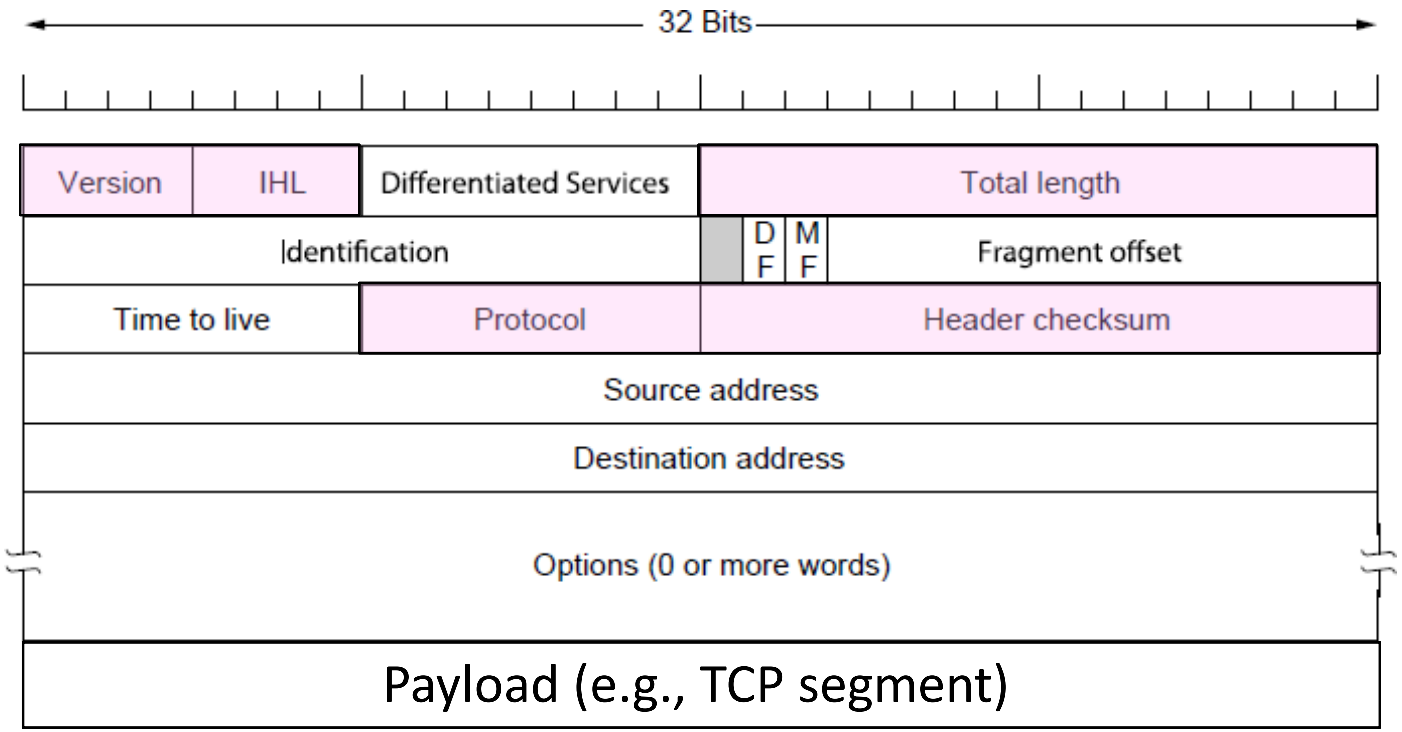


IP as a Lowest Common Denominator

- Suppose only some networks support QOS or security etc.
  - Difficult for internetwork to support
- Pushes IP to be a “lowest common denominator”
  - Asks little of lower-layer networks
  - Gives little as a higher layer service

IPv4 (Internet Protocol)

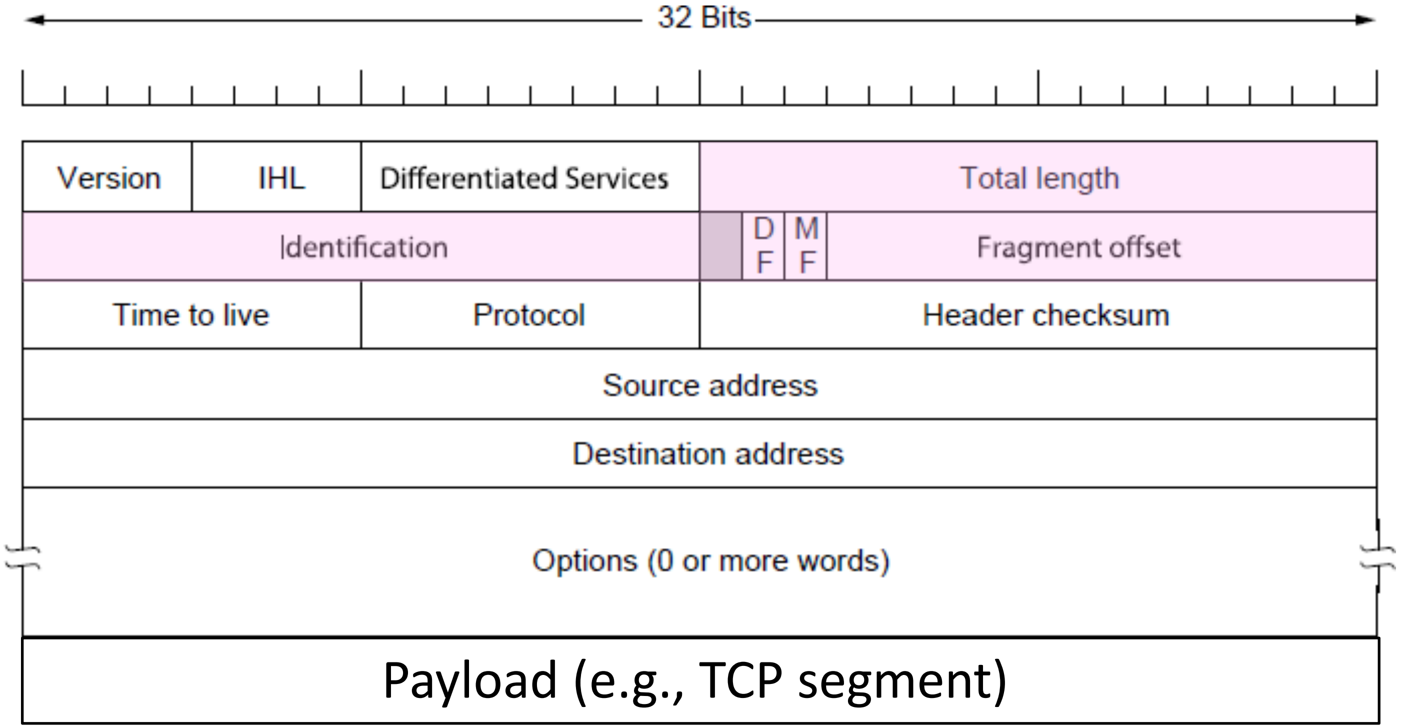
- Various fields to meet straightforward needs
  - Version, Header (IHL), Total length, Protocol, and Header Checksum



how much overhead?
- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

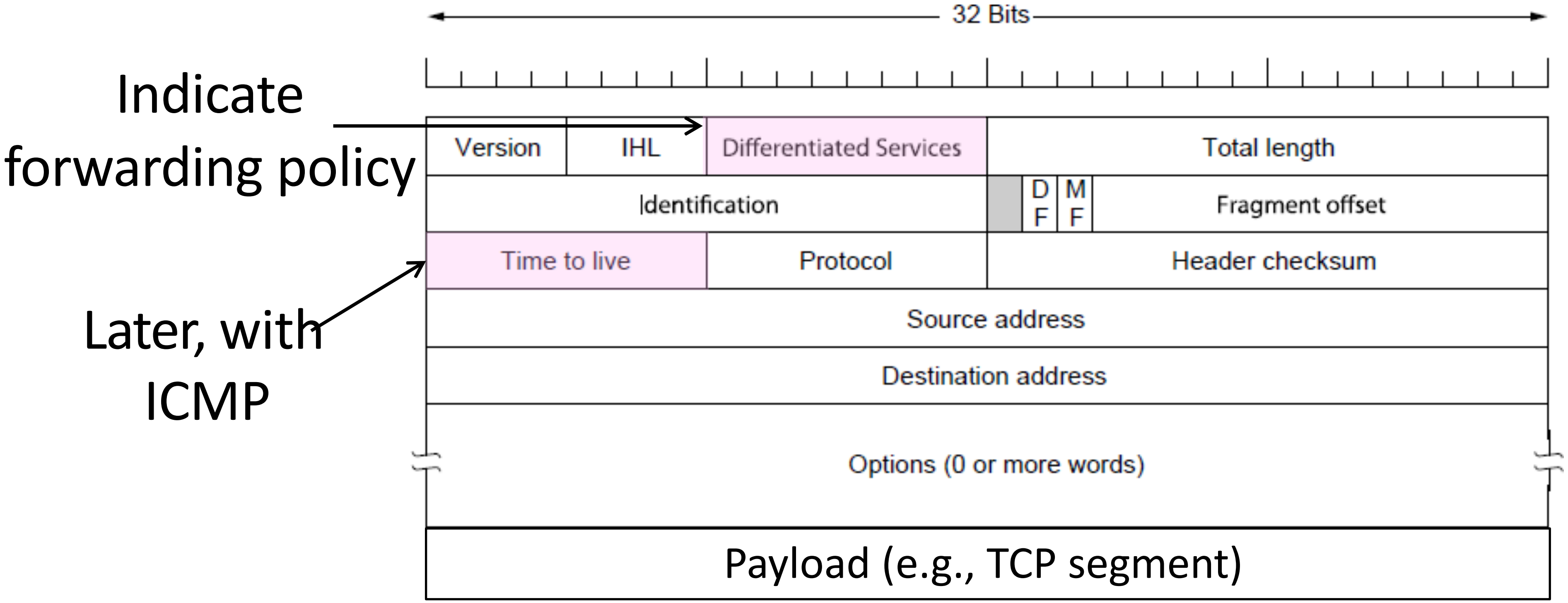
IPv4

- Some fields to handle packet size differences (later)
    - Identification, Fragment offset, Fragment control bits



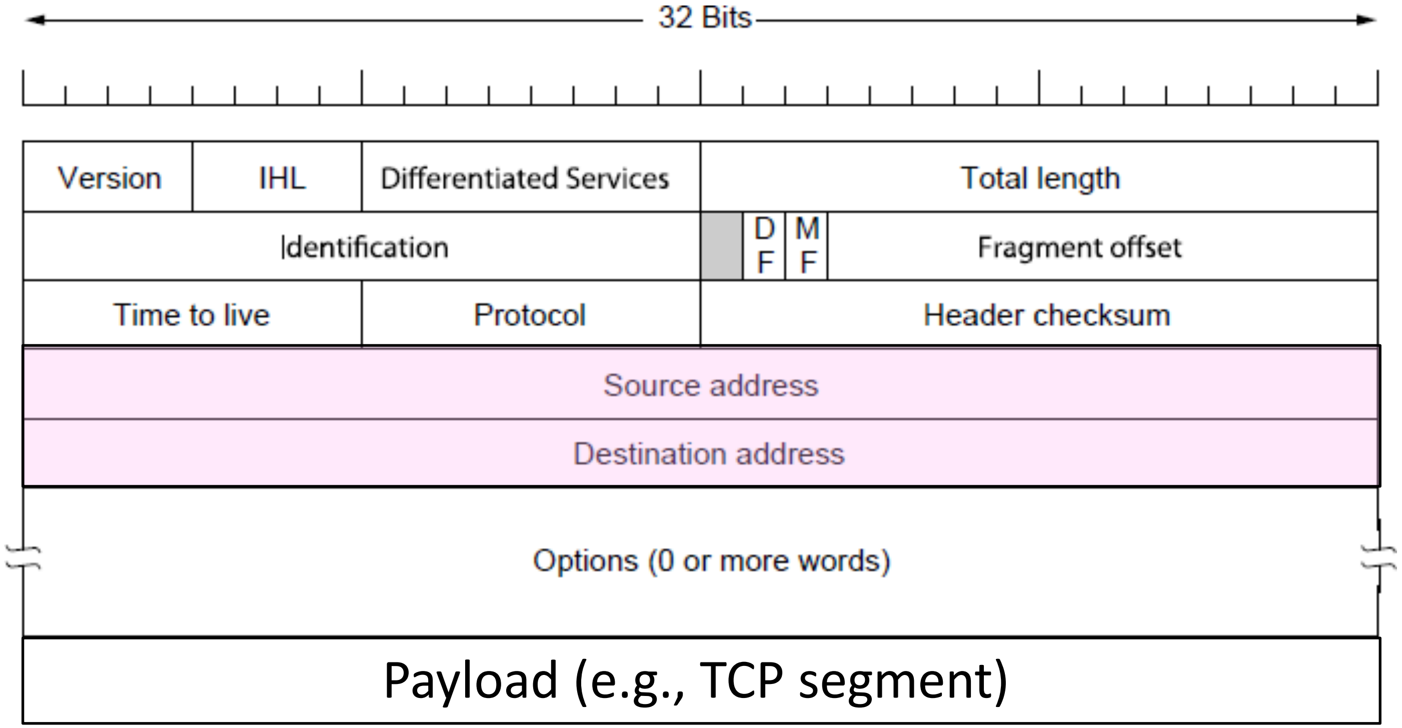
IPv4

- Other fields to meet other needs (later, later)
  - Differentiated Services, Time to live (TTL)



IPv4

- Network layer of the Internet, uses datagrams
  - Provides a layer of addressing above link addresses (next)



IP Addresses

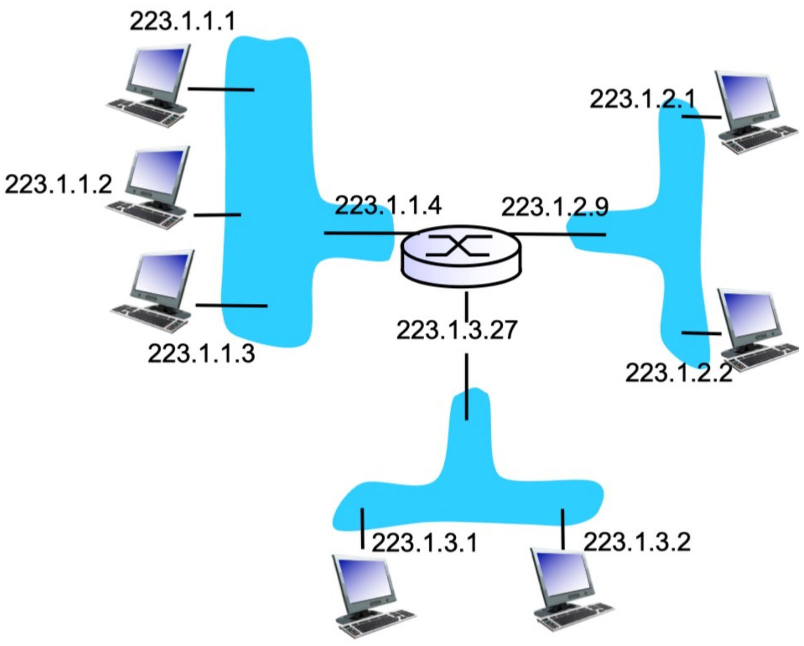
- IPv4 uses 32-bit addresses
  - IPv6 uses 128-bit addresses
- Written in “dotted quad” notation
  - Four 8-bit numbers separated by dots

$$\underbrace{\texttt{aaaaaaaa}}^{\text{8 bits}}\underbrace{\texttt{bbbbbbbb}}^{\text{8 bits}}\underbrace{\texttt{cccccccc}}^{\text{8 bits}}\underbrace{\texttt{dddddddd}}^{\text{8 bits}} \leftrightarrow \texttt{A.B.C.D}$$

$$\texttt{00010010}\,|\,\texttt{00011111}\,|\,\texttt{00000000}\,|\,\texttt{00000001} \leftrightarrow \text{??}$$

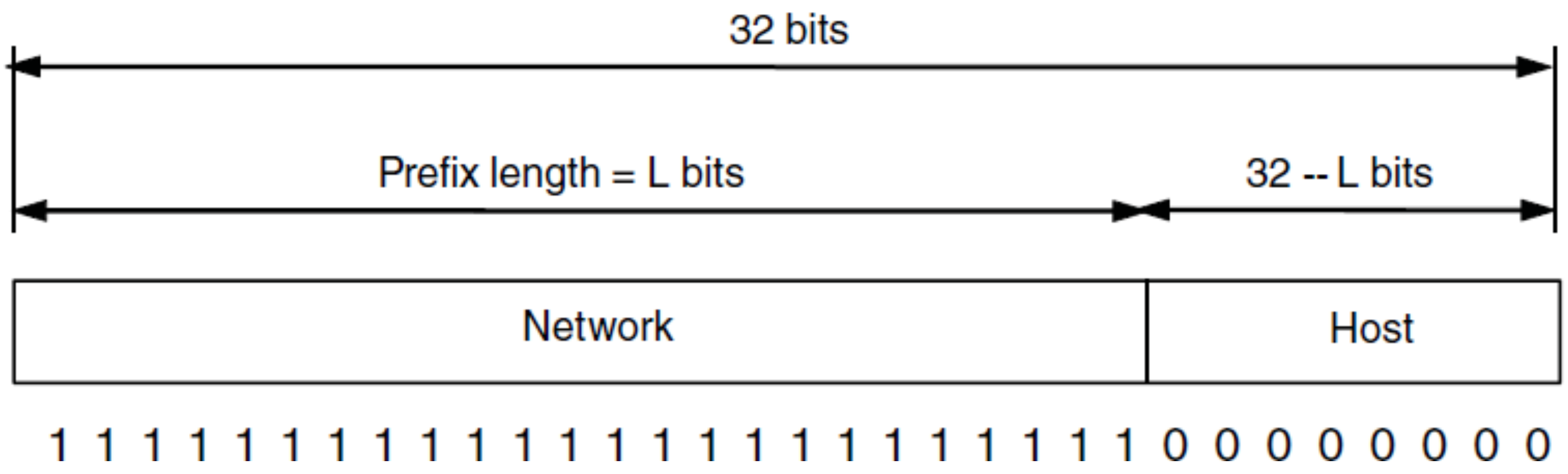
IP Addresses

- IP addresses associated with each interface
  - Interface: connection between host/router and physical link
  - A router have multiple interfaces
  - A host usually has one or two (wired, wireless)



IP Prefixes

- Addresses are allocated in blocks called prefixes (subnet, later)
  - Addresses in an L-bit prefix have the same top L bits
  - There are 2^{32-L} addresses aligned on 2^{32-L} boundary



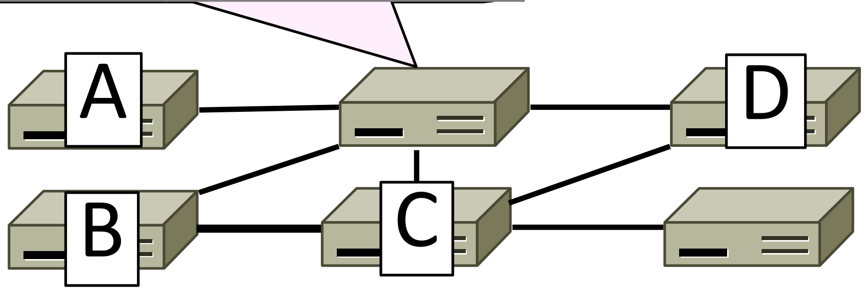
IP Prefixes

- Written in “IP address/length” notation
  - Address is lowest address in the prefix, length is prefix bits
  - E.g., 128.13.0.0/16 is 128.13.0.0 to 128.13.255.255
  - So a /24 (“slash 24”) is 256 addresses and /32 is 1 address

IP Forwarding

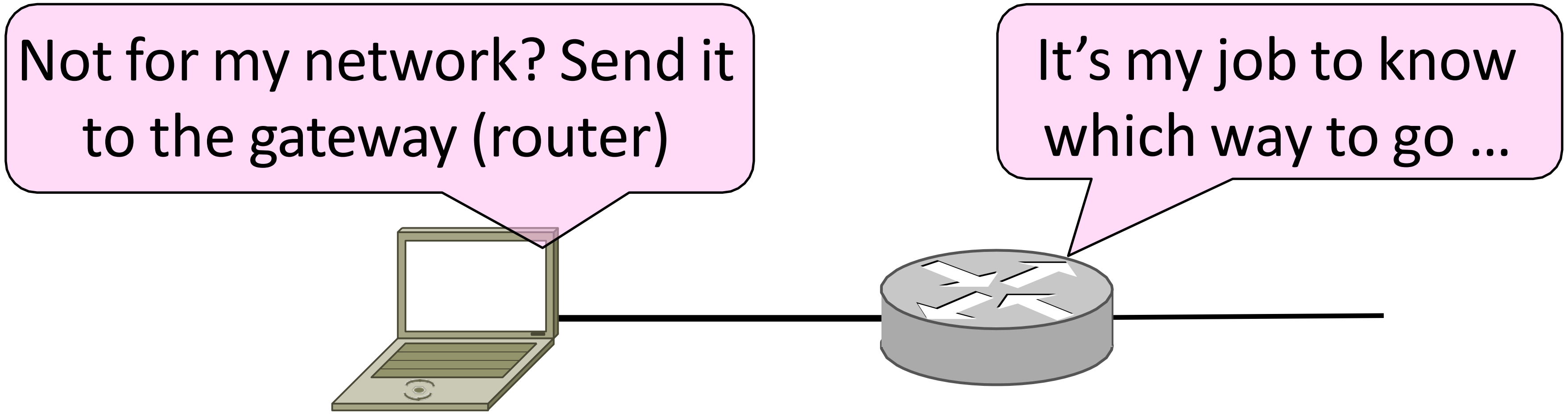
- Nodes use a table that lists the next hop for prefixes
- Lookup the destination address's prefix in the table

Prefix	Next Hop
102.24.0.0/19	D
192.24.12.0/22	B



Host/Router Distinction

- In the Internet:
  - Routers do the routing, know way to all destinations
  - Hosts send remote traffic (out of prefix) to nearest router

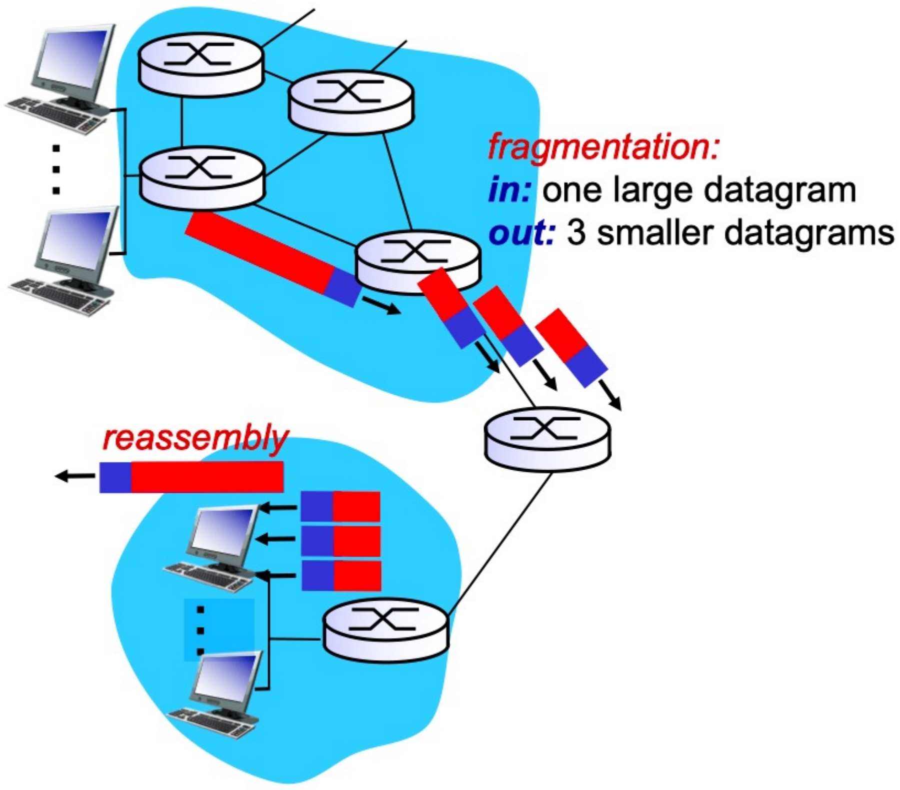


Host Networking

- Consists of 4 pieces of data:
  - IP Address
  - Subnet Mask
    - Defines local addresses
  - Gateway
    - Who (local) to send non-local packets to for routing
  - DNS Server (Later)

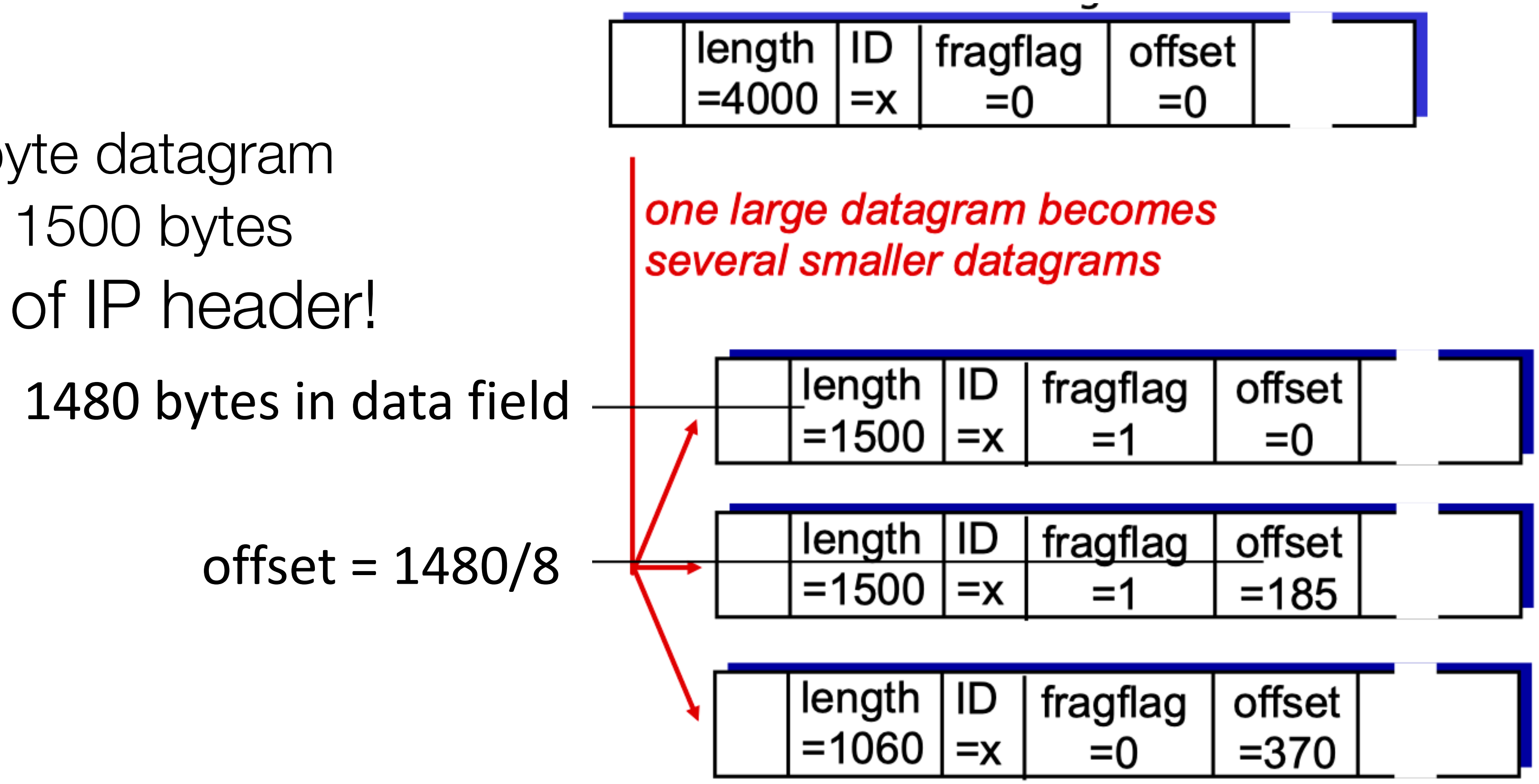
IP fragmentation

- Network links have MTU (max transfer size)
  - largest possible link-level frame
  - different link types, different MTUs
- Large IP datagram divided (fragmented) within net
  - One datagram becomes several datagrams
  - Reassembled only at final destination



IP fragmentation

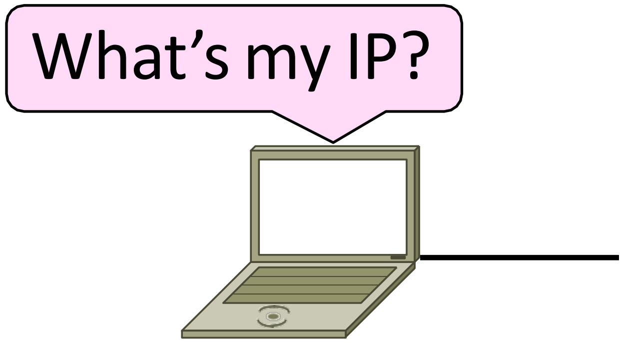
- Example
  - 4000 byte datagram
  - MTU = 1500 bytes
- 20 bytes of IP header!



Dynamic Host Configuration Protocol (DHCP)

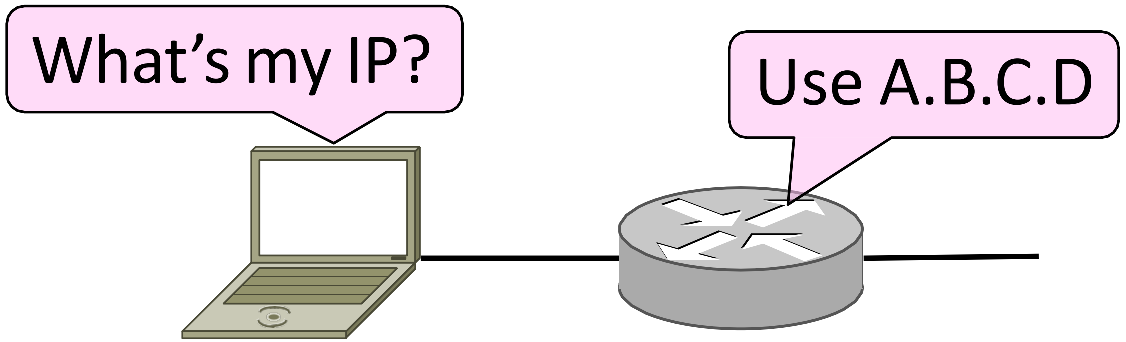
Bootstrapping

- Problem:
  - A node wakes up for the first time …
  - What is its IP address? What's the IP address of its router?
  - At least Ethernet address is on NIC



Bootstrapping

- Manual configuration (old days)
  - Can't be factory set, depends on use
- DHCP: Automatically configure addresses



DHCP: Dynamic Host Configuration Protocol

- Invented around 1993, widely used now
- It leases IP address to nodes
- Provides other parameters too
    - Network prefix
    - Address of local router
    - DNS server, time server, etc.

DHCP Protocol Stack

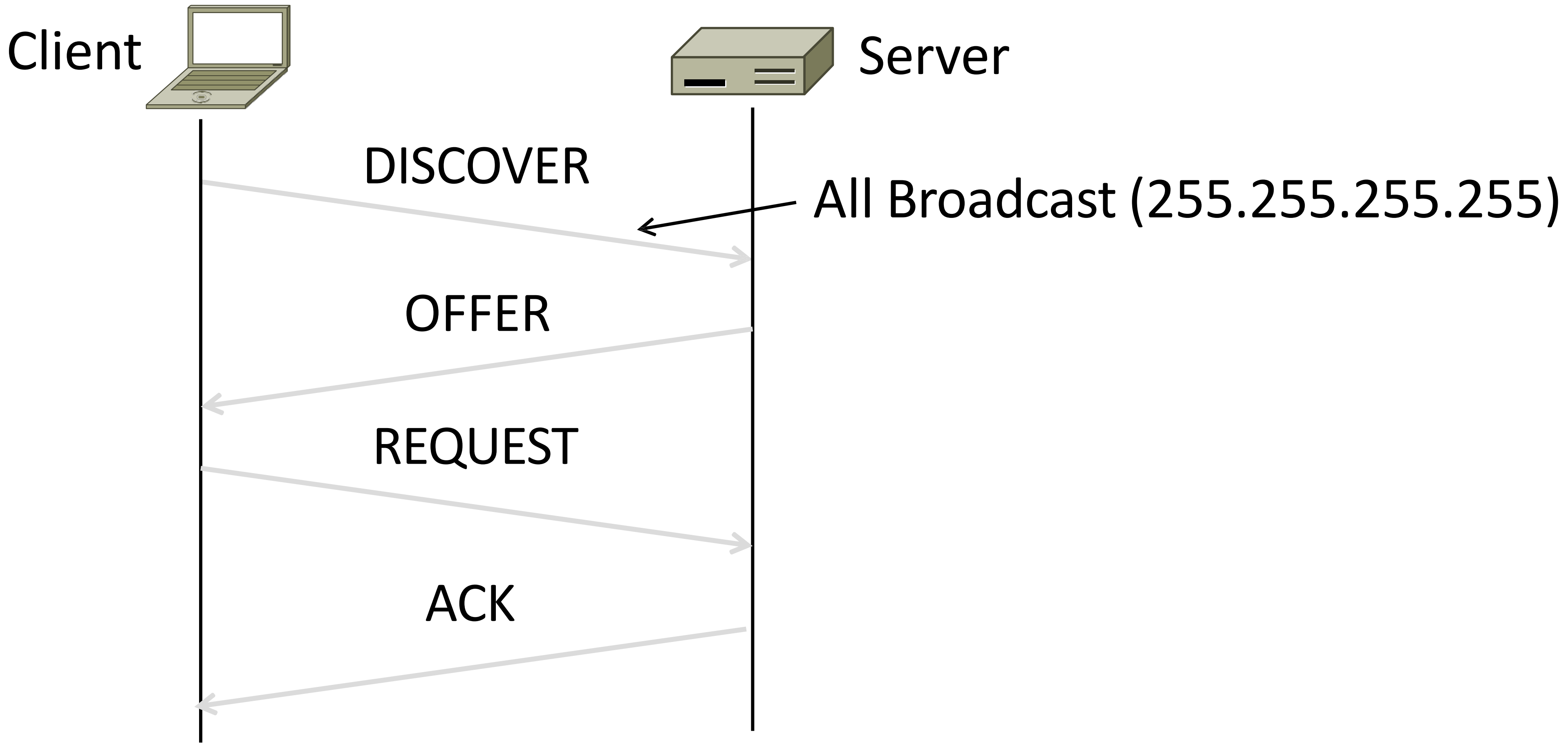
- DHCP is a client-server application
    - Uses UDP ports 67, 68

DHCP
UDP
IP
Ethernet

DHCP Addressing

- Bootstrap issue:
  - How does node send a message to DHCP server before it is configured?
- Answer:
  - Node sends broadcast messages that delivered to all nodes on the link-level network
  - Broadcast address is all 1s
  - IP (32 bit): 255.255.255.255
  - Ethernet (48 bit): ff:ff:ff:ff:ff:ff

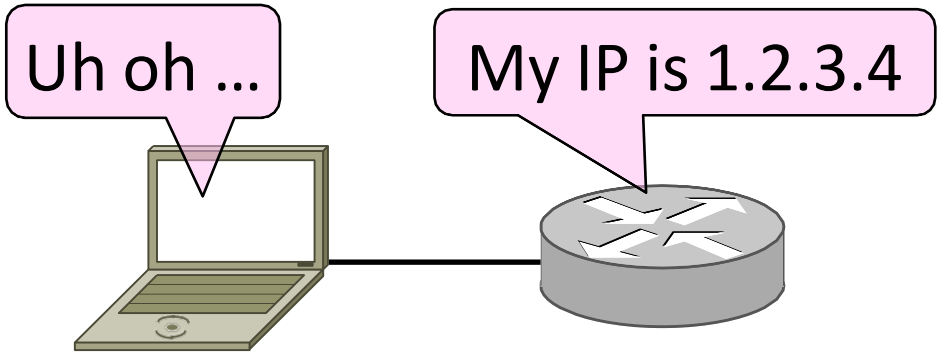
DHCP Messages



Address Resolution Protocol (ARP)

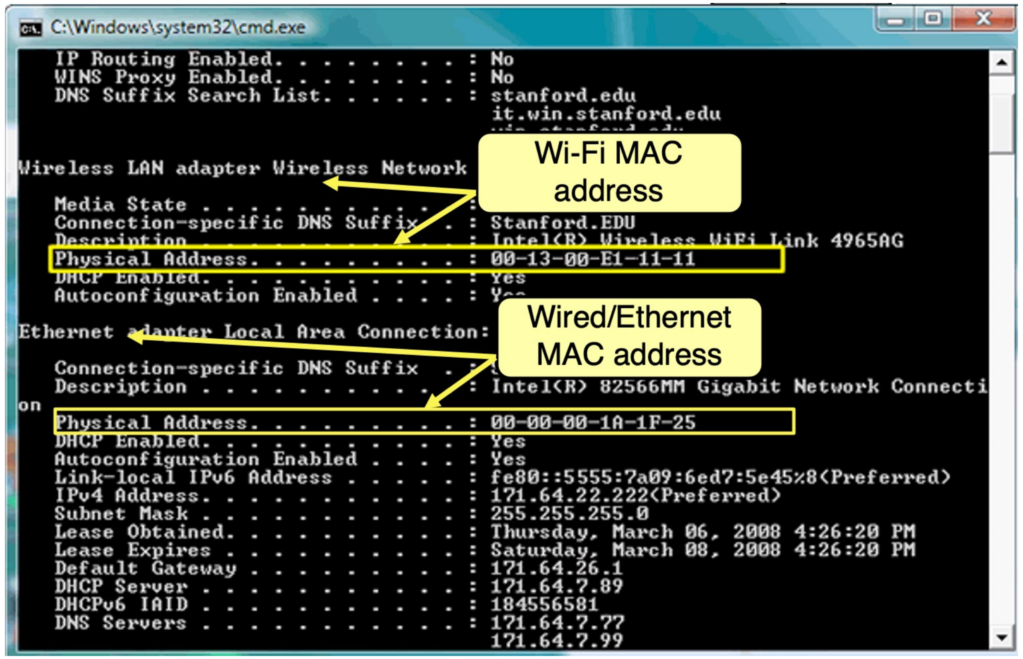
Sending an IP Packet

- Problem:
  - A node needs Link layer addresses (physical addresses) to send a frame over the local link
  - How does it get the destination link address from a destination IP address?



Physical address

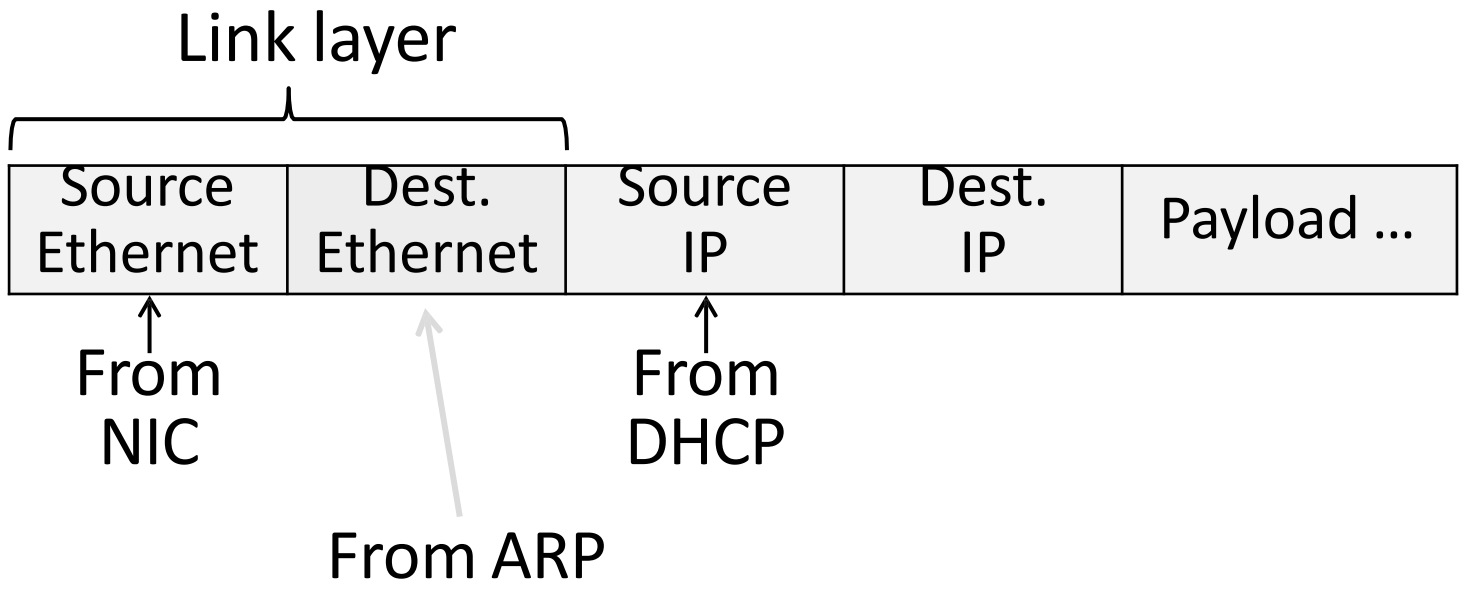
- MAC (Media Access Control) address in link layer
  - 48-bit physical address for hardware interface
  - Every device has a unique address
- IP vs. MAC
  - Analogy: MAC -> ID; IP -> home address



IP address changes as the device location changes

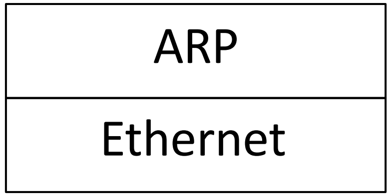
ARP (Address Resolution Protocol)

- Node uses to map a local IP address to its Link layer addresses

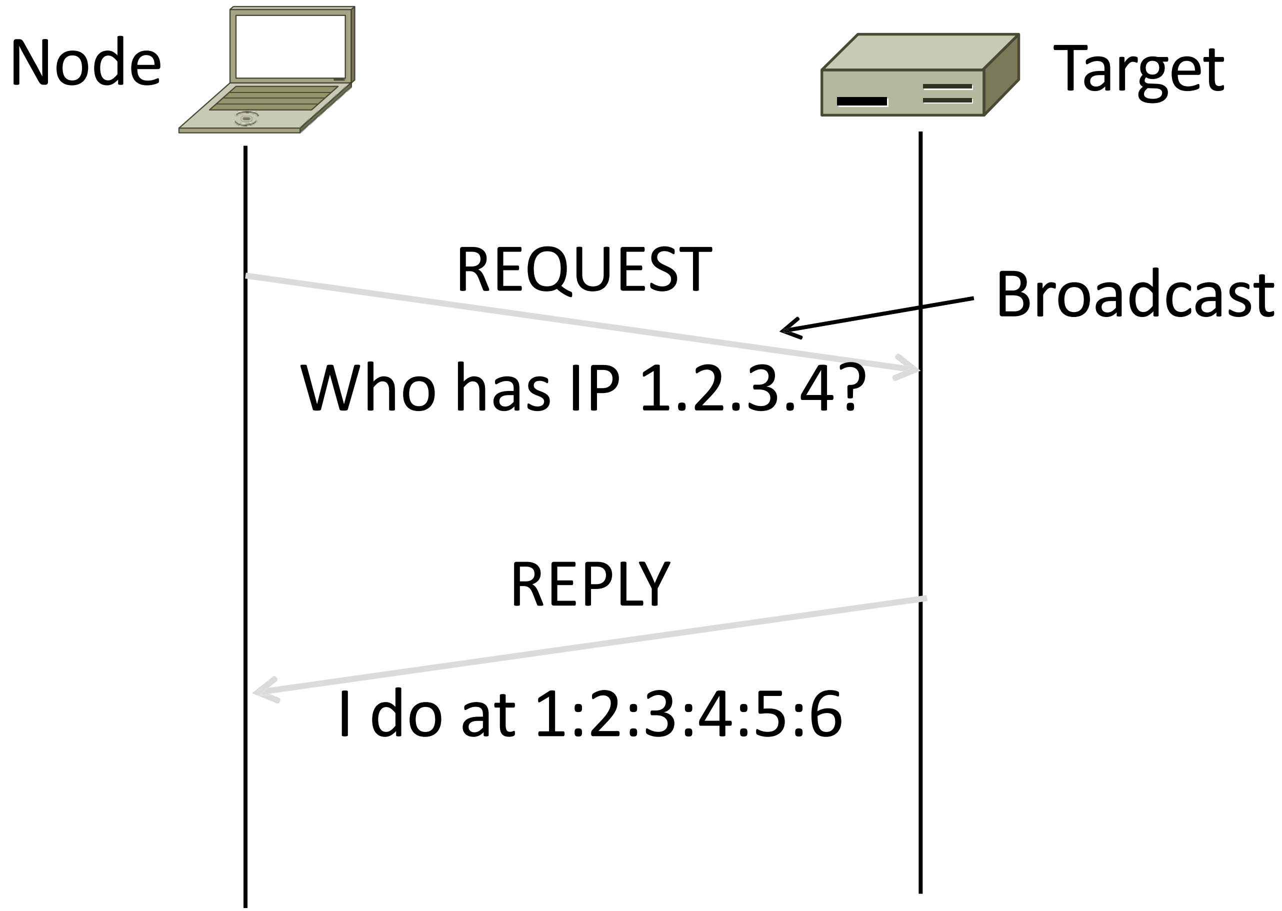


ARP Protocol Stack

- ARP sits right on top of link layer
  - No servers, just asks node with target IP to identify itself
  - Uses broadcast to reach all nodes



ARP Messages



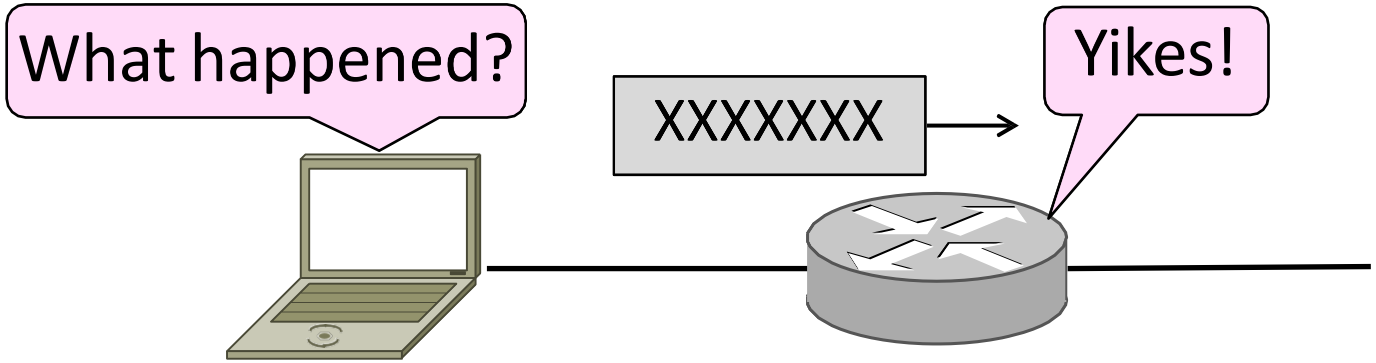
ARP Table

```
[Ratuls-MacBook-Pro:19wi ratul$ arp -a | grep 192
? (192.168.88.1) at e4:8d:8c:54:0:52 on en0 ifscope [ethernet]
```

Internet Control Message Protocol (ICMP)

Internet Control Message Protocol

- Problem: What happens when something goes wrong during forwarding?
  - Need to be able to find the problem



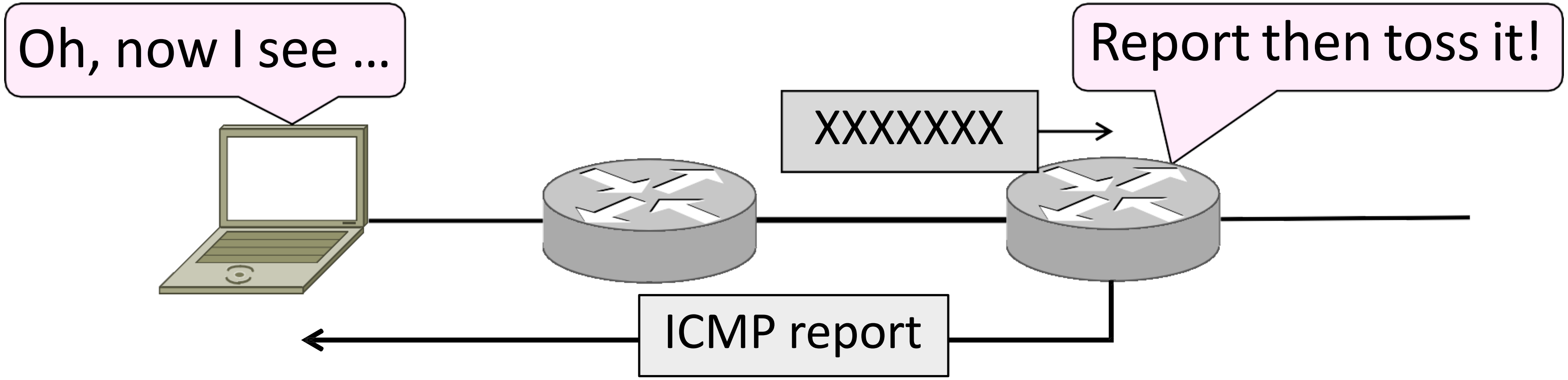
Internet Control Message Protocol

- ICMP is a companion protocol to IP
  - They are implemented together
  - Sits on top of IP (IP Protocol=1)

- Provides error report and testing
  - Error is at router while forwarding
  - Also testing that hosts can use

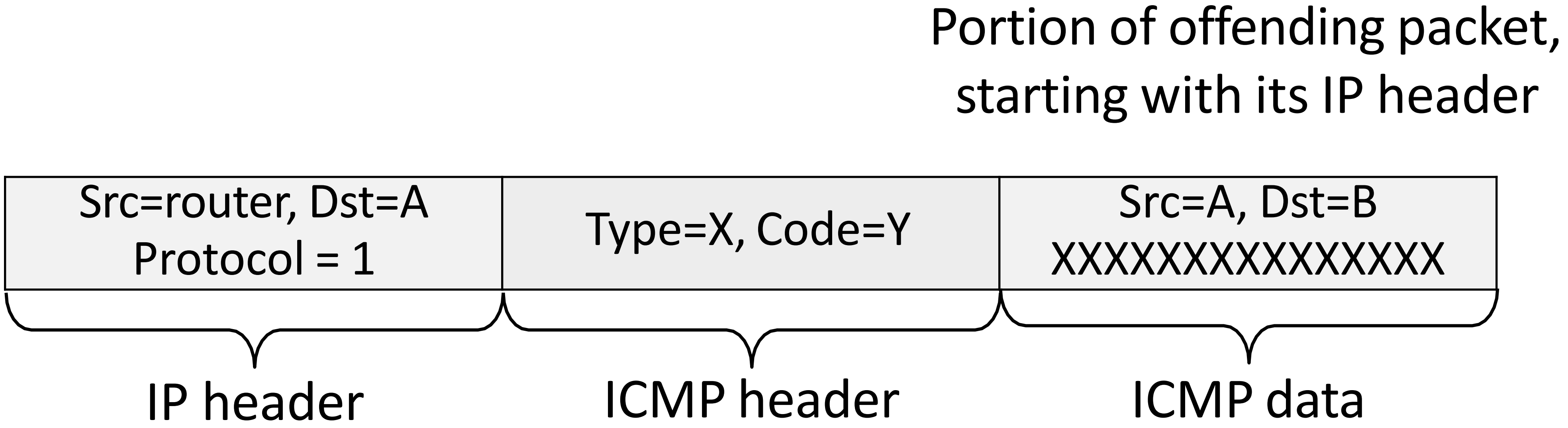
ICMP Errors

- When router encounters an error while forwarding:
  - It sends an ICMP error report back to the IP source
  - It discards the problematic packet; host needs to rectify



ICMP Message Format

- Each ICMP message has a Type, Code, and Checksum
- Often carry the start of the offending packet as payload
- Each message is carried in an IP packet



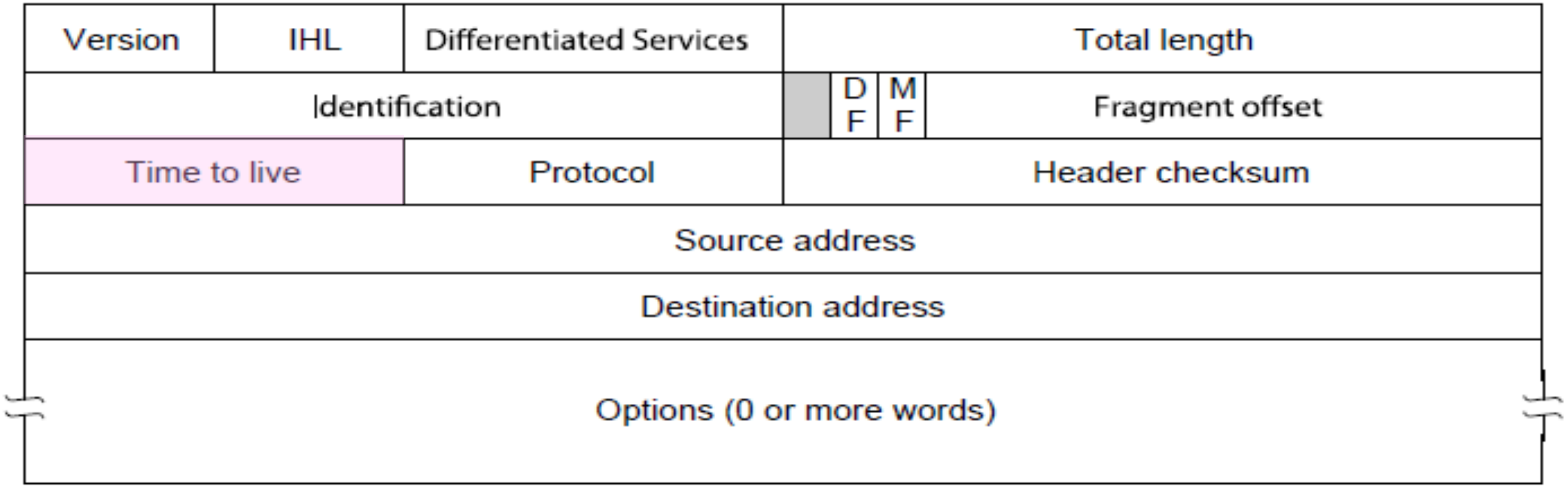
Example ICMP Messages

Name	Type / Code	Usage
Dest. Unreachable (Net or Host)	3 / 0 or 1	Lack of connectivity
Dest. Unreachable (Fragment)	3 / 4	Path MTU Discovery
Time Exceeded (Transit)	11 / 0	Traceroute
Echo Request or Reply	8 or 0 / 0	Ping

Testing, not a forwarding error: Host sends Echo Request, and destination responds with an Echo Reply

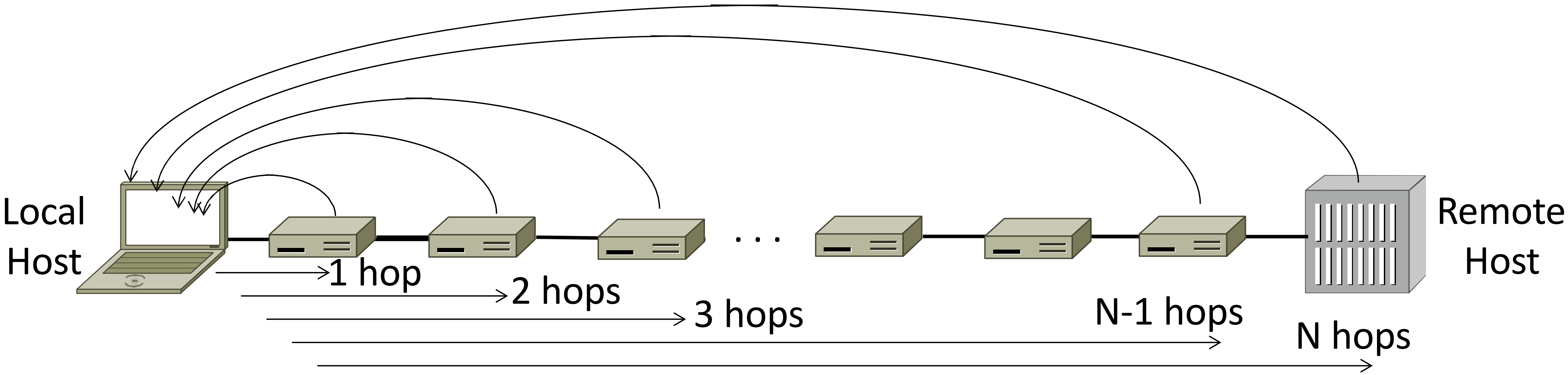
Traceroute

- IP header contains TTL (Time to live) field
  - Decremented every router hop, with ICMP error at zero
  - Protects against forwarding loops



Traceroute

- Traceroute repurposes TTL and ICMP functionality
  - Sends probe packets increasing TTL starting from 1
  - ICMP errors identify routers on the path



Network Address Translation (NAT)

Problem: Internet’s success

Today, Internet connects

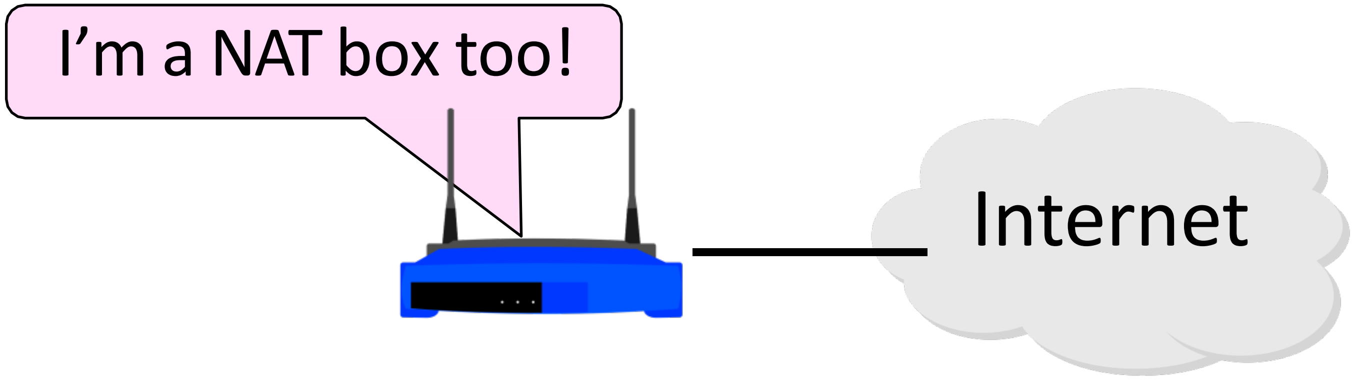
- 4B people
- 50B devices

And we’re using 32-bit addresses!

- 2B unique addresses

Network Address Translation (NAT)

- Basic idea: Map many “Private” IP addresses to one “Public” IP.
- Allocate IPs for private use (192.168.x, 10.x)



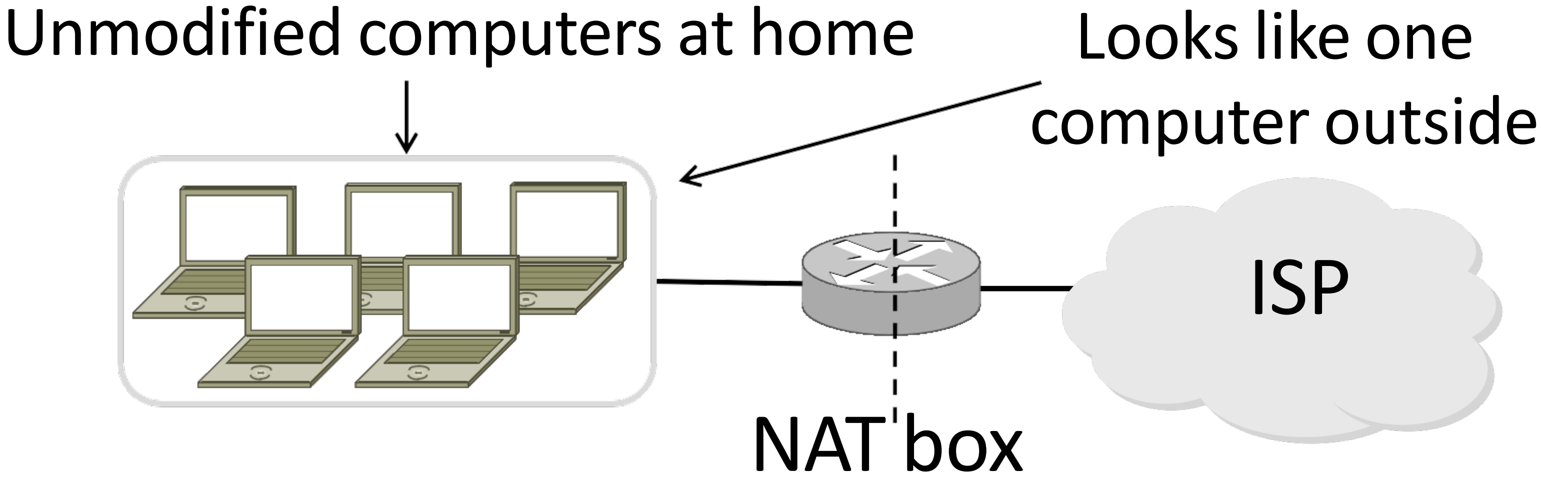
NAT (Network Address Translation) Box

- NAT box maps an internal IP to an external IP
  - Many internal hosts connected using few external addresses
  - Middlebox that “translates addresses”

- Motivated by IP address scarcity
  - Controversial at first, now accepted

NAT

- Common scenario:
  - Home computers use “private” IP addresses
  - NAT (in AP/firewall) connects home to ISP using a single external IP address



How NAT Works

Keeps an internal/external translation table

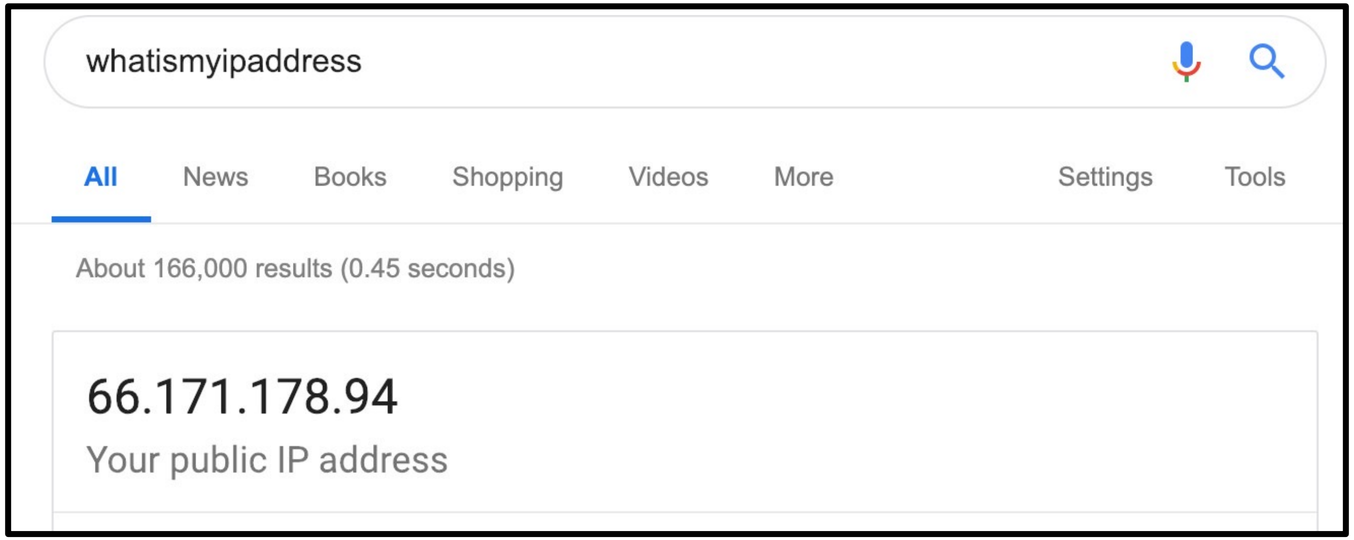
- Typically uses IP address + TCP port
- This is address and port translation

What host thinks	What ISP thinks
Internal IP:port	**External IP : port**
192.168.1.12 : 5523	44.25.80.3 : 1500
192.168.1.13 : 1234	44.25.80.3 : 1501
192.168.2.20 : 1234	44.25.80.3 : 1502

- Need ports to make mapping 1-1 since there are fewer external IPs

NAT in action

```
Ratuls-MacBook-Pro:19wi ratul$ ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether f0:18:98:a5:f9:cc
        inet6 fe80::440:e511:c06f:78f9%en0 prefixlen 64 secured scopeid 0xa
        inet 192.168.88.14 netmask 0xffffff00 broadcast 192.168.88.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```



NAT Downsides

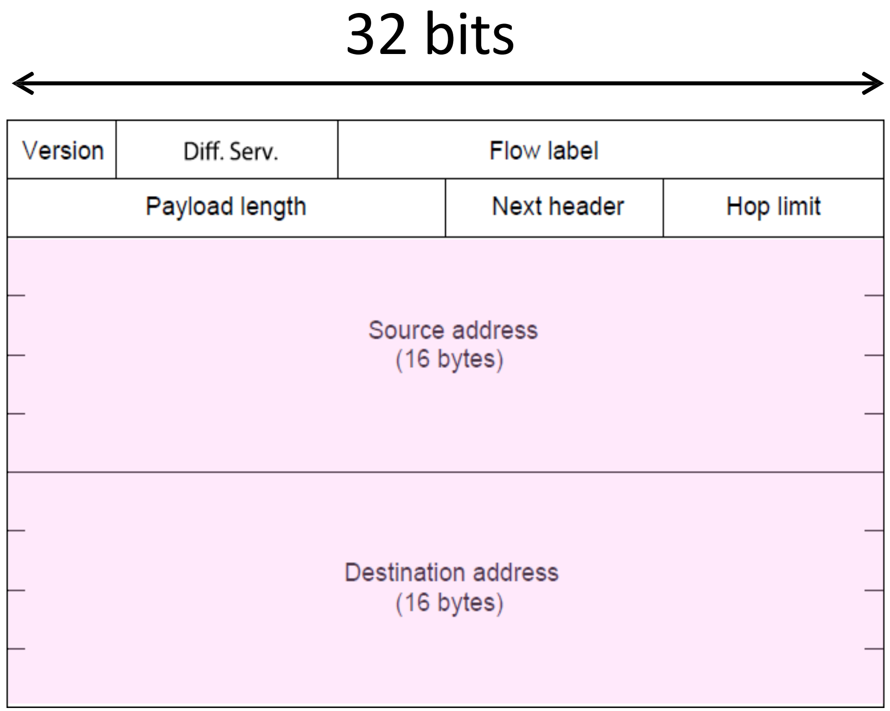
- Connectivity has been broken!
  - Can only send incoming packets after an outgoing connection is set up
  - Difficult to run servers or peer-to-peer apps
- Doesn't work if return traffic by passes the NAT
- Breaks apps that expose their IP addresses (FTP)

NAT Upsides

- Relieves much IP address pressure
  - Many home hosts behind NATs
- Easy to deploy
  - Rapidly, and by you alone
- Useful functionality
  - Firewall, helps with privacy

IP Version 6 (IPv6) to the Rescue

- Features large addresses
  - 128 bits, most of header
- New notation
  - 8 groups of 4 hex digits (16 bits)
  - Omit leading zeros, groups of zeros



Ex: 2001:0db8:0000:0000:0000:ff00:0042:8329

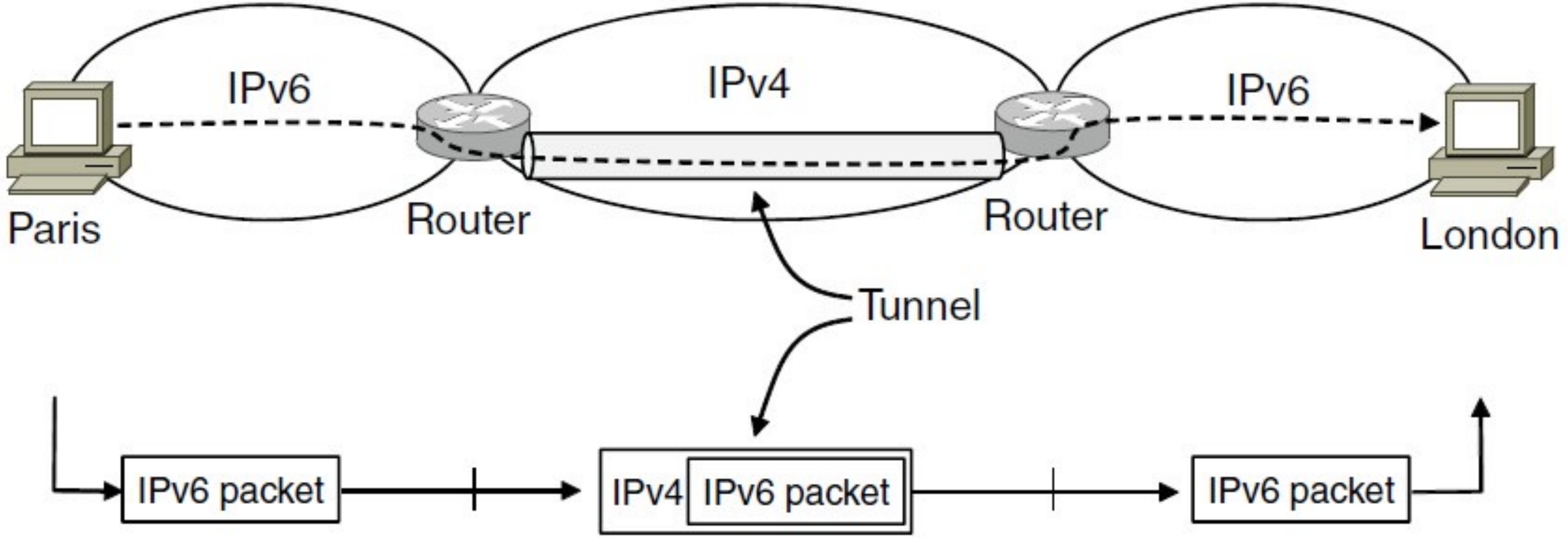
→ 2001:db8::ff00:42:8329

IPv6 Transition

- The Big Problem:
  - How to deploy IPv6?
  - Fundamentally incompatible with IPv4
- Dozens of approaches proposed
  - Dual stack (speak IPv4 and IPv6)
  - Translators (convert packets)
  - Tunnels (carry IPv6 over IPv4)

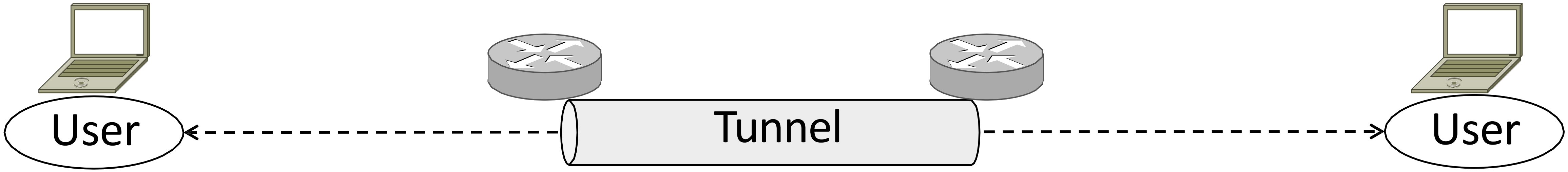
Tunneling

- Native IPv6 islands connected via IPv4
    - Tunnel carries IPv6 packets across IPv4 network



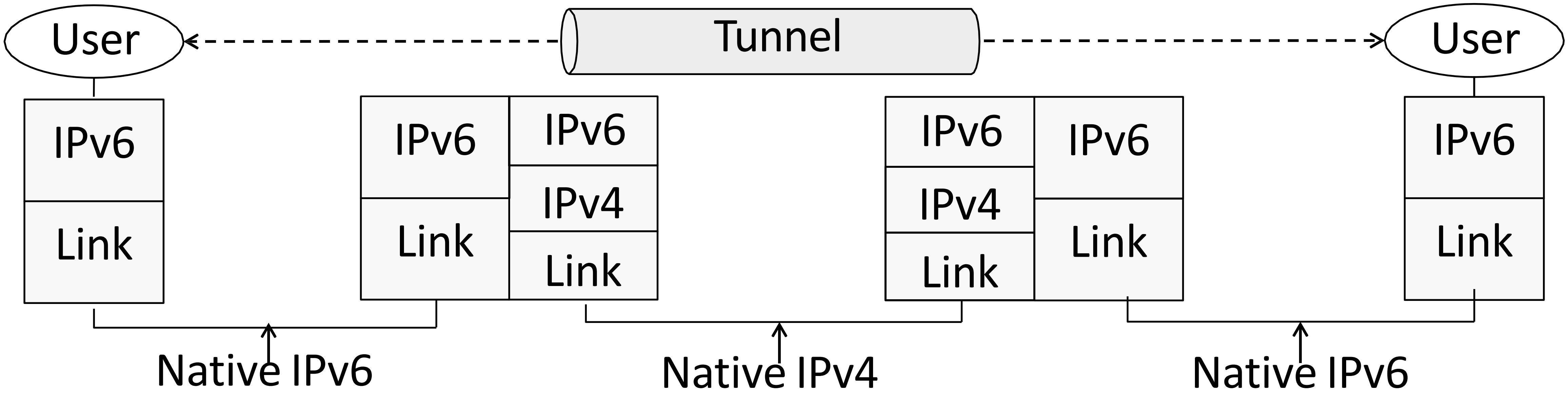
Tunneling

- Tunnel acts as a single link across IPv4 network



Tunneling

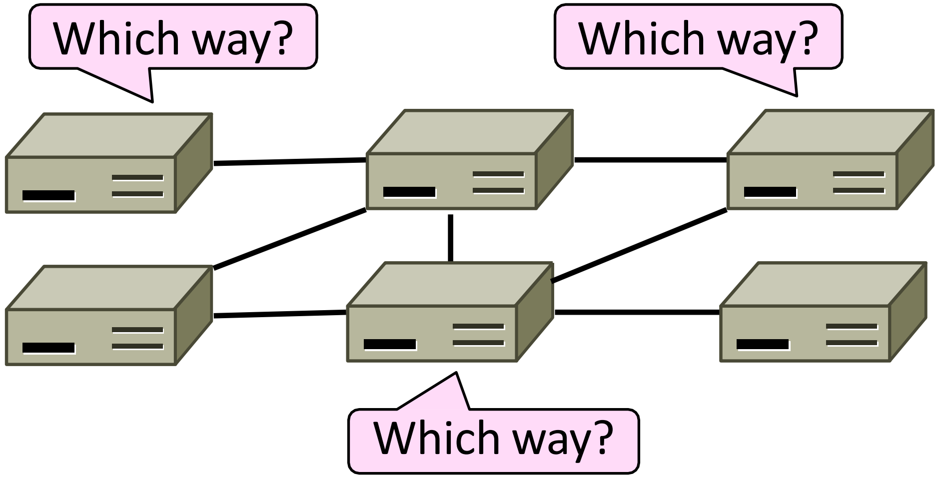
- Tunnel acts as a single link across IPv4 network



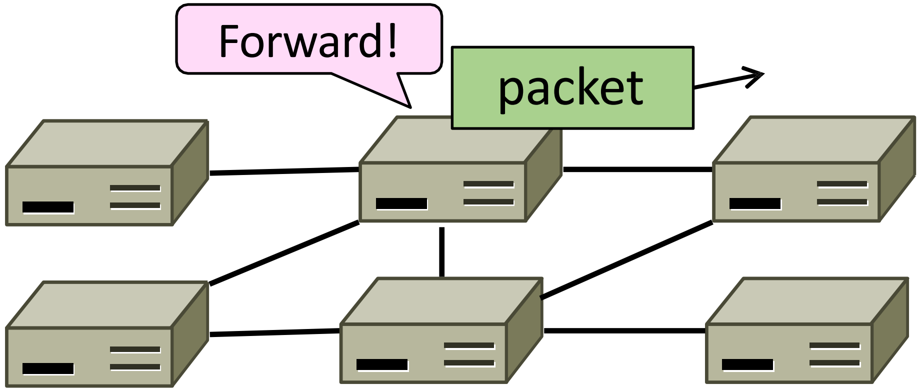
Routing and Forwarding

Routing versus Forwarding

- Routing: deciding the direction to send traffic



- Forwarding: sending a packet on its way



Overview of Routing and Forwarding

- Hosts on same network have IPs in the same IP prefix
- Hosts send off-network traffic to the gateway router

- Routers discover routes to different prefixes (routing)
- Routers use **longest prefix matching** to send packets to the right next hop (forwarding)

Longest Prefix Matching

- Prefixes in the forwarding table can overlap

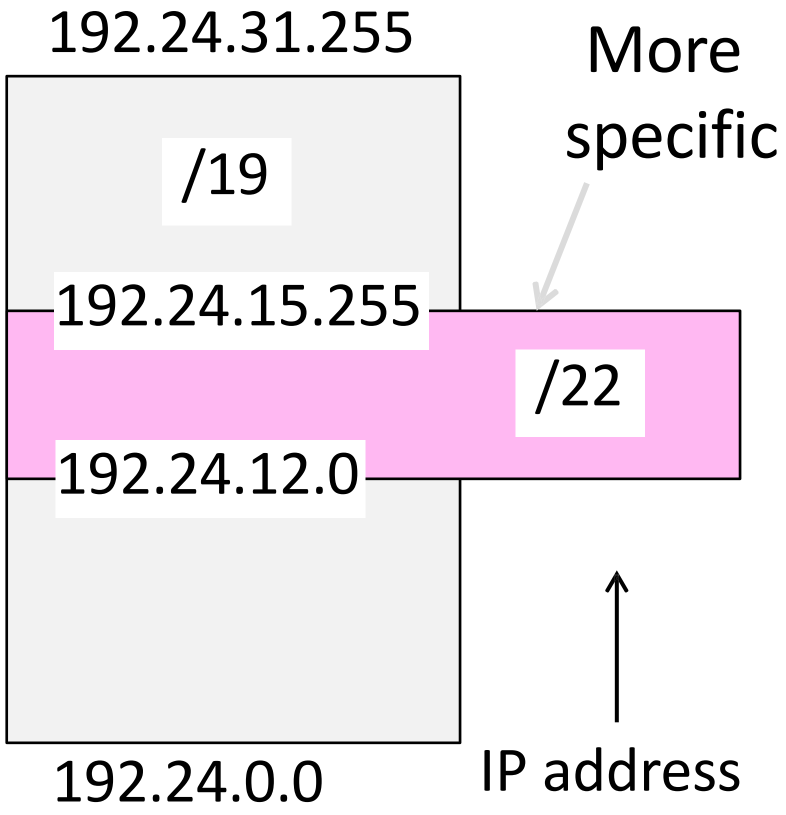
Prefix	Next Hop
0.0.0.0/0	A
192.24.0.0/19	B
192.24.12.0/22	C

- Longest prefix matching forwarding rule:
  - For each packet, find the longest prefix that contains the destination address, i.e., the most specific entry
  - Forward the packet to the next hop router for that prefix

Longest Prefix Matching

Prefix	Next Hop
192.24.0.0/19	D
192.24.12.0/22	B

192.24.6.0 → D
192.24.14.32 → B
192.24.54.0 → ?



Flexibility of Longest Prefix Matching

- Can provide default behavior, with less specifics
    - Send traffic going outside an organization to a border router (gateway)
- Can special case behavior, with more specifics
    - For performance, economics, security, …

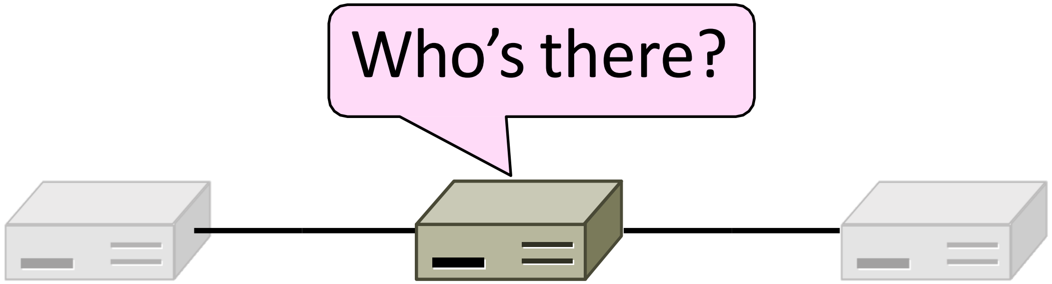
Goals of Routing Algorithms

- We want several properties of any routing scheme:

Property	Meaning
Correctness	Finds paths that work
Efficient paths	Uses network bandwidth well
Fair paths	Doesn’t starve any nodes
Fast convergence	Recovers quickly after changes
Scalability	Works well as network grows large

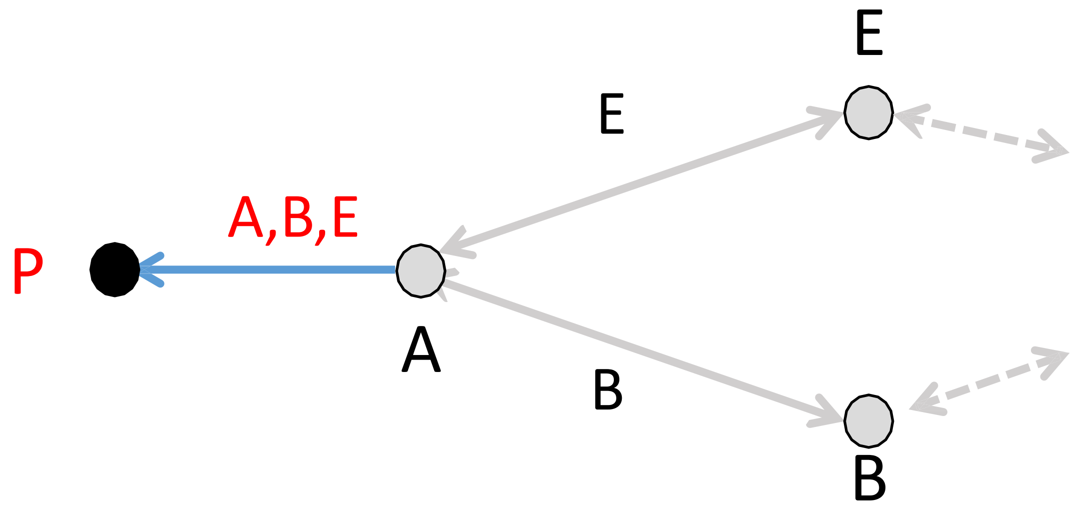
Rules of Fully Distributed Routing

- All nodes are alike; no controller
- Nodes learn by exchanging messages with neighbors
- Nodes operate concurrently
- There may be node/link/message failures



Simple routing that obeys the rules

- All routers find a path to all hosts
- But scales poorly!

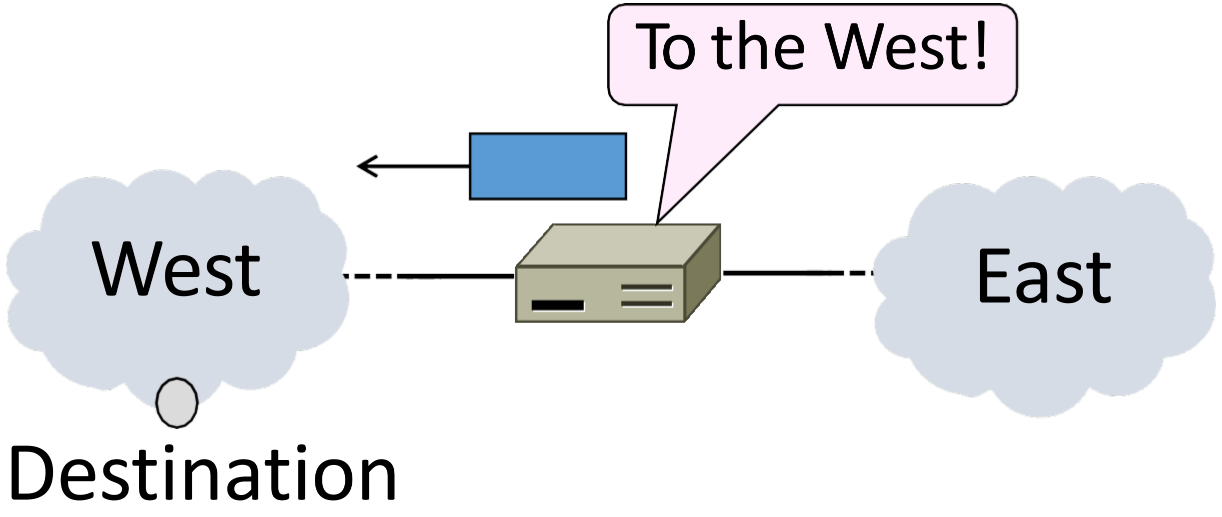


Techniques to Scale Routing

- First: Network hierarchy
  - Route to network regions
- Next: IP prefix aggregation
  - Combine, and split, prefixes

Hierarchical Routing

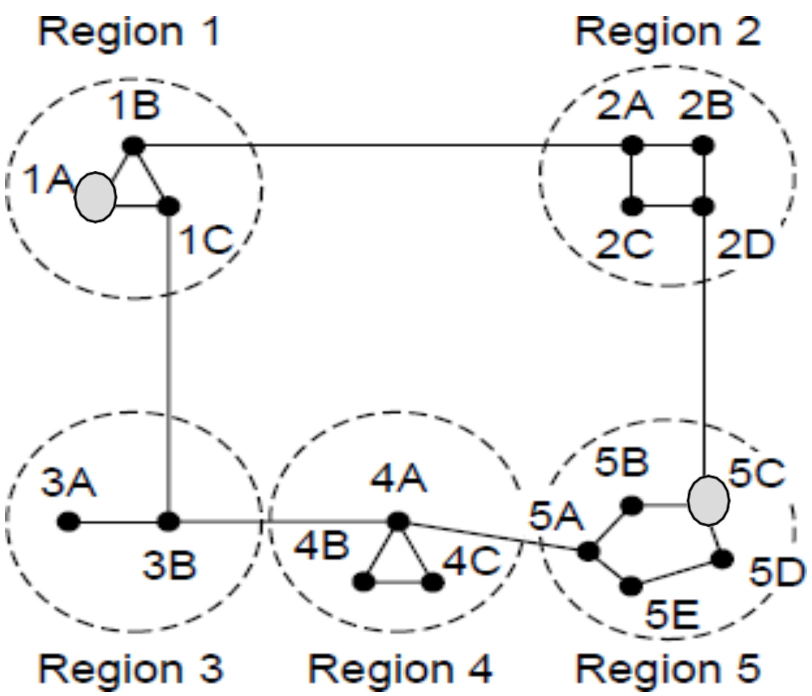
- Scale routing using hierarchy with regions
  - Route to regions, not individual nodes



Hierarchical Routing

- Introduce a larger routing unit
  - Region, e.g., ISP network
- Route first to the region, then to the IP prefix within the region
  - Hide details within a region from outside of the region

Hierarchical Routing



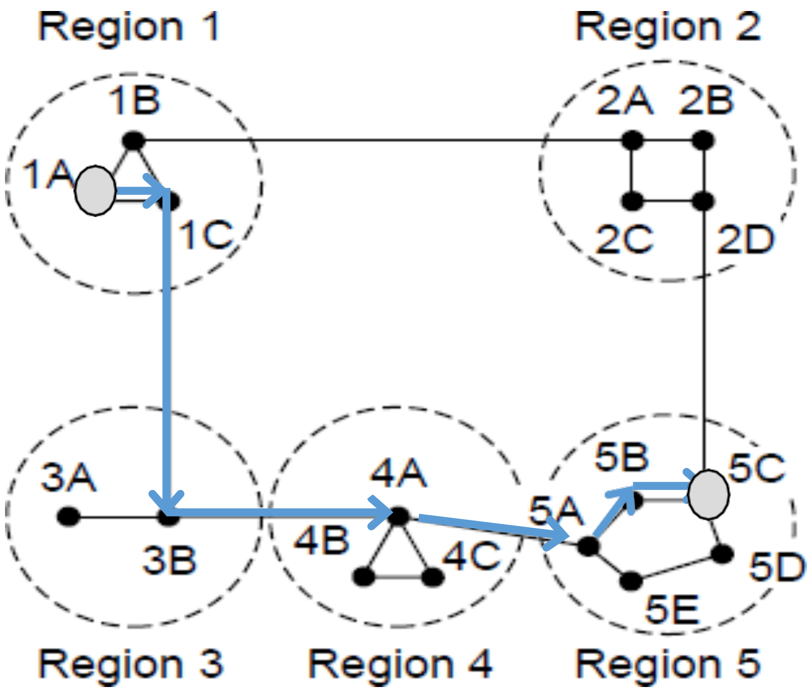
Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Hierarchical Routing



Full table for 1A

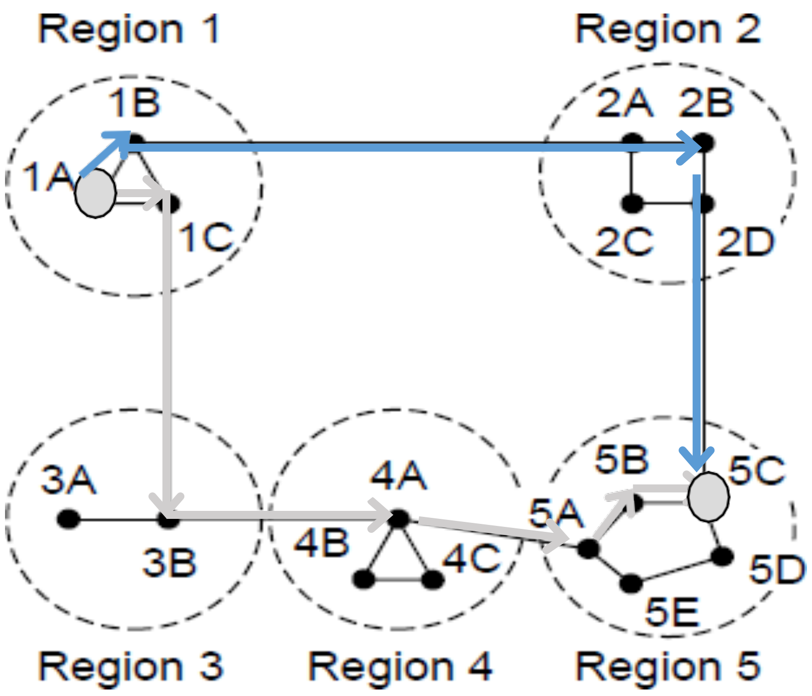
Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Hierarchical Routing

- Penalty is longer paths



Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

1C is best route to region 5, except for destination 5C

Observations

- Outside a region, nodes have one route to all hosts within the region
  - This gives savings in table size, messages and computation
- However, each node may have a different route to an outside region
  - Routing decisions are still made by individual nodes; there is no single decision made by a region

How to generate routing table

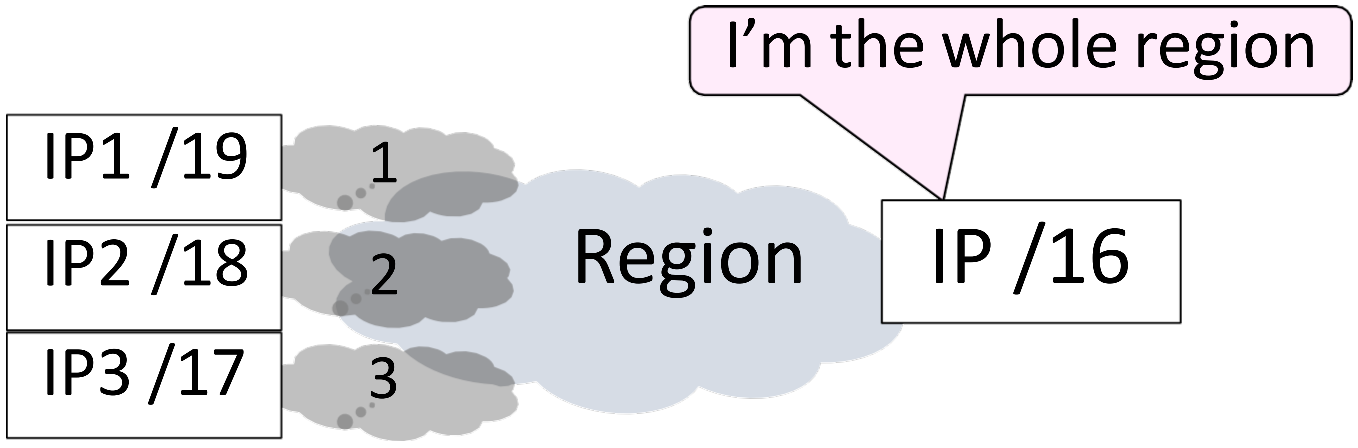
- Static routing
  - Manual Configuration
  - Simplicity and Control
- Dynamic routing
  - Routing protocols to automatically learn and share routing information
  - Common protocols: RIP (Routing Information Protocol), OSPF (Open Shortest Path First), and BGP (Border Gateway Protocol)

Enabling hierarchical routing:

IP Prefix Aggregation and Subnets

Idea

- Scale routing by adjusting the size of IP prefixes
  - Split (subnets) and join (aggregation)



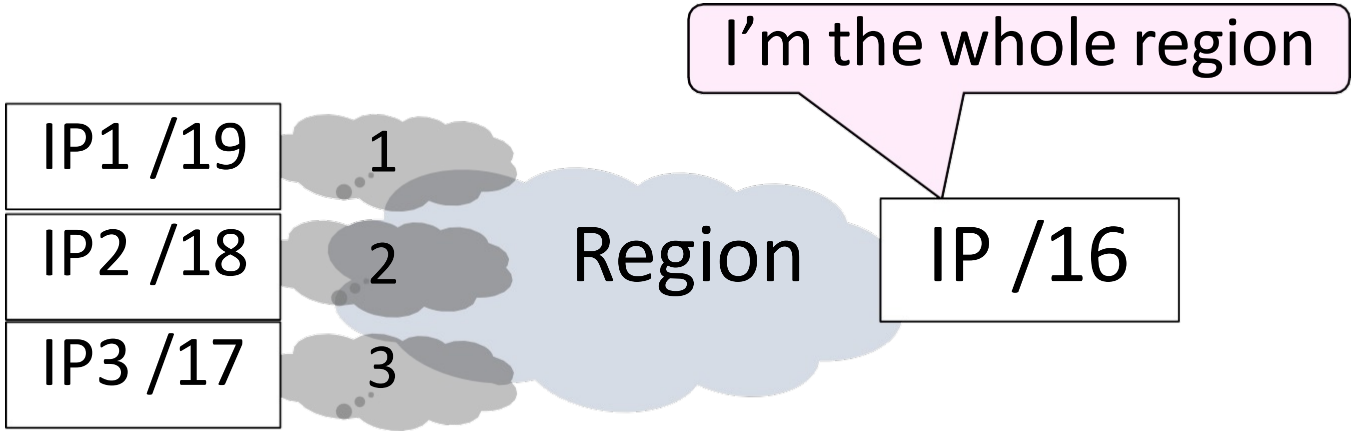
Recall

- IP addresses are allocated in blocks called IP prefixes, e.g., 18.31.0.0/16
  - Hosts on one network in same prefix
- “/N” prefix has the first N bits fixed and contains 2^{32-N} addresses
  - E.g., a “/24” has 256 addresses
- Routers keep track of prefix lengths
  - Use it as part of longest prefix matching

Routers can change prefix lengths without affecting hosts

Prefixes and Hierarchy

- IP prefixes help to scale routing, but can go further
  - Use a less specific (larger) IP prefix as a name for a region



Subnets and Aggregation

- Two use cases for adjusting the size of IP prefixes; both reduce routing table

Subnets

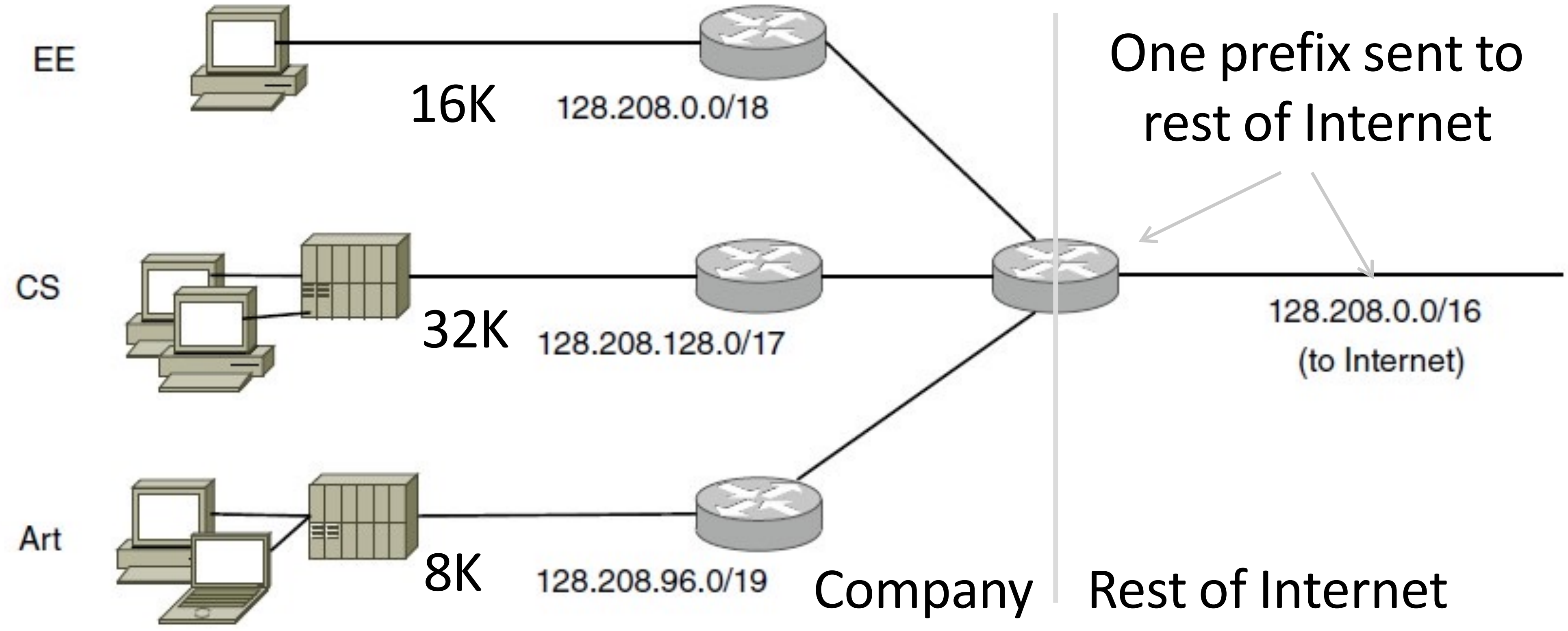
- Internally split one large prefix into multiple smaller ones

Aggregation

- Join multiple smaller prefixes into one large prefix

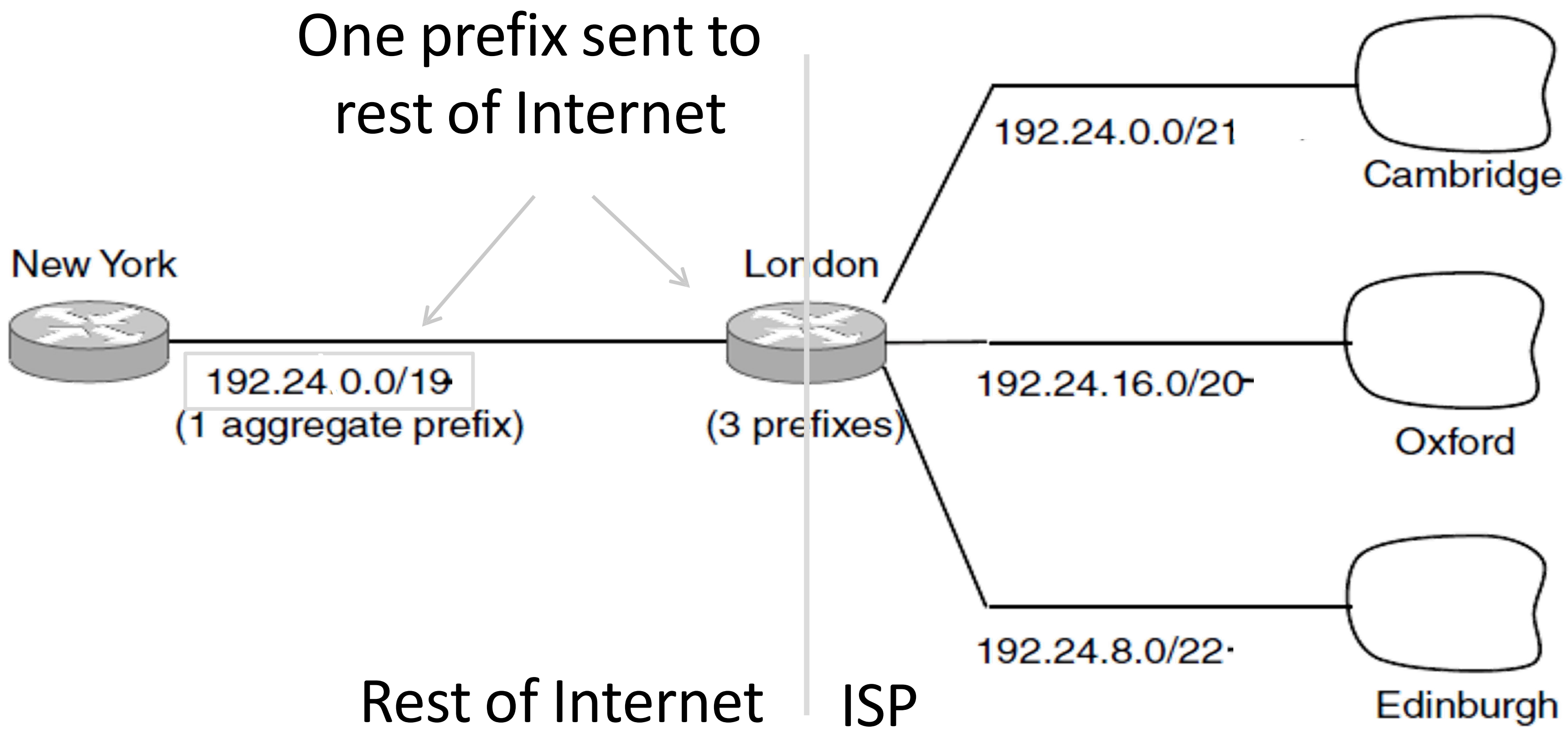
Subnets

- Internally split up one IP prefix



Aggregation

- Externally join multiple separate IP prefixes



Credits

- Some slides are adapted from course slides of CSE 461 in UW