CSE160: Computer Networks

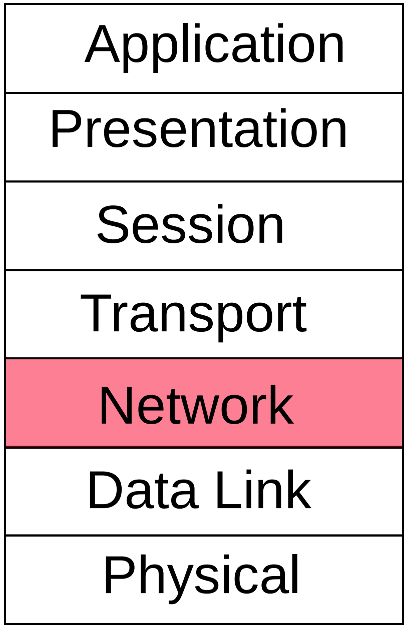
Lecture #12 – IP Addressing

2020-10-06

Professor
Alberto E. Cerpa

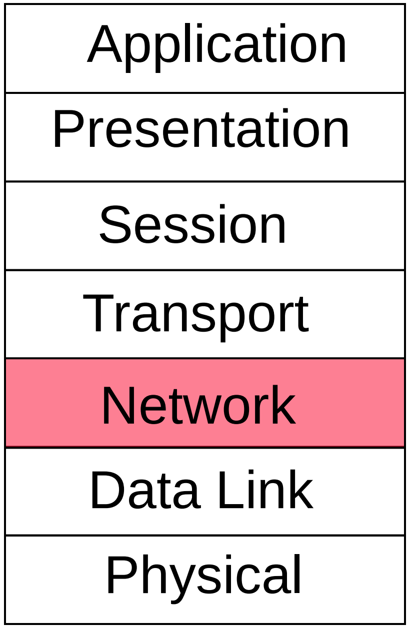
Last Time

- Focus
  - How do we make routing scale?
- Inter-domain routing
  - Hierarchical Routing
  - ASes and BGP
  - Routing Policies



This Lecture

- Focus
  - How do we make routing scale?
- IP Addressing
  - Address formats
  - Hierarchy, Classful Addresses
  - Classless Inter-Domain Routing
  - Subnets and Aggregation



Scalability Concerns

- Routing burden grows with size of an internetwork
  - Size of routing tables
  - Volume of routing messages
  - Amount of routing computation

- To scale to the size of the Internet, apply:
  - Use of structural hierarchy (AS/BGP)
  - Hierarchical addressing
  - Route aggregation

Scalability Concerns

- Use of structural hierarchy, i.e. BGP routing
  - Size of routing tables
    - Pros: reduces the size of the routing tables for many routers inside the AS. If IP address outside AS, route through BGP border router.
    - Cons: size of routing tables in the border routers still huge!
  - Volume of routing messages
    - Pros: allows routing algorithms to scale by running multiple types of algorithms at different levels of the hierarchy, i.e. reduces the size of the graph.
    - Cons: size of routing messages at the border router still big.
  - Amount of routing computation
    - Pros: smaller graph size, implies reduced routing computation for the majority of routers.
    - Cons: border router still have to calculate routes to potentially billions of destination IP addresses (end2end connectivity)!

Humans Location Naming Structure

- How do humans deal with scalability complexity for worldwide locations?
- We use hierarchical *naming*
  - Street number and name, city name, zip code, country name, continent, planet (future?!)
  - Close neighbors should have similar addresses. E.g. 345 Pine St., Merced, CA 95341 and 347 Pine St., Merced, CA 95341
  - The further you move from the source address, the more different your address should be.
  - In order to get to a location, the further away you are, the least detail information you need. E.g.: first state, then city, then street name/number.

Postal Service Routing

- How does the Postal Service deal with worldwide parcel routing?
- It uses multi-level *hierarchical routing* based on *hierarchical naming*
  – Merced resident sends a letter with full destination address, e.g. 210 W 57th St., New York, NY 10019.
  – Local Merced USPS employee put in the East Coast bin. She does not know how to route to NY.
  – Airport USPS Employee sorts East Coast bin and put NY letter into the appropriate plane.
  – Letter arrives to JFK, and it is put in the Manhattan bin.
  – USPS Manhattan gets it and put it in the Central Park bin.
  – Central Park Office gives it to employee who delivers the letter in his route.

IP Addresses

- Reflect location in topology; used for scalable routing
  - Unlike “flat” Ethernet addresses
- Interfaces on same network share prefix
  - Prefix administratively assigned (IANA or ISP)
  - Addresses globally unique
- Routing only advertises entire networks by prefix
  - Local delivery in a single “network” doesn’t involve router
  - (will make “network” precise later on)

Getting an IP address (recap)

- Old fashioned way: sysadmin configured each machine

- Dynamic Host Configuration Protocol (DHCP)
  - One DHCP server with the bootstrap info
    - Host address, gateway/router address, subnet mask, …
    - Find it using broadcast
  - Addresses may be leased; renew periodically

- "Stateless" Autoconfiguration (in IPv6)
  - Get rid of server – reuse Ethernet addresses for lower portion of address (uniqueness) and learn higher portion from routers

Address Resolution Protocol (ARP recap)

- On a single link, need Ethernet addresses to send a frame … source is a given, but what about destination?
  - Requires mapping from IP to MAC addresses
- ARP is a dynamic approach to learn mapping
  - Node A sends broadcast query for IP address X
  - Node B with IP address X replies with its MAC address M
  - A caches (X, M); old information is timed out (~15 mins)
  - Also: B caches A's MAC and IP addresses, other nodes refresh

IP Addresses

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

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

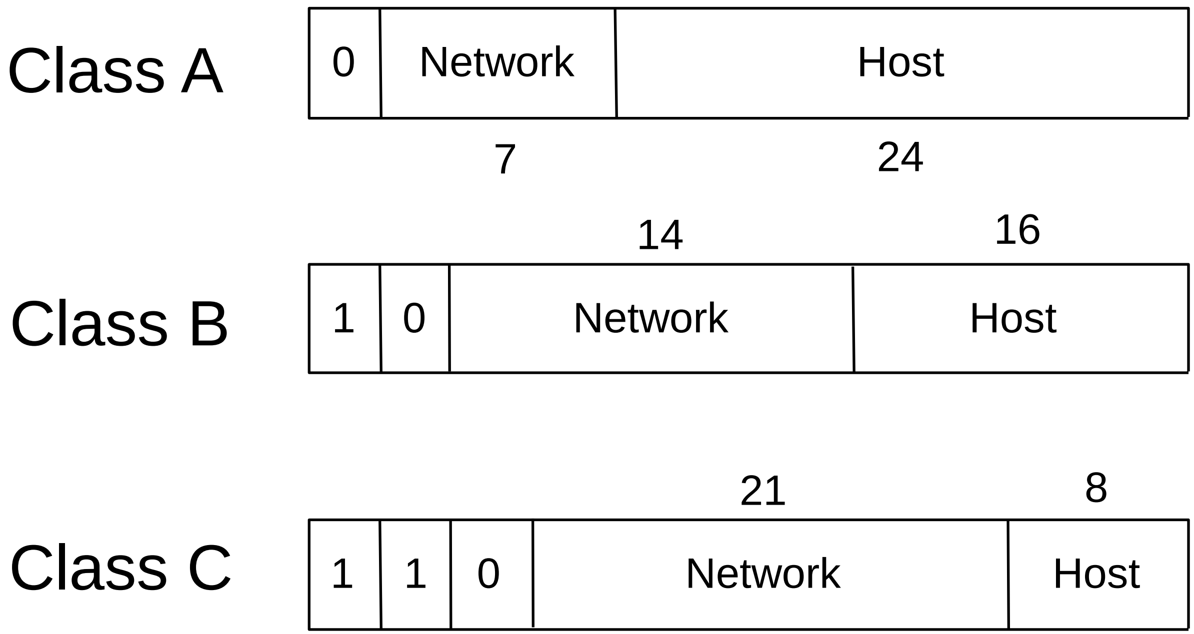
128 64 32 16 8 4 2 1

00010010 | 00011111 | 00000000 | 00000001 ↔ **18.31. 0.1**

18 **31** **0** **1**

Classful IP Addressing

- Originally, IP addresses came in fixed size blocks with the class/size encoded in the high-order bits



- Class A: 0.0.0.0 – 127.255.255.255 (0000 0000 = 0 – 0111 1111 = 127)
- Class B: 128.0.0.0 – 191.255.255.255 (1000 0000 = 128 – 1100 0000 = 192)
- Class C: 192.0.0.0 – 223.255.255.255 (1100 0000 = 192 – 1110 0000 = 224)

IPv4 Assignment Circa 1982

- This is an ARPANET assignment

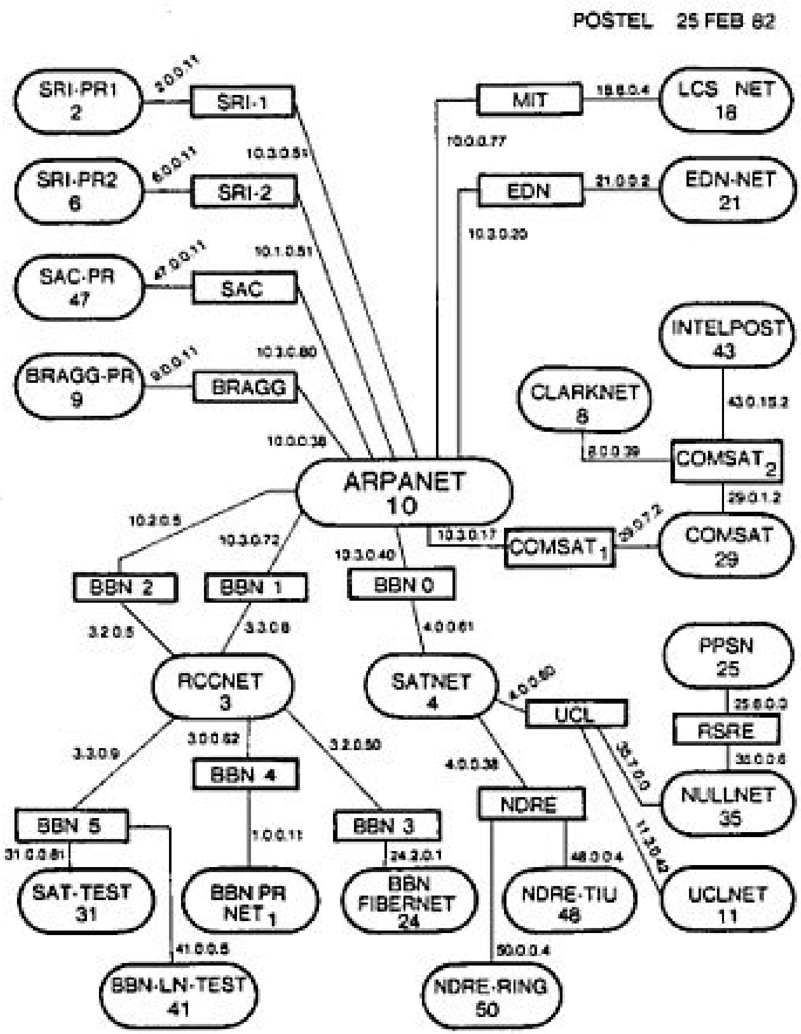
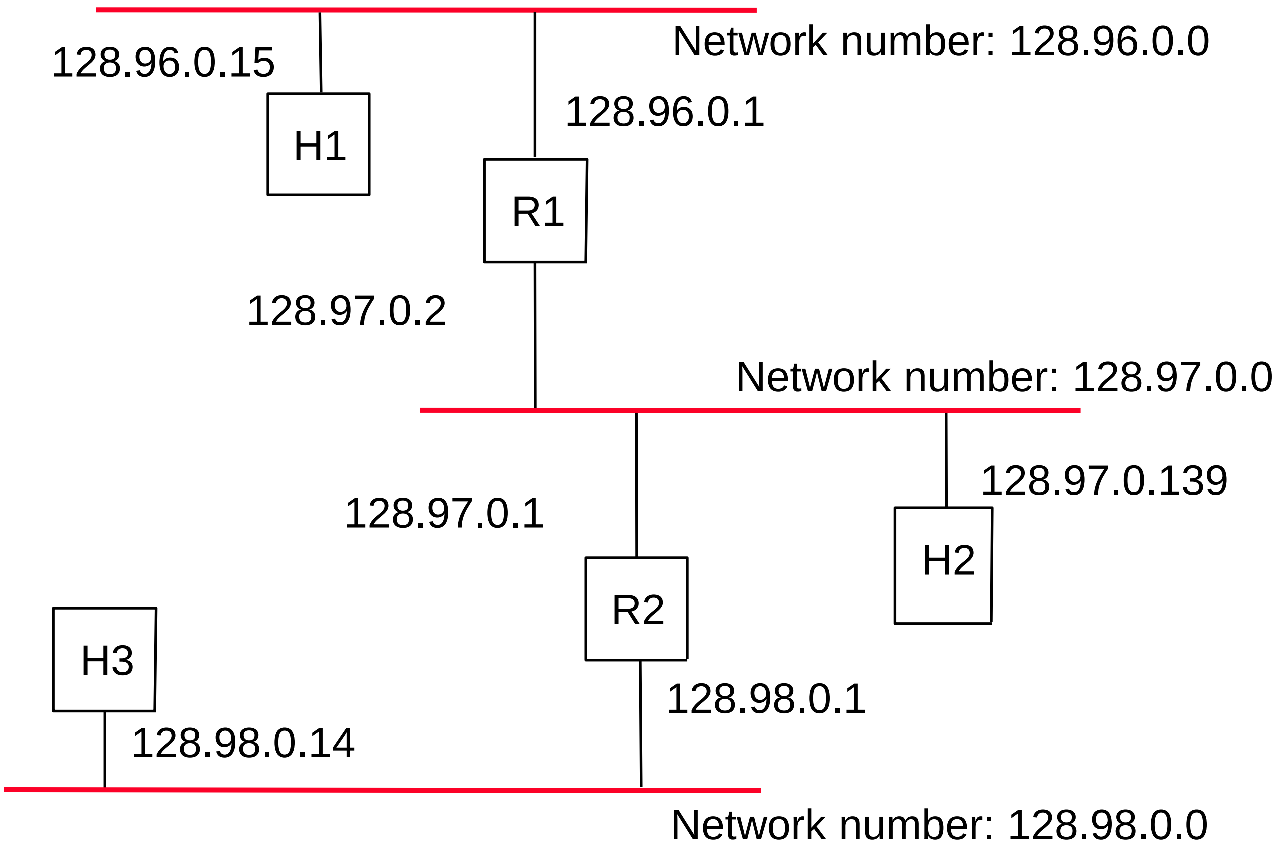


Image in public domain

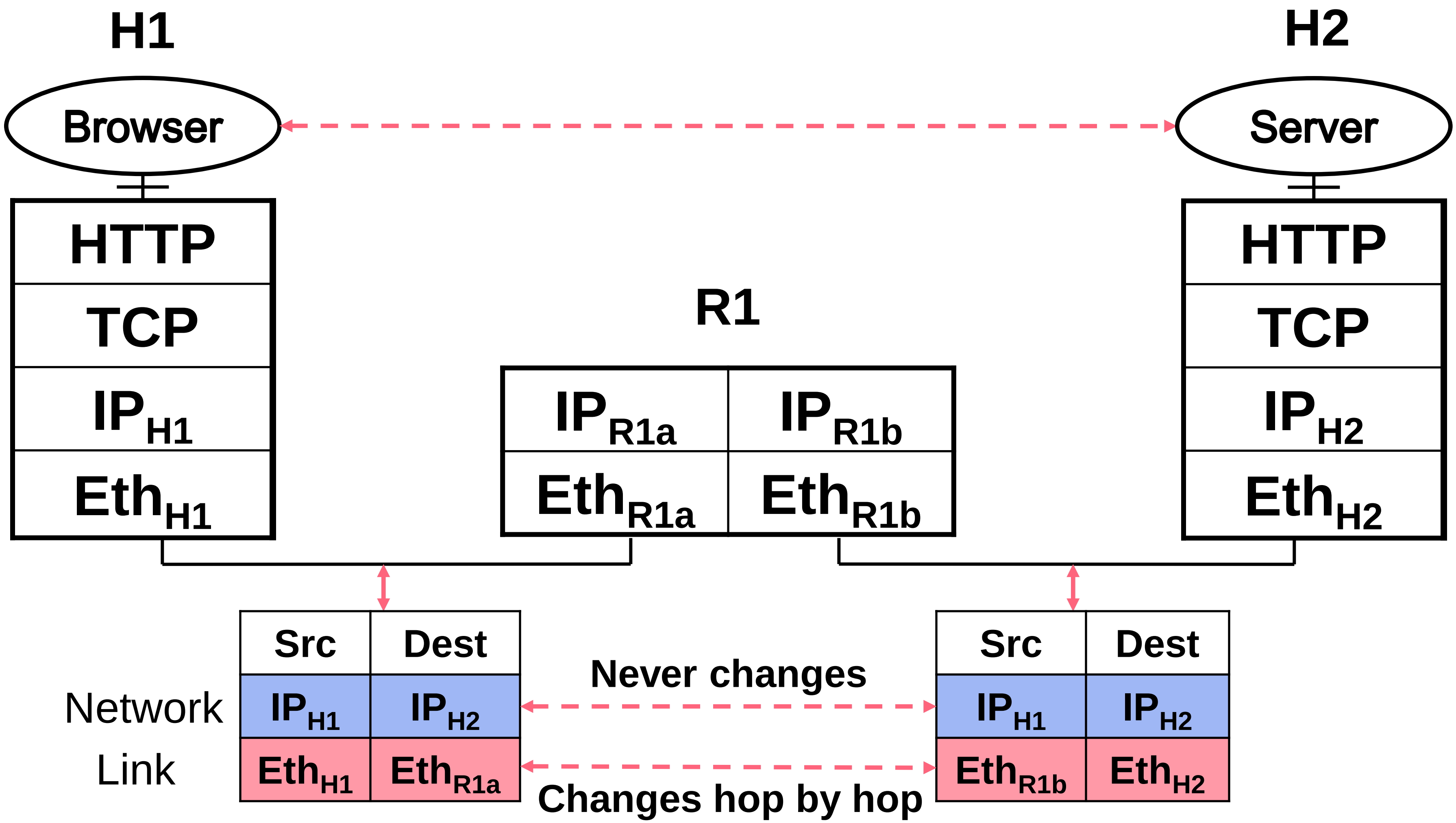
Network Example



Updated Forwarding Routine

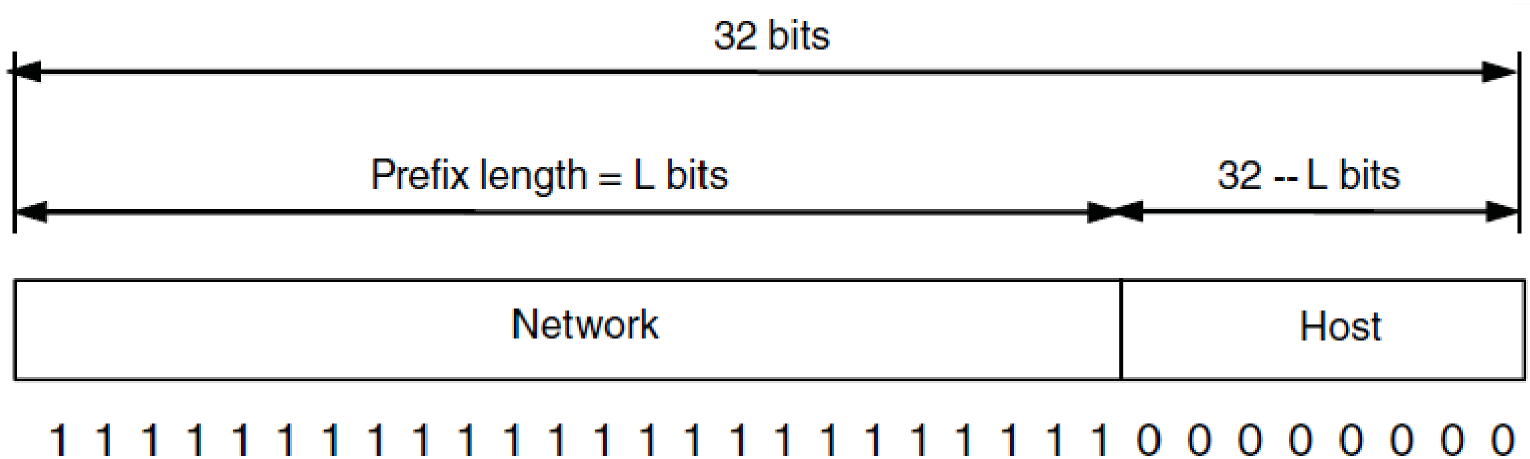
- Used to be “look up destination address for next hop”
- Now addresses have network and host portions:
  - If host: if destination network is the same as the host network, then ***deliver locally*** (without router). Otherwise ***send to the router***.
  - If router: look up destination network in routing table to find next hop and *send to* ***next router***. If destination network is directly attached then ***deliver locally***.
- (Note that it will get a little more complicated later)
- How does the router knows it has to deliver locally?
  - It has same network address than one of its interfaces!
- How does a host send a packet to the router?

How to send a packet to a router?



Classless Inter-Domain Routing (CIDR)

- Generalize class A, B, C into prefixes of arbitrary length; now must carry prefix length with address
- Addresses are allocated in blocks called prefixes
  - Address in an L-bit prefix have the same top L bits
  - There are 2^{32-L} addresses aligned on 2^{32-L} boundary



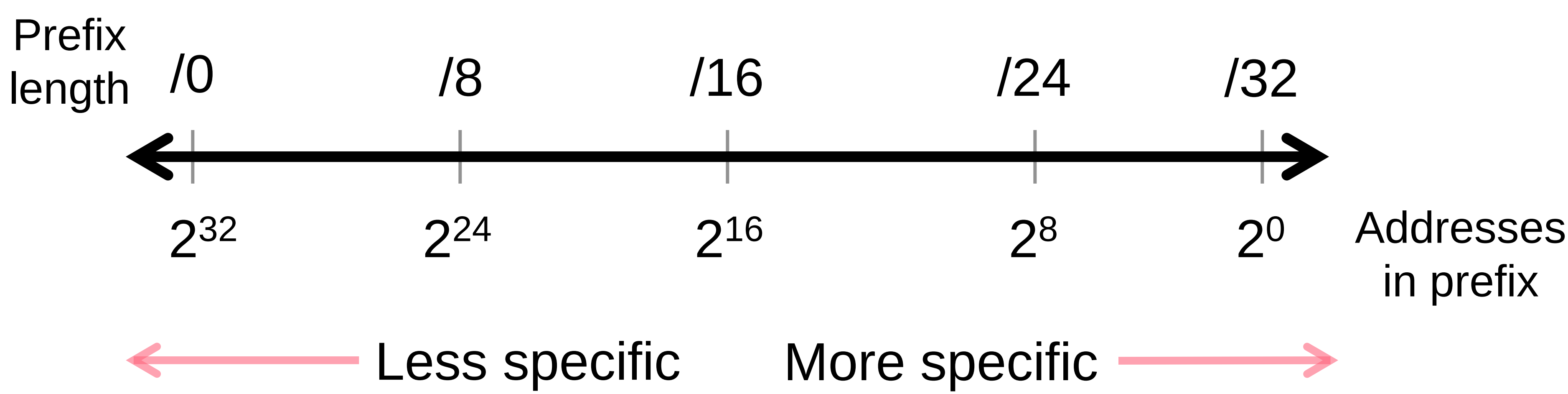
IP Prefixes

- 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 a /32 is one address

00010010	00011111	00000000	xxxxxxxx	↔	18.31.0.0/24
10000000	00001101	xxxxxxxx	xxxxxxxx	↔	128.13.0.0/16

IP Prefixes (2)

- More specific prefix
  - Has longer prefix, hence a smaller number of IP addresses
- Less specific prefix
  - Has shorter prefix, hence a lager number of IP address



Public/Private Addresses

- Public IP addresses, e.g., 18.31.0.1
  - Valid destination on the global Internet
  - Must be allocated to you before you use
  - Mostly exhausted … time for IPv6!
- Private IP addresses
  - Can be used freely within private networks (home, small company)
  - 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16
  - Need public IP address(es) and NAT to connect to global Internet

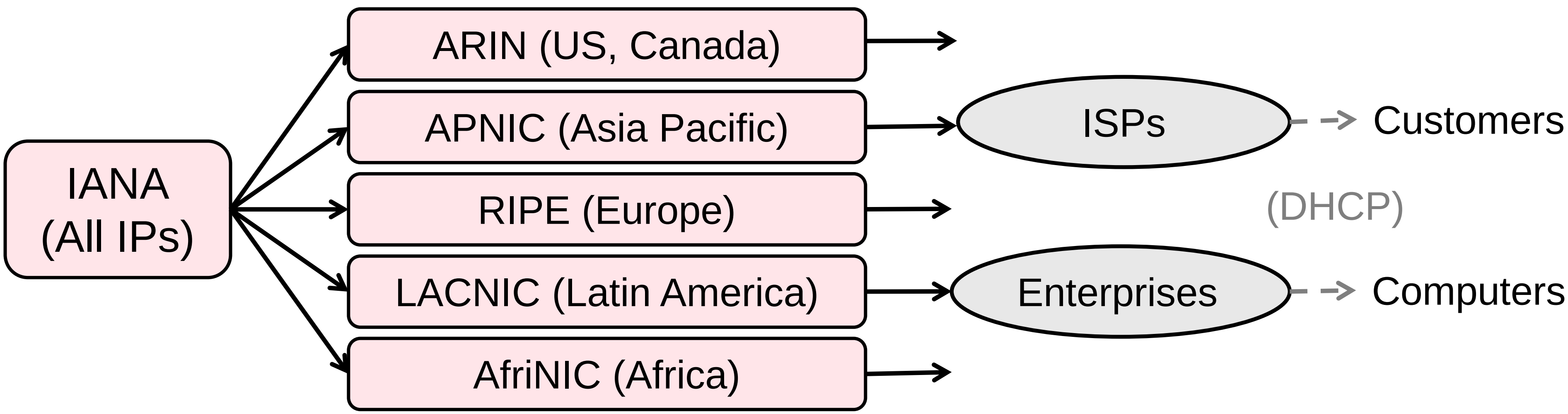
Private Addresses

- Private block of IP address for:
  - Computers not connected to the Internet (e.g. factory machines for process control)
  - Computers behind a NAT/firewall (e.g. your home computers)
- Three non-overlapping ranges of IPv4 address for private networks
- These addresses are not routed on the Internet, so no coordination with an IP address registry
- For IPv6 they are called unique local address (ULAs), with routing prefix fc00::/7

Name	CIDR block	Address range	Total addresses	Classful description
24-bit block	10.0.0.0/8	10.0.0.0 – 10.255.255.255	16,777,216	Single Class A
20-bit block	172.16.0.0/12	172.16.0.0 – 172.31.255.255	1,048,576	16 Class B blocks
16-bit block	192.168.0.0/16	192.168.0.0 – 192.168.255.255	65,536	256 Class C blocks

Allocating Public IP Addresses

- Follows a hierarchical process
  - IANA delegates to regional bodies (RIRs)
  - RIRs delegate to companies in their region
  - Companies assign to their customers/computers (DHCP)



IPv6 Address

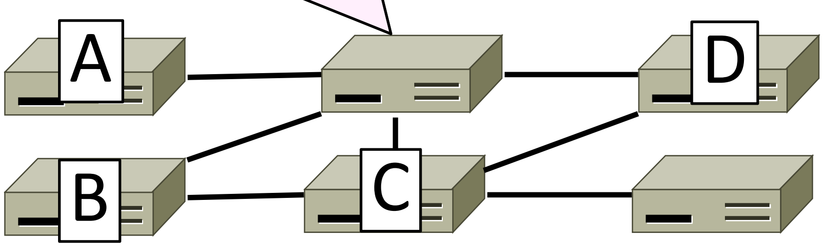
Routing prefix (48 bits or more)	SubnetID (16 bits or fewer)	InterfaceID (64 bits)

- Unicast and anycast address format (for others consult your textbook)
  - Network prefix (routing and subnet id combined) is the most significant 64 bits
  - Routing prefix may vary, a larger prefix size means a smaller subnet id size
  - The 64-bit interface identifier is either:
    - Automatically generated from the MAC address using modified EUI-64 format
    - Obtained from a DHCPv6 server
    - Automatically established randomly
    - Assigned manually
  - 128 bits written in 16-bit hexadecimal chunks
  - Still hierarchical, just more levels

IP Forwarding

- Addresses on one network belong to the same prefix
- Node uses a table that lists the next hop for IP prefixes

Prefix	Next Hop
192.24.0.0/18	D
192.24.12.0/22	B



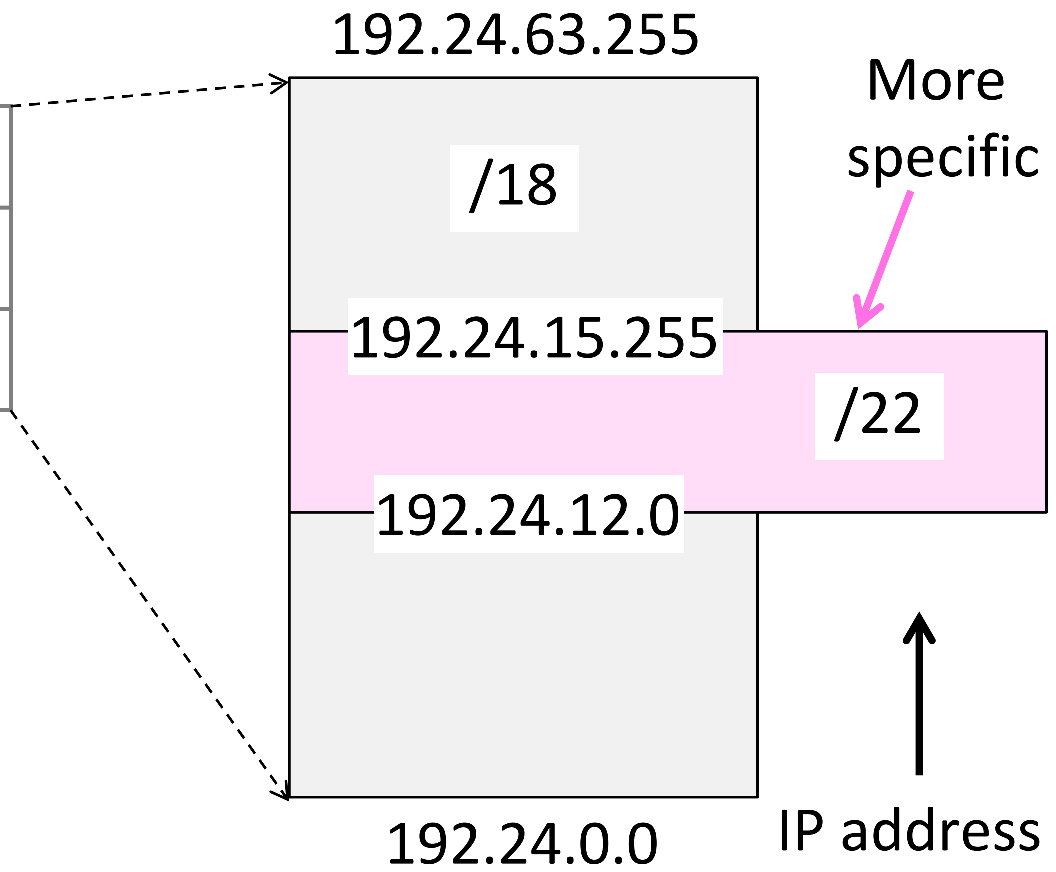
Longest Matching Prefix

- Prefixes in the table might overlap!
  - Combines hierarchy with flexibility

- Longest matching prefix 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 Matching Prefix Example

Prefix	Next Hop
192.24.0.0/18	D
192.24.12.0/22	B



192.24.6.0 →
192.24.14.32 →
192.24.54.0 →

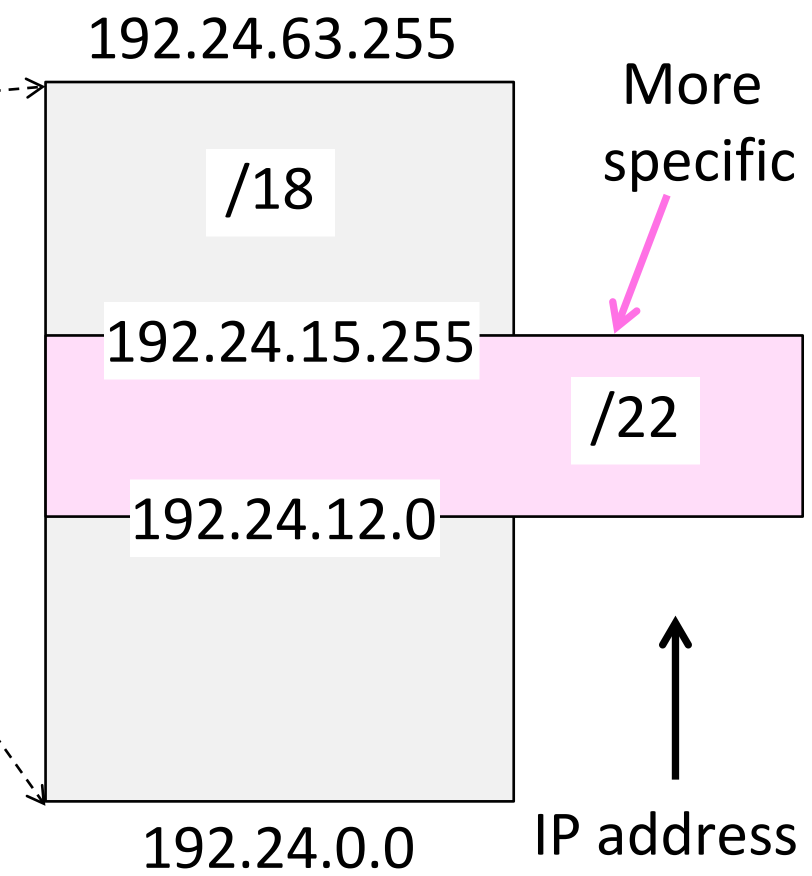
IP Address Work Slide

- Route to D = 192.00011000.00xxxxxx.xxxxxxxx
- Route to B = 192.00011000.000011xx.xxxxxxxx
- 192.24.6.0 = 192.00011000.00000110.00000000
- 192.24.14.32 = 192.00011000.00001110.00010000
- 192.24.54.0 = 192.00011000.00110110.00000000

Longest Matching Prefix Example (2)

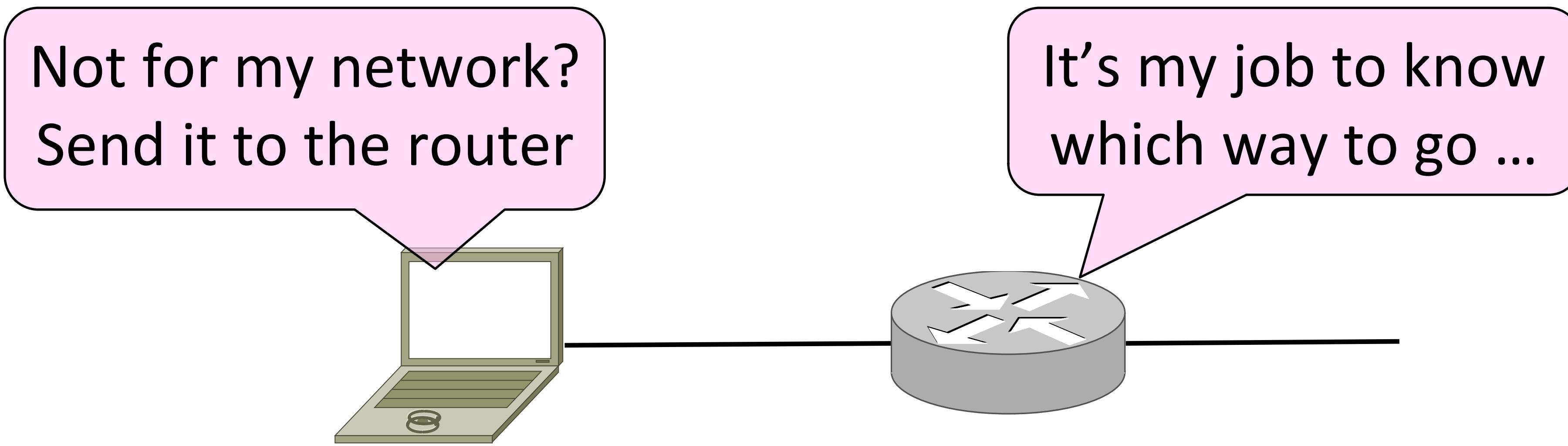
Prefix	Next Hop
192.24.0.0/18	D
192.24.12.0/22	B

192.24.6.0 → **D**
192.24.14.32 → **B**
192.24.54.0 → **D**



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 Forwarding Table

- Give using longest matching prefix
  - 0.0.0.0/0 is a default route that catches all IP addresses

Prefix	Next Hop
My network prefix	Send to that IP
0.0.0.0/0	Send to my router

Flexibility of Longest Matching Prefix

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

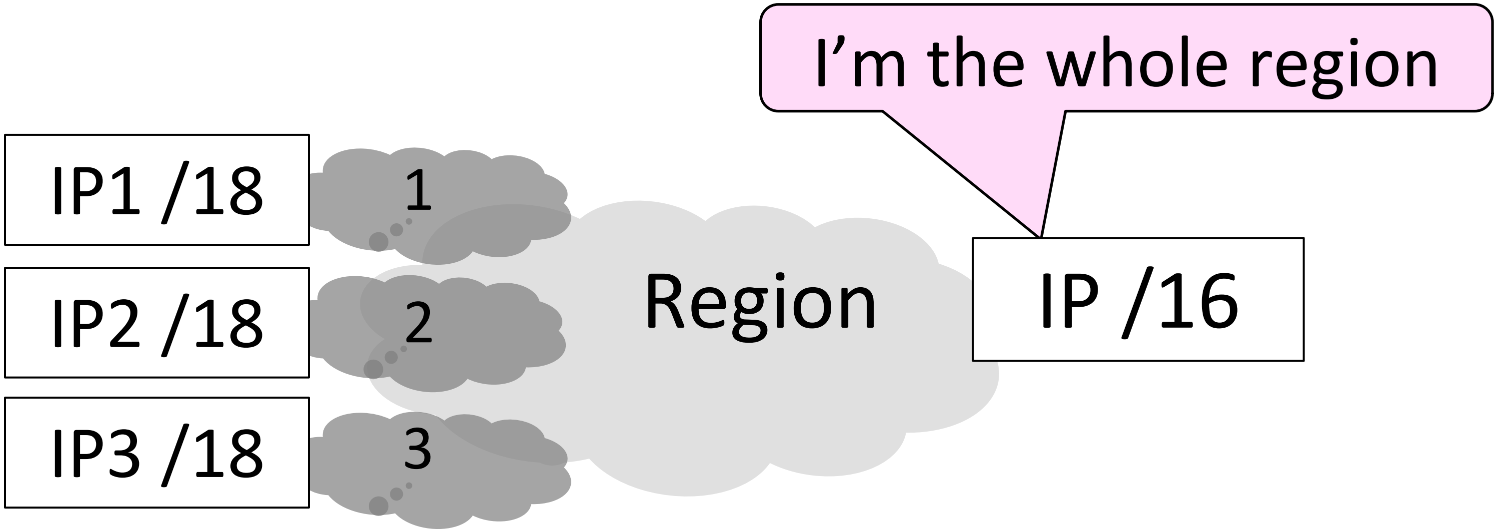
Performance of Longest Matching Prefix

- Uses hierarchy for a compact table
  - Benefits from less specific prefixes

- Lookup more complex than table
  - Used to be a concern for fast routers
  - Not an issue in practice these days

Prefixes and Hierarchy

- IP prefixes already help to scale routing, but we can go further
  - Routers can change prefix lengths without affecting hosts!
  - We can use a less specific (smaller) IP prefix as a name for a region



Subnets and Aggregation

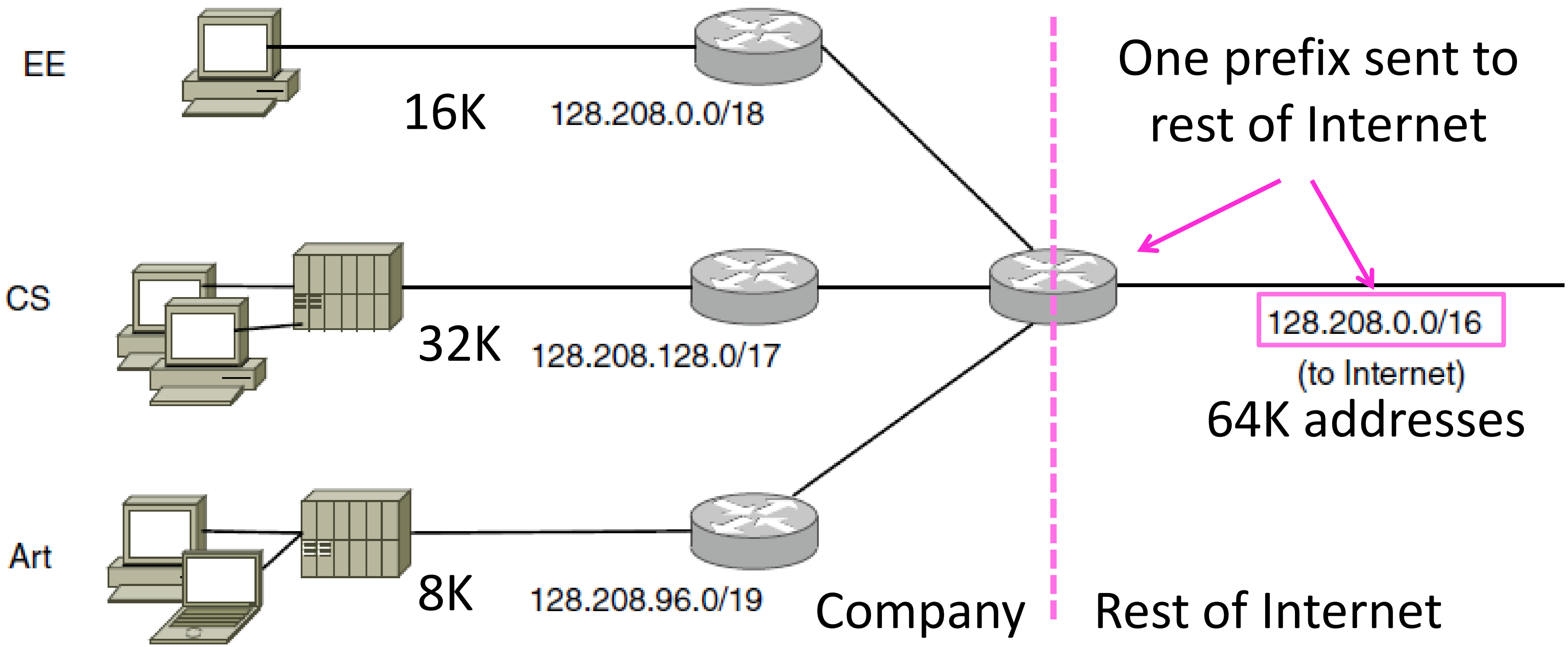
- Two use cases for adjusting the size of IP prefixes; both reduce routing table

1. Subnets
   – Internally split one large prefix into multiple smaller ones

2. Aggregation
   – Externally join multiple smaller prefixes into one large prefix

Subnets

- Internally split up one IP prefix



Subnetting – Mechanism

- Split up one network number into multiple physical networks by taking part of the host number as subnet ID
- 'AND' operation between IP and mask
- Helps allocation efficiency -- can hand out subnets
- Rest of internet does not see subnet structure
  - subnet is purely internal to network
  - aggregates routing info

Network number	Host number

/16 address

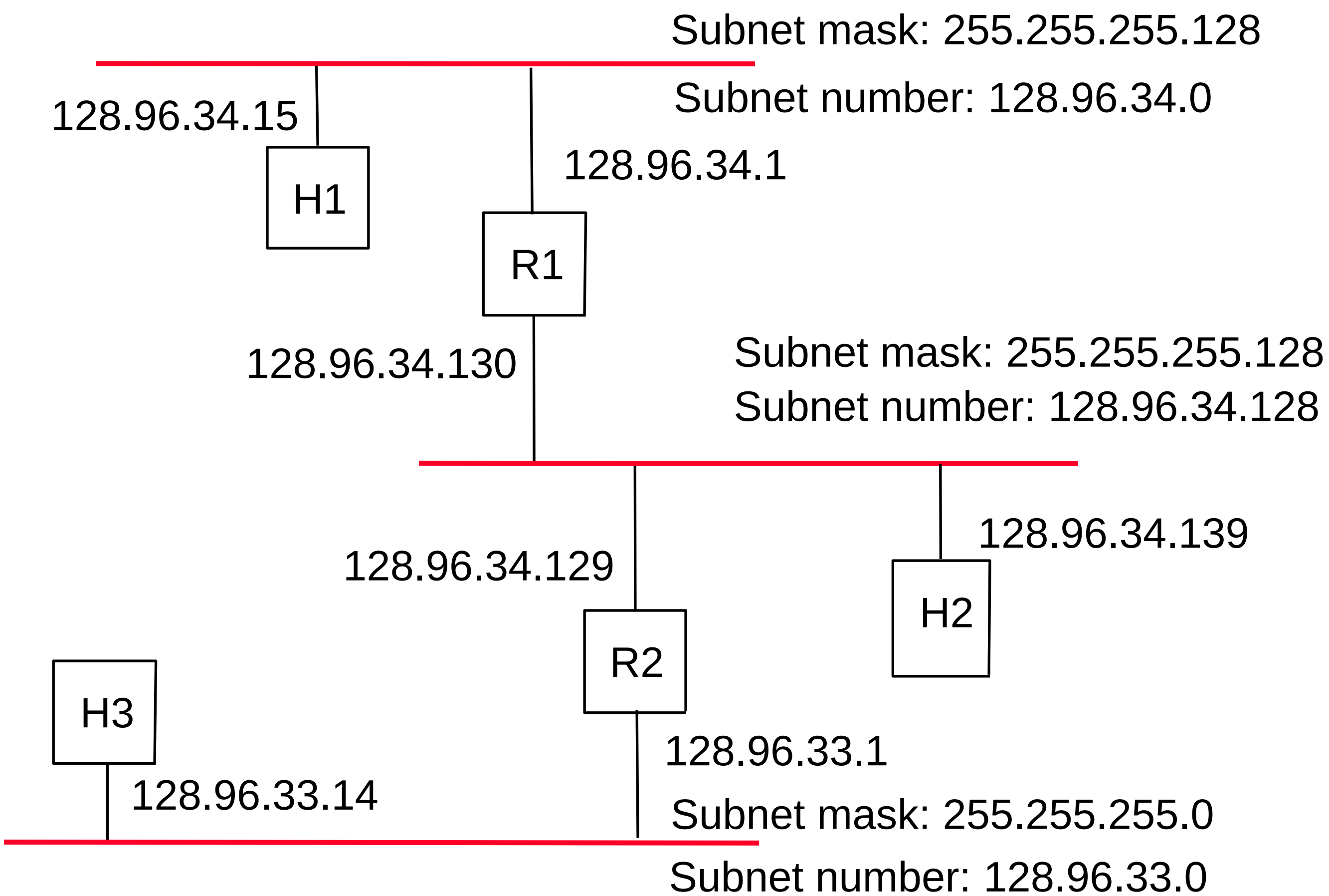
11111111 11111111 11111111	00000000

Subnet mask (255.255.255.0)

Network number	Subnet ID	Host ID

Subnetted address

Subnet Example



Updated Forwarding Routine

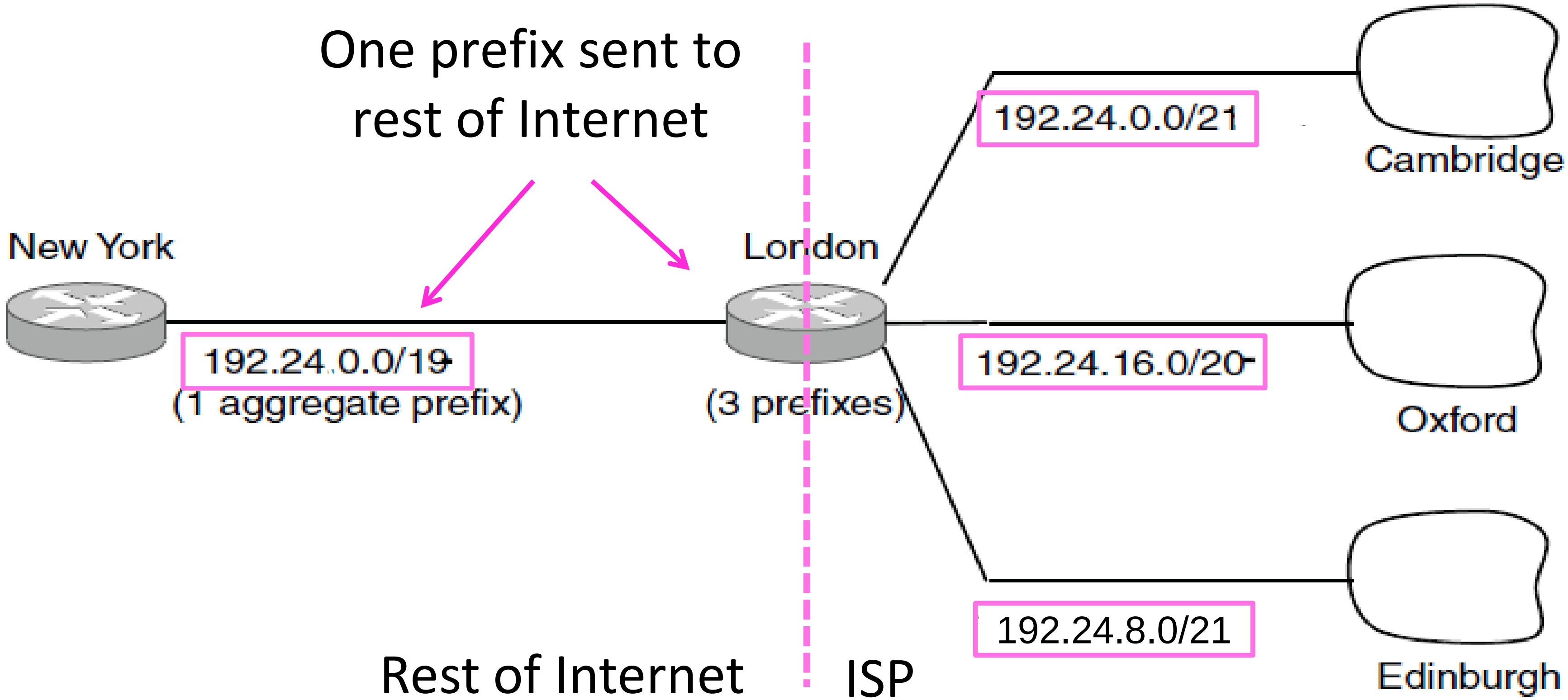
- Used to know network from address (class A, B, C)

- Now need to "search" routing table for right subnet
  - If host: easy, just substitute "subnet" for "network"
  - If router: search routing table for the subnet that the destination belongs to, and use that to forward as before

- (Note that it will get a little more complicated later :-)

Aggregation

- Externally join multiple separate IP prefixes



- 192.24.0.0/21 = 192.24.00000xxx.x
- 192.24.8.0/21 = 192.24.00001xxx.x

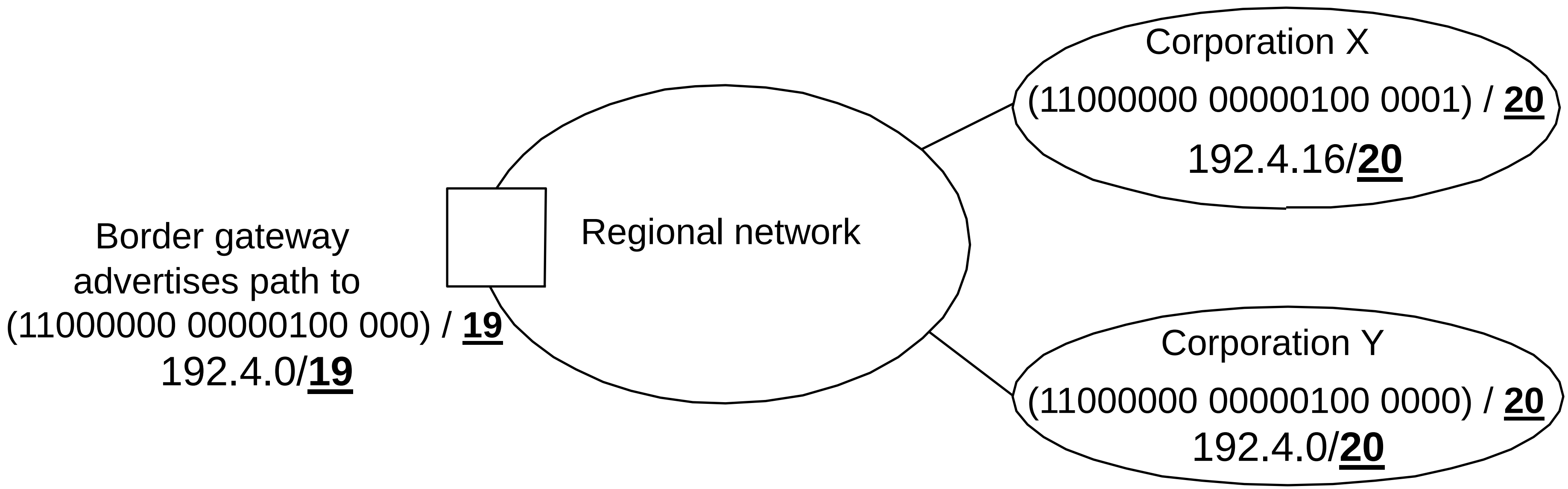
192.24.0.0/20 = 192.24.0000xxxx.x

- 192.24.16.0/20 = 192.24.0001xxxx.x
- 192.24. 0.0/20 = 192.24.0000xxxx.x

192.24.0.0/19 = 192.24.000xxxxx.x

Aggregation Example

- X and Y routes can be aggregated because they form a bigger contiguous range.



- But aggregation isn't always possible. Why?
  - can only aggregate power of 2

IP Forwarding Revisited

- Routing table now contains routes to “prefixes”
  - IP address and length indicating what bits are fixed
- Now need to “search” routing table for longest matching prefix, only at routers
  - Search routing table for the prefix that the destination belongs to, and use that to forward as before
  - There can be multiple matches; take the longest prefix (why?)
- This is the IP forwarding routine used at routers.

Scalability Concerns

- Size of routing tables
  - Hierarchical routing: reduces the size of the routing tables for many routers inside the AS.
  - Hierarchical naming and route aggregation: reduce the size of the routing tables at the border routers.
- Volume of routing messages
  - Hierarchical routing: routing messages are fewer, since fewer routes need to be updated in a smaller graph (AS).
  - Hierarchical naming and route aggregation: routing messages are fewer by transmitting fewer aggregating entries.
- Amount of routing computation
  - Hierarchical routing: smaller graph size, implies reduced routing computation for all routers.
  - Hierarchical naming and route aggregation: border router needs to calculate fewer routes to aggregate destinations (block of IPs addresses).

Key Concepts

- Hierarchical address allocation helps routing scale
  - Addresses are constrained by topology
  - Only need to advertise and compute routes for networks
  - Hide internal structure within a domain via subnets
  - Keep host simple and let routers worry about routing
- The combination of both hierarchical routing and naming to provide topological and address structure is the secret to scale to very large number of nodes!