# CS 3640: Introduction to Networks and Their Applications

Fall 2023, Lecture 10: Addressing in the network layer

Instructor: Rishab Nithyanand
Teaching Assistant: Manisha Keim

# Announcements

- **Assignment 2 due 9/28.**

- **Assignment 3**
  - Releasing on 9/28. Due on 10/12.
  - First group programming project.
    - Socket programming to implement ping and traceroute.
    - 2-5/group. Declare groups to Manisha and me by ICON message before 10/2.

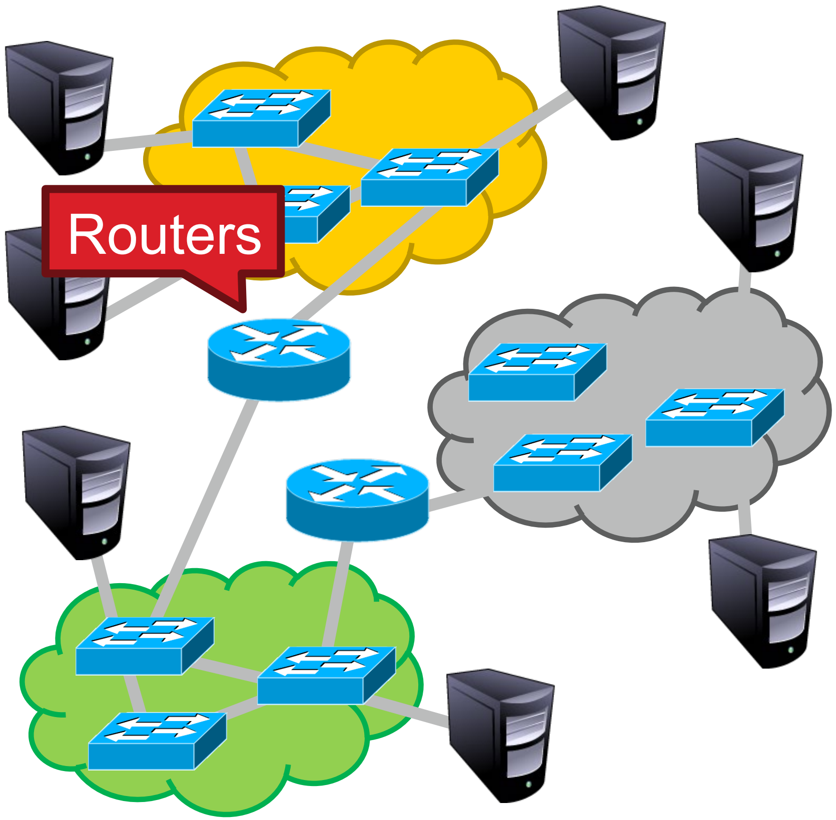
# Today’s class

1. The role of the network layer

2. Network layer addressing

# What the network layer does

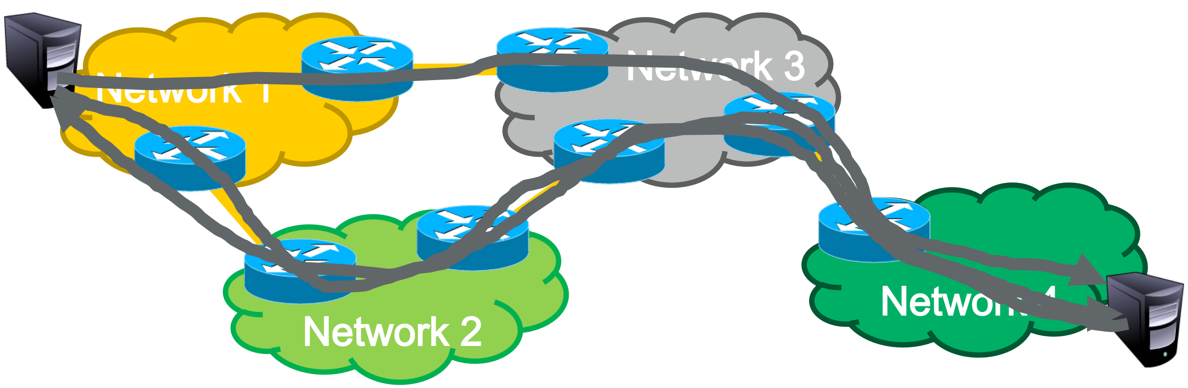
- **It connects multiple LANs with each other.**
  - Facilitate communication between incompatible link layers (e.g., Wi-Fi and Ethernet networks).
  - The connected networks are an "internetwork". Example: The Internet.
  - Interconnections are made using network-layer devices called "routers".




IOWA

# The structure of the Internet

- **The Internet is a collection of interconnected networks.**
  - It is not organized to have a specific topology and technologies in each network vary significantly.
  - Packets travel from source to destination by hopping through networks
    - Routers "peer" (connect) with different networks
    - At each hop a router sends packets to one of its peer networks
  - Remember: The Internet is packet-switched
    - **Discuss: What are the implications of this?**
      - Packets between the same source and destination can take different routes.



# Forwarding (link layer) vs. Routing (network layer)

- **Discuss: How do you plan a long road trip from Iowa City to New York City?**

- You plan the trip at a high-level: We'll drive from Iowa to Illinois to Ohio to Pennsylvania to New Jersey to New York.
  - This is routing: What should be the next state (network) we travel to?

- Then you plan the day-to-day bits: We'll use Hwy 6 to get to the I-80 ramp, then take exit X to get on the I-76.
  - This is forwarding: How can we reach the next state (network) from where we are?

# Functions of the network layer

- **What are the functions of the network layer?**
  - Addressing: How do we specify who we want to talk to on the Internet?
  - Routing: How do we make sure our packets get to their destination?
- **Discuss: Why not use the link layer? It does both of these for LANs.**
  - Link layer protocols were not designed to be scalable to billions of end-hosts.
  - Link layers can be different in different networks!
- **What makes the network layer special?**
  - The core protocols are implemented in every host and router on the "internet"! There is no communication between networks without the network layer.
  - **Discuss: What kind of service guarantees can the network layer provide if it has to serve every network on the Internet?**
    - Serve the lowest common denominator. Best-effort but no guarantees.

# Today’s class

1. The role of the network layer

2. Network layer addressing

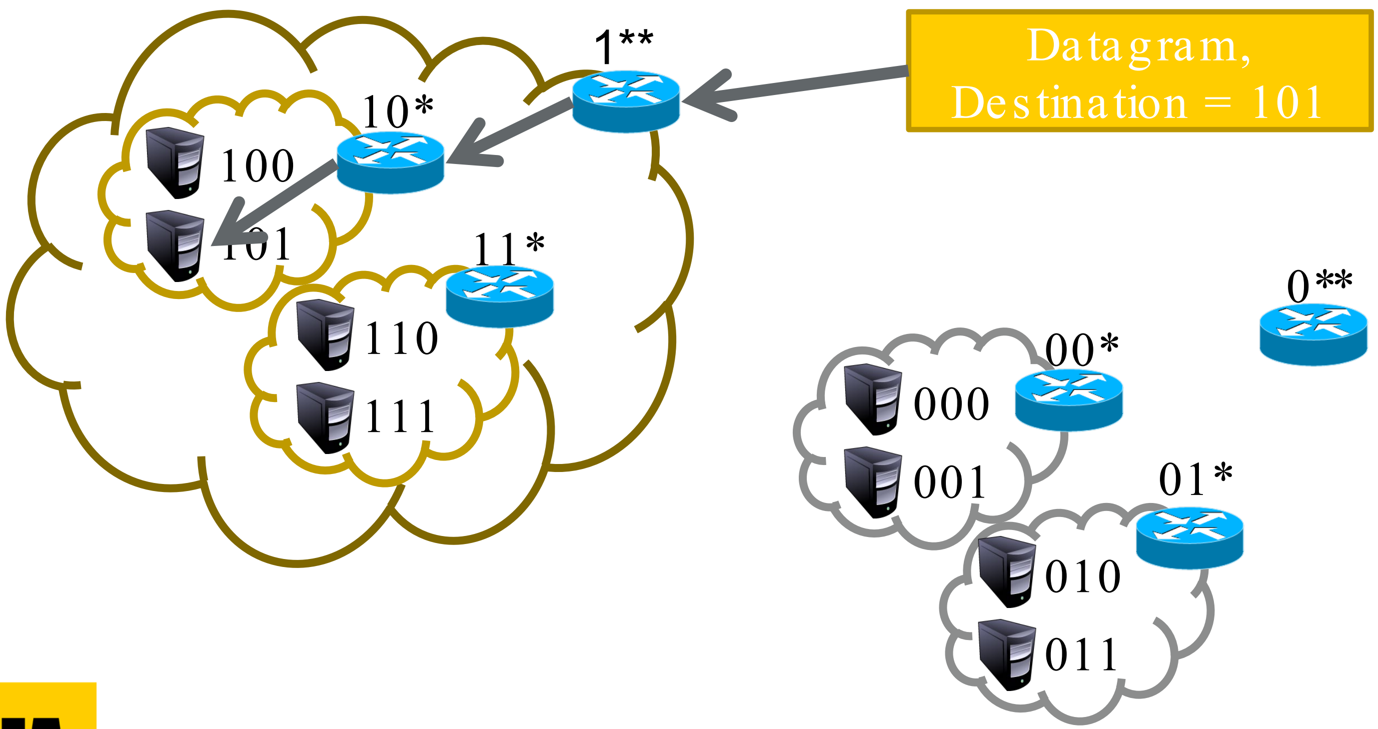
# Addressing options

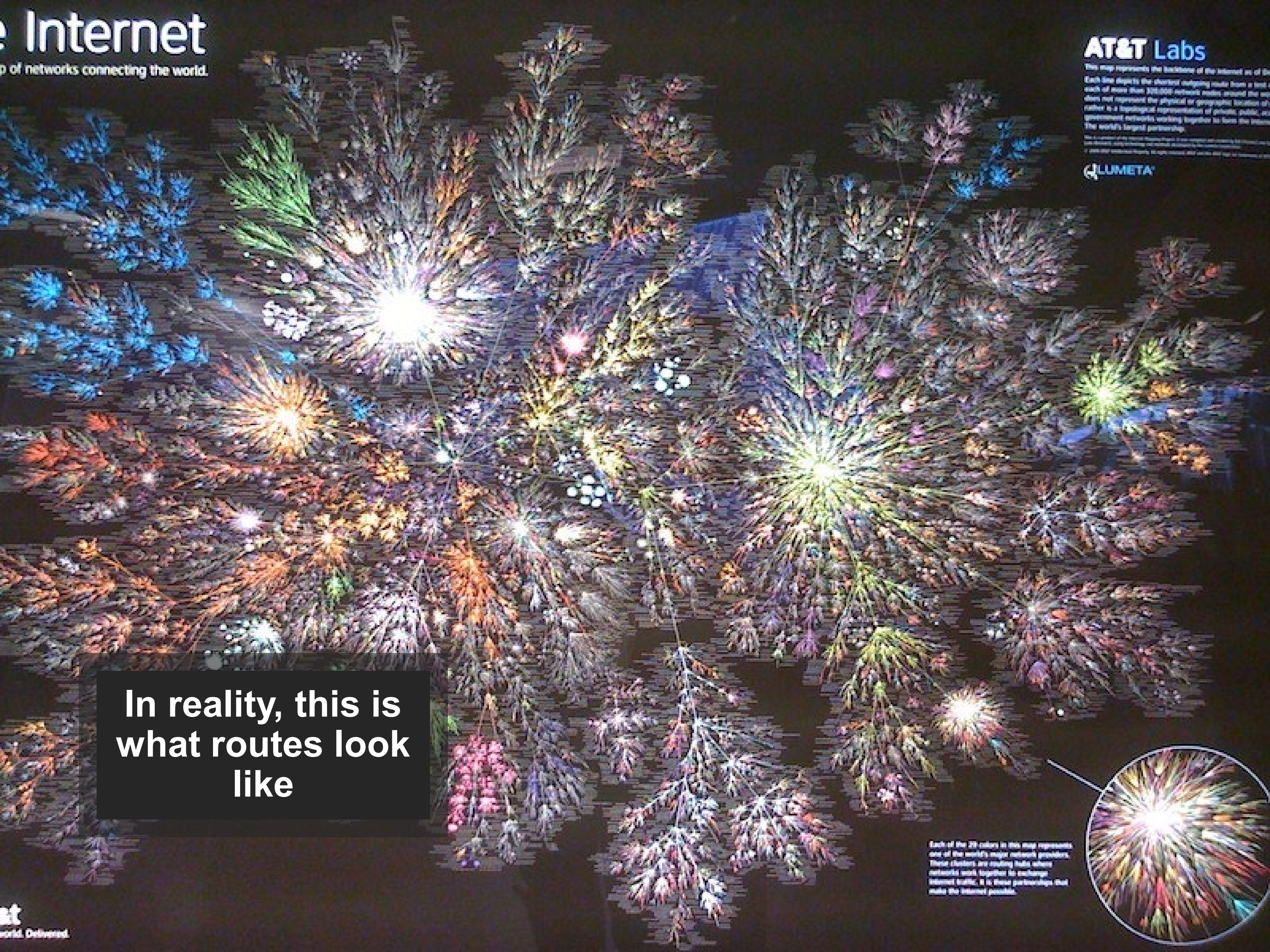
- **Discuss: How did LANs deal with addressing? Why doesn't that work on the Internet?**
  - LANs relied on a "flat" addressing scheme.
  - A unique, essentially **random**, 48-bit string for each device.
  - Problem: Each router has to remember how to reach every other device on the Internet (~23 Billion).

- **Discuss: How do we deal with addressing in the physical world?**
  - We use a hierarchical addressing scheme.
  - Country (least specific) – State – City – Street – Building – Apartment – Person (most specific).
  - Mail routing offices along the way only need to know how to reach the region at some specificity.
    - E.g., UPS New York only needs to know how to get the parcel to UPS Iowa. UPS Iowa needs to get the parcel to UPS Iowa City.

# Network layer addressing

- The Internet uses a hierarchical addressing scheme.






In reality, this is what routes look like

# IP addressing

- **Every device on the Internet has an "IP address"**
  - IP (Internet Protocol) is **THE** network layer protocol.
  - There is no Internet without IP!

- **IPv4: 32 bit strings**
  - They are usually written in dotted notation. Example: 192.168.0.1
  - Each of these numbers is a byte.

0 8 16 24 31

| | | | | |
|---|---|---|---|---|
| Decimal | 192 | 168 | 21 | 76 |
| Hex | C0 | A8 | 15 | 4C |
| Binary | 11000000 | 10101000 | 00010101 | 01001100 |

# Routing tables

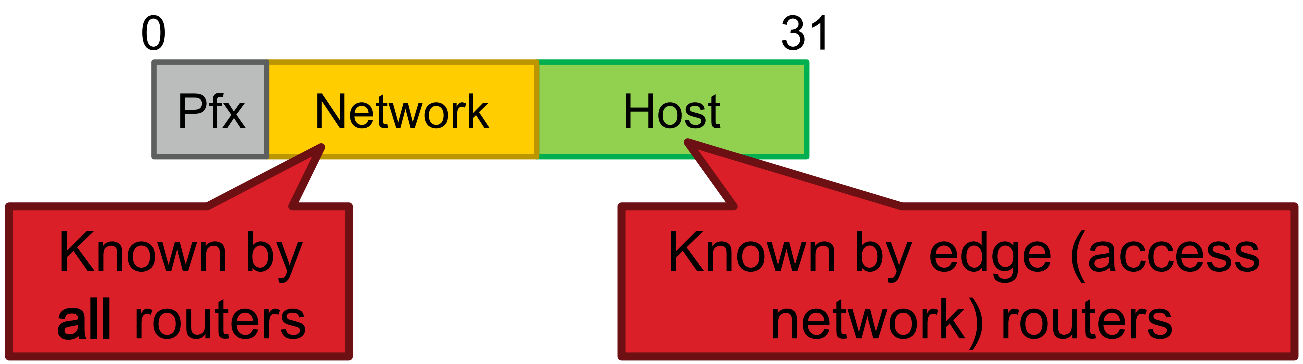
- **Routing tables contain information to help packets get to their destination.**
  - Requirement: For ANY IP address, be able to give the next hop.

- **32 bit addresses: $2^{32}$ addresses! Cannot store next hop info for each address.**

- **Problem isn't just storage. Also speed of lookup.**
  - 4 10G ports operating at capacity will require 176Gbps memory bandwidth!
    - DRAM is 6 Gbps.
  - Need to hit caches and have smaller tables.

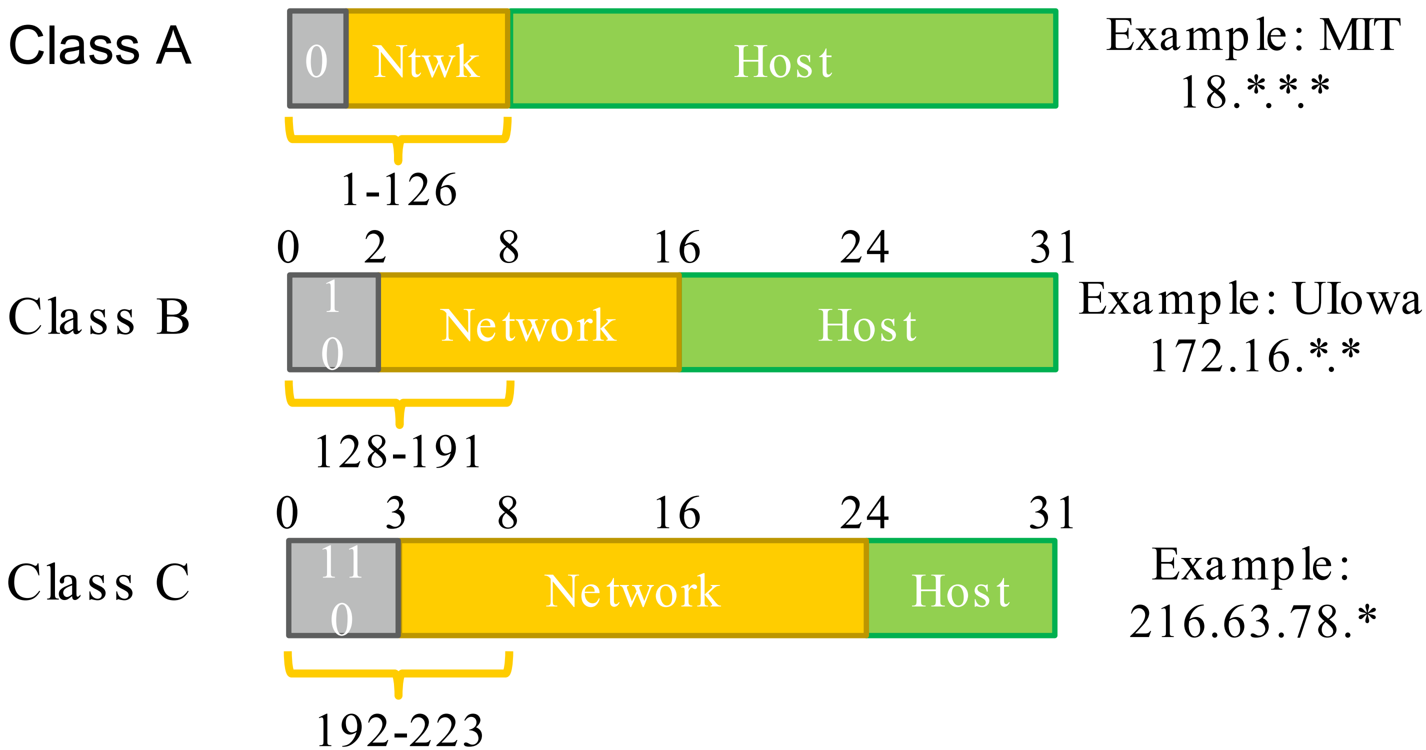
# Hierarchy in IP addressing

- **Each IP address has a "network" string known to all routers and a "host" string known to edge routers (access networks).**




- **How long is the network string and host string?**
  - Depends on the "class" of the network.

# Classes of IP addresses/networks

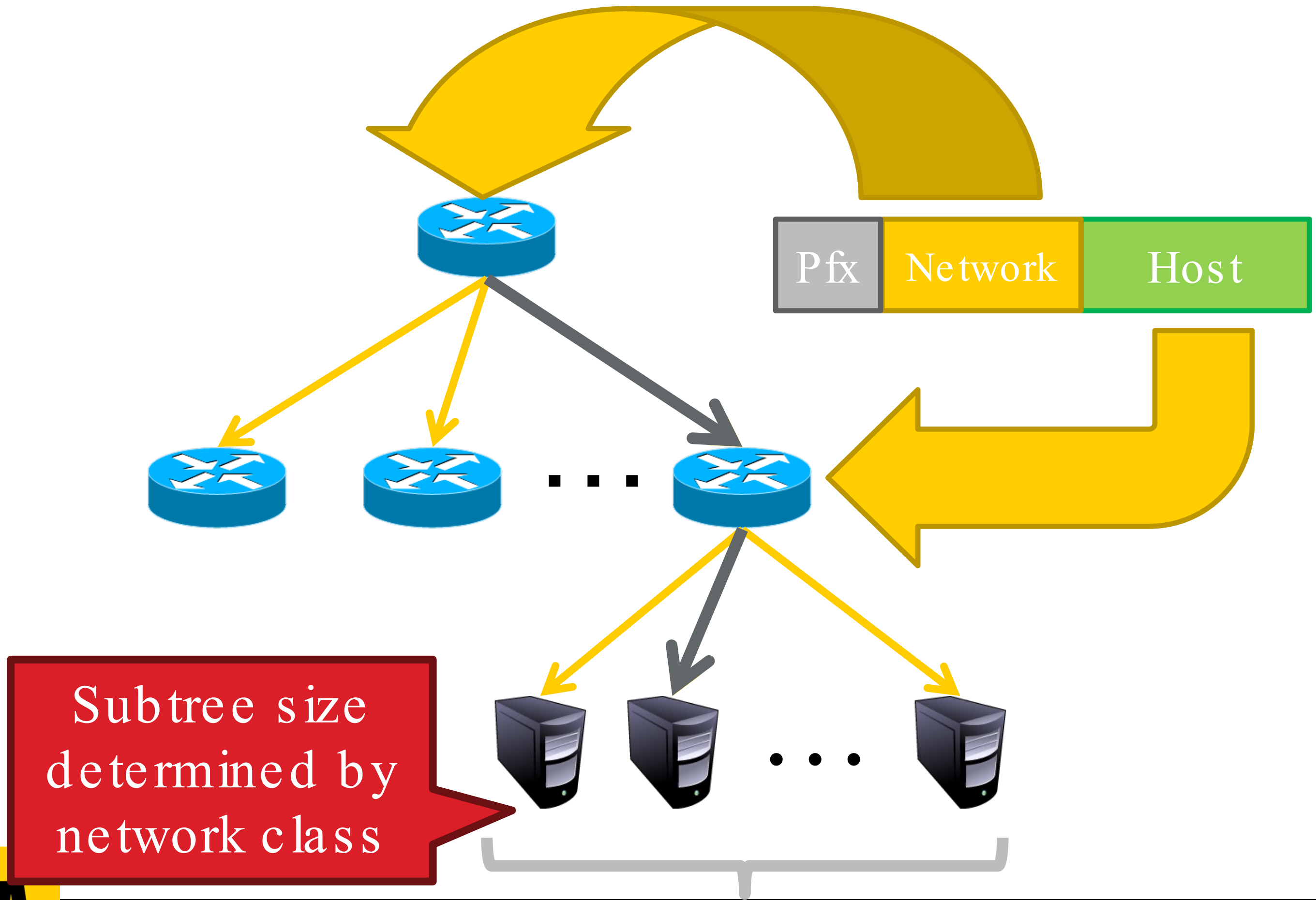



- **IANA (Internet Assigned Number Authority) assigns IP addresses.**
- **You can apply, get a class, and start installing routers that advertise routes to addresses in your class.**

IOWA

# Hierarchy in IP addressing

The hierarchy of IP addresses



# Hierarchy in IP addressing

- Discuss: Do IP address classes solve our problem of memory on routing devices? Why or why not?

Way too big for access networks to remember

| Class | Prefix Bits | Network Bits | Number of network classes | Hosts per network clas |
|---|---|---|---|---|
| A | 1 | 7 | $2^7 - 2 = 126$ (0 and 127 are reserved) | $2^{24} - 2 = 16{,}777{,}214$ (All 0 and all 1 are reserved) |
| B | 2 | 14 | $2^{14} = 16{,}398$ | $2^{16} - 2 = 65{,}534$ (All 0 and all 1 are reserved) |
| C | 3 | 21 | $2^{21} = 2{,}097{,}512$ | $2^8 - 2 = 254$ (All 0 and all 1 are reserved) |
| | | | Total: 2,114,036 | |

Too many network IDs for every router to remember

# The problem with class-based hierarchies

- **Problem: Too many hosts to remember for Class A and Class B access network routers.**
  - **Discuss: How would you fix this?**

Way too big

| Class | Prefix Bits | Network Bits | Number of network classes | Hosts [illegible] network class |
|---|---|---|---|---|
| A | 1 | 7 | $2^7 - 2 = 126$<br>(0 and 127 are reserved) | $2^{24} - 2 = 16{,}777{,}214$<br>(All 0 and all 1 are reserved) |
| B | 2 | 14 | $2^{14} = 16{,}398$ | $2^{16} - 2 = 65{,}534$<br>(All 0 and all 1 are reserved) |
| C | 3 | 21 | $2^{21} = 2{,}097{,}512$ | $2^8 - 2 = 254$<br>(All 0 and all 1 are reserved) |
| | | | Total: 2,114,036 | |

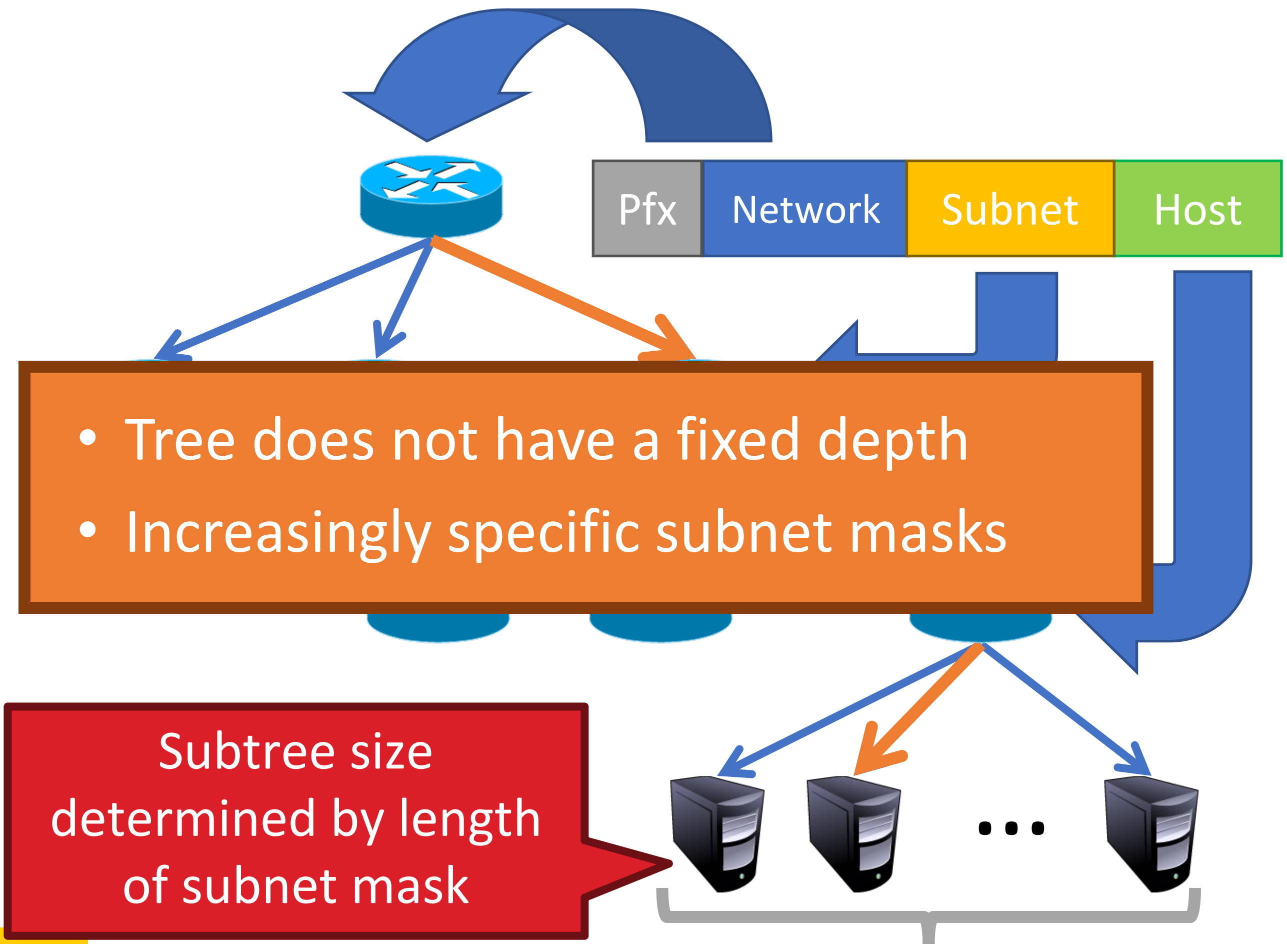
# Reducing router storage with subnets

- **Problem: Too many hosts to remember for Class A and Class B access network routers.**

- **Solution: Tack on another layer to the hierarchy.**
  - Each "network" manages many "subnetworks".
    - The number of hosts in each subnetwork is much smaller.
    - Each access network router only needs to know how to reach a subnet.
    - Each subnet router only needs to know how to reach hosts in it.
  - If it seems like everything on the Internet is a patched solution, its because it is! The designers did not foresee needing to store 1000s of entries on routers.


IOWA

# Reducing storage with subnets



| Pfx | Ntwk | Subnet | Host |
| --- | --- | --- | --- |

# Subnet masks and routing tables

- **A subnet "mask" is a 32 bit string.**
  - 1s as long as the prefix+network+subnet parts of the IP address. It is used by routers to indicate which hosts are reachable in a subnet.

| Address Prefix | Subnet Mask | Prefix After Masking (in Binary) | Next Hop |
| --- | --- | --- | --- |
| 0.0.0.0 | 0.0.0.0 | ******** ******** ******** ******** | Port 4 |
| 18.0.0.0 | 255.0.0.0 | 00010010 ******** ******** ******** | Port 2 |
| 128.42.0.0 | 255.255.0.0 | 10000000 00101010 ******** ******** | Port 3 |
| 128.42.128.0 | 255.255.128.0 | 10000000 00101010 1******* ******** | Port 5 |
| 128.42.222.0 | 255.255.255.0 | 10000000 00101010 11011110 ******** | Port 1 |

- **Discuss: We need to send a packet to 128.42.222.198. This entry is satisfied by 4 entries in the routing table. Which entry should I use? Why?**
  - Match the longest prefix. The row with the longest string of 1s in the subnet mask. It gets me closest to the destination.

IOWA

# The limitations of subnetting

- **Discuss: Do subnets solve the problem of having too many network classes?**
  - No! We have less number of hosts in a class, but a router still needs to know (at least) every network which is allocated a class (at least 2.1 M).

| Class | Prefix Bits | Network Bits | Number of network classes | Hosts per network class |
|---|---|---|---|---|
| A | 1 | 7 | $2^7 - 2 = 126$<br>(0 and 127 are reserved) | $2^{24} - 2 = 16,777,214$<br>(All 0 and all 1 are reserved) |
| B | 2 | 14 | $2^{14} = 16,398$ | $2^{16} - 2 = 65,534$<br>(All 0 and all 1 are reserved) |
| C | 3 | 21 | $2^{21} = 2,097,512$ | $2^8 - 2 = 254$<br>(All 0 and all 1 are reserved) |
| | | | Total: 2,114,036 | |

Way too big

IOWA

# Reducing routing table sizes with CIDR (Classless Inter Domain Routing)

- **Key idea: Get rid of IP classes in routing tables.**
- **Create an arbitrary split between networks and hosts.**
  - The split is specified by a bitmask (net mask).
- **CIDR representation example:**
  - Let 129.10.0.1 be our host IP. Let 255.255.0.0 be our net mask
    - Host IP: 10000001 00001010 00000000 00000001
    - Net mask: 11111111 11111111 00000000 00000000
  - CIDR representation: 129.10.0.1 & 255.255.0.0: 129.10.0.0/16
  - Saying you can route to 129.10.0.0/16 means you're saying you can reach all IPs which have a CIDR representation of 129.10.0.0/16
    - this includes 129.10.0.0/17, …, 129.19.0.0/32.

IOWA

# Reducing routing table sizes with CIDR (Classless Inter Domain Routing)

| Prefix | Netmask | Prefix After Masking (in Binary) | IP Address Range |
|---|---|---|---|
| 207.46.0.0 | 19 | 11001111 00101110 000***** ******** | 207.46.0 – 31.* |
| 207.46.32.0 | 19 | 11001111 00101110 001***** ******** | 207.46.32 – 63.* |
| 207.46.64.0 | 19 | 11001111 00101110 010***** ******** | 207.46.64 – 95.* |
| 207.46.128.0 | 18 | 11001111 00101110 10****** ******** | 207.46.128 – 191.* |
| 207.46.192.0 | 18 | 11001111 00101110 11****** ******** | 207.46.192 – 255.* |

Hole in the Routing Table: No coverage for 96 – 127

Missing entry: 207.46.96.0/19

# Reducing routing table sizes with CIDR (Classless Inter Domain Routing)

Same length netmask

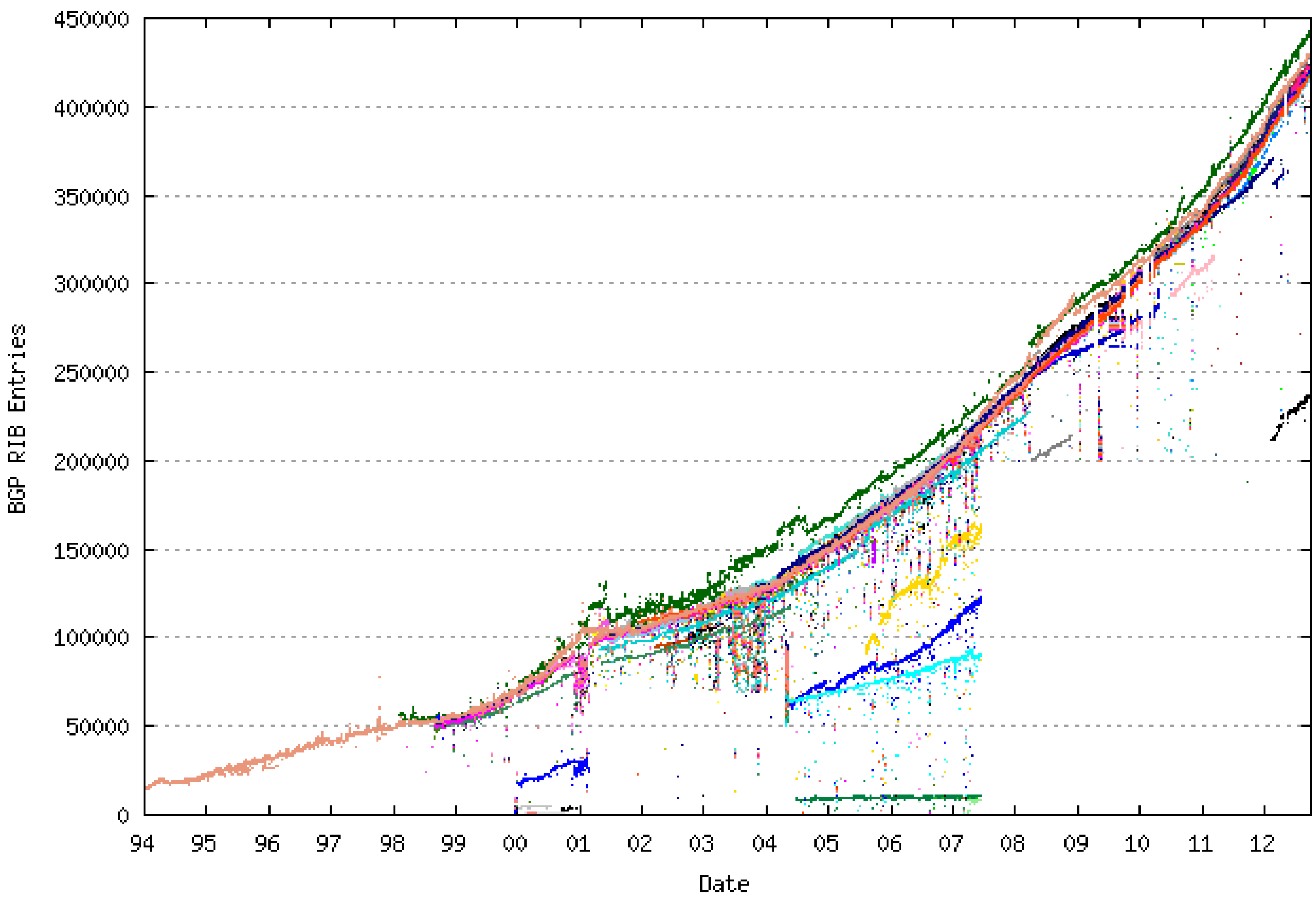
All bits match except for the last one

Same port

| Prefix | etmask | Prefix After Maskin Binary) | Next Hop |
|---|---|---|---|
| 207.46.0.0 | 17 | 11001111 00101110 0******* ******** | Port 1 |
| 207.46.128.0 | 18 | 11001111 00101110 10****** ******** | Port 2 |
| 207.46.192.0 | 18 | 11001111 00101110 11****** ******** | Port 3 |

Aggregation allows multiple routes to be compressed together to shrink the size of the routing table

# Size of CIDR routing tables: ~450K entries for the entire internet



# What you should remember from this lecture

- **Why doesn’t “flat addressing” work on the Internet? What does?**
  - Flat addressing is not scalable. Too many routing table entries for every router.
  - Hierarchical addressing scales better. Don’t need to have entries for all routers.
    - Need highly specific information only for routers in their own network.

- **Class-based addressing is too coarse.**
  - Class A networks are too big for access network routers.
  - Class C networks are too many for everyone else.

- **CIDR introduces arbitrary hierarchies using netmasks.**
  - Improves scalability by allowing more aggregation at routers.

- **At each hop routing decisions are made using “longest prefix matching”.**