IP Addressing and Forwarding

COS 461: Computer Networks
Spring 2011

Mike Freedman

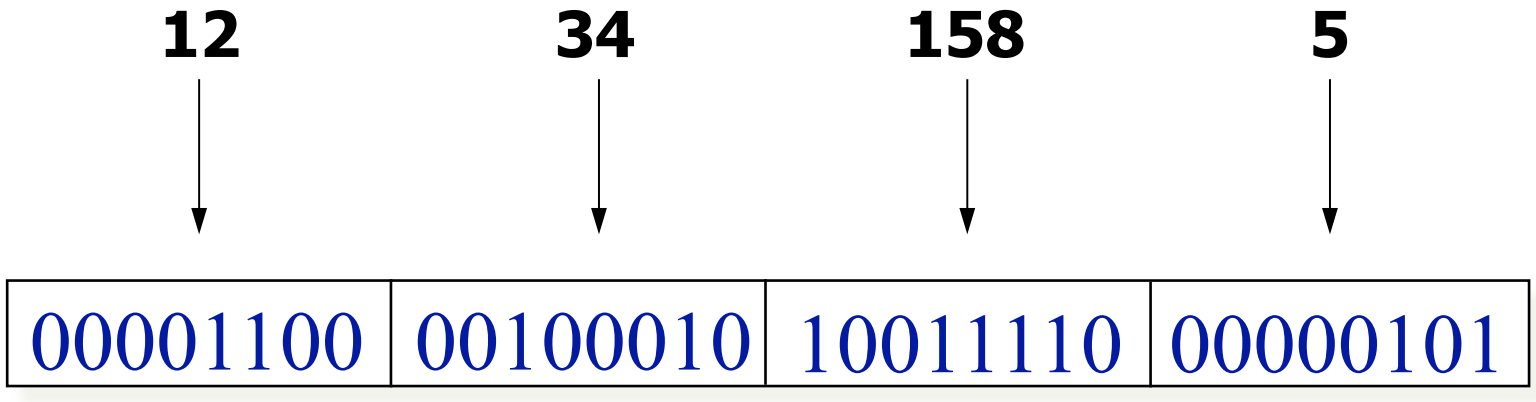
http://www.cs.princeton.edu/courses/archive/spring11/cos461/

Goals of Today's Lecture

- IP addresses
  - Dotted-quad notation
  - IP prefixes for aggregation
- Address allocation
  - Classful addresses
  - Classless InterDomain Routing (CIDR)
  - Growth in the number of prefixes over time
- Packet forwarding
  - Forwarding tables
  - Longest-prefix match forwarding
  - Where forwarding tables come from

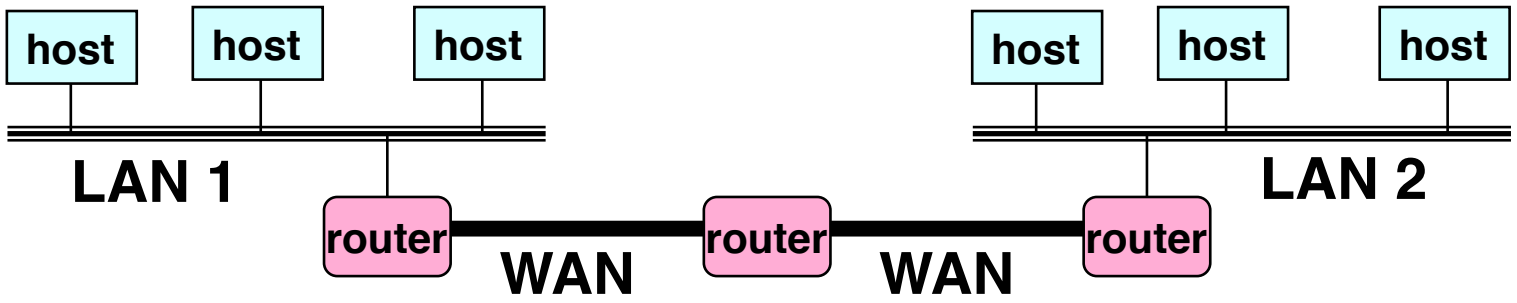
IP Address (IPv4)

- A unique 32-bit number
- Identifies an interface (on a host, on a router, …)
- Represented in dotted-quad notation



Grouping Related Hosts

- The Internet is an "inter-network"
  - Used to connect *networks* together, not *hosts*
  - Needs way to address a network (i.e., group of hosts)

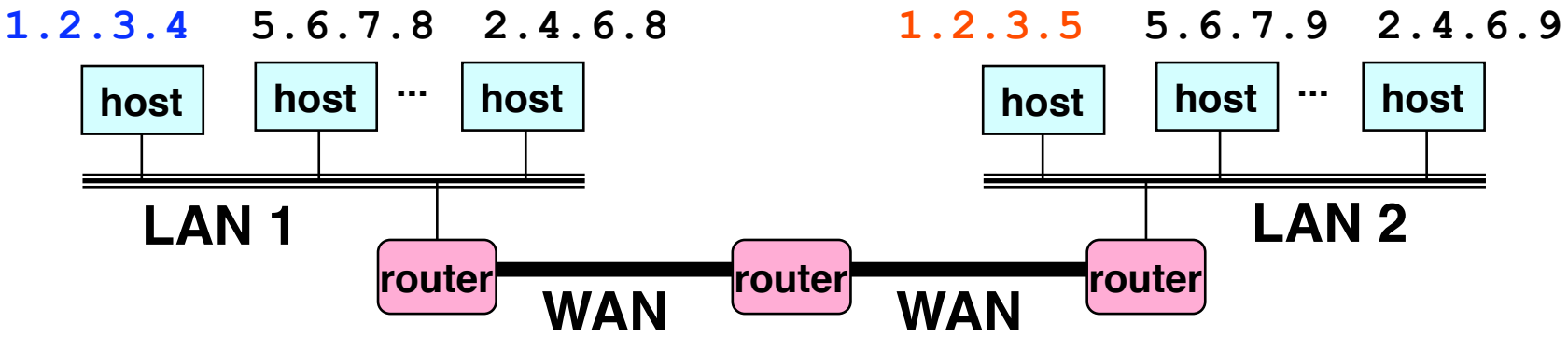


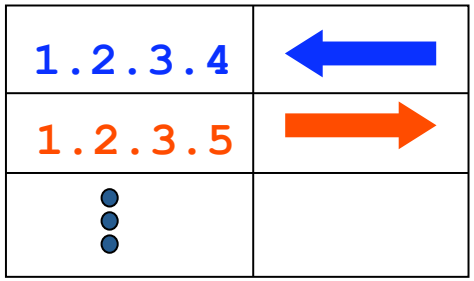
LAN = Local Area Network

WAN = Wide Area Network

Scalability Challenge

- Suppose hosts had arbitrary addresses
  - Then every router would need a lot of information
  - …to know how to direct packets toward *every* host





forwarding table a.k.a. FIB (forwarding information base)

Scalability Challenge

- Suppose hosts had arbitrary addresses
  - Then every router would need a lot of information
  - …to know how to direct packets toward *every* host

- Back of envelop calculations
  - 32-bit IP address: 4.29 billion (2^{32}) possibilities
  - How much storage?
    - Minimum: 4B address + 2B forwarding info per line
    - Total: 24.58 GB just for forwarding table
  - What happens if a network link gets cut?

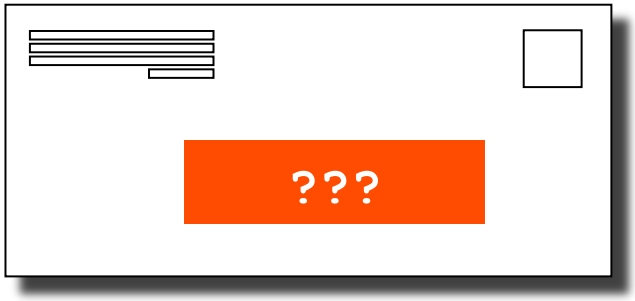
Standard CS Trick

Have a scalability problem?

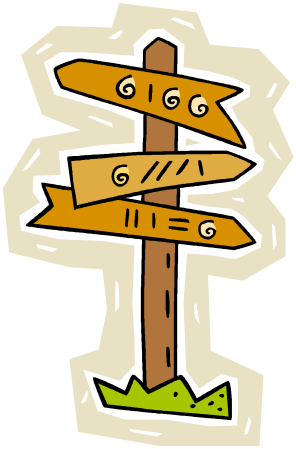
Introduce hierarchy…

Hierarchical Addressing in U.S. Mail

- Addressing in the U.S. mail
  - Zip code: 08540
  - Street: Olden Street
  - Building: 35
  - Room: 308
  - Occupant: Mike Freedman

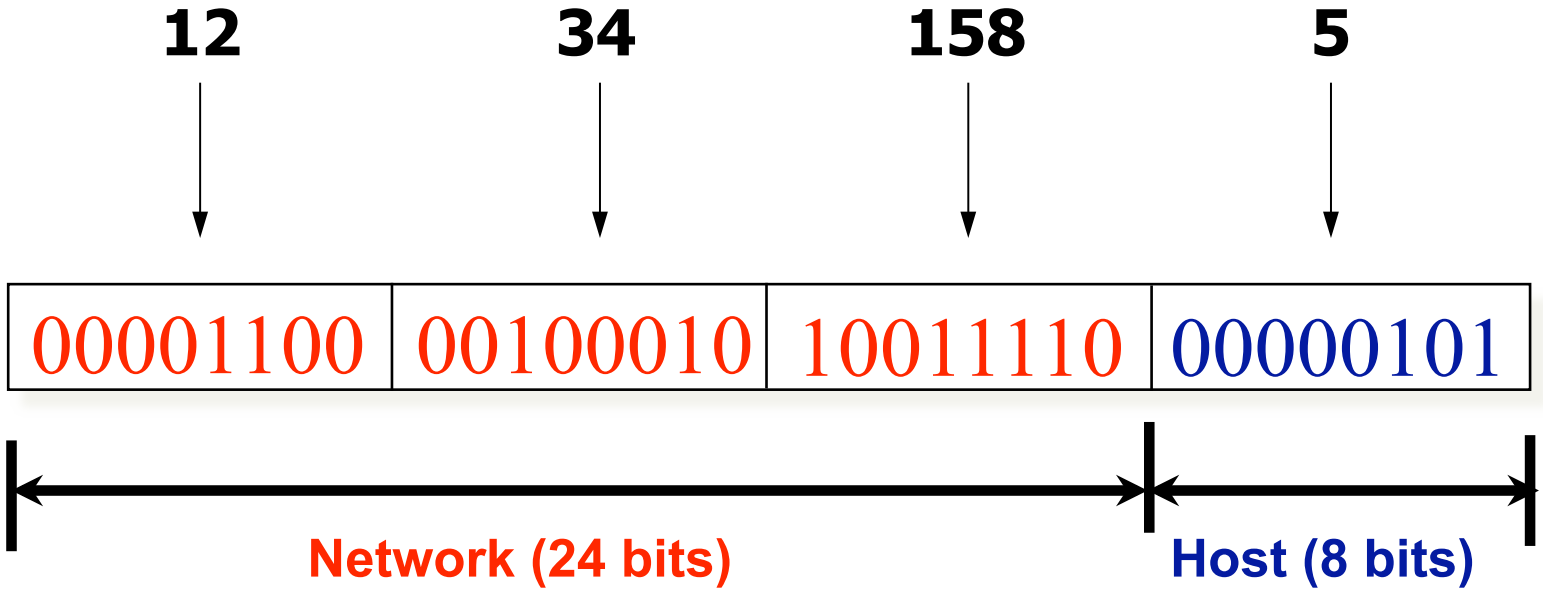


- Forwarding the U.S. mail
  - Deliver to post office in zip code
  - Assign to mailman covering street
  - Drop into mailbox for building/room
  - Give to appropriate person



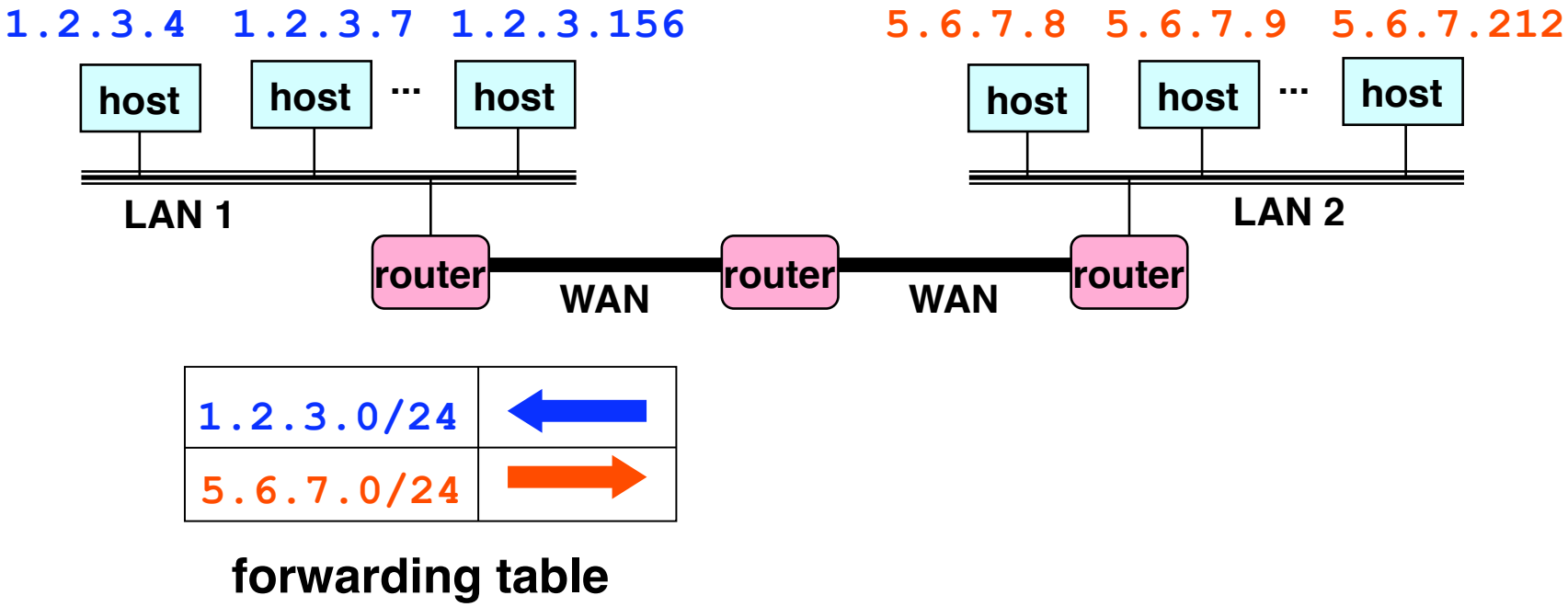
Hierarchical Addressing: IP Prefixes

- IP addresses can be divided into two portions
  - Network (left) and host (right)
- 12.34.158.0/24 is a 24-bit **prefix**
  - Which covers 2^8 addresses (e.g., up to 255 hosts)



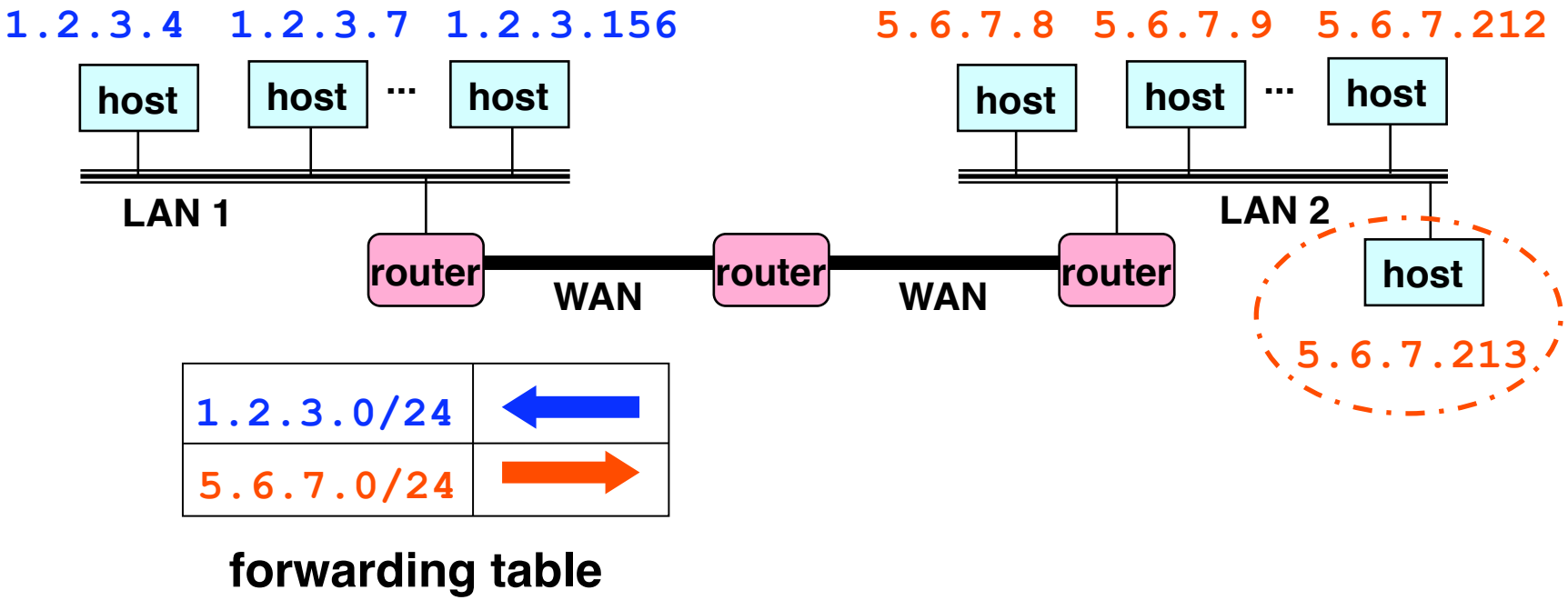
Scalability Improved

- Number related hosts from a common subnet
  - 1.2.3.0/24 on the left LAN
  - 5.6.7.0/24 on the right LAN



Easy to Add New Hosts

- No need to update the routers
  - E.g., adding a new host 5.6.7.213 on the right
  - Doesn't require adding a new forwarding-table entry



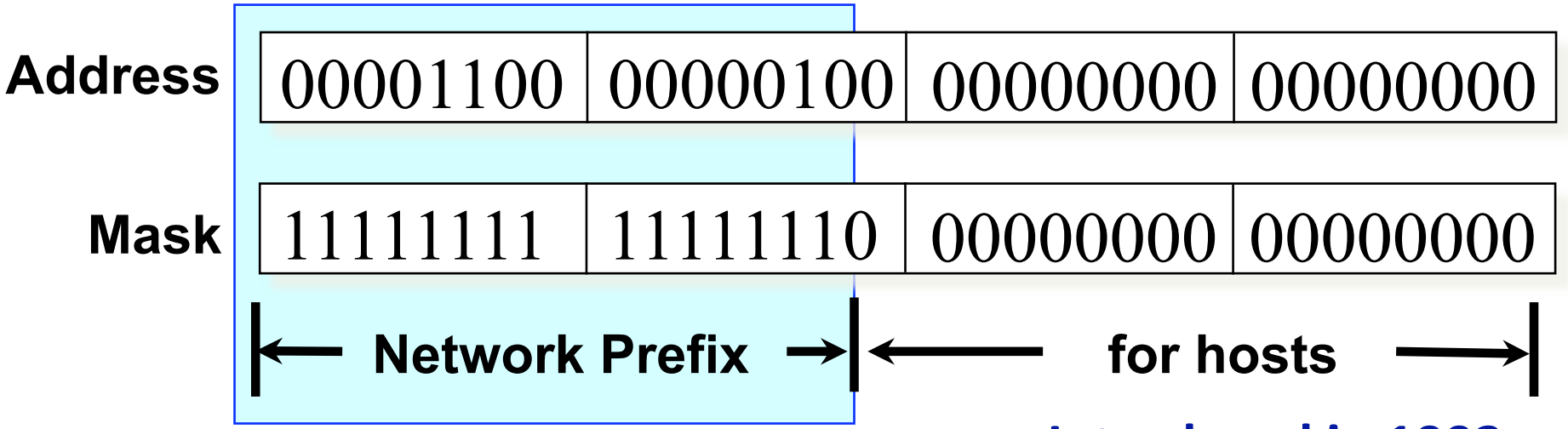
Address Allocation

Classful Addressing

- In olden days, only fixed allocation sizes
  - Class A: 0* : Very large /8 blocks (MIT has 18.0.0.0/8)
  - Class B: 10* : Large /16 blocks (Princeton has 128.112.0.0/16)
  - Class C:110* : Small /24 blocks
  - Class D: 1110* : Multicast groups
  - Class E:11110* : Reserved for future use
- Why folks use dotted-quad notation!
- Position of "first 0" made it easy to determine class of address in hardware (hence, how to parse)

Classless Inter-Domain Routing (CIDR)

- IP prefix = IP address (AND) subnet mask
- IP Address : 12.4.0.0, Mask: 255.254.0.0



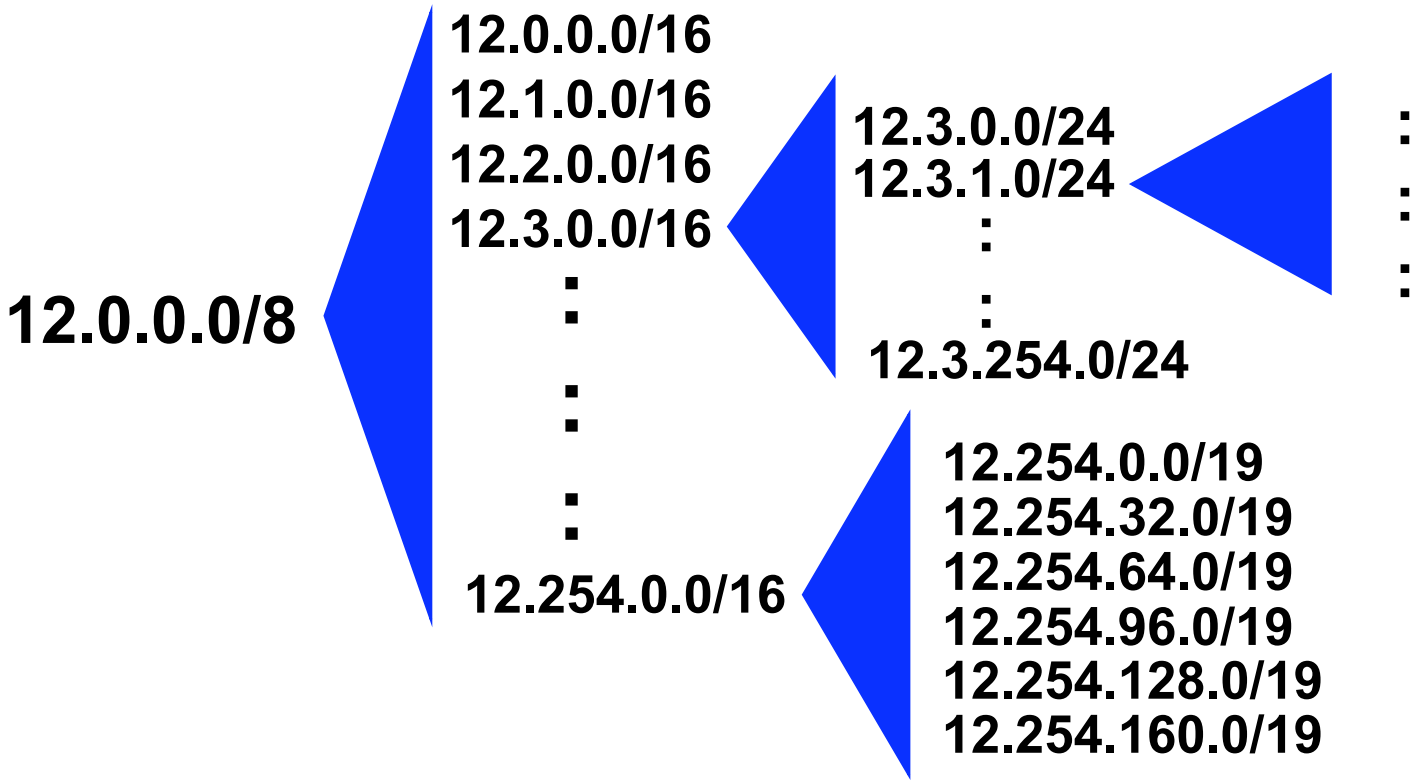
Written as 12.4.0.0/15

Introduced in 1993
RFC 1518-1519

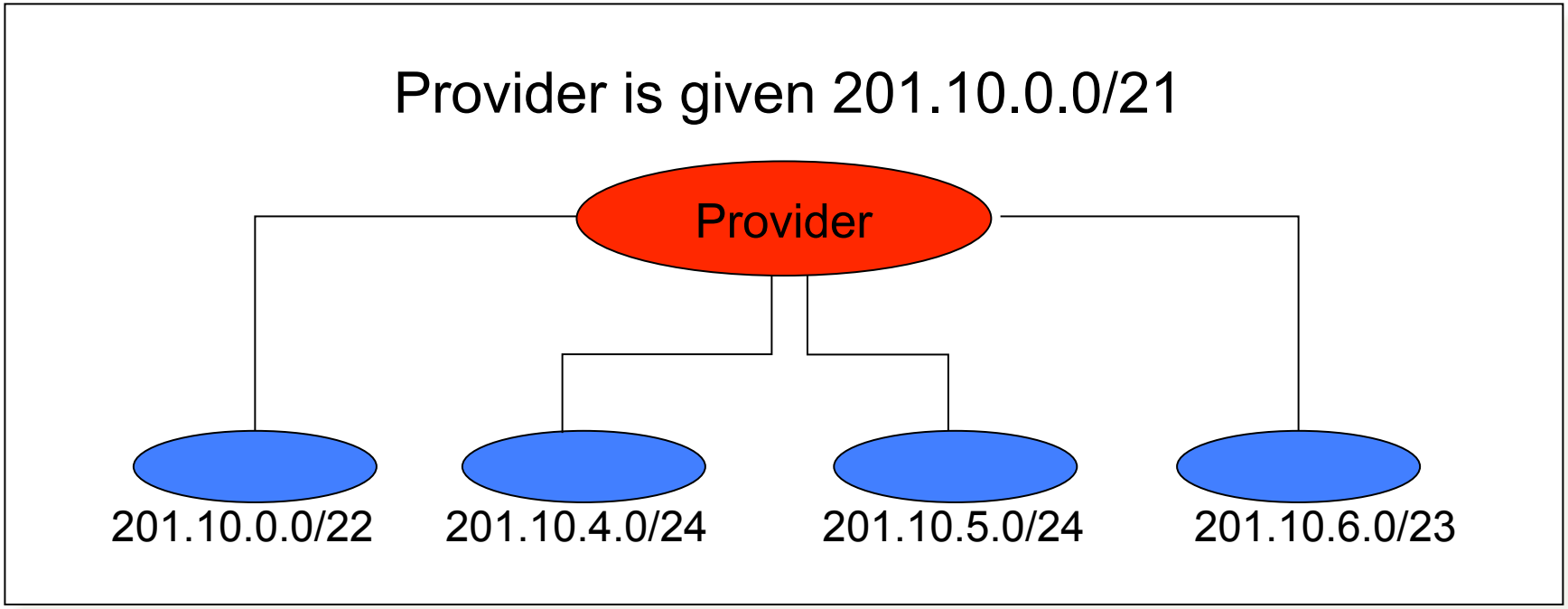
```
$ ifconfig
en1: flags=8863<UP,BROADCAST,…,MULTICAST> mtu 1500
     inet 192.168.1.1 netmask 0xffffff00 broadcast 192.168.1.255
     ether 21:23:0e:f3:51:3a
```

CIDR: Hierarchal Address Allocation

- Prefixes are key to Internet scalability
  - Address allocated in contiguous chunks (prefixes)
  - Routing protocols and packet forwarding based on prefixes
  - Today, routing tables contain ~350,000 prefixes (vs. 4B)

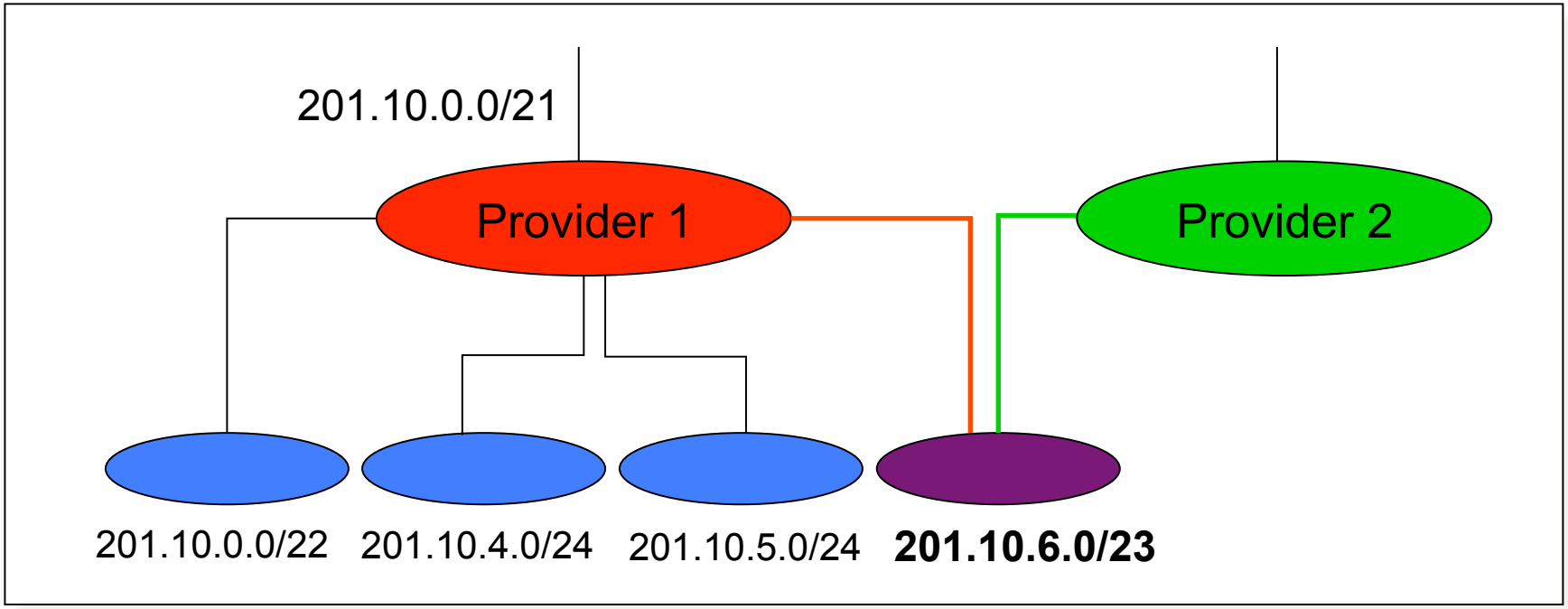


Scalability: Address Aggregation



- Other Internet Routers just know how to reach 201.10.0.0/21
- Provider can direct IP packets to appropriate customer

But, Aggregation Not Always Possible

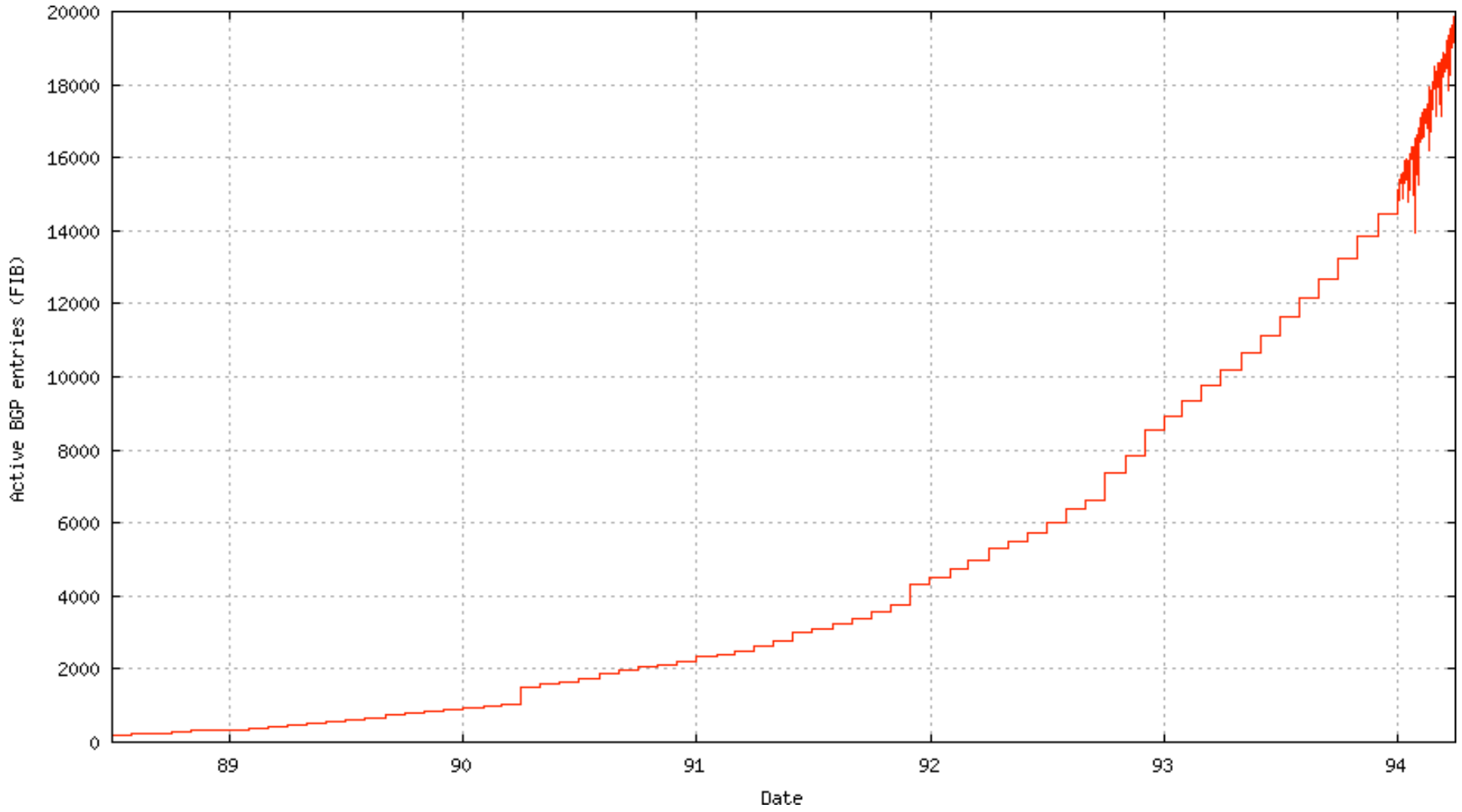


- *Multi-homed* customer (201.10.6.0/23) has two providers
- Other parts of Internet need to know how to reach destinations through *both* providers

Scalability Through Hierarchy

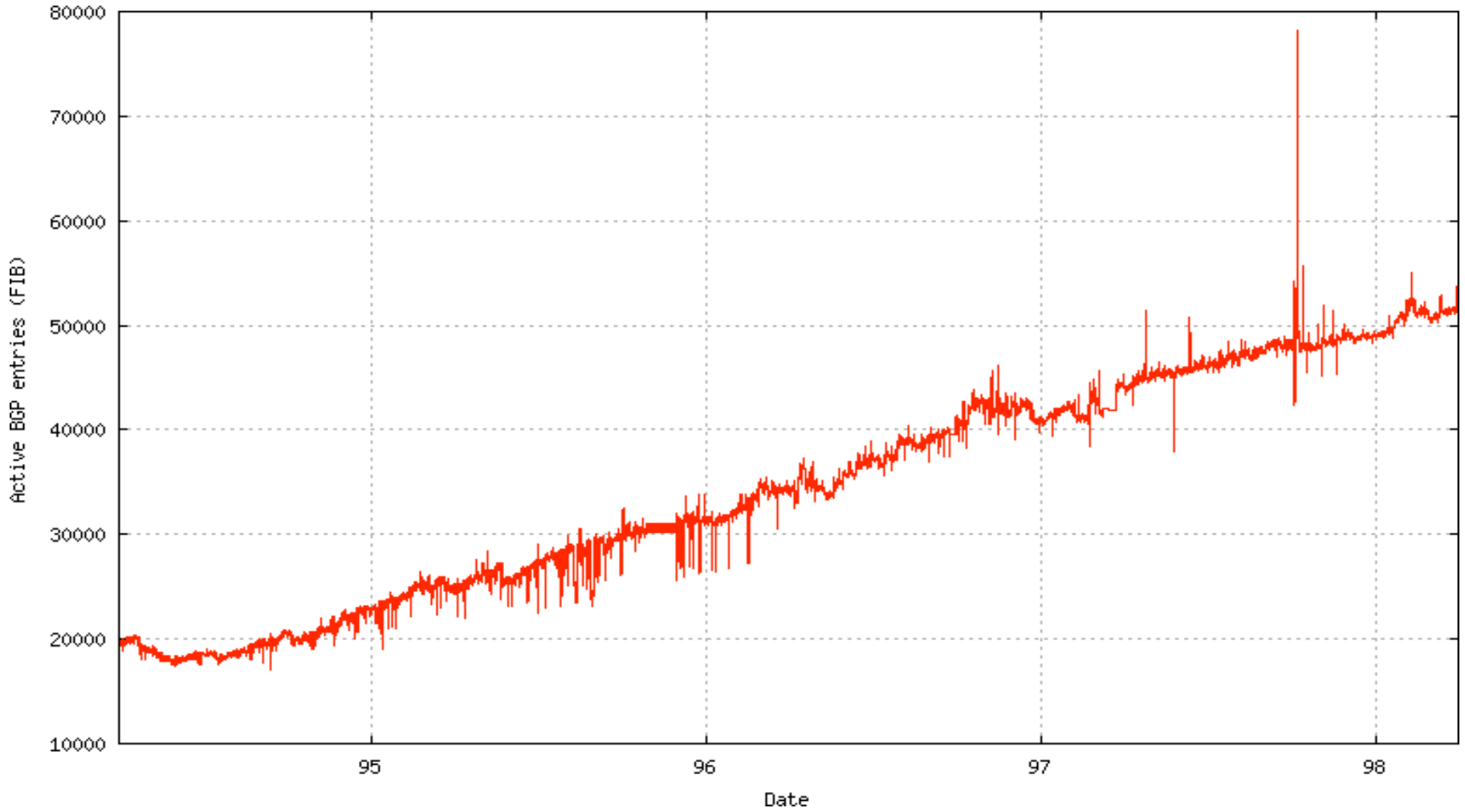
- Hierarchical addressing
  - Critical for scalable system
  - Don't require everyone to know everyone else
  - Reduces amount of updating when something changes
- Non-uniform hierarchy
  - Useful for heterogeneous networks of different sizes
  - Initial class-based addressing was far too coarse
  - Classless InterDomain Routing (CIDR) helps
- Next few slides
  - History of the number of globally-visible prefixes
  - Plots are # of prefixes vs. time

Pre-CIDR (1988-1994): Steep Growth



Growth faster than improvements in equipment capability

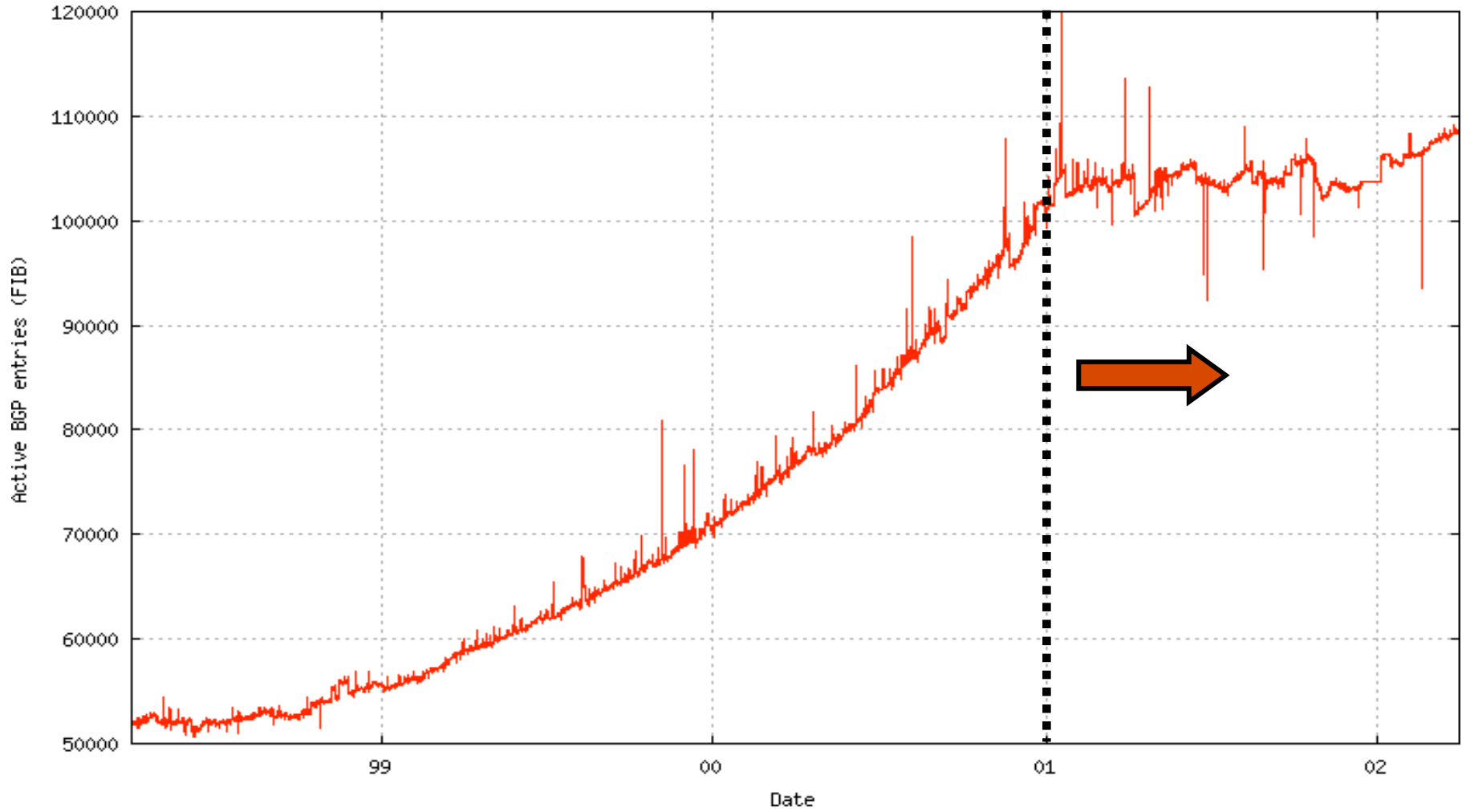
CIDR Deployed (1994-1998): Much Flatter



Efforts to aggregate (even decreases after IETF meetings!)

Good use of aggregation, and peer pressure in CIDR report

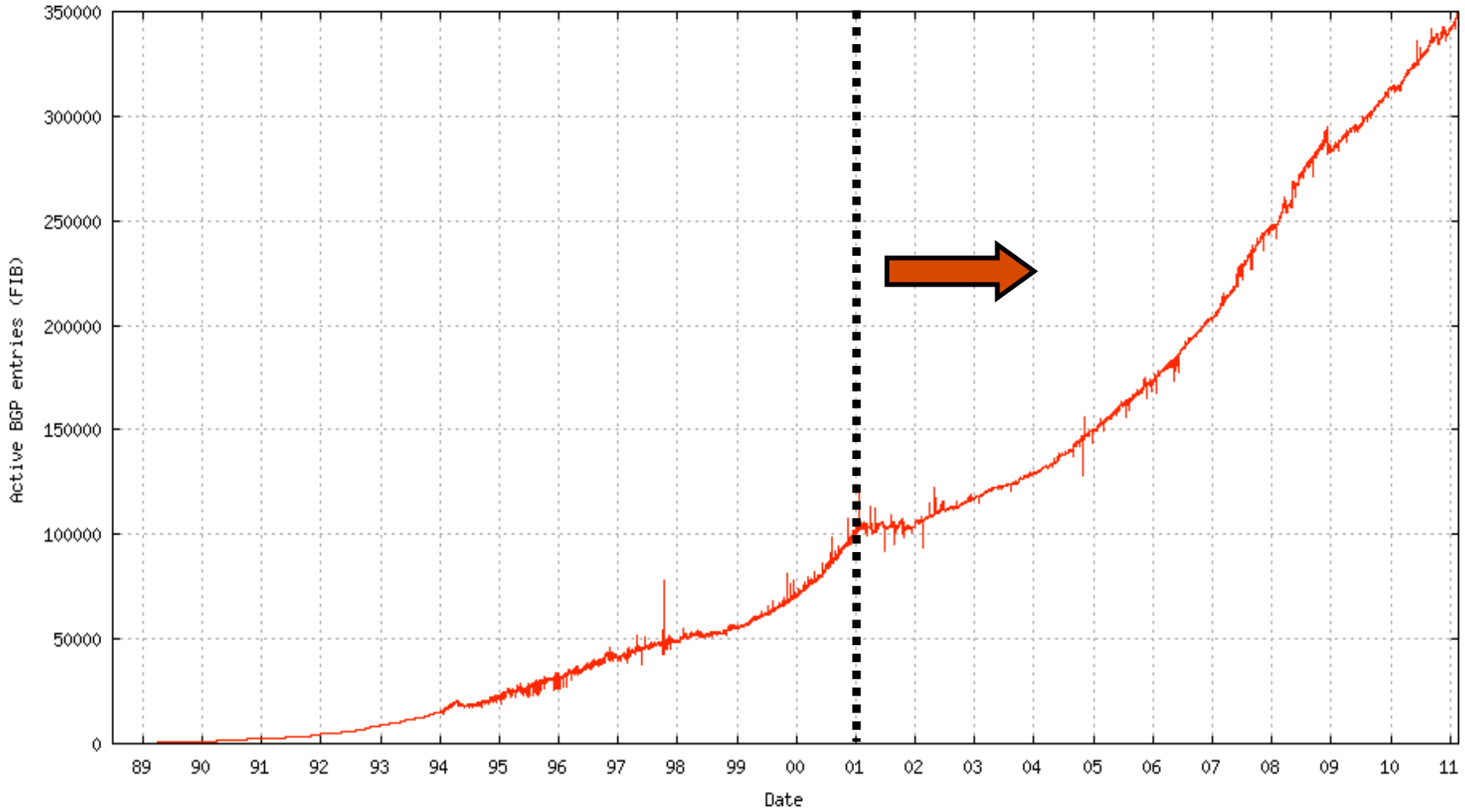
Boom Period (1998-2001): Steep Growth



Internet boom and increased multi-homing

"Dot-com" bubble of 2001 saw slow down

Long-Term View (1989-2011): Post-Boom



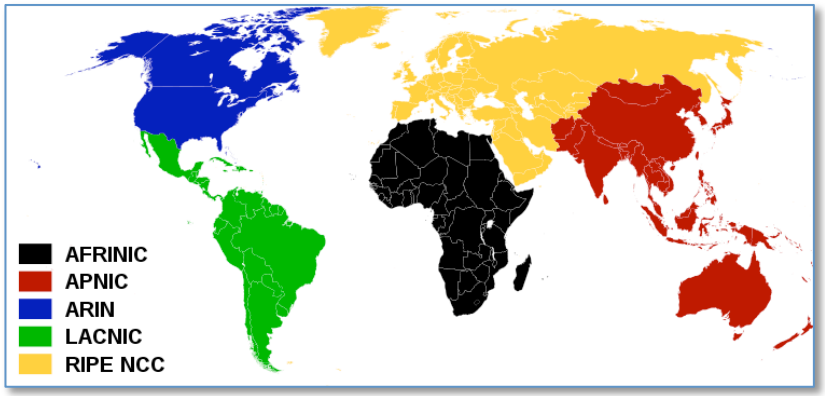
Obtaining a Block of Addresses

- Separation of control
  - Prefix: assigned *to* an institution
  - Addresses: assigned *by* the institution to their nodes
- Who assigns prefixes?

Internet Corp. for Assigned Names and Numbers (IANA)

Regional Internet Registries (RIRs)

Internet Service Providers (ISPs)



Figuring Out Who Owns an Address

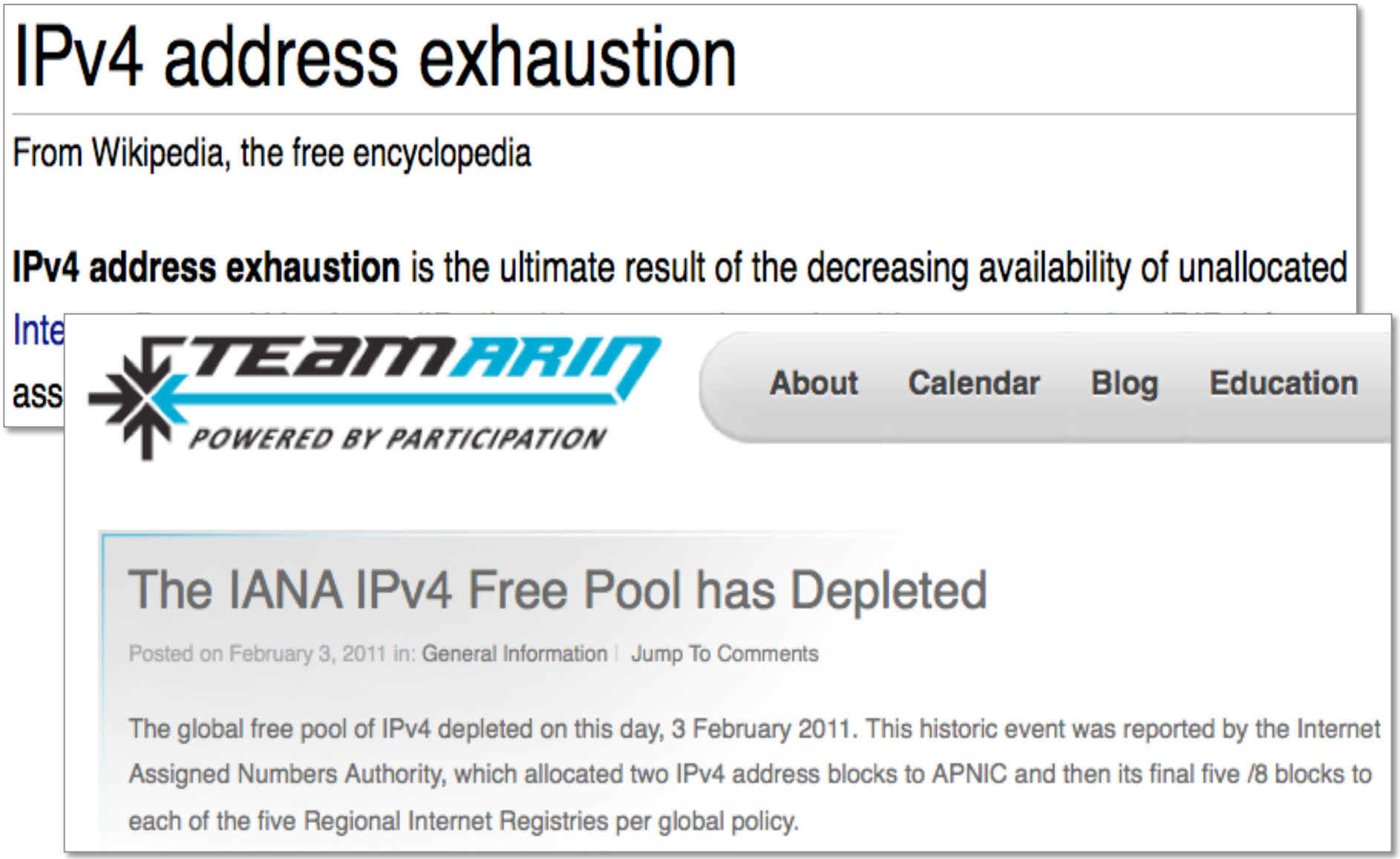
- Address registries
  - Public record of allocations
  - ISPs should update when allocating to customers
  - Records often out-of-date
- Ways to query
  - UNIX: "whois –h whois.arin.net 128.112.136.35"
  - http://www.arin.net/whois/
  - …

```
OrgName:      Princeton University
OrgID:        PRNU
Address:      Office of Info Tech
Address:      87 Prospect Avenue
City:         Princeton
StateProv:    NJ
PostalCode:   08540
Country:      US

NetRange:     128.112.0.0 –
                128.112.255.255
CIDR:         128.112.0.0/16
NetName:      PRINCETON
NetHandle:    NET-128-112-0-0-1
Parent:       NET-128-0-0-0-0
NetType:      Direct Allocation
NameServer:   DNS.PRINCETON.EDU
NameServer:   NS1.FAST.NET
NameServer:   NS2.FAST.NET
NameServer:   NS1.UCSC.EDU
NameServer:   ARIZONA.EDU
NameServer:   NS3.NIC.FR

Comment:
RegDate:      1986-02-24
Updated:      2007-02-27
```

Are 32-bit Addresses Enough?

- Not all that many unique addresses
  - 2^{32} = 4,294,967,296 (just over four billion)
  - Some are reserved for special purposes
  - Addresses are allocated non-uniformly
    - My fraternity/dorm at MIT has as many IP addrs as Princeton!
- More devices need addr's: smartphones, toasters, …
- Long-term solution: a larger address space
  - IPv6 has 128-bit addresses ($2^{128} = 3.403 \times 10^{38}$)
- Short-term solutions: limping along with IPv4
  - Private addresses (RFC 1918):
    - 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16
  - Network address translation (NAT)
  - Dynamically-assigned addresses (DHCP)

No more IPv4 prefixes for IANA!



IPv4 address exhaustion

From Wikipedia, the free encyclopedia

IPv4 address exhaustion is the ultimate result of the decreasing availability of unallocated

Inte

ass

TEAM ARIN

POWERED BY PARTICIPATION

About Calendar Blog Education

The IANA IPv4 Free Pool has Depleted

Posted on February 3, 2011 in: General Information | Jump To Comments

The global free pool of IPv4 depleted on this day, 3 February 2011. This historic event was reported by the Internet Assigned Numbers Authority, which allocated two IPv4 address blocks to APNIC and then its final five /8 blocks to each of the five Regional Internet Registries per global policy.

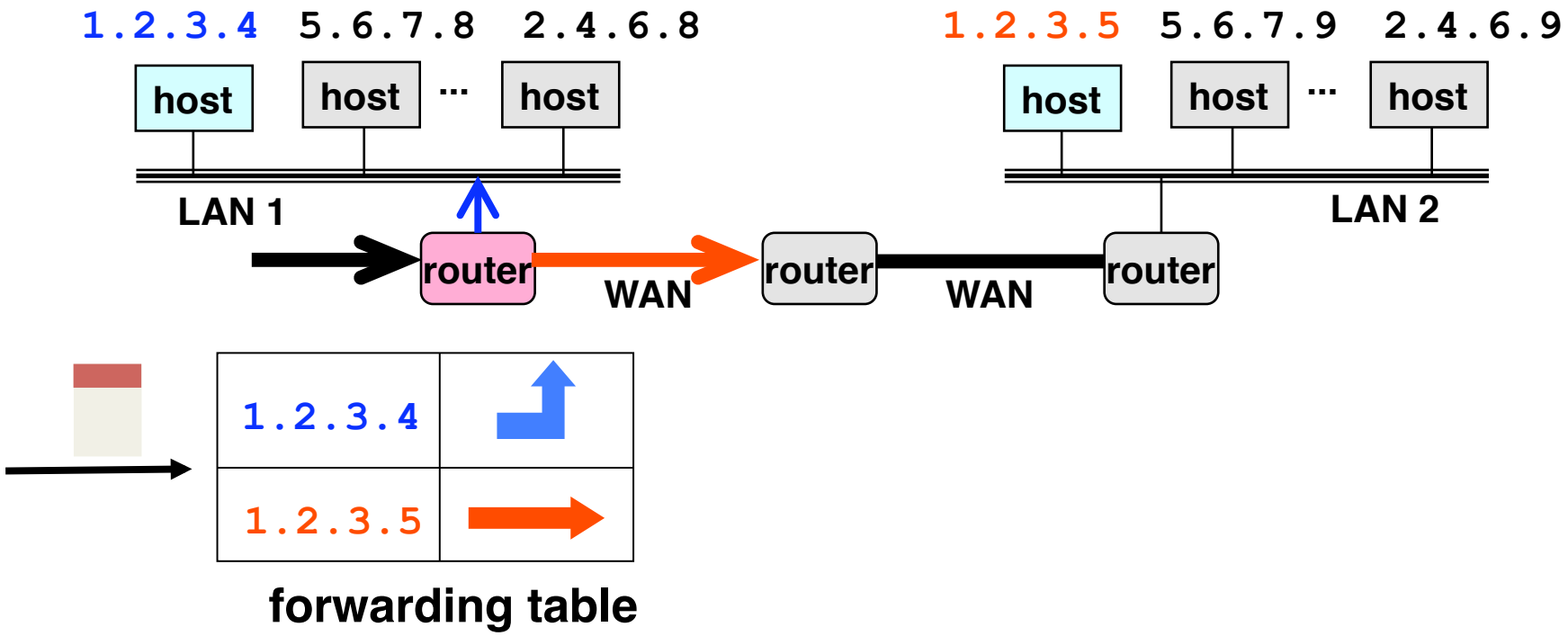
Hard Policy Questions

- How much address space per geographic region?
  - Equal amount per country?
  - Proportional to the population?
  - What about addresses already allocated?
    - MIT still has >> IP addresses than most countries?
- Address space portability?
  - Keep your address block when you change providers?
  - Pro: avoid having to renumber your equipment
  - Con: reduces the effectiveness of address aggregation
- Keeping the address registries up to date?
  - What about mergers and acquisitions?
  - Delegation of address blocks to customers?
  - As a result, the registries are horribly out of date

Hop-by-Hop Packet Forwarding

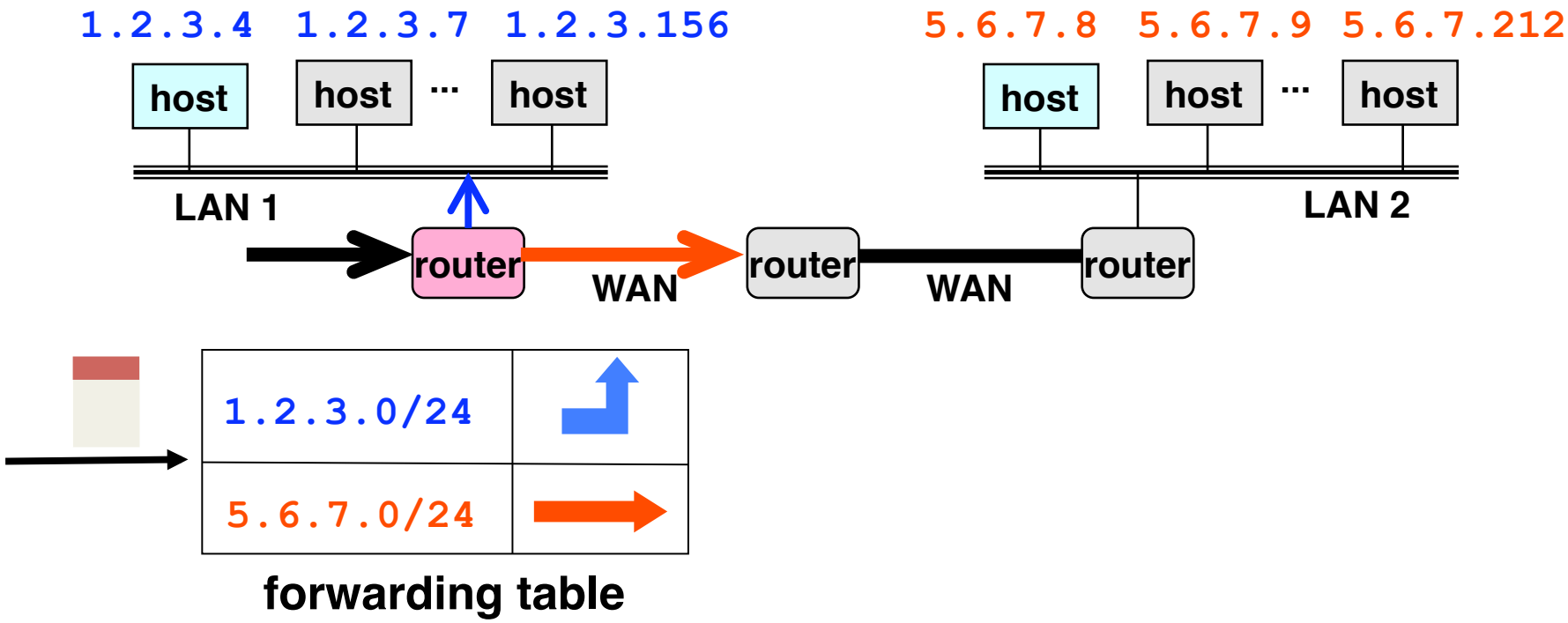
Separate Entry Per Address

- If router had a forwarding entry per IP addr
  - Match *destination addr* of incoming packet
  - Uniquely determine *outgoing interface*



Separate Entry Per 24-bit Prefix

- If router had an entry per 24-bit prefix
  - Look only at the top 24 bits of destination addr
  - Index into table to determine next-hop interface

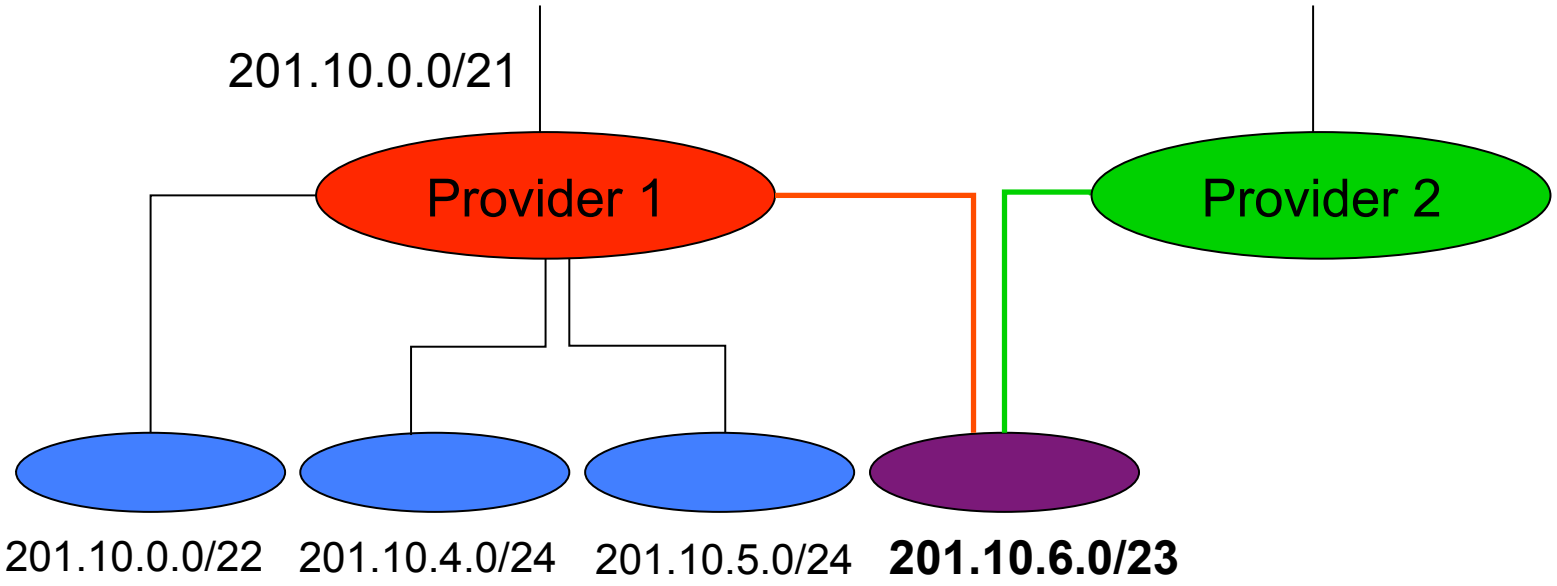


Separate Entry Classful Address

- If the router had an entry per classful prefix
  - Mixture of Class A, B, and C addresses
  - Depends on the first couple of bits of the destination
- Identify the mask automatically from the address
  - First bit of 0: class A address (/8)
  - First two bits of 10: class B address (/16)
  - First three bits of 110: class C address (/24)
- Then, look in the forwarding table for the match
  - E.g., If addr is 1.2.3.4, lookup up entry for 1.2.3.0/24
- So far, everything is exact matching

CIDR Makes Packet Forwarding Harder

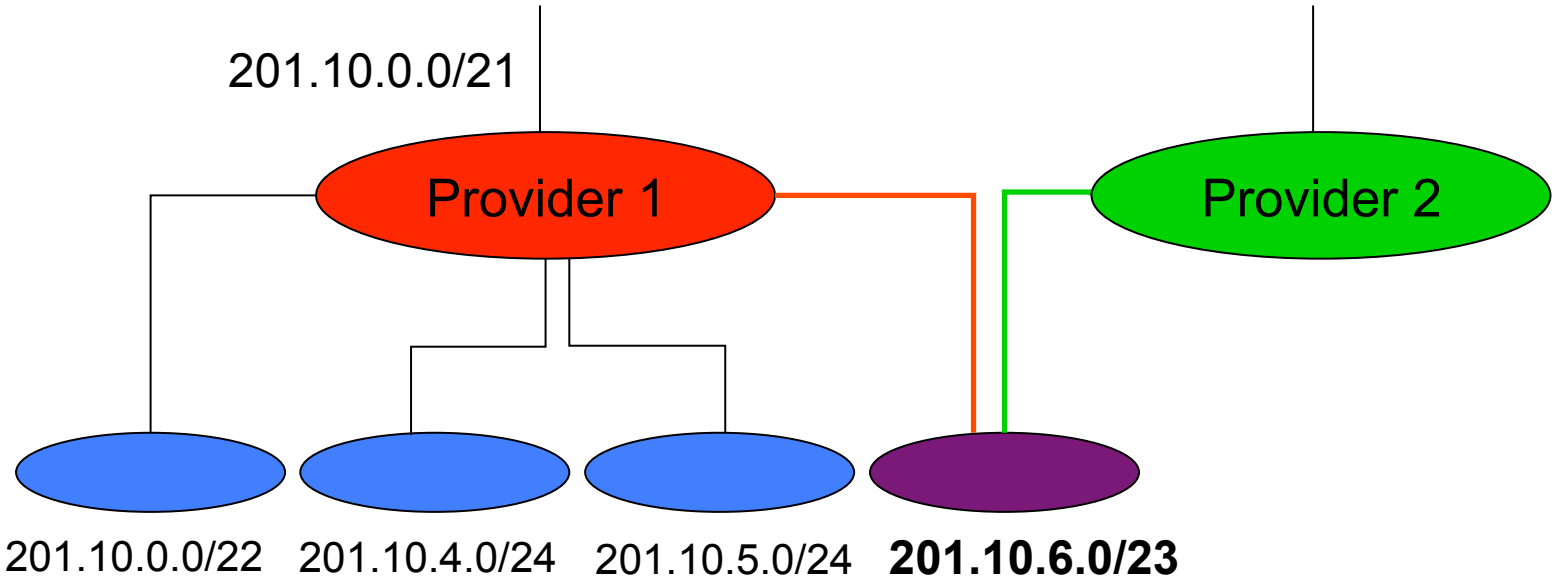
- Efficient use of address space vs. overlapping rules
- Forwarding table may have many matches
  - 201.10.6.17 matches both 201.10.0.0/21 and 201.10.6.0/23
  - Entries may map to different outgoing interfaces



Another reason FIBs get large

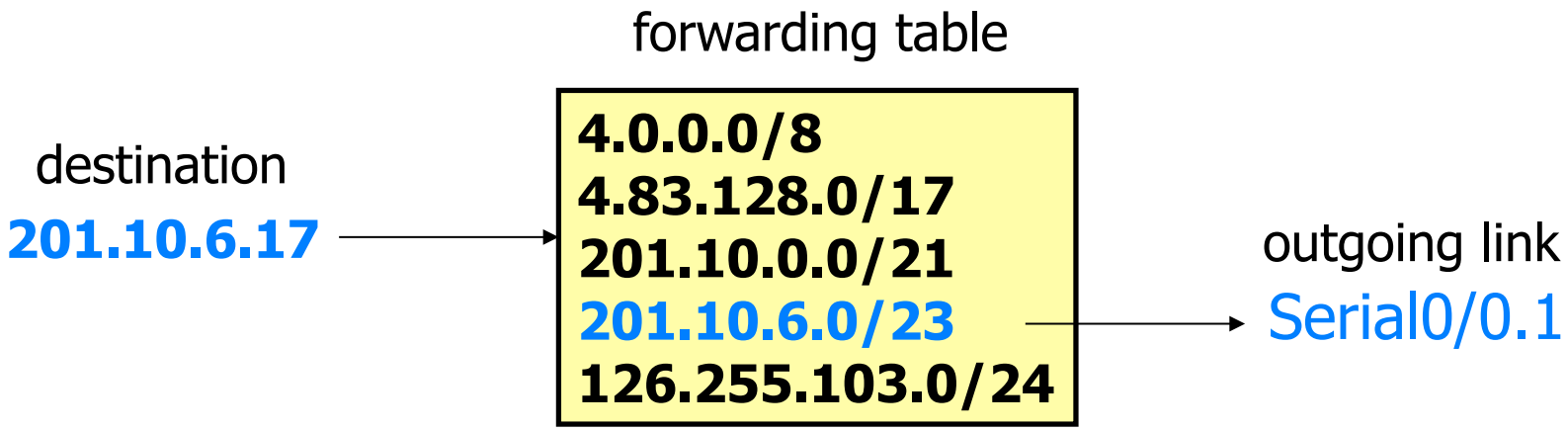
- If customer 201.10.6.0/23 prefers to receive traffic from Provider 1 (it may be cheaper), then P1 needs to announce 201.10.6.0/23, not 201.10.0.0/21

- Can't always aggregate! [See "Geographic Locality of IP Prefixes" M. Freedman, M. Vutukuru, N. Feamster, and H. Balakrishnan. Internet Measurement Conference (IMC), 2005



Longest Prefix Match Forwarding

- How to resolve multiple matches?
  - Router identifies most specific prefix:
    longest prefix match (LPM)
  - Cute algorithmic problem to achieve fast lookups

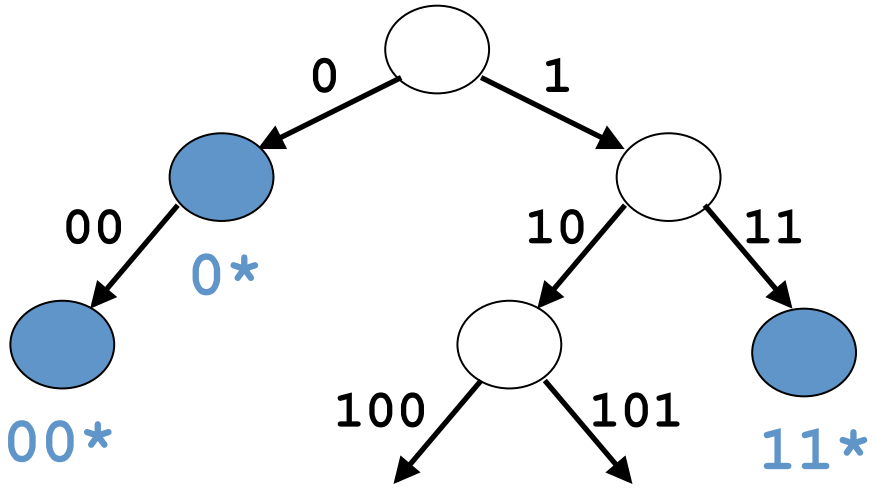


Simplest Algorithm is Too Slow

- Scan the forwarding table one entry at a time
  - Keep track of entry with longest-prefix (by netmask)
- Overhead is linear in size of forwarding table
  - Today, that means 350,000 entries!
  - How much time do you have to process?
    - Consider 10Gbps routers and 64B packets
    - 10^{10} / 8 / 64: 19,531,250 packets per second
    - 51 nanoseconds per packet
- Need greater efficiency to keep up with *line rate*
  - Better algorithms
  - Hardware implementations

Patricia Tree (1968)

- Store prefixes as a tree
  - One bit for each level of tree
  - Some nodes correspond to valid prefixes
  - ... which have next-hop interfaces in a table
- When a packet arrives
  - Traverse tree based on destination address
  - Stop upon reaching longest matching prefix



Even Faster Lookups

- Patricia tree is faster than linear scan
  - Proportional to number of bits in address
  - Speed-up further by time vs. space tradeoff
    - Each node in 4-ary tree has 4 children, cuts depth by half
- Still somewhat slow, major concern in mid-to-late 1990s
  - … after CIDR was introduced and LPM major bottleneck
  - Reintroduction of circuit switching via pre-established paths: individual paths named by labels added to packets (MPLS)
- Innovation of special hardware
  - Content Addressable Memories (CAMs): assoc. array in h/w
    - Compares key in parallel to each entry
  - Ternary CAMs (TCAMS): Stored data is 0, 1, <don't care>
    - Least sig. bits represented by <don't care> (netmask=0)

Where do Forwarding Tables Come From?

- Entries can be statically configured
  - E.g., "map 12.34.158.0/24 to Serial0/0.1"
- But, this doesn't adapt
  - To failures, new equipment, …
  - To need to balance load, …
- That is where other technologies come in…
  - Routing protocols, DHCP, and ARP (later in course)

How Do End Hosts Forward Packets?

- End host with single network interface
- Don't need a routing protocol
  - Packets to host itself (e.g., 1.2.3.4/32)
    - Delivered locally
  - Packets to other hosts on LAN (e.g., 1.2.3.0/24)
    - Sent out interface: Broadcast medium!
  - Packets to external hosts (e.g., 0.0.0.0/0)
    - Sent out interface to local gateway
- How is information learned?
  - Static setting of address, subnet mask, and gateway
  - Dynamic Host Config Protocol (DHCP): Local server tells you settings when you join network

Conclusions

- IP addresses
  - Dotted-quad notation
  - IP prefixes for aggregation
- Address allocation
  - Classful addr's, Classless routing (CIDR), FIB growth
- Packet forwarding
  - Forwarding tables
  - Longest-prefix match forwarding
  - Where forwarding tables come from
- Next lecture: Transport protocols (UDP and TCP)
- Routing protocols come later