15-441 Computer Networking

Lecture 4 –Internet design and IP Addressing

Peter Steenkiste

Justine Sherry

Fall 2017

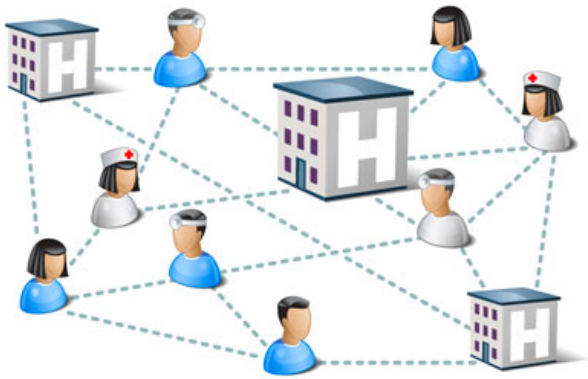
www.cs.cmu.edu/~prs/15-441-F16

Muchas gracias to Sylvia Ratnasamy and Scott Shenker (Berkeley) for some slide material

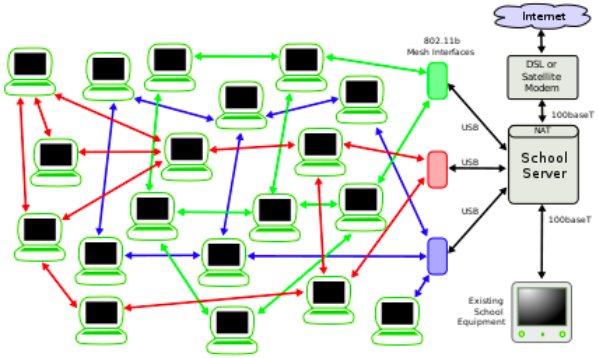
Outline

- IP design goals
- Traditional IP addressing
  - Addressing approaches
  - Class-based addressing
  - Subnetting
  - CIDR
- Routing

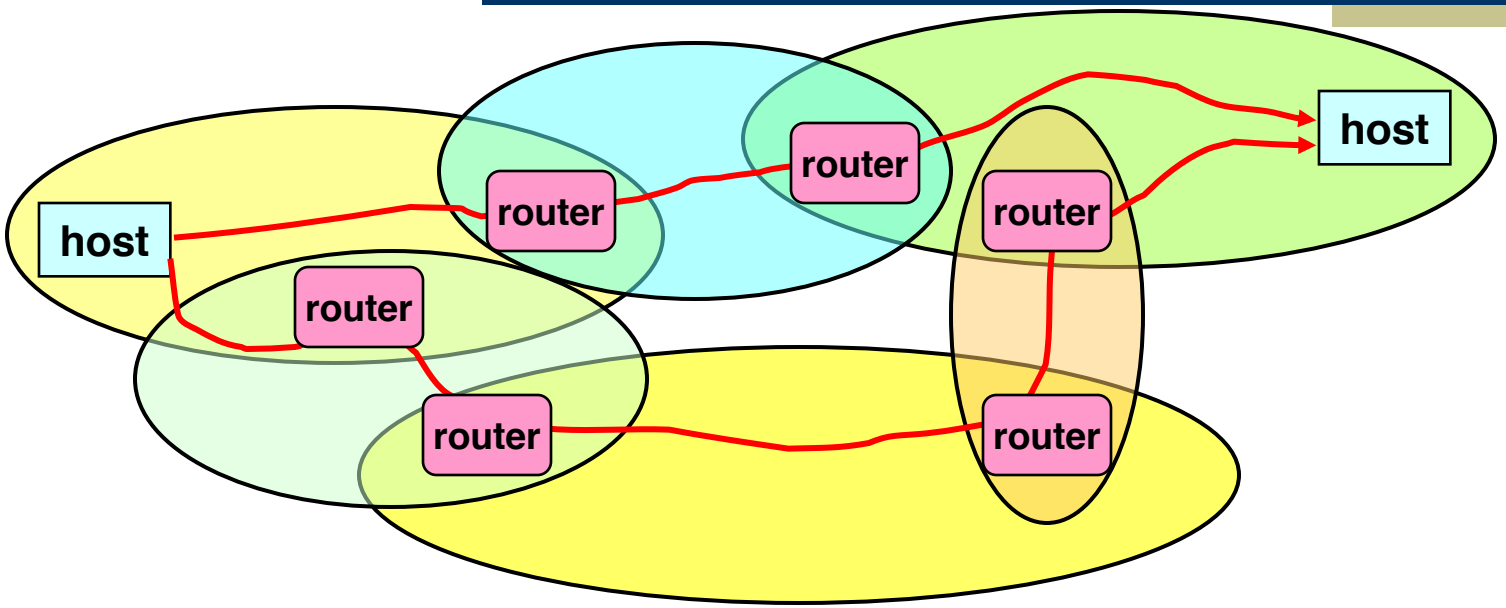
So far you know how to build a Local Area Network



How do we get them to talk to each other?



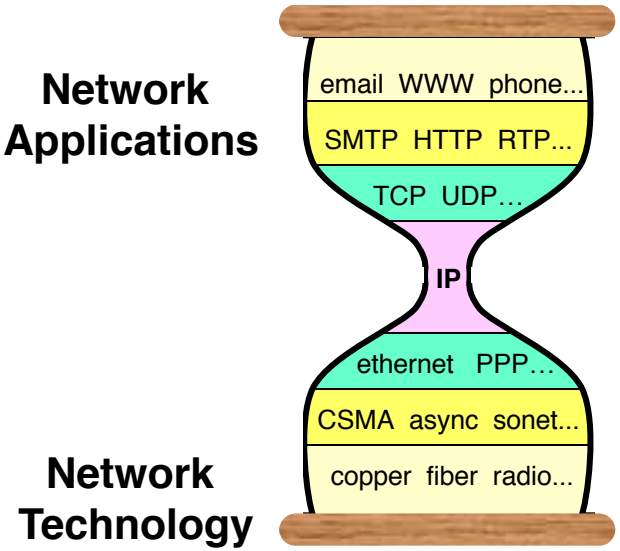
Logical Structure of an Internet



- Interconnection of separately managed networks using routers
  - Individual networks can use different (layer 1-2) technologies
- Send packets from source to destination by hopping through networks
  - “Network” layer responsibility
- How do routers connect heterogenous network technologies?

Solution: Internet Protocol (IP)

- Inter-network connectivity provided by the Internet protocol
- Hosts use Internet Protocol to send packets destined across networks.
- IP creates abstraction layer that hides underlying technology from network application software
  - Allows range of current & future technologies
  - WiFi, traditional and switched Ethernet, personal area networks, …



The Packet as an Envelope

Host wants to send…

Packet Payload:
GET nyan.cat….

The Packet as an Envelope

So needs to add local addressing header.

Local Address Header (Ethernet) To: Destination Host From: Sender Host
Packet Payload: GET nyan.cat….

The Packet as an Envelope

Local Address Header (Ethernet) To: Router From: Sender Host
IP Header To: 123.45.67.89 (Destination Host) From: 169.229.49.157 (Sender Host)
Packet Payload: GET nyan.cat….

Add another header! IP address tells us where to send in another network.

The Packet as an Envelope

Local Address Header (Token Ring) To: Destination Host From: Local Router
IP Header To: 123.45.67.89 (Destination Host) From: 169.229.49.157 (Sender Host)
Packet Payload: GET nyan.cat….

At the receiver, might have a different local address header type –
But IP layer remains unchanged.

What are the Goals?

- LANs: “Connect hosts” → switching:
  - “Wire” abstraction: behaves like Ethernet
  - Only has to scale up a “LAN size”
  - Availability
- Internet: “Connect networks” → routing:
  - Scalability
  - Manageability of individual networks
  - Availability
- Affects addressing, protocols, routing

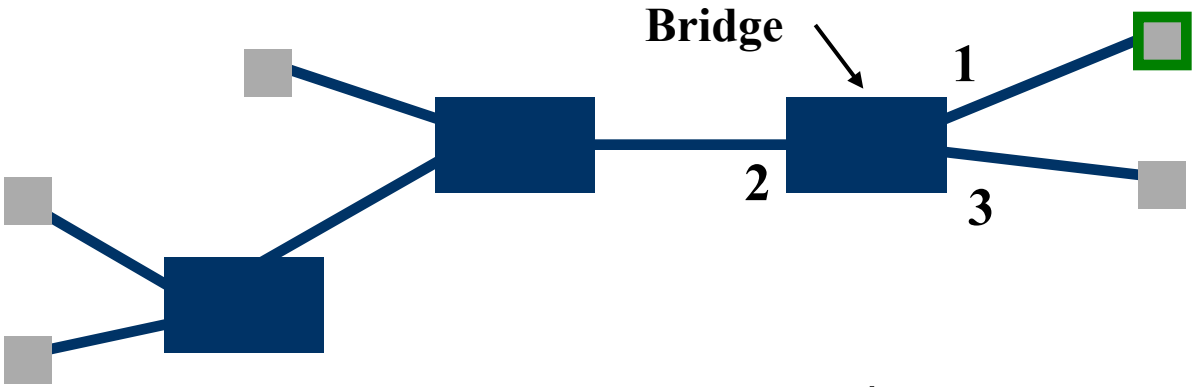
Outline

- IP design goals
- Traditional IP addressing
  - Addressing approaches
  - Class-based addressing
  - Subnetting
  - CIDR
- Routing

Addressing and Forwarding

- Flat address space with smart routers
  - Packets carry destination
  - Routers know location of every host
- Flat address space with dumb routers
  - Packet carries a path
- Heirarchical Routing Space
  - What we actually do in IP
- (Table of virtual circuits ids)
  - More on this later, but not today

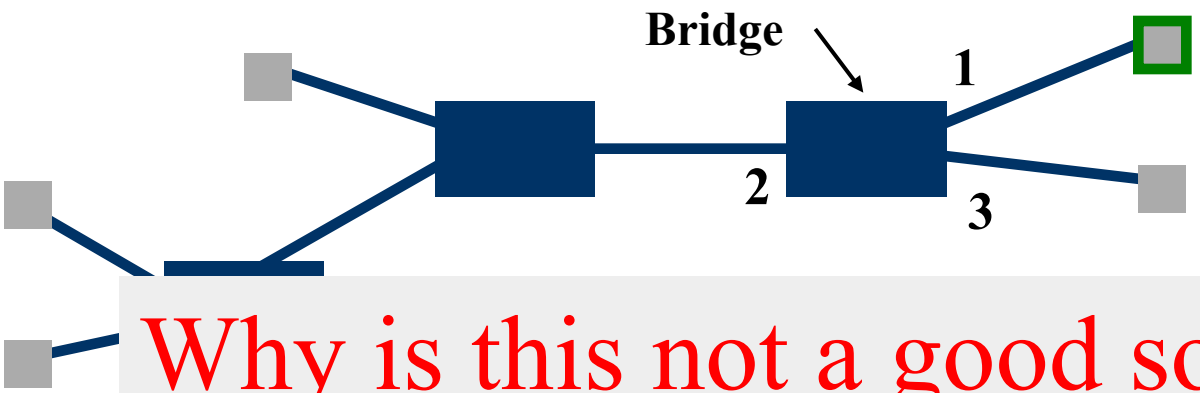
Flat Address Forwarding



MAC Address	Port	Age
A21032C9A591	1	36
99A323C90842	2	01
8711C98900AA	2	15
301B2369011C	2	16
695519001190	3	11

- Bridge/switch has a table that shows for each MAC Address which port to use for forwarding
- For every packet, the bridge "looks up" the entry for the packets destination MAC address and forwards the packet on that port.
  - Other packets are broadcast – why?
- Timer is used to flush old entries

Flat Address Forwarding



Why is this not a good solution for the Internet?

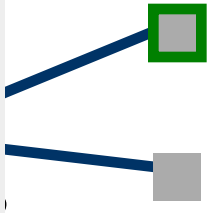
MAC Addr	Port	Age
A21032C9A591	1	36
99A323C90842	2	01
8711C98900AA	2	15
301B2369011C	2	16
695519001190	3	11

which port to use for forwarding

- For every packet, the bridge "looks up" the entry for the packets destination MAC address and forwards the packet on that port.
  - Other packets are broadcast – why?
- Timer is used to flush old entries

Flat Address Forwarding

Each router tracking 2^32 addresses = scalability nightmare



MAC Address	Port	Age
A21032C9A591	1	36
99A323C90842	2	01
8711C98900AA	2	15
301B2369011C	2	16
695519001190	3	11

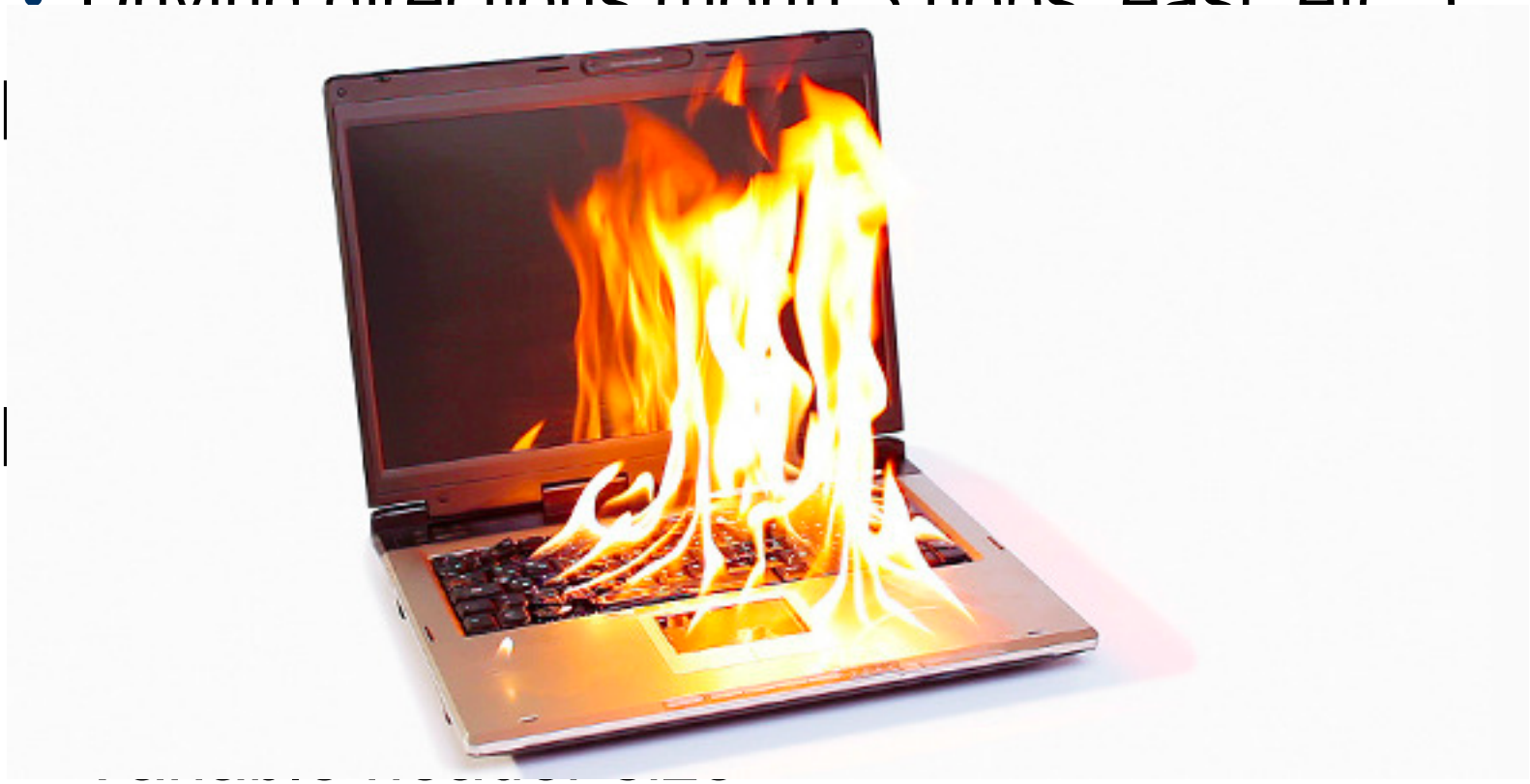
- Brid
  sho ich
  port
- For ks
  up”
  des
  forw
  - Other packets are broadcast – why?
- Timer is used to flush old entries

Source Routing

- List entire path in packet
  - Driving directions (north 3 hops, east, etc..)
- Router processing
  - Strip first step from packet
  - Examine next step in directions and forward
- Rarely used
  - End points need to know a lot about network
  - Economic and security concerns
  - Variable header size

Source Routing

- List entire path in packet
  - Driving directions (north 3 hops, east, etc.)
- |
- |



Heirarchical Addressing

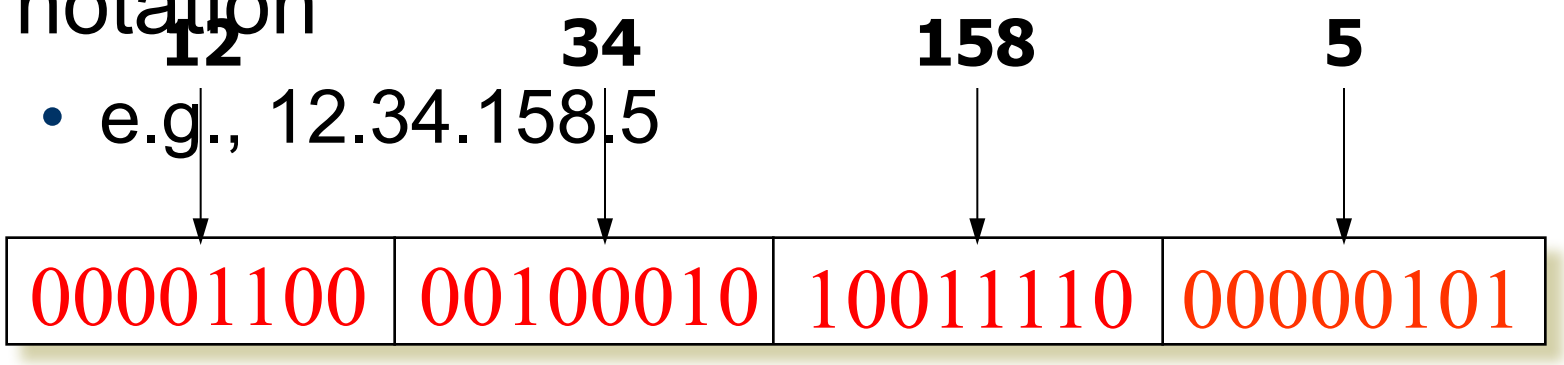
- Flat addresses – one address for every host
  - Peter Steenkiste: 123-45-6789
  - Does not scale – router table size explodes
  - 630M (1/09) entries,doubling every 2.5 years
  - Why does it work for Ethernet?
- Hierarchical – add structure
  - Pennsylvania / Pittsburgh / Oakland / CMU / Steenkiste
    or Peter Steenkiste: (412)268-0000
  - Common “trick” to simplify forwarding, reduce forwarding table
- What type of Hierarchy?
  - How many levels?
  - Same hierarchy depth for everyone?
  - Who controls the hierarchy?

IP Addresses (IPv4)

- Unique 32-bit number associated with a host

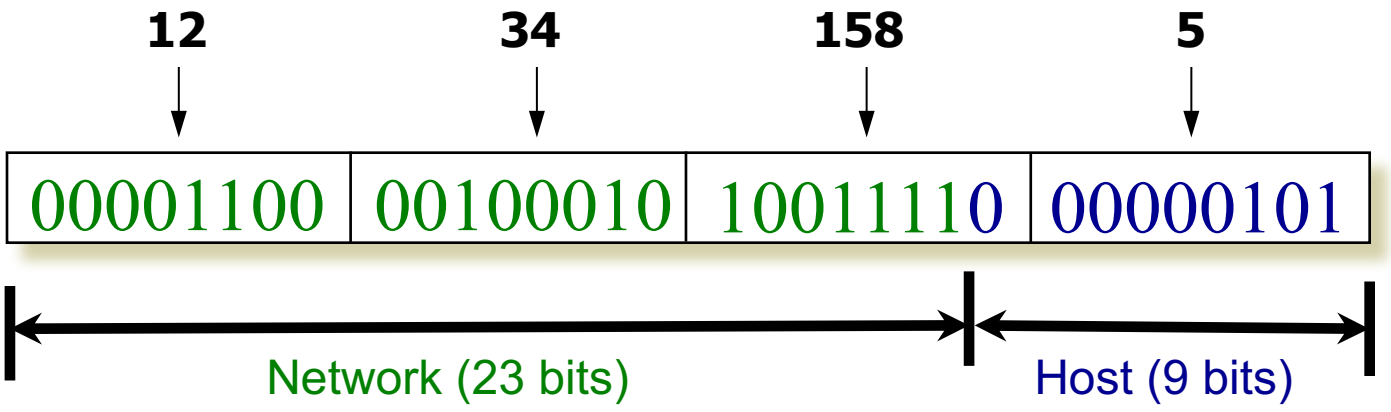
00001100 00100010 10011110 00000101

- Represented with the "dotted quad" notation
  - e.g., 12.34.158.5



Hierarchy in IP Addressing

- 32 bits are partitioned into a prefix and suffix components
- Prefix is the network component; suffix is host component



- Interdomain routing operates on the network prefix

History of Internet Addressing

- Always dotted-quad notation
- Always network/host address split
- But nature of that split has changed over time

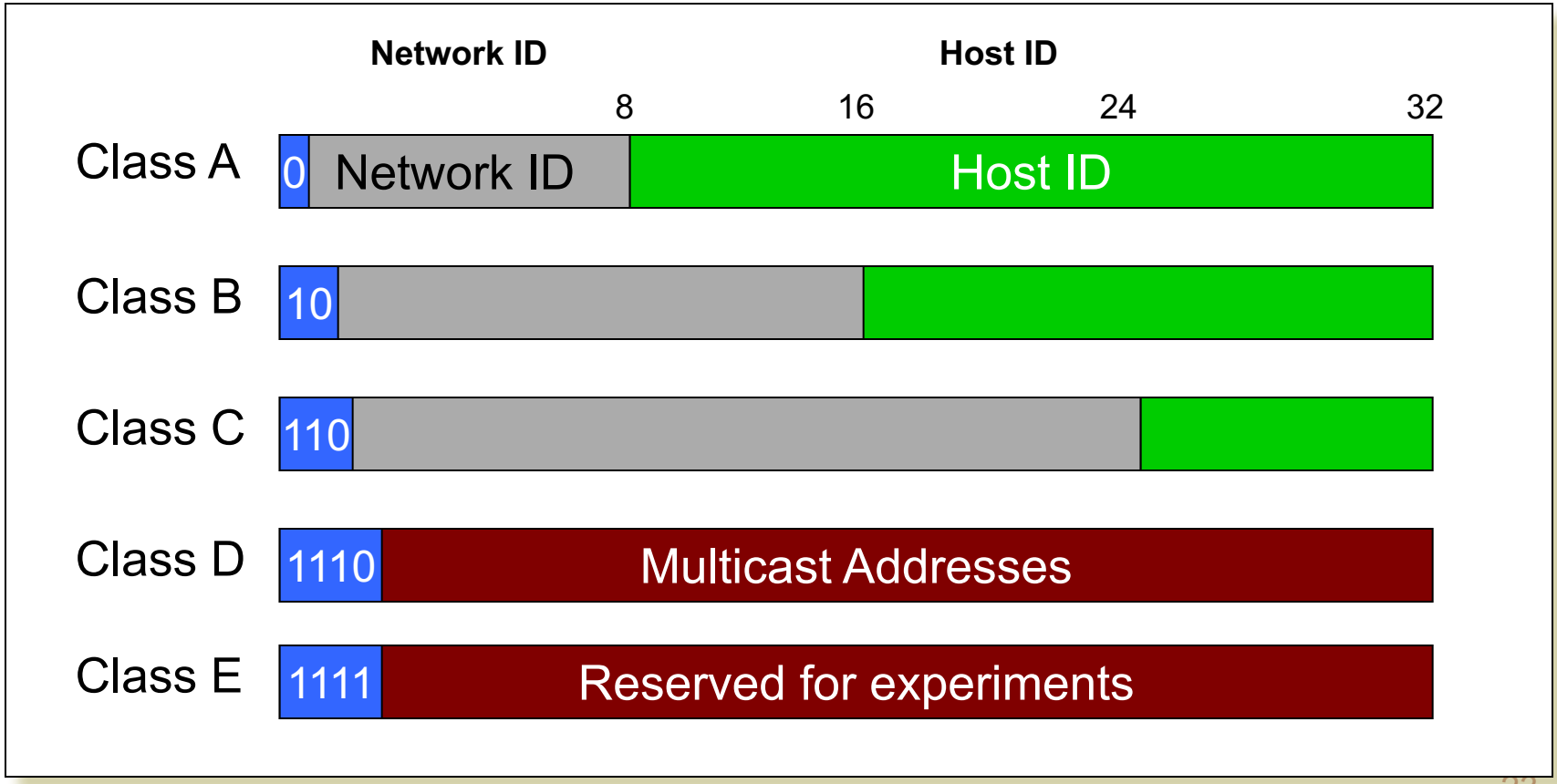
Original Internet Addresses

- First eight bits: network component
- Last 24 bits: host component

Assumed 256 networks were more than enough!

IP Address Structure, ca 1981

Routers know how to get to network ID, but not individual hosts.



IP Route Lookup, ca 1981

- Address specifies prefix for forwarding table
  - Extract address type and network ID
- Forwarding table contains
  - List of class+network entries
  - A few fixed prefix lengths (8/16/24)
  - Prefix – part of address that really matters for routing
- www.cmu.edu address 128.2.11.43
  - Class B address – class + network is 128.2
  - Lookup 128.2 in forwarding table for class B
- Tables are still large!
  - 2 Million class C networks

IP Address Problem (1991)

- Address space depletion
  - Suppose you need $2^{16} + 1$ addresses?
  - Class A too big for all but a few domains
  - Class C too small for many domains but they don't need a class B address
  - Class B address pool allocated at high rate
  - Many allocated address block are sparsely used
- Developed a strategy based on a three solutions
  - Switch to a "classless" addressing model
  - Network address translation
  - Definition of IPv6 with larger IP addresses

Today's Addressing: CIDR

- CIDR = Classless Interdomain Routing
- Idea: Flexible division between network and host addresses
- Motivation: offer a better tradeoff between size of the routing table and efficient use of the IP address space

CIDR (example)

- Suppose a network has fifty computers
  - allocate 6 bits for host addresses (since $2^5 < 50 < 2^6$)
  - remaining 32 - 6 = 26 bits as network prefix

- Flexible boundary means the boundary must be explicitly specified with the network address!
  - informally, "slash 26" → 128.23.9/26
  - formally, prefix represented with a 32-bit mask: 255.255.255.192 where all network prefix bits set to "1" and host suffix bits to "0"

Classful vs. Classless addresses

- Example: an organization needs 500 addresses.
  - A single class C address not enough (254 hosts).
  - Instead a class B address is allocated. (~65K hosts)
  - That's overkill, a huge waste!
- CIDR allows an arbitrary prefix-suffix boundary
  - Hence, organization allocated a single /23 address (equivalent of 2 class C's)
- Maximum waste: 50%

Hence, IP Addressing: Hierarchical

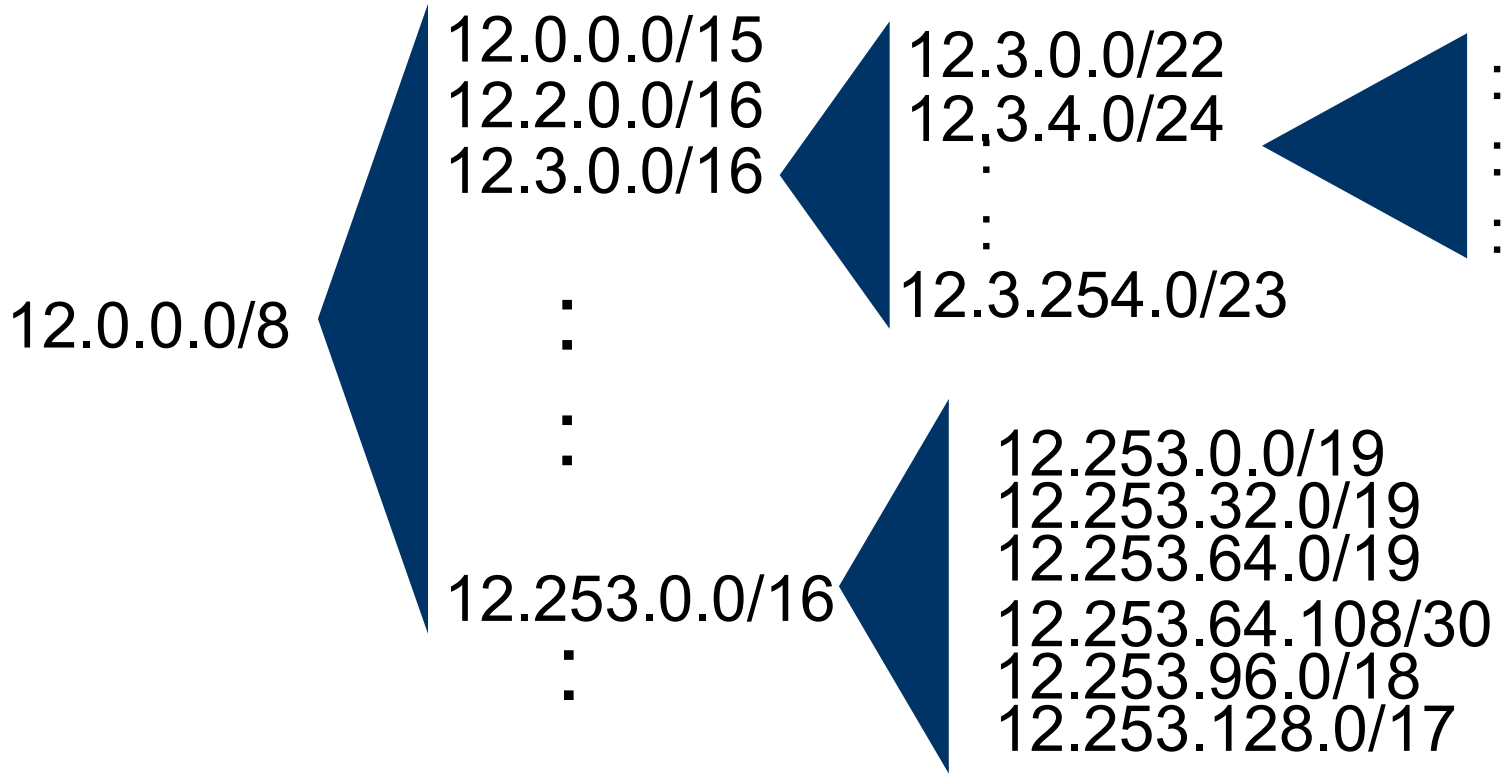
- Hierarchical address structure
- Hierarchical address allocation
- Hierarchical addresses and routing scalability

Allocation Done Hierarchically

- Internet Corporation for Assigned Names and Numbers (ICANN) gives large blocks to…
- Regional Internet Registries, such as the American Registry for Internet Names (ARIN), which give blocks to…
- Large institutions (ISPs), which give addreses to…
- Individuals and smaller institutions

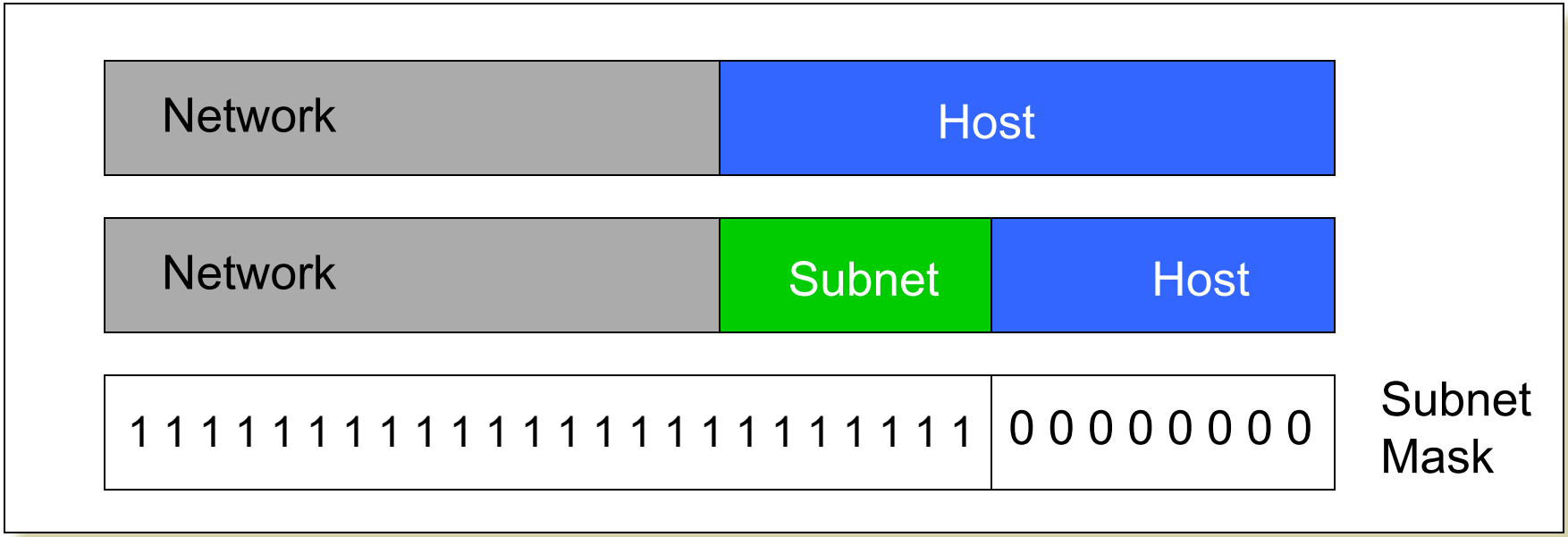
CIDR: Addresses allocated in contiguous prefix chunks

Recursively break down chunks as get closer to host



Subnetting

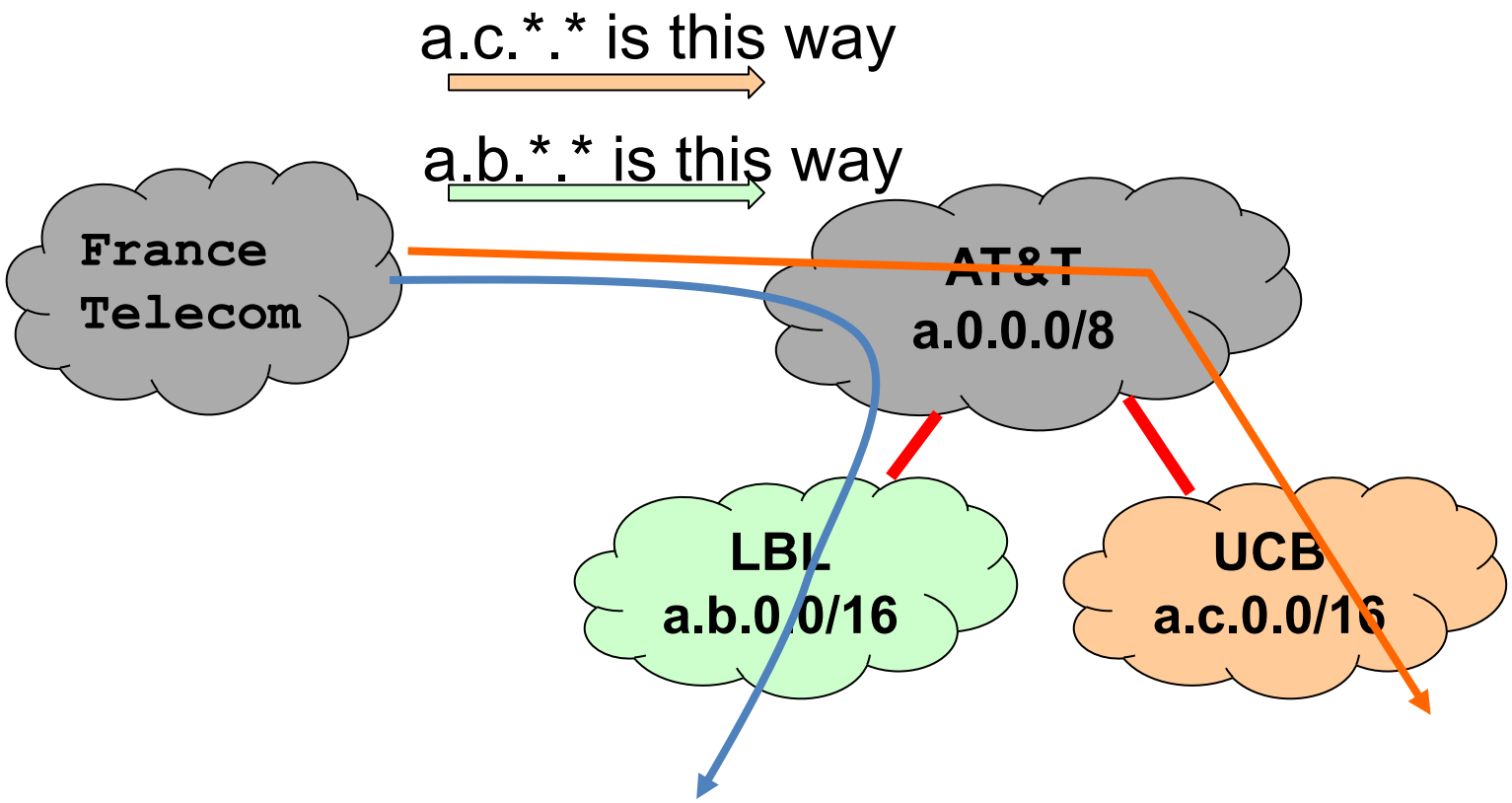
- Add another layer to hierarchy
- Variable length subnet masks
  - Could subnet a network internally into several chunks
- Subnetting is done internally in the organization
  - It is not visible outside – important for management



Subnet Addressing RFC917 (1984)

- Some “LANs” are very big
  - Large companies, universities, …
  - Internet became popular quickly
- Cannot manage this as a single LAN
  - Hard to manage, becomes inefficient
- Need simple way to partition large networks
  - Partition into multiple IP networks that share the same prefix – called a “subnet”, part of a network
- CMU case study in RFC
  - Chose not to adopt – concern that it would not be widely supported ☺

IP addressing → scalable routing?

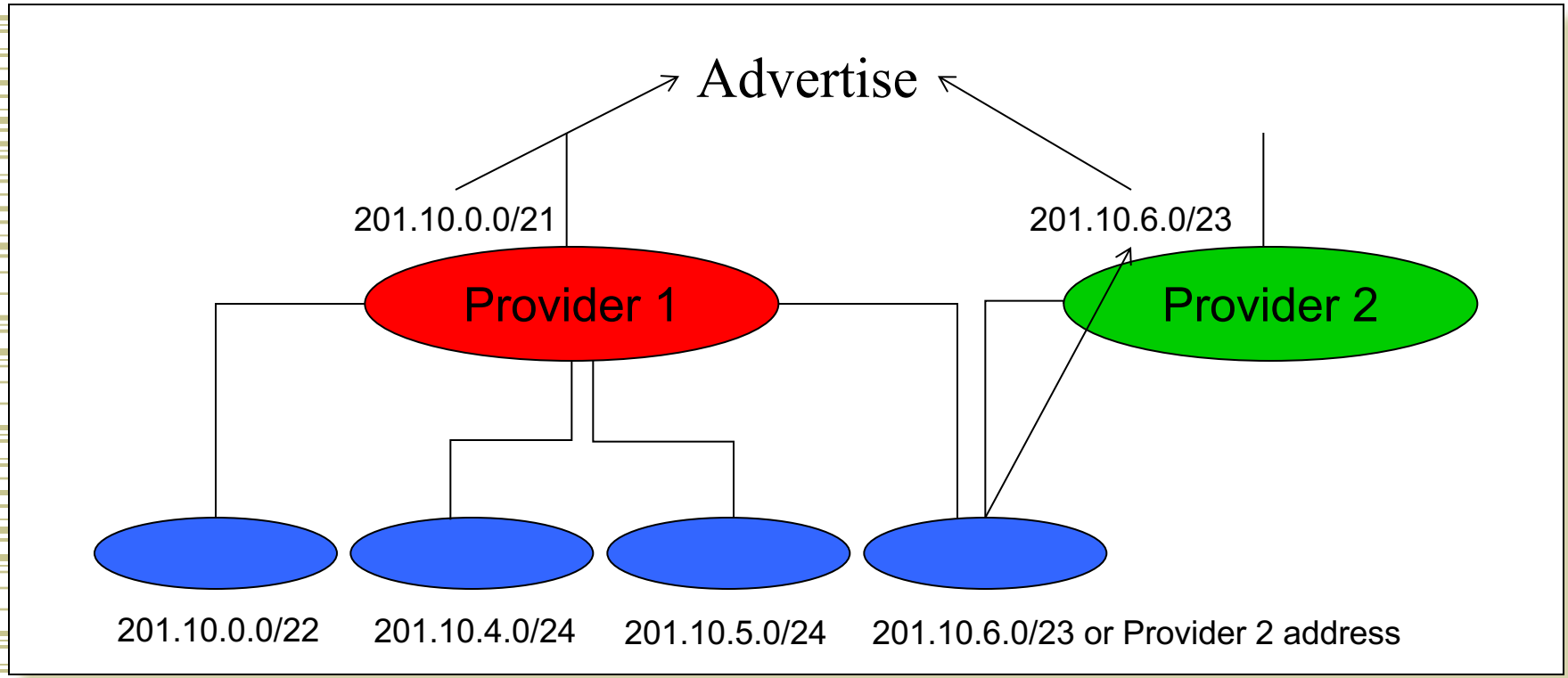


Outline

- IP design goals
- Traditional IP addressing
  - Addressing approaches
  - Class-based addressing
  - Subnetting
  - CIDR
- Routing

CIDR Implication: Longest Prefix Match

- How to deal with multi-homing, legacy addresses, …



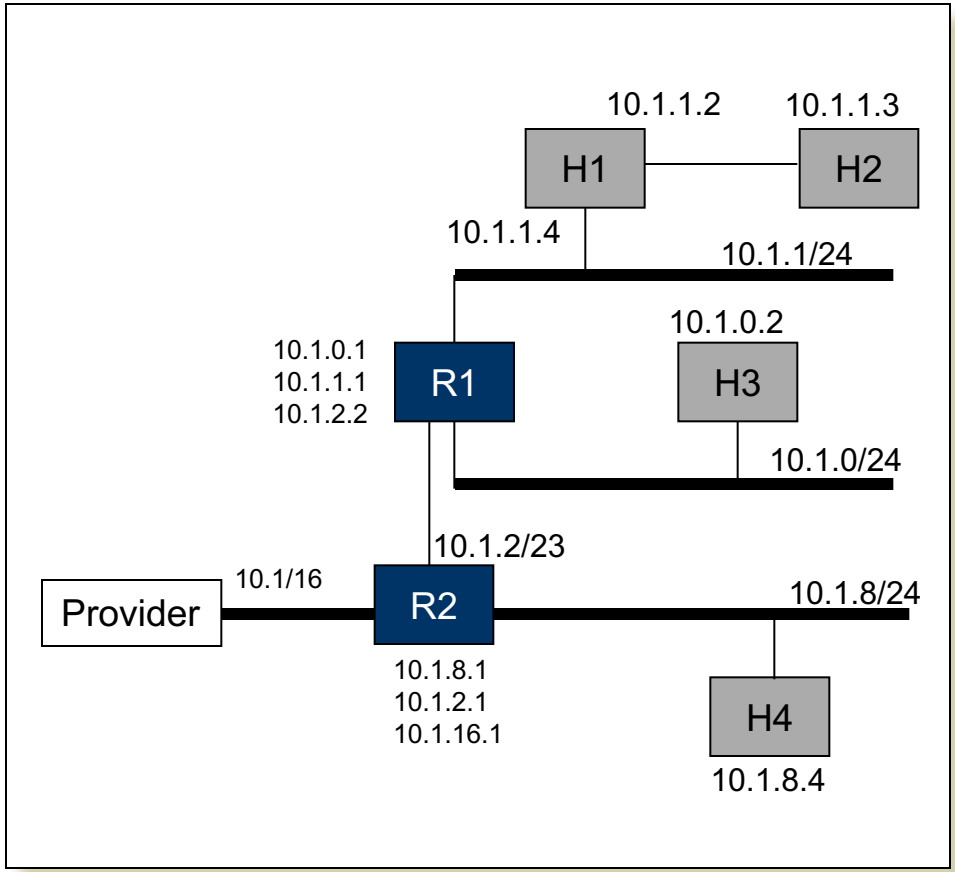
Host Routing Table Example

Destination	Gateway	Genmask	Iface
128.2.209.100	0.0.0.0	255.255.255.255	eth0
128.2.0.0	0.0.0.0	255.255.0.0	eth0
127.0.0.0	0.0.0.0	255.0.0.0	lo
0.0.0.0	128.2.254.36	0.0.0.0	eth0

- From “netstat –rn”
- Host 128.2.209.100 when plugged into CS ethernet
- Dest 128.2.209.100 → routing to same machine
- Dest 128.2.0.0 → other hosts on same ethernet
- Dest 127.0.0.0 → special loopback address
- Dest 0.0.0.0 → default route to rest of Internet
  - Main CS router: gigrouter.net.cs.cmu.edu (128.2.254.36)

Routing to the Network

- Packet to 10.1.1.3 arrives
- Path is R2 – R1 – H1 – H2
- H1 serves as a router for the 10.1.1.2/31 network

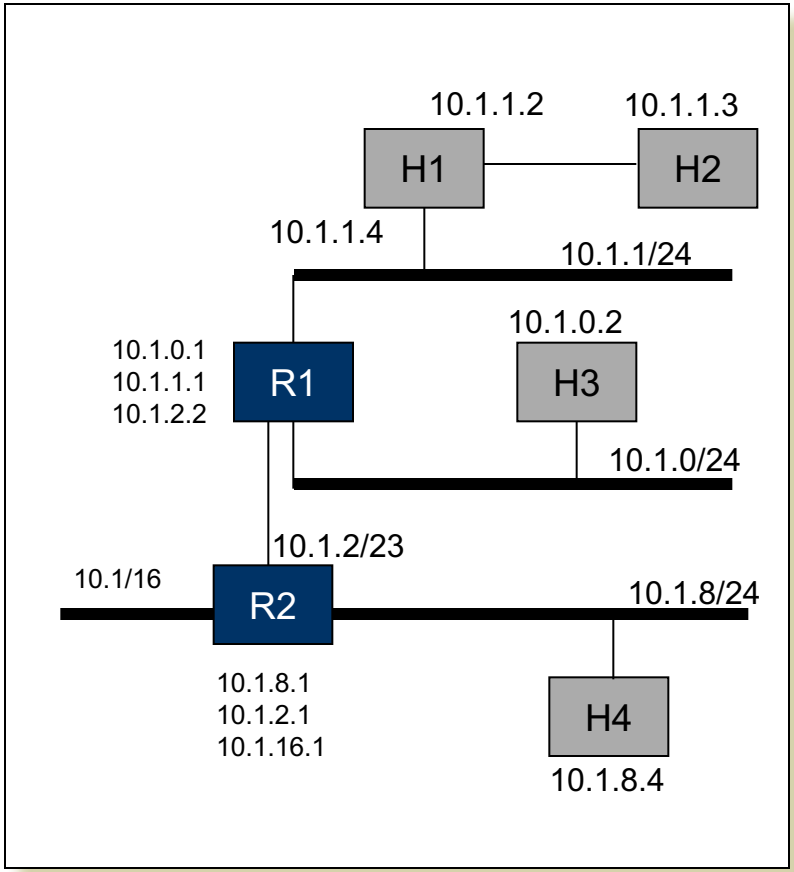


Routing Within the Subnet

- Packet to 10.1.1.3
- Matches 10.1.0.0/23

Routing table at R2

Destination	Next Hop	Interface
127.0.0.1	-	lo0
Default or 0/0	provider	10.1.16.1
10.1.8.0/24	-	10.1.8.1
10.1.2.0/23	-	10.1.2.1
10.1.0.0/23	10.1.2.2	10.1.2.1

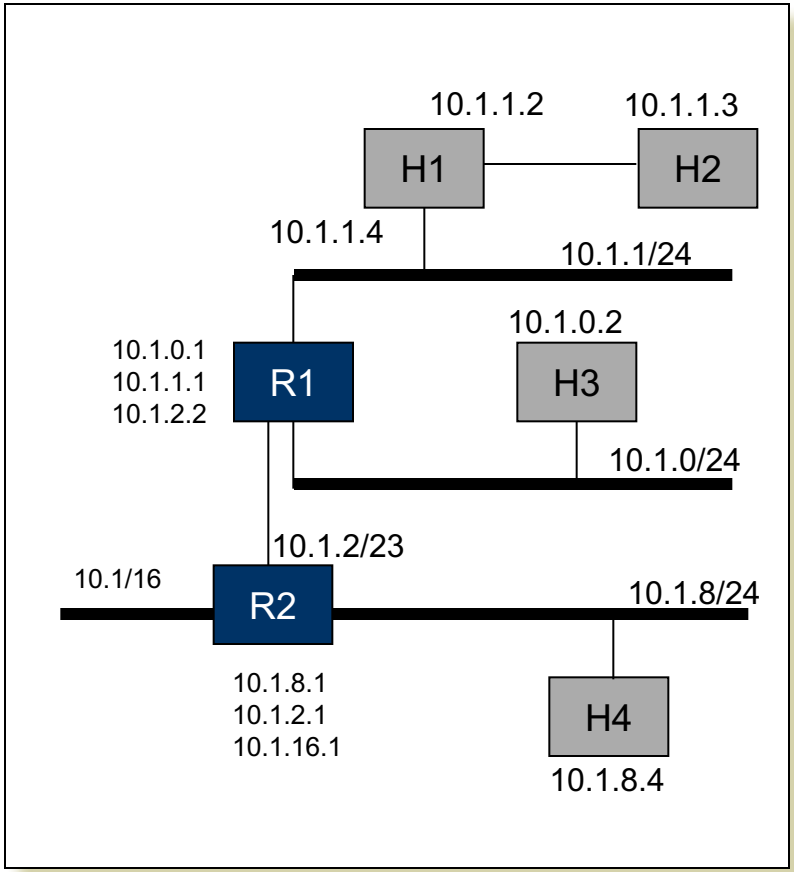


Routing Within the Subnet

- Packet to 10.1.1.3
- Matches 10.1.1.2/31
  - Longest prefix match

Routing table at R1

Destination	Next Hop	Interface
127.0.0.1	-	lo0
Default or 0/0	10.1.2.1	10.1.2.2
10.1.2.0/23	10.1.2.1	10.1.2.2
10.1.0.0/24	-	10.1.0.1
10.1.1.0/24	-	10.1.1.1
10.1.1.2/31	10.1.1.4	10.1.1.1

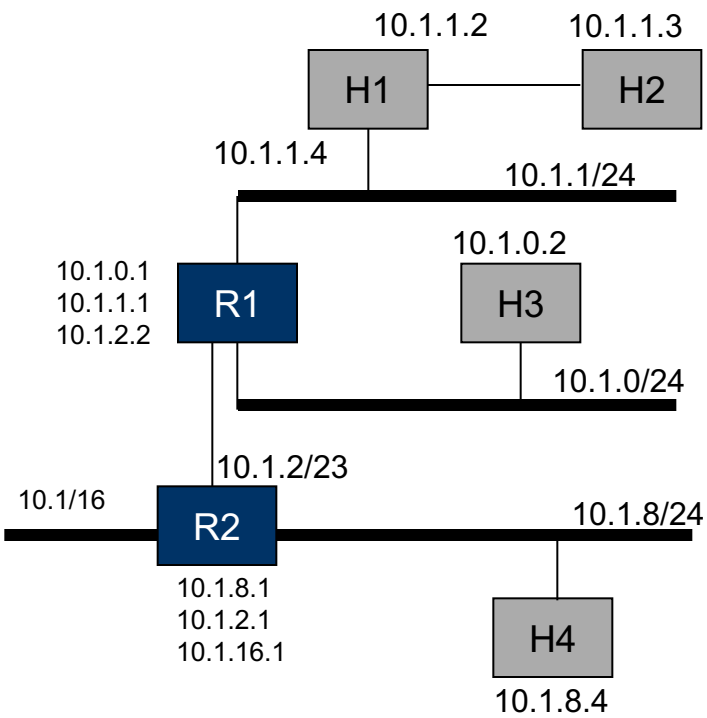


Routing Within the Subnet

- Packet to 10.1.1.3
- Direct route
  - Longest prefix match

Routing table at H1

Destination	Next Hop	Interface
127.0.0.1	-	lo0
Default or 0/0	10.1.1.1	10.1.1.4
10.1.1.0/24	-	10.1.1.1
10.1.1.2/31	-	10.1.1.2



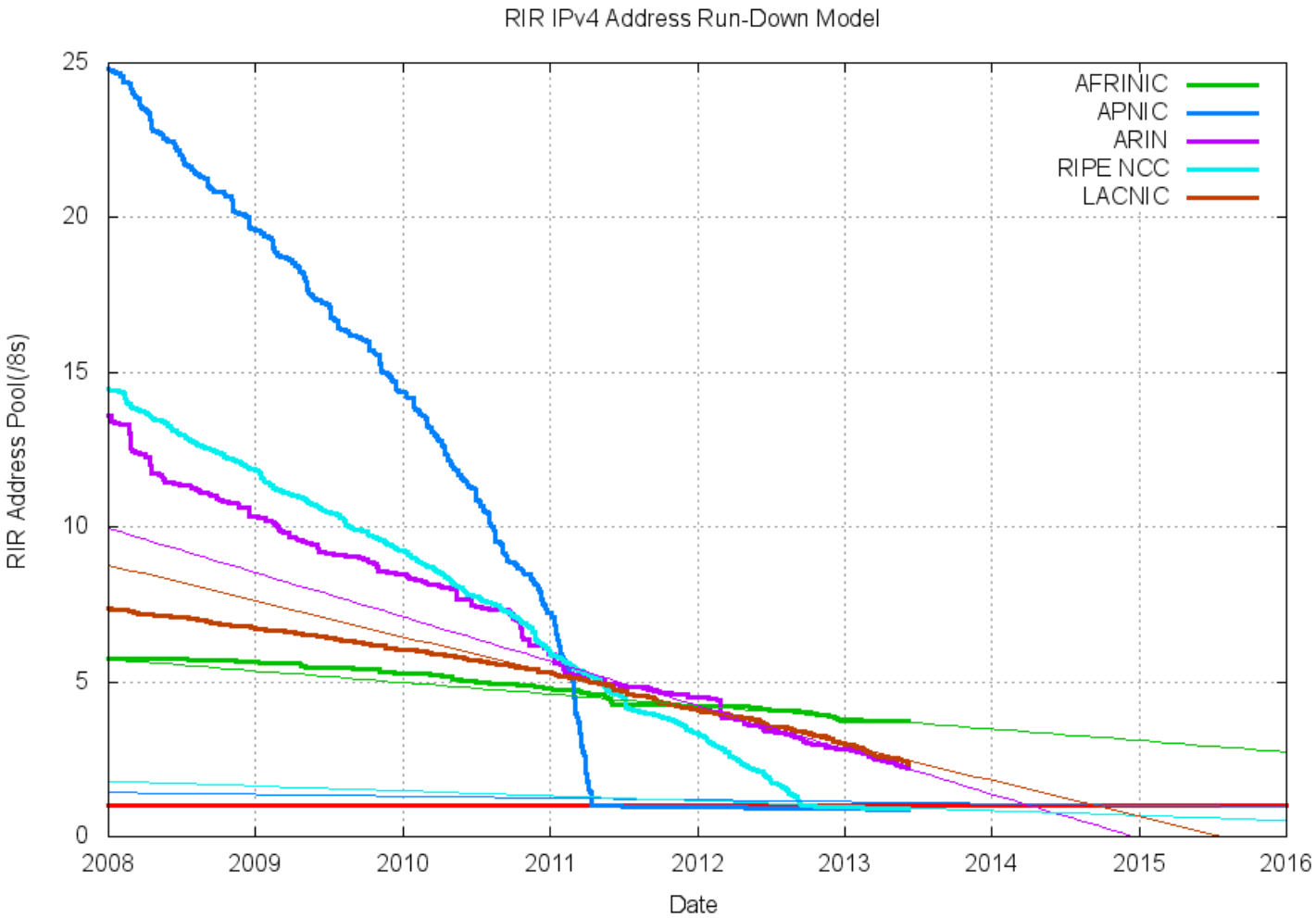
Important Concepts

- Hierarchical addressing critical for scalable system
  - Don’t require everyone to know everyone else
  - Reduces number of updates when something changes
- Classless inter-domain routing supports more efficient use of address space
  - Adds complexity to routing, forwarding, …
    - But Scalable!

IP Addresses: How to Get One?

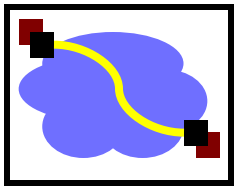
- How does an ISP get block of addresses?
  - From **Regional Internet Registries** (RIRs)
    - ARIN (North America, Southern Africa), APNIC (Asia-Pacific), RIPE (Europe, Northern Africa), LACNIC (South America)
- How about a single host?
  - Assigned by sys admin (static or dynamic)
  - DHCP: Dynamic Host Configuration Protocol: dynamically get address: "plug-and-play"
    - Host broadcasts "DHCP discover" msg
    - DHCP server responds with "DHCP offer" msg
    - Host requests IP address: "DHCP request" msg
    - DHCP server sends address: "DHCP ack" msg

IP Address Availability Remains a Major Challenge

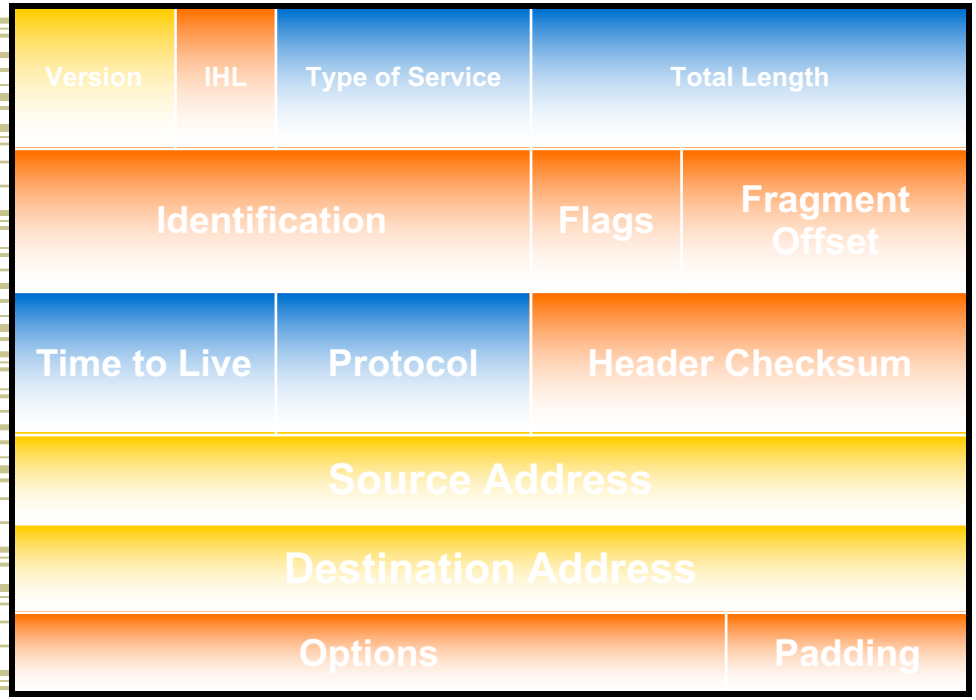


- Some are in big trouble!
- APNIC:
  - Asia
- AFRINIC:
  - Africa
- ARIN:
  - North America
- LACNIC:
  - Latin America
- RIPE NCC:
  - Europe, Middle East, parts of central Asia

IPv4 and IPv6 Header Comparison



IPv4



IPv6

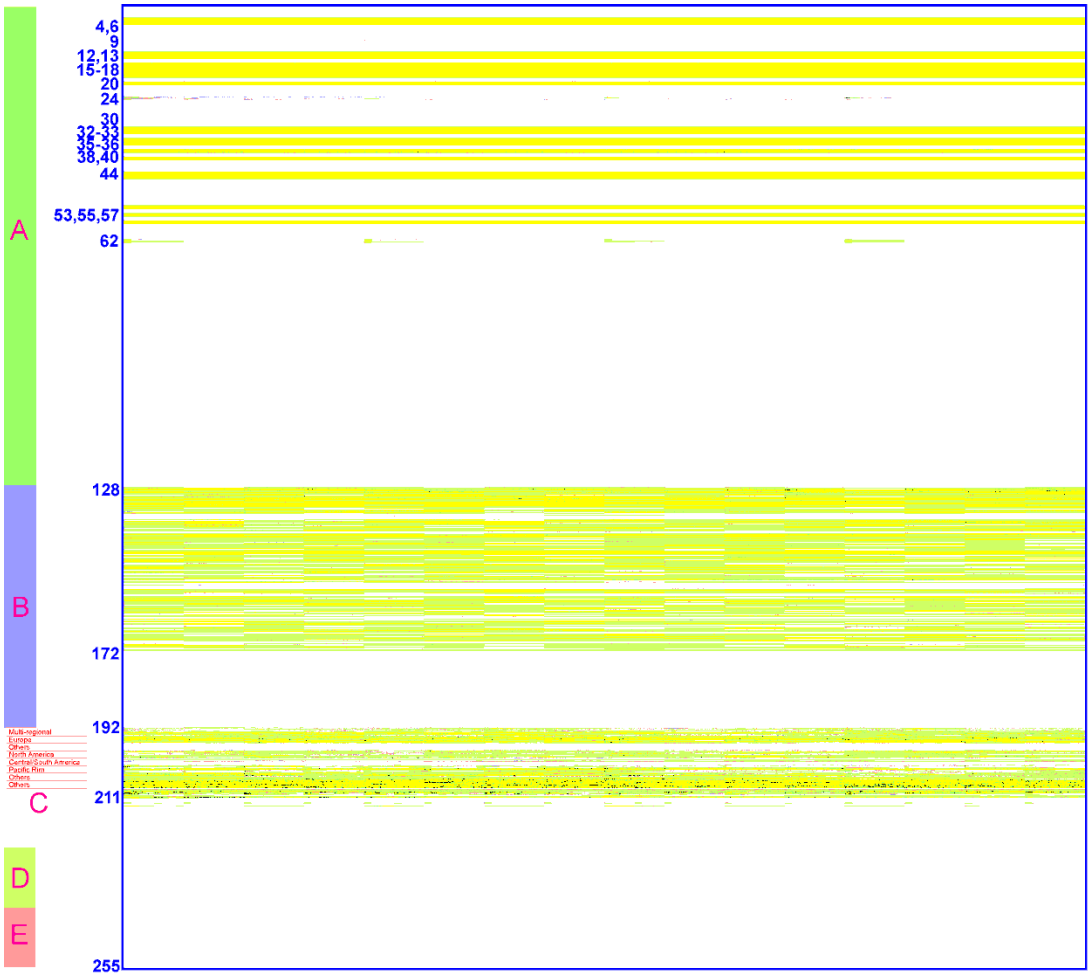
- Field name kept from IPv4 to IPv6
- Fields not kept in IPv6
- Name & position changed in IPv6
- New field in IPv6

Tuesday:
Network Address Translation

- (No Spoilers)

EXTRA SLIDES

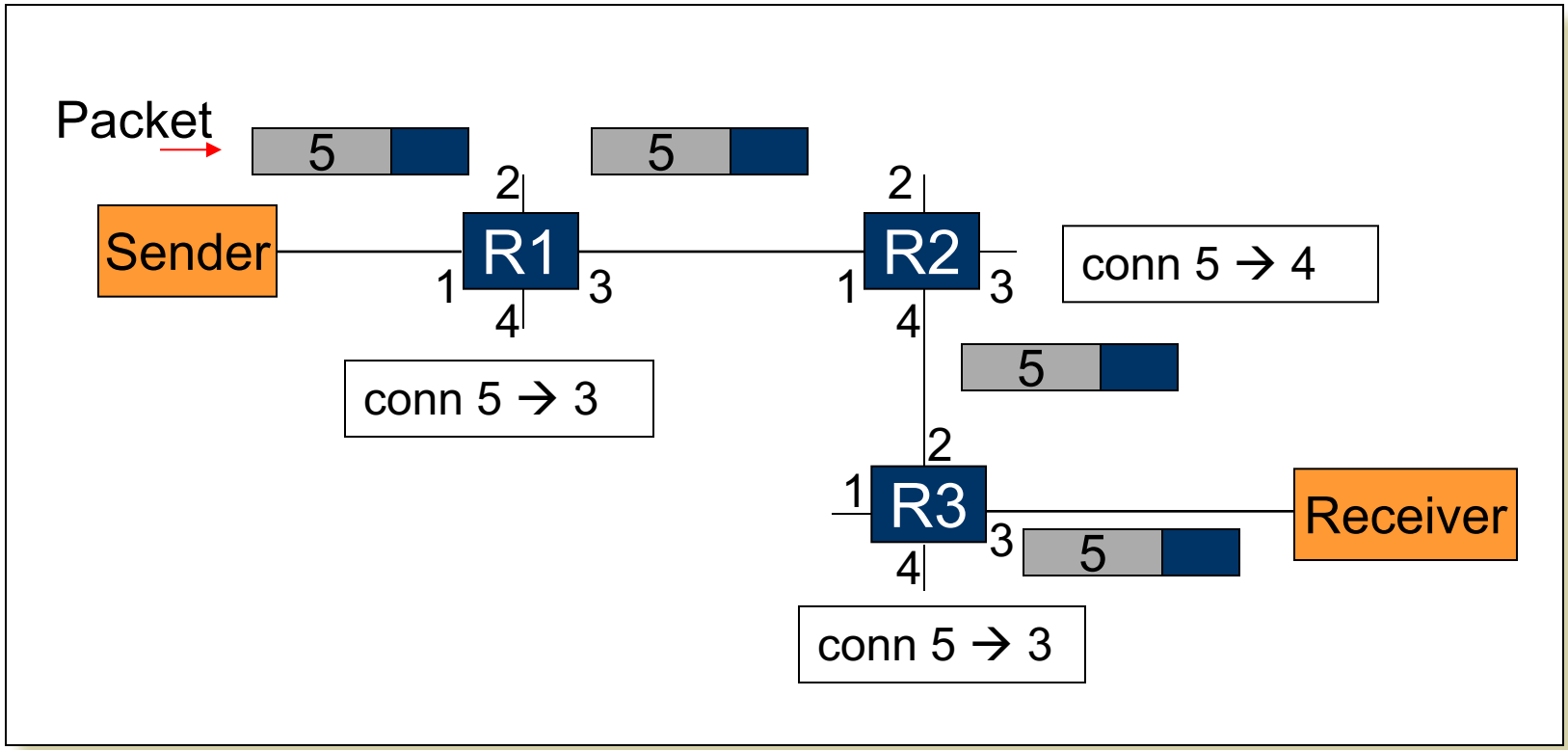
IP Address Utilization ('97)



Simplified Virtual Circuits

- Connection setup phase
  - Use other means to route setup request
  - Each router allocates flow ID on local link
- Each packet carries connection ID
  - Sent from source with 1st hop connection ID
- Router processing
  - Lookup flow ID – simple table lookup
  - Replace flow ID with outgoing flow ID
  - Forward to output port

Simplified Virtual Circuits Example



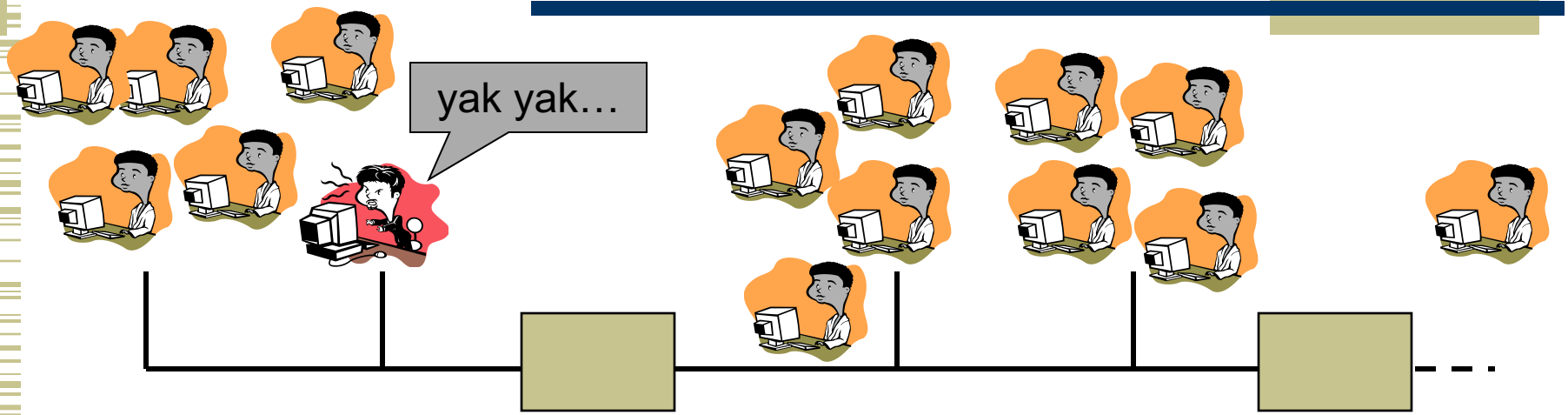
Virtual Circuits

- Advantages
  - Efficient lookup (simple table lookup)
  - Can reserve bandwidth at connection setup
  - Easier for hardware implementations
- Disadvantages
  - Still need to route connection setup request
  - More complex failure recovery – must recreate connection state
- Typical use → fast router implementations
  - ATM – combined with fix sized cells
  - MPLS – tag switching for IP networks

Some Special IP Addresses

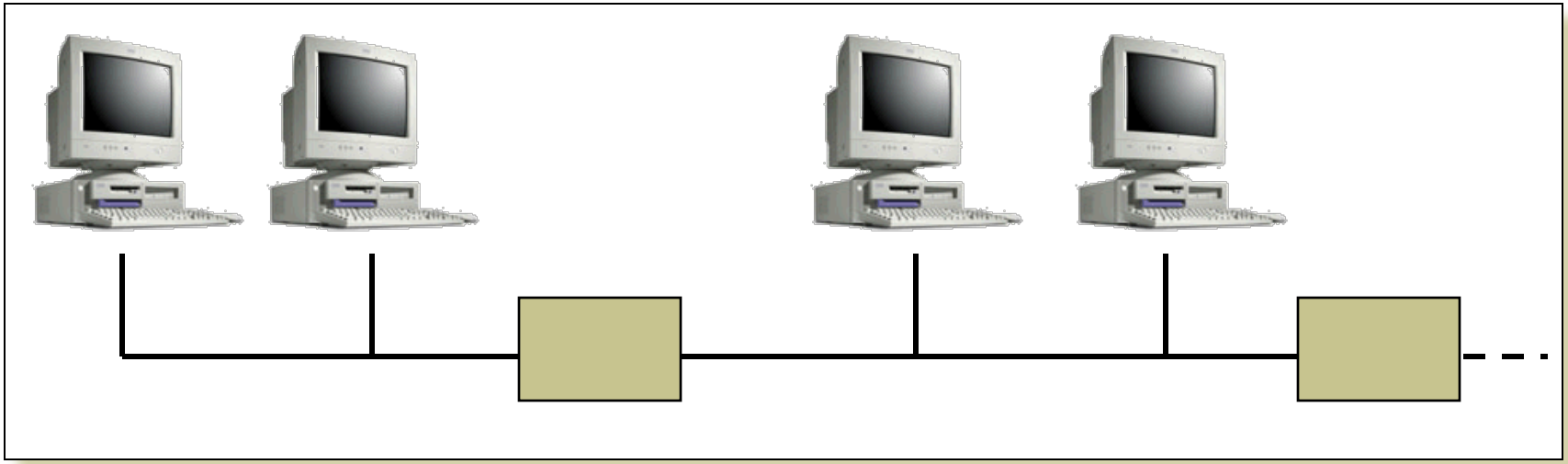
- 127.0.0.1: local host (a.k.a. the loopback address
- Host bits all set to 0: network address
- Host bits all set to 1: broadcast address

Problem 1 – Reconnecting LANs



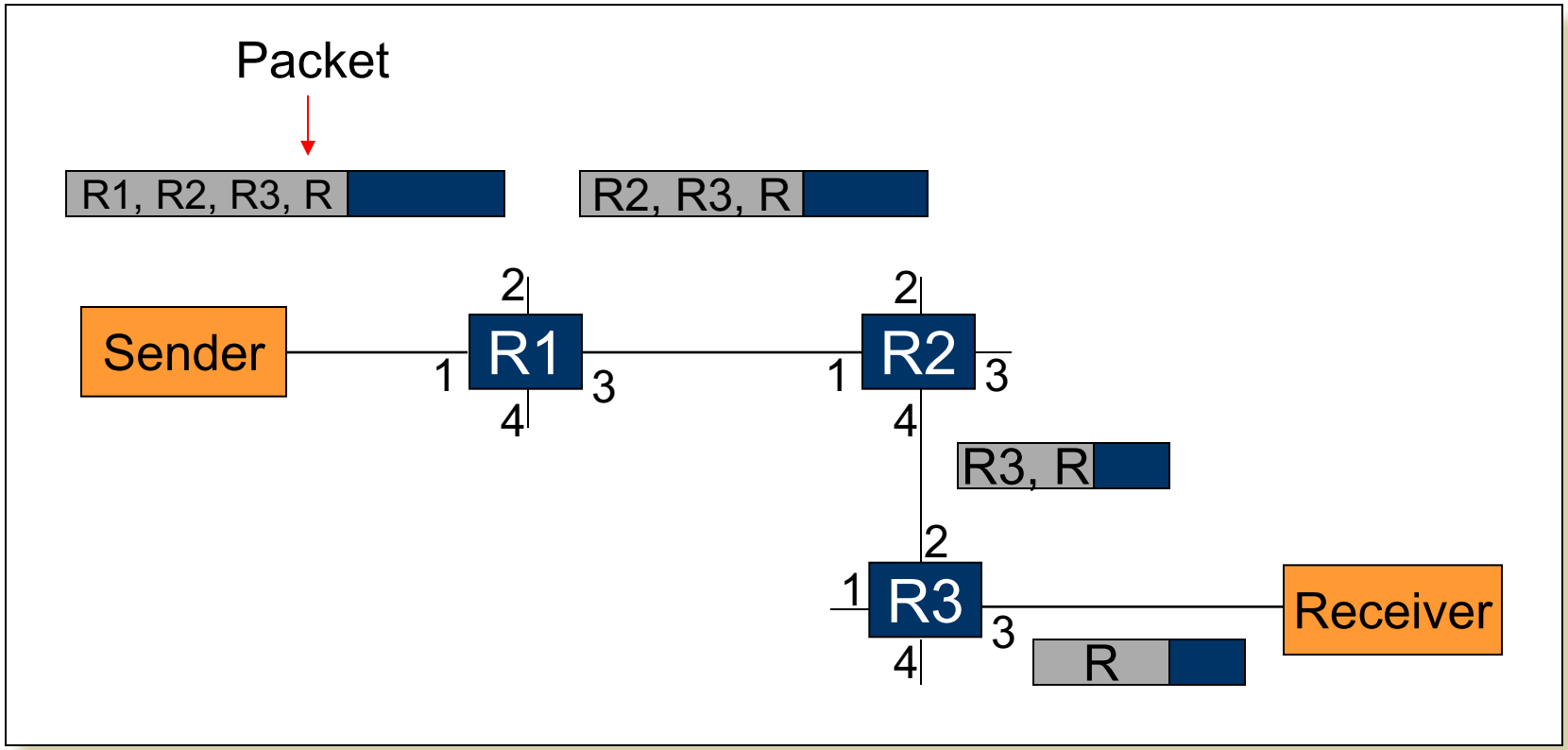
- When should these boxes forward packets between wires?
- How do you specify a destination?
- How does your packet find its way?

Problem 2 – Bridging Weaknesses



- Doesn’t handle incompatible LAN technologies
- How well does it scale?

Source Routing Example



Global Addresses (IP)

- Each packet has destination address
- Each router has forwarding table of destination → next hop
  - At v and x: destination → east
  - At w and y: destination → south
  - At z: destination → north
- Distributed routing algorithm for calculating forwarding tables

Source Routing

- Advantages
  - Switches can be very simple and fast
- Disadvantages
  - Variable (unbounded) header size
  - Sources must know or discover topology (e.g., failures)
- Typical uses
  - Ad-hoc networks (DSR)
  - Machine room networks (Myrinet)

Comparison

	Source Routing	Global Addresses	Virtual Circuits
Header Size	Worst	OK – Large address	Best
Router Table Size	None	Number of hosts (prefixes)	Number of circuits
Forward Overhead	Best	Prefix matching (Worst)	Pretty Good
Setup Overhead	None	None	Connection Setup
Error Recovery	Tell all hosts	Tell all routers	Tell all routers and Tear down circuit and re-route

To Do

- Add some of material from PKU course
  - What is required?
  - Scalability as a driver: modularity
  - Other components of the architecture is a design that embeds choices
    - Other options possible
  - Choices used for the Internet
    - E-E principle
    - Emphasis on availability, simplicity (check Clark)
- Something on: name, IP addr, MAC addr?

Outline

- IP design concepts
  - “The design philosophy of the DARPA Internet Protocols”, Dave Clark, SIGCOMM 88
  - “End-to-end arguments in system design”, Saltzer, Reed, and Clark, ACM Transactions on Computer Systems, November 1984
- Traditional IP addressing
- CIDR

Goals of the Internet [Clark88]

0 **Connect existing networks**

initially ARPANET and ARPA packet radio network

1. Survivability

ensure communication service even in the presence of network and router failures

2. Support multiple types of services
3. Must accommodate a variety of networks
4. Allow distributed management
5. Allow host attachment with a low level of effort
6. Be cost effective
7. Allow resource accountability

Goal 0: Connecting Networks

- How to internetwork various network technologies
  - ARPANET, X.25 networks, LANs, satellite networks, packet networks, serial links…
- Many differences between networks
  - Address formats
  - Performance – bandwidth/latency
  - Packet size
  - Loss rate/pattern/handling
  - Routing

IP Standardization

- Minimum set of assumptions that underlying networks must meet to be part of the Internet
  - Minimum packet size, addressing, header format, ..
  - Very simply service model (more on this later)
- Alternative: translation “gateways” – N^2 solution!
- Important non-assumptions:
  - Perfect reliability
  - Support for broadcast, multicast, or other services
  - Priority handling of traffic
  - Internal knowledge of delays, speeds, failures, etc
- No assumption about how each network works internally

Goal 1: Survivability

- If network is disrupted and reconfigured…
  - Communicating entities should not care!
  - No higher-level state reconfiguration
- How to achieve such reliability?
- Key question: where to keep communication state?

	Store in Network	Store on Host
Failure handling	Replicate the state	Natural “Fate sharing”
Switches …	Must maintain state	Are stateless
Net Engineering	Tough	Simple
Trust in host	Less	More

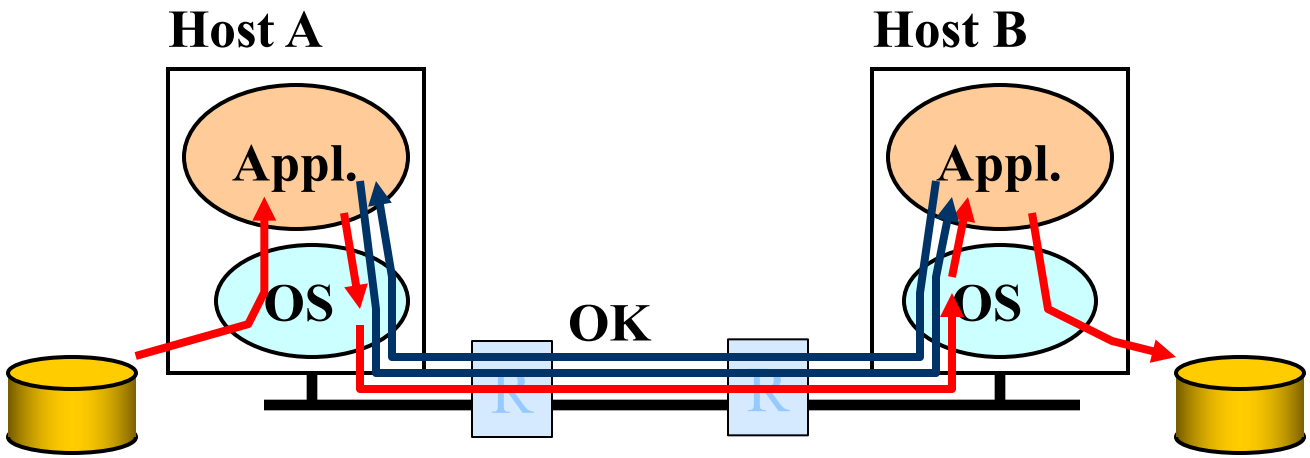
Principle: Soft-state

- How can I not have state in the network, e.g., forwarding tables? Kind of silly.
- Solution: Soft-state
  - Announce state
  - Refresh state
  - Timeout state
- Loss of state results in loss of performance, not loss of connectivity
  - E.g., timeout increases latency
- Survivability is more important than performance

Principle: End-to-End Argument (Saltzer'81)

- Argument: Some functions can only be correctly implemented by the endpoints – do not try to implement these elsewhere
  - Not a law – more of a "best practices"
- Deals with where to place functionality
  - Inside the network (in switching elements)
  - At the edges
- Focus of the paper is "distributed system"
  - Not a pure networking paper

Example: Reliable File Transfer



- Solution 1: make each step reliable, and then concatenate them
  - Expensive, complex, may backfire
- Solution 2: end-to-end check and retry
  - Simpler and cheaper – cost failure is loss of performance
- Take-away: no need to make packet delivery reliable!

Other Examples Throughout Course

- What should be done at the end points, and what by the network?
  - Reliable/sequenced delivery?
  - Addressing/routing?
  - Security?
  - Multicast?
  - Real-time guarantees?
  - Routing?

The “Other” goals

2. Types of service: only best effort service
   - Other services are optional, e.g., QoS
   - Or at end-points: TCP versus UDP
4. Decentralization: Internet – see BGP lecture
   - Allow distributed management of resources
5. Attaching a host
   - Host must implement hard part ☹ → transport services
6. Cost effectiveness: minimalist approach to IP
   - Packet overhead less important by the year
7. Accountability: “accounting” for resources
   - Accounting for billing purposes versus
   - Security: huge problem in the Internet today!

Changes Over Time → New Principles?

- Developed in simpler times
  - Common goals, consistent vision
- With success came changes in Internet goals
  - ISPs must talk to provide connectivity but are fierce competitors
  - Privacy of users vs. government's need to monitor
  - User's desire to exchange files vs. copyright owners
  - Security breaches vs. universal connectivity

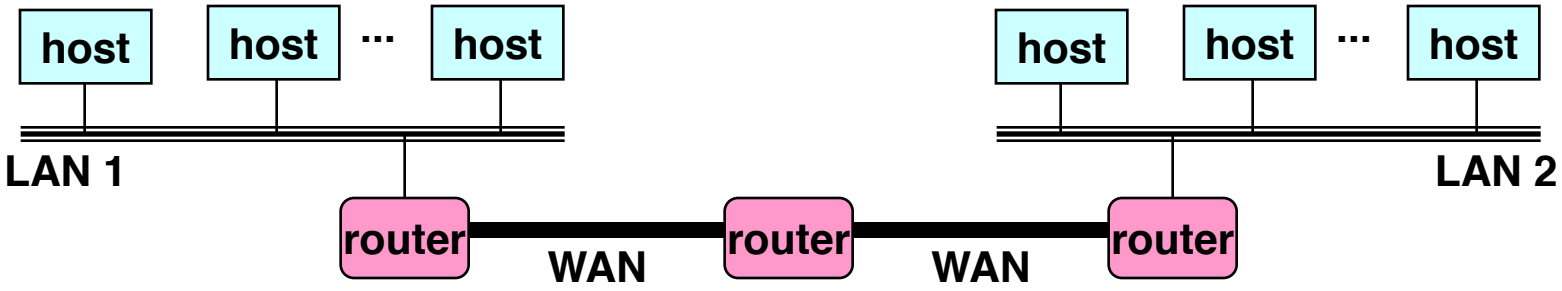
- Provide choice → allow all parties to make choices on interactions – "tussle"
  - Can be tuned at different times: runtime, contract, hardware, ..
  - Creates competition, adjust for different contexts, ..
  - Fear between providers helps shape the tussle

Summary: Minimalist Approach

- Dumb network – focus on basic connectivity
  - IP provides minimal functionality: Addressing, forwarding, routing
- Smart end system – all other (complex) functions
  - Transport, application layers: sophisticated functionality
    - Flow control, error control, congestion control
- Advantages
  - Accommodate heterogeneous technologies (Ethernet, modem, satellite, wireless)
  - Support diverse applications (telnet, ftp, Web, X windows)
  - Decentralized network administration
- But the Internet has evolved – revisit at end of course

What is an Internetwork?

- Multiple incompatible LANs can be physically connected by specialized computers called *routers*
- The connected networks are called an *internetwork*
  - The "*Internet*" is one (very big & successful) example of an internetwork



LAN 1 and LAN 2 might be completely different, totally incompatible LANs (e.g., Ethernet and ATM)