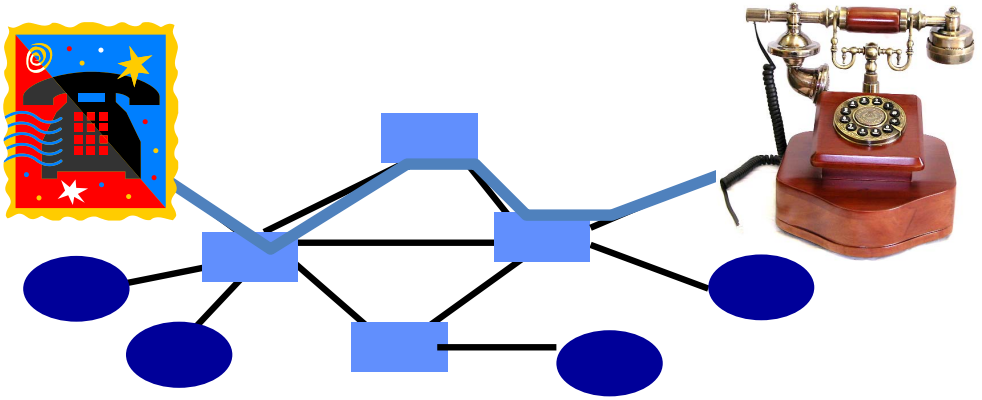
IP Protocol Stack: Key Abstractions

Best-Effort Global Packet Delivery

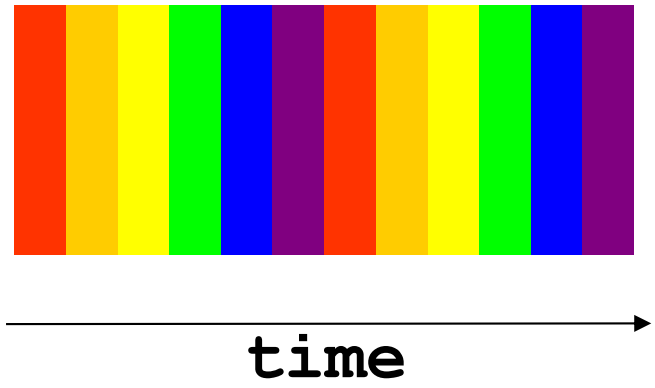
Circuit Switching (e.g., Phone Network)

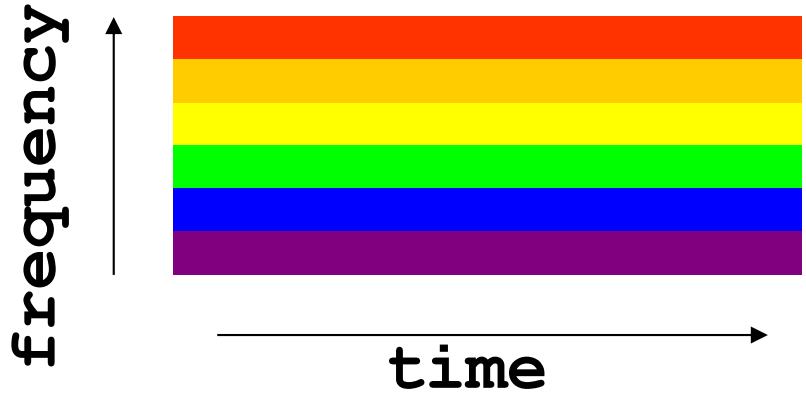
- Source establishes connection
  - Reserve resources along hops in the path
- Source sends data
  - Transmit data over the established connection
- Source tears down connection
  - Free the resources for future connections



Circuit Switching: Static Allocation

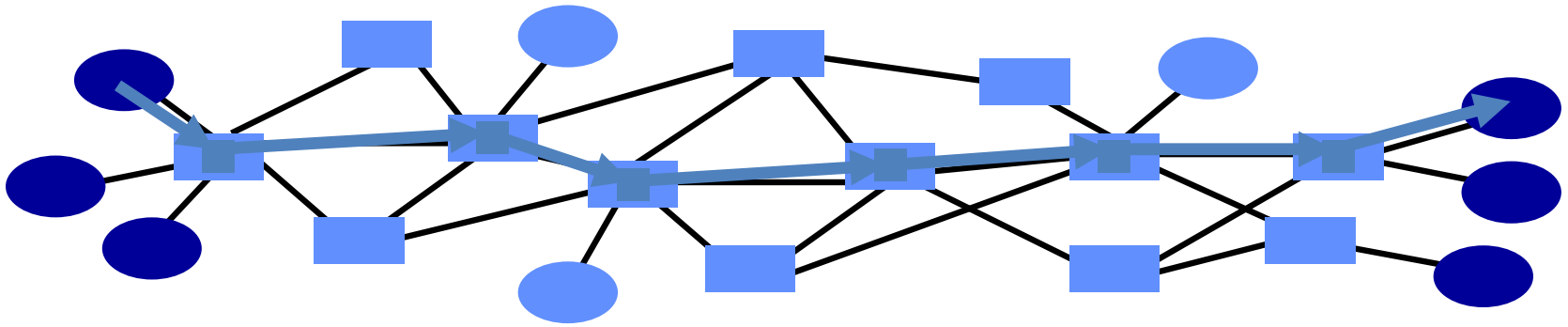
- Time-division
  - Each circuit allocated certain time slots
- Frequency-division
  - Each circuit allocated certain frequencies



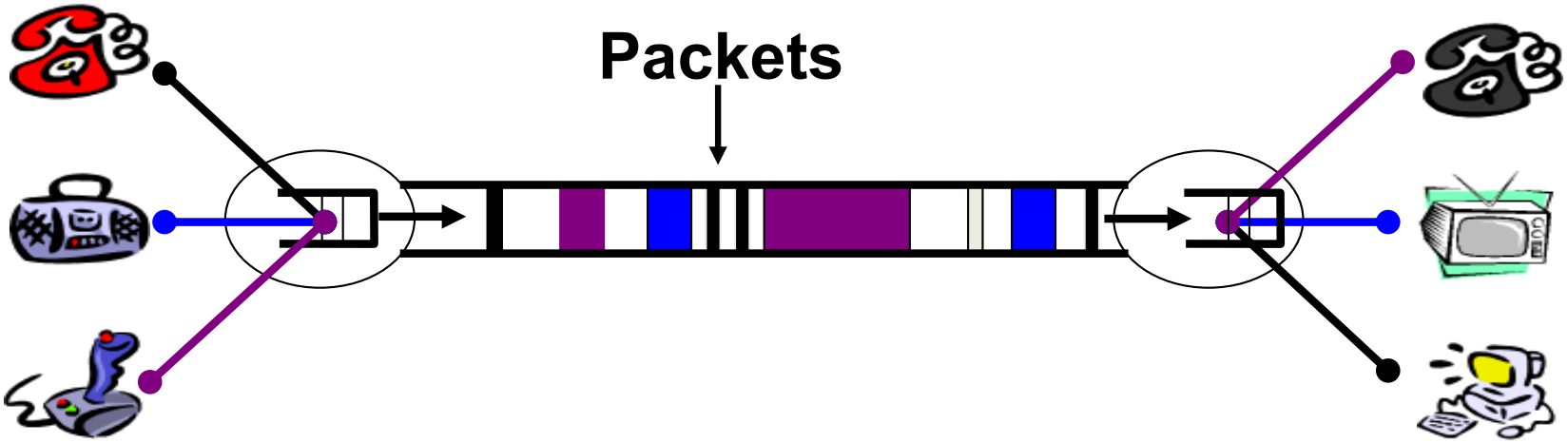


Packet Switching

- Message divided into packets
  - Header identifies the destination address
- Packets travel separately through the network
  - Forwarding based on the destination address
  - Packets may be buffered temporarily
- Destination reconstructs the message



Packet Switching: Statistical (Time Division) Multiplexing



- Intuition: Traffic by computer end-points is bursty!
  - Versus: Telephone traffic not bursty (e.g., constant 56 kbps)
  - One can use network while others idle
- Packet queuing in network: tradeoff space for time
  - Handle short periods when outgoing link demand > link speed

Is Best Effort Good Enough?

- Packet loss and delay
  - Sender can resend
- Packet corruption
  - Receiver can detect, and sender can resend
- Out-of-order delivery
  - Receiver can put the data back in order
- Packets follow different paths
  - Doesn't matter
- Network failure
  - Drop the packet
- Network congestion
  - Drop the packet

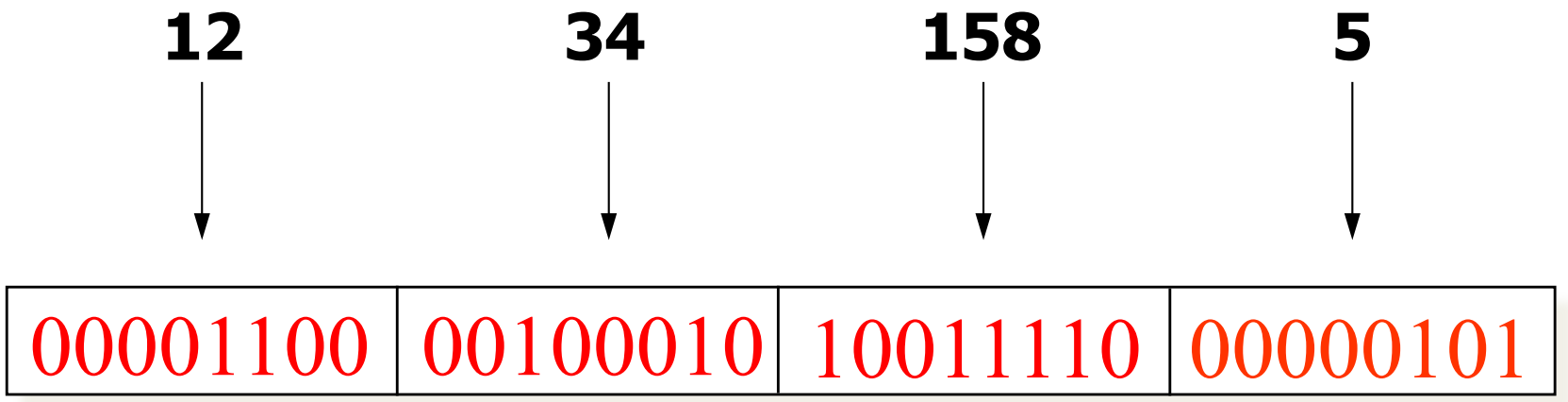
Packet (Y) vs. Circuit Switching (A)?

• Predictable performance	Circuit
• Network never blocks senders	Packet
• Reliable, in-order delivery	Circuit
• Low delay to send data	Packet
• Simple forwarding	Circuit
• No overhead for packet headers	Circuit
• High utilization under most workloads	Packet
• No per-connection network state	Packet

Network Addresses

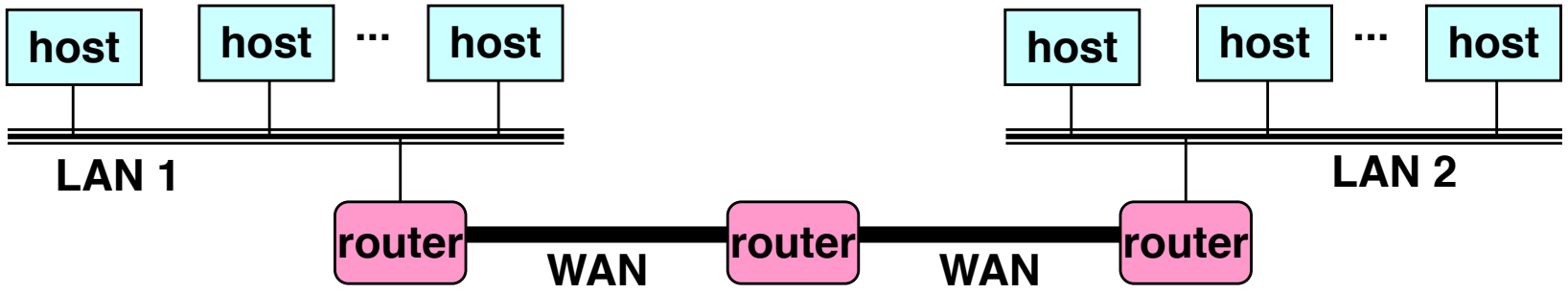
IP Address (IPv4)

- 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
  - Need to address a network (i.e., group of hosts)

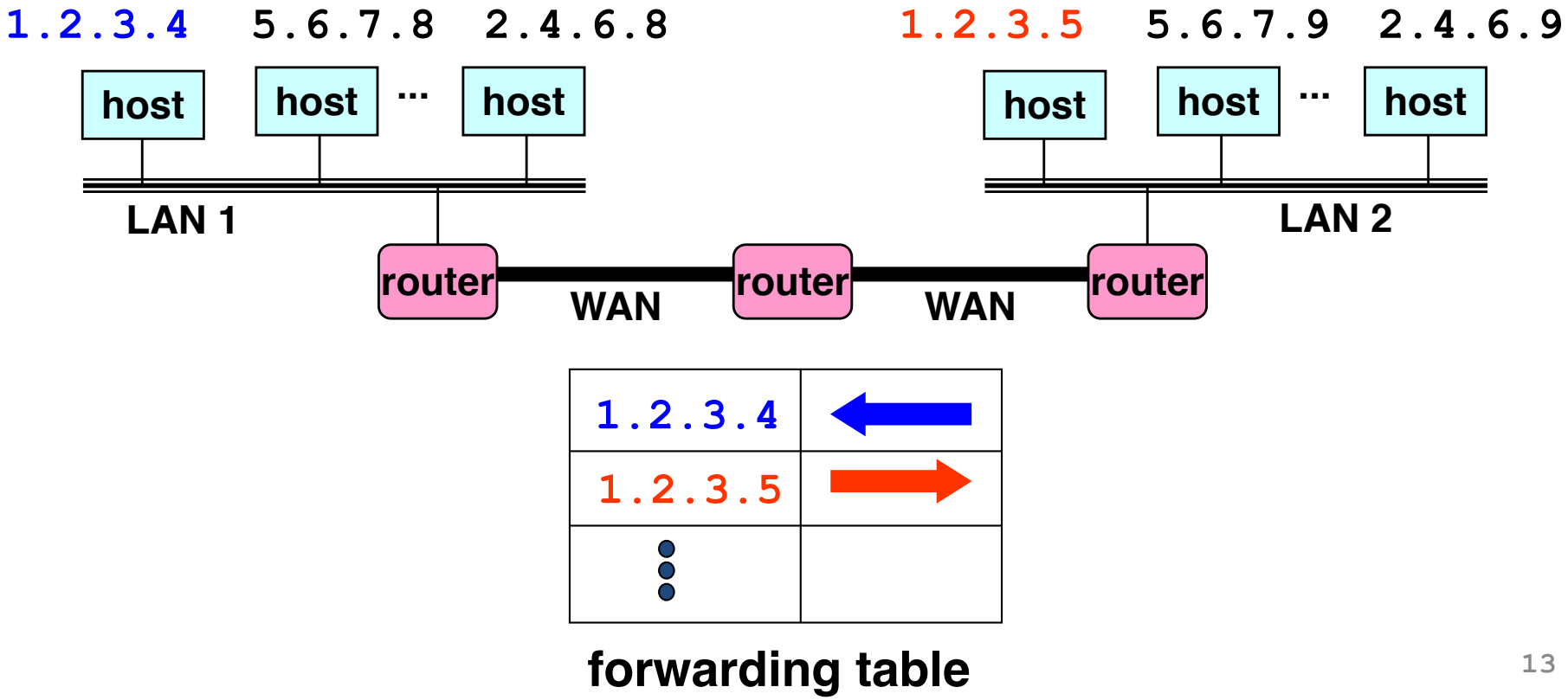


LAN = Local Area Network

WAN = Wide Area Network

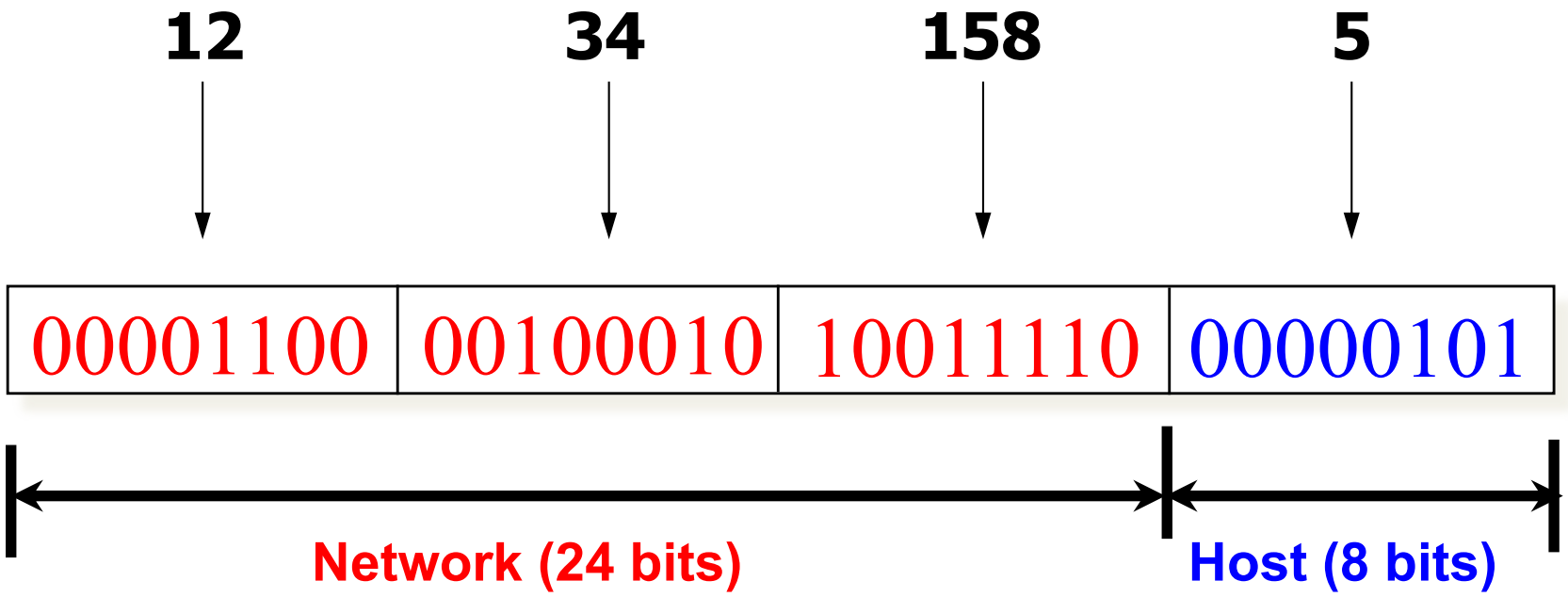
Scalability Challenge

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



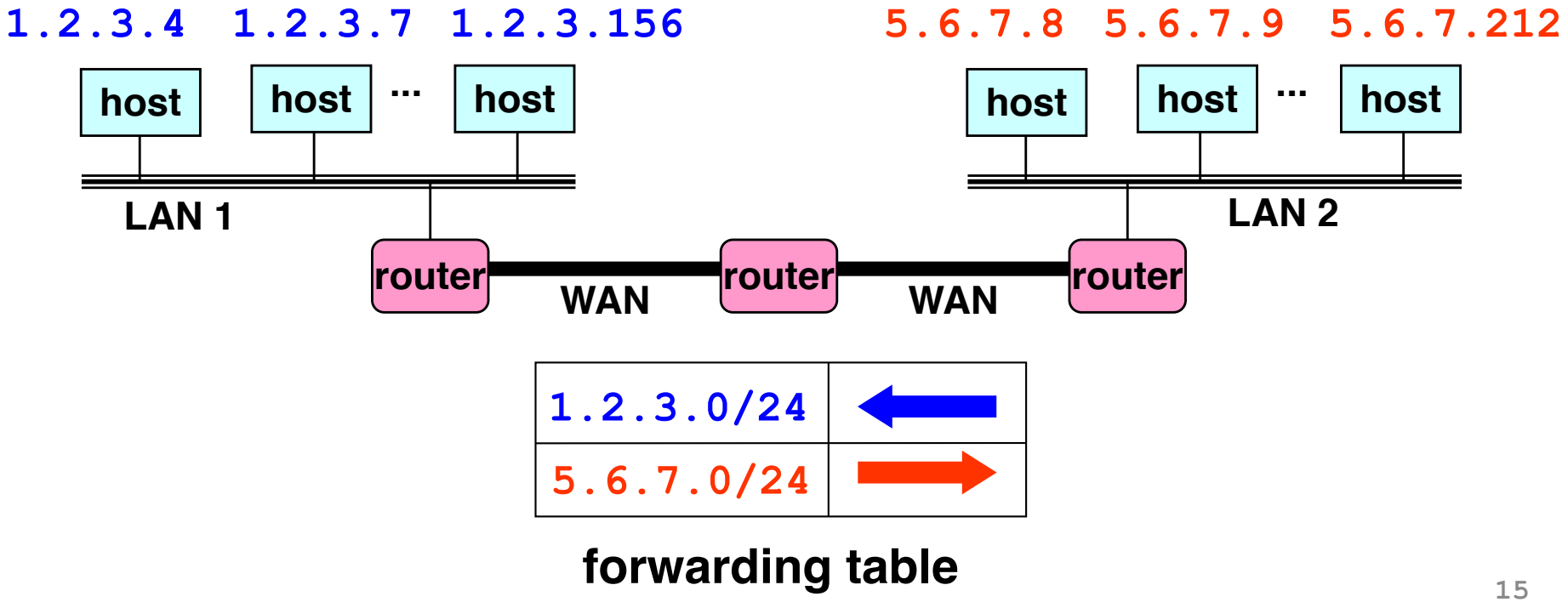
Hierarchical Addressing: IP Prefixes

- Network and host portions (left and right)
- 12.34.158.0/24 is a 24-bit **prefix** with 2^8 addresses



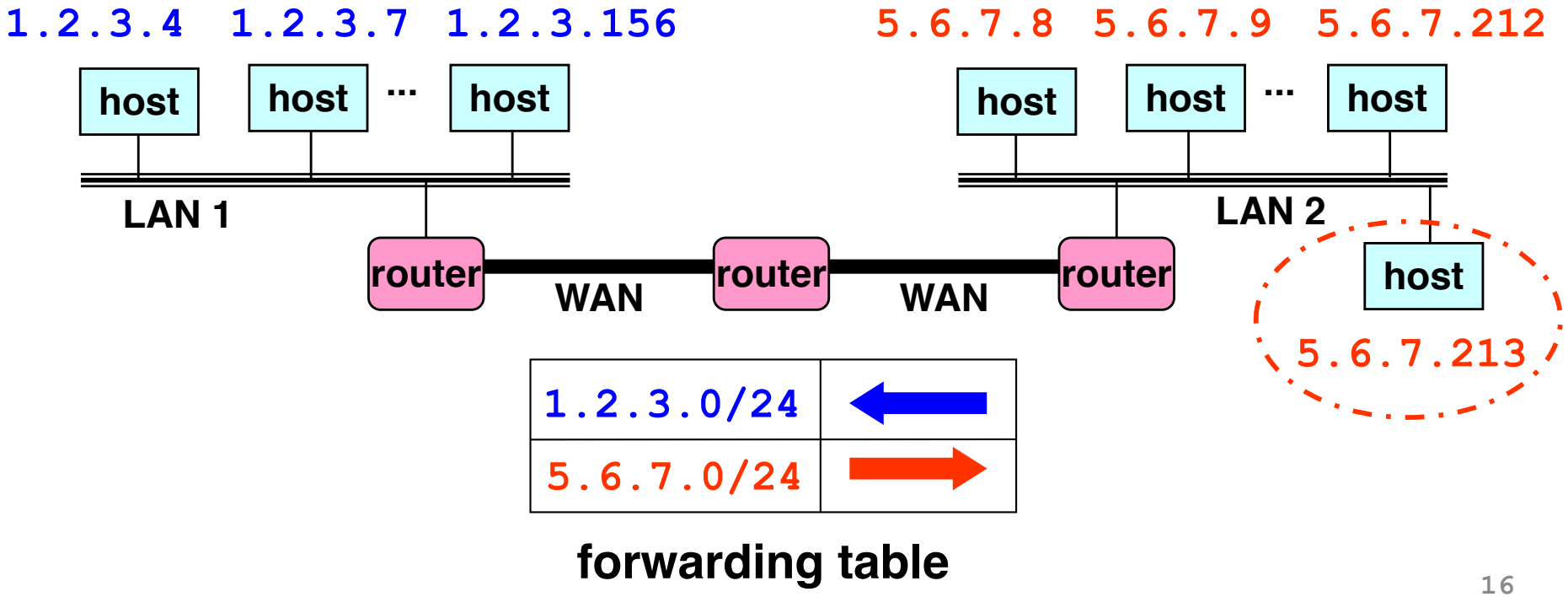
Scalability Improved

- 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



History of IP Address Allocation

Classful Addressing

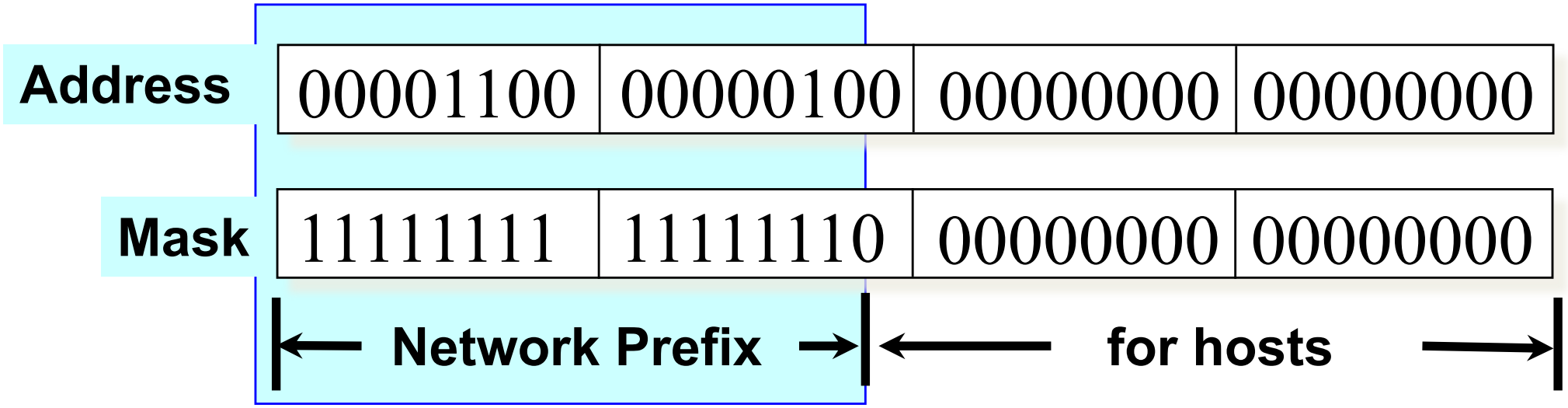
- In the olden days, only fixed allocation sizes
  - Class A: 0*
    - Very large /8 blocks (e.g., MIT has 18.0.0.0/8)
  - Class B: 10*
    - Large /16 blocks (e.g,. Princeton has 128.112.0.0/16)
  - Class C: 110*
    - Small /24 blocks (e.g., AT&T Labs has 192.20.225.0/24)
  - Class D: 1110* for multicast groups
  - Class E: 11110* reserved for future use
- This is why folks use dotted-quad notation!

Classless Inter-Domain Routing (CIDR)

- Use two 32-bit numbers to represent network:

  Network number = IP address + Mask

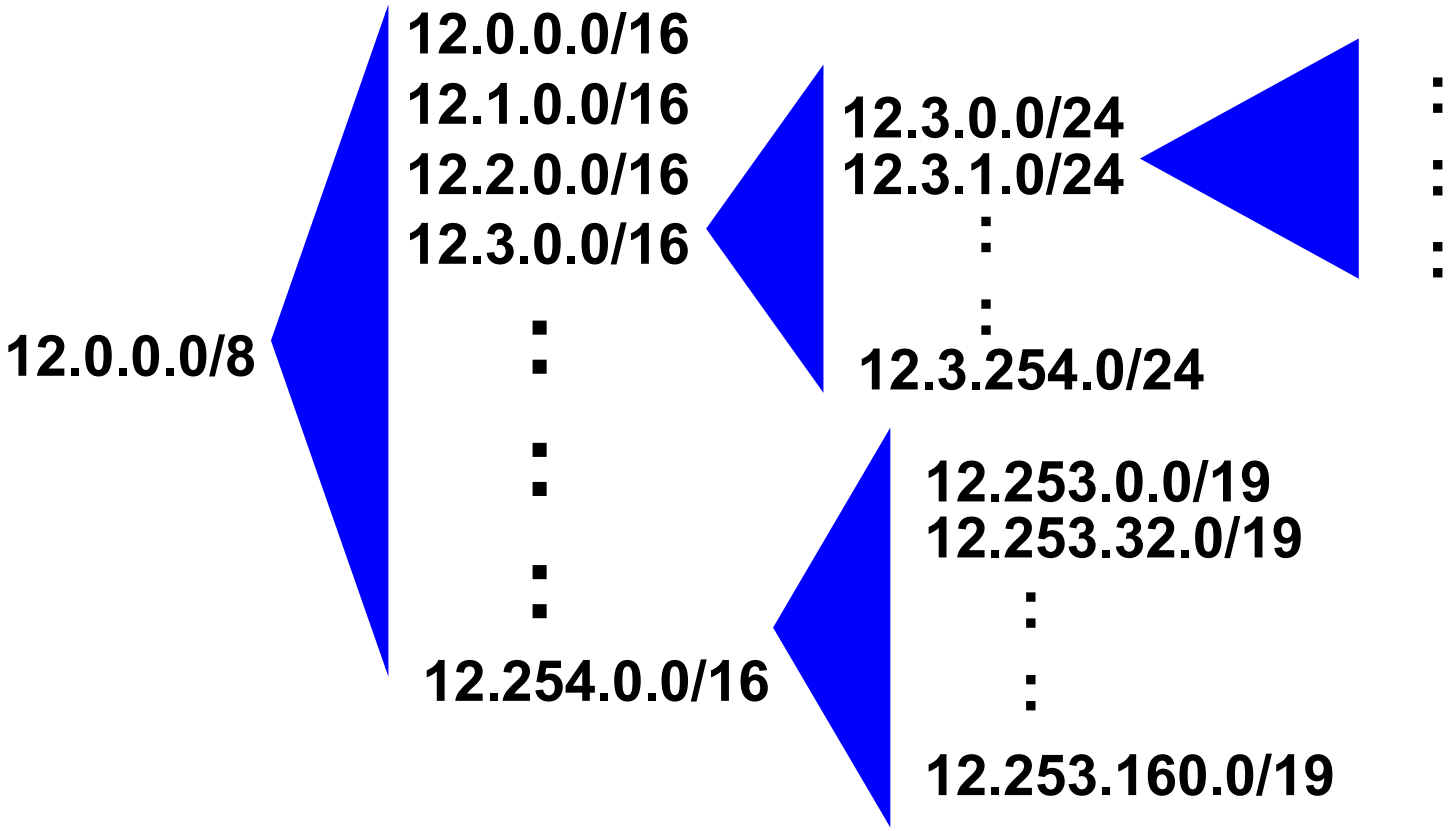
IP Address : 12.4.0.0 IP Mask: 255.254.0.0



Written as 12.4.0.0/15

Hierarchical Address Allocation

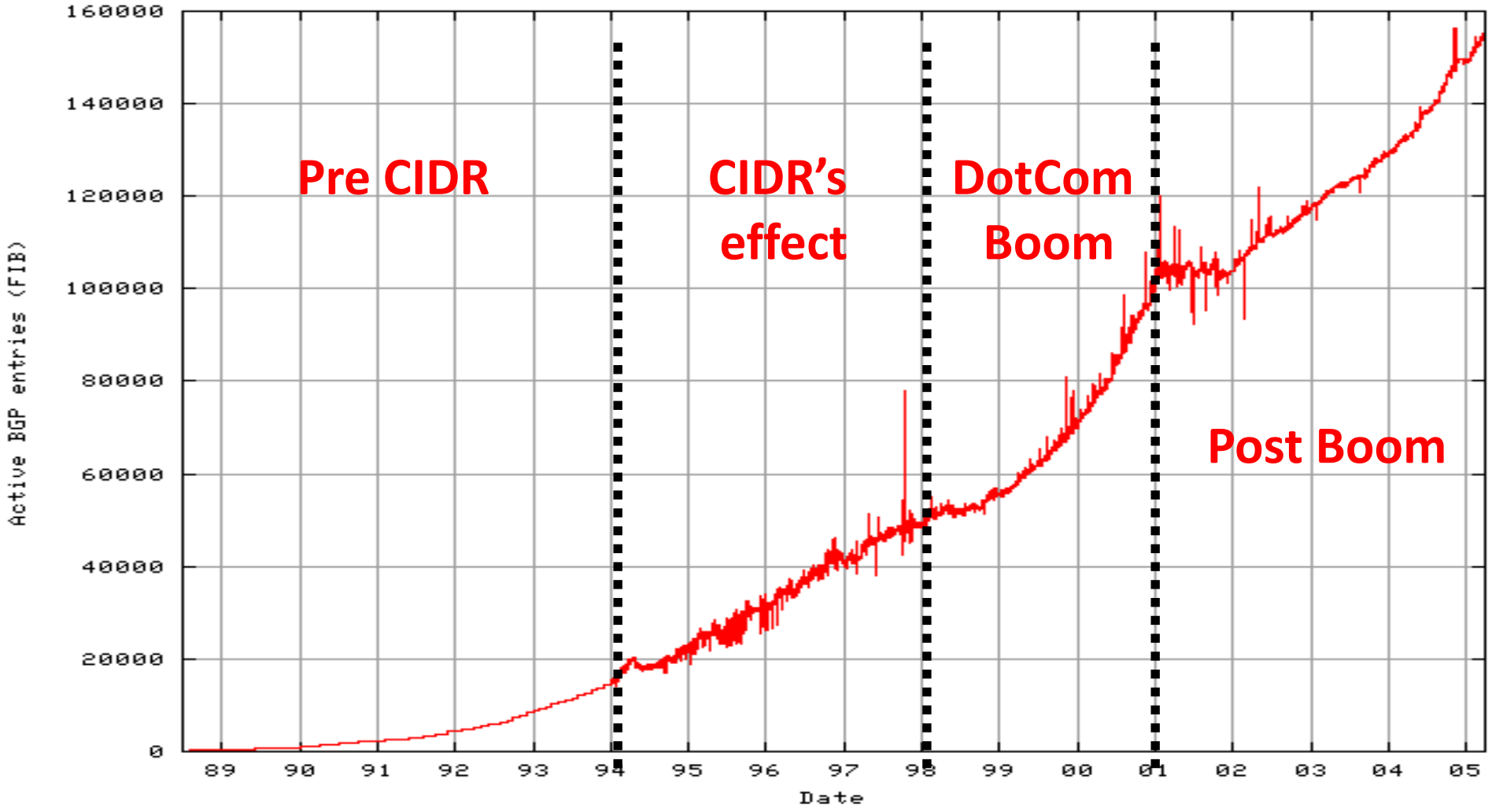
- Hierarchy is key to scalability
  - Address allocated in contiguous chunks (prefixes)
  - Today, the Internet has about 600-800,000 prefixes



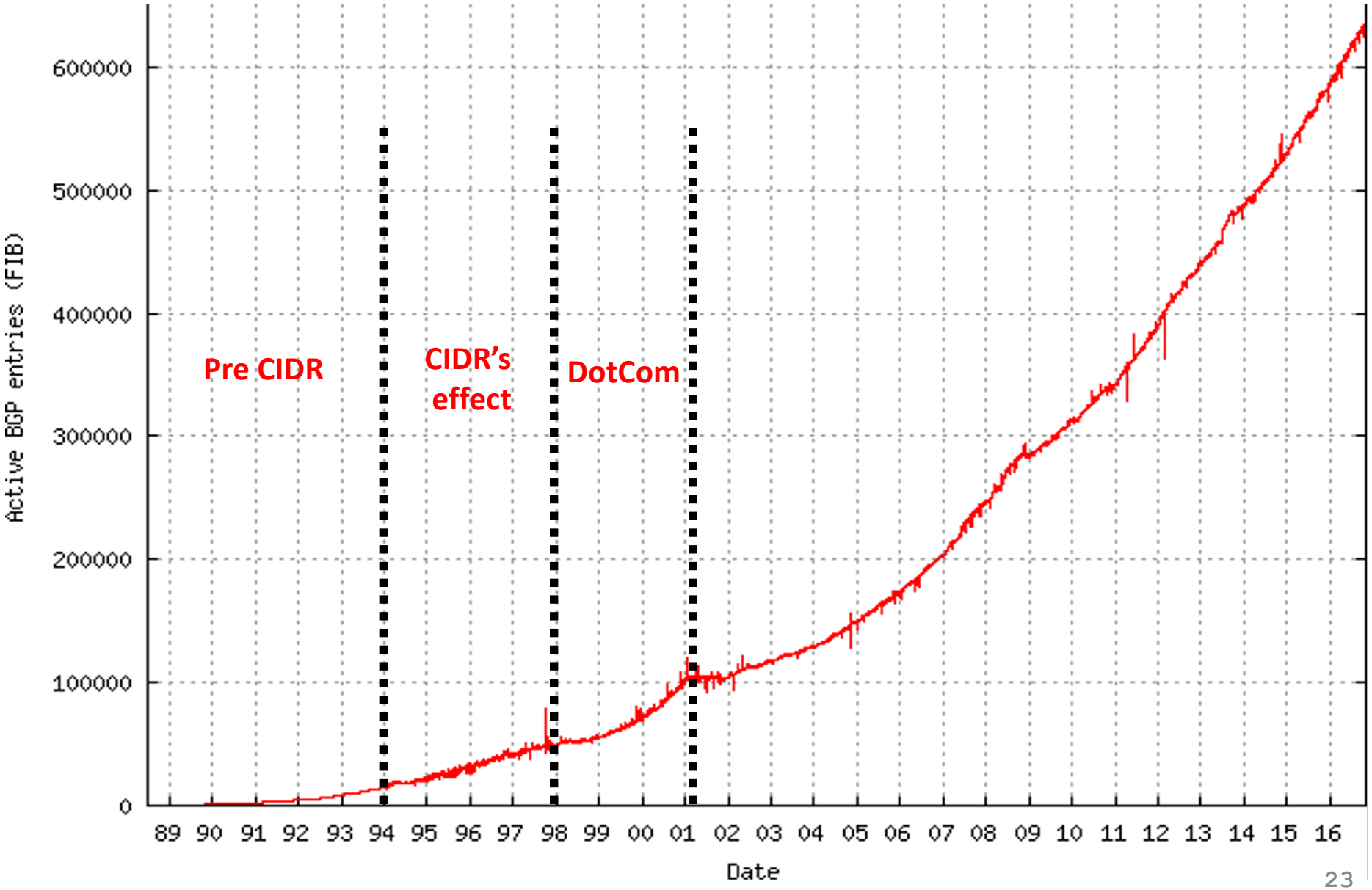
Obtaining a Block of Addresses

- Internet Corporation for Assigned Names and Numbers (ICANN)
  - Allocates large blocks to Regional Internet Registries
- Regional Internet Registries (RIRs)
  - E.g., ARIN (American Registry for Internet Numbers)
  - Allocates to ISPs and large institutions
- Internet Service Providers (ISPs)
  - Allocate address blocks to their customers
  - Who may, in turn, allocate to their customers…

Long Term Growth (1989-2005)



Long Term Growth (1989-2017)



Network addresses

- Classless addressing
  - Can we have exactly same IP addresses with different CIDR notations?
    - IP address would be 194.24.0.20 (regardless of subnet mask)
      - 194.24.0.20/15: 194.24.0.0 - 194.25.255.255
      - 194.24.0.20/16: 194.24.0.0 - 194.24.255.255 [subnet of above]
      - 194.24.0.20/20: 194.24.0.0 - 194.24.15.255 [subnet of above]
      - 194.24.0.20/21: 194.24.0.0 - 194.24.7.255 [subnet of above]
      - 194.24.0.20/24: 194.24.0.0 - 194.24.0.255 [subnet of above]
      - 194.24.0.20/29: 194.24.0.16 - 194.24.0.23 [subnet of above]

Network addresses

- Special addresses in each block
  - First and last network address
    - E.g., 192.168.5.0/24 (subnet mask 255.255.255.0)
      - Network address: 192.168.5.0
      - Directed broadcast address: 192.168.5.255

	Binary form	Dot-decimal
Network address	11000000.10101000.00000101.00000000	192.168.5.0
Directed broadcast address	11000000.10101000.00000101.11111111	192.168.5.255

Network addresses

- Special addresses in each block
  - First and last network address
    - Can address ending with 0 or 255 be used as a host address?
      - **Yes!** only exception are **FIRST** and **LAST** addresses of network
        - E.g., 192.168.0.0/16 (subnet mask 255.255.0.0)
          - Network address: 192.168.0.0
          - Directed broadcast address: 192.168.255.255
          - Addresses usable for hosts
            - 192.168.**1.**0,
            - 192.168.**2.**0,
            - …
            - 192.168.**1.**255,
            - 192.168.**2.**255,
            - …

Network addresses

- Special addresses in each block
  - First and last network address
    - Network address/directed broadcast address always end with 0/255?
      - **No!**
        - E.g., CIDR 203.0.113.16/28
          - Network address: 203.0.113.16
          - Directed broadcast address: 203.0.113.31

	Binary form	**Dot-decimal**
Network address	11001011.00000000.01110001.00010000	203.0.113.16
Directed broadcast address	11001011.00000000.01110001.00011111	203.0.113.31

Network addresses

- Special addresses in each block
  - First and last network address
    - A special case is /31 network (subnet mask 255.255.255.254)
      - Capacity for just two hosts
      - Typically, used for point-to-point connections
      - No network address or directed broadcast address

Network addresses

- Special addresses in each block
  - First and last network address
    - A special case is /32 network (subnet mask 255.255.255.255)
      - Capacity for just one host
      - Cannot be used for assigning address to network links
        - Need more than one address per link
      - Is strictly reserved for use on links that can have only one address
        - E.g., loopback interface
      - No network address or directed broadcast address

Network addresses

- Special blocks of addresses
  - 0.0.0.0/8
    - RFC1700, page 4: *“0.0.0.0/8 - Addresses in this block refer to source hosts on "this" network. Address 0.0.0.0/32 may be used as a source address for this host on this network; other addresses within 0.0.0.0/8 may be used to refer to specified hosts on this network.”*

    - Non-routable address, describes an invalid or unknown target

Network addresses

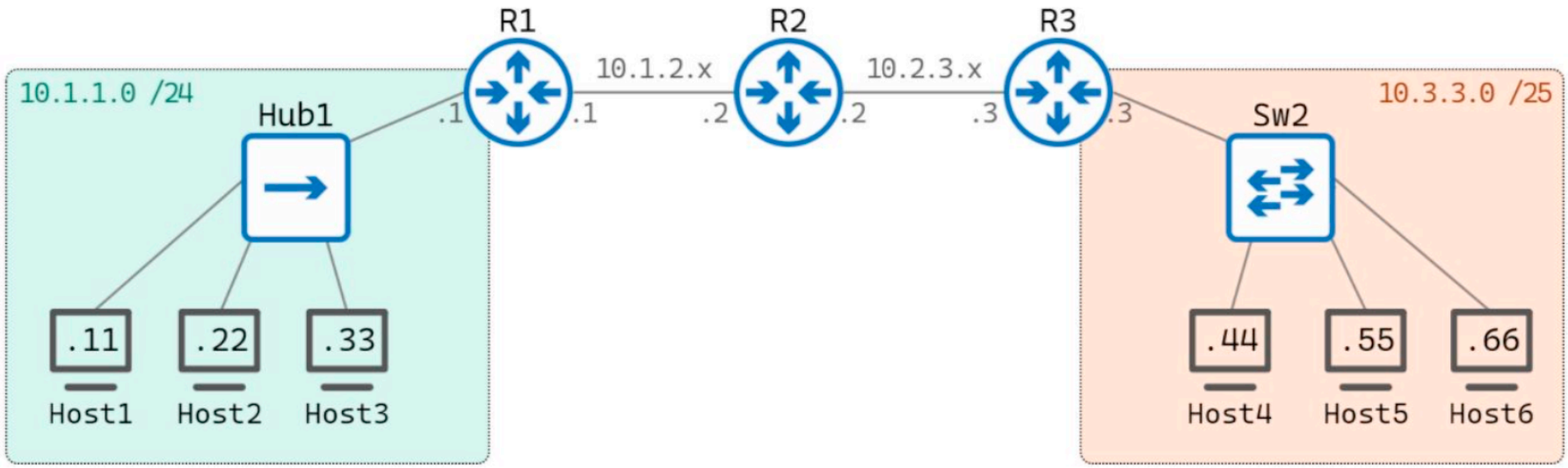
- Special blocks of addresses
  - 0.0.0.0/8
    - 0.0.0.0/0
      - "All addresses"
      - Covers every IP on Internet
      - Used in routing (specify default gateway)
      - Used in firewalls (specify default rules)
    - Is different from
    - 0.0.0.0/32 (same as 0.0.0.0)
      - Used on application-level as uninitialized IP address
      - "Unspecified address" (host without IP uses it in DHCPDISCOVER)
      - INADDR_ANY (configuring server to bind listening sockets)

Network addresses

- Special blocks of addresses
  - Link-local addresses
    - What happens when
      - Host is unable to get IP address from DHCP server
      - And, no IP address assigned manually
        - Host assigns itself an IP address from link-local addresses
    - 169.254.0.0/16
      - 169.254.0.0 - 169.254.255.255 (65,536 addresses)
    - Normally used when
      - No external, stateful address config mechanism (e.g., DHCP) exists
      - Or, primary configuration method has failed

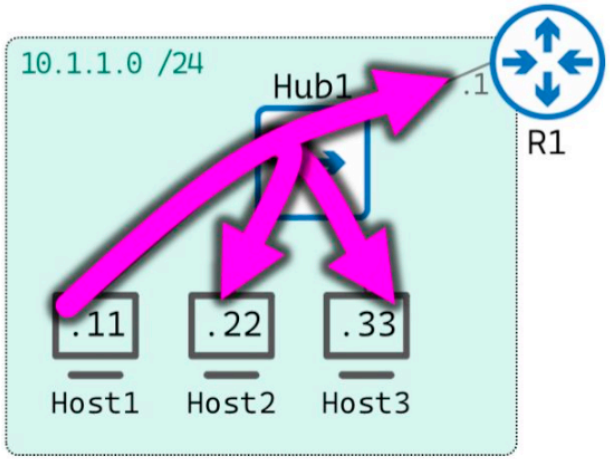
Network addresses

- Broadcast addresses
  - E.g., consider the following topology



Network addresses

- Broadcast addresses
  - Local broadcast (255.255.255.255)



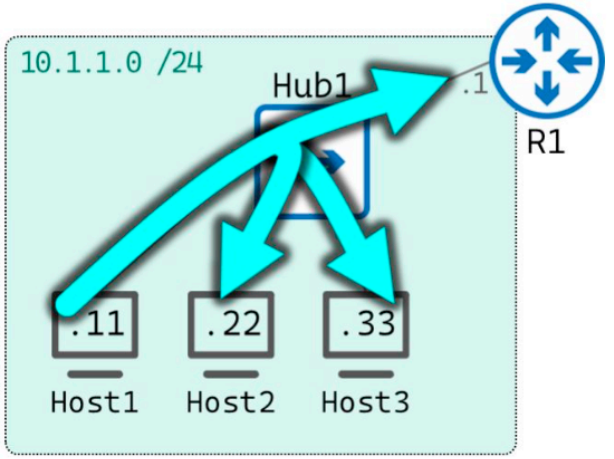

```
Host1# ping 255.255.255.255
PING 255.255.255.255 (255.255.255.255): 56 data bytes
64 bytes from 10.1.1.11: seq=0 ttl=64 time=0.044 ms
64 bytes from 10.1.1.33: seq=0 ttl=64 time=0.944 ms (DUP!)
64 bytes from 10.1.1.22: seq=0 ttl=64 time=1.108 ms (DUP!)
64 bytes from 10.1.1.1: seq=0 ttl=255 time=1.324 ms (DUP!)
^C
--- 255.255.255.255 ping statistics ---
1 packets transmitted, 1 packets received, 3 duplicates, 0% packet loss
round-trip min/avg/max = 0.044/0.855/1.324 ms
Host1#
```

Protocol	Source	Destination	Info
ICMP	10.1.1.11	255.255.255.255	Echo (ping) request id=0x6801, seq=0/0, ttl=64 (broadcast)
ICMP	10.1.1.33	10.1.1.11	Echo (ping) reply id=0x6801, seq=0/0, ttl=64
ICMP	10.1.1.22	10.1.1.11	Echo (ping) reply id=0x6801, seq=0/0, ttl=64
ICMP	10.1.1.1	10.1.1.11	Echo (ping) reply id=0x6801, seq=0/0, ttl=255

```
> Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
> Ethernet II, Src: ee:ee:ee:11:11:11, Dst: ff:ff:ff:ff:ff:ff
> Internet Protocol Version 4, Src: 10.1.1.11, Dst: 255.255.255.255
> Internet Control Message Protocol
```

Network addresses

- Broadcast addresses
  - Directed broadcast (10.1.1.255)



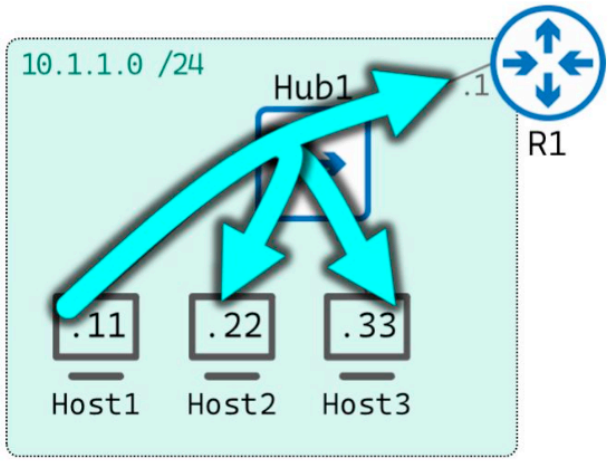

```
Host1# ping 10.1.1.255
PING 10.1.1.255 (10.1.1.255): 56 data bytes
64 bytes from 10.1.1.11: seq=0 ttl=64 time=0.046 ms
64 bytes from 10.1.1.33: seq=0 ttl=64 time=0.615 ms (DUP!)
64 bytes from 10.1.1.22: seq=0 ttl=64 time=0.835 ms (DUP!)
64 bytes from 10.1.1.1: seq=0 ttl=255 time=1.261 ms (DUP!)
^C
--- 10.1.1.255 ping statistics ---
1 packets transmitted, 1 packets received, 3 duplicates, 0% packet loss
round-trip min/avg/max = 0.046/0.689/1.261 ms
Host1#
```

Protocol	Source	Destination	Info
ICMP	10.1.1.11	10.1.1.255	Echo (ping) request id=0x6901, seq=0/0, ttl=64 (no response …
ICMP	10.1.1.33	10.1.1.11	Echo (ping) reply id=0x6901, seq=0/0, ttl=64
ICMP	10.1.1.22	10.1.1.11	Echo (ping) reply id=0x6901, seq=0/0, ttl=64
ICMP	10.1.1.1	10.1.1.11	Echo (ping) reply id=0x6901, seq=0/0, ttl=255

```
> Frame 7: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
> Ethernet II, Src: ee:ee:ee:11:11:11, Dst: ff:ff:ff:ff:ff:ff
> Internet Protocol Version 4, Src: 10.1.1.11, Dst: 10.1.1.255
> Internet Control Message Protocol
```

Network addresses

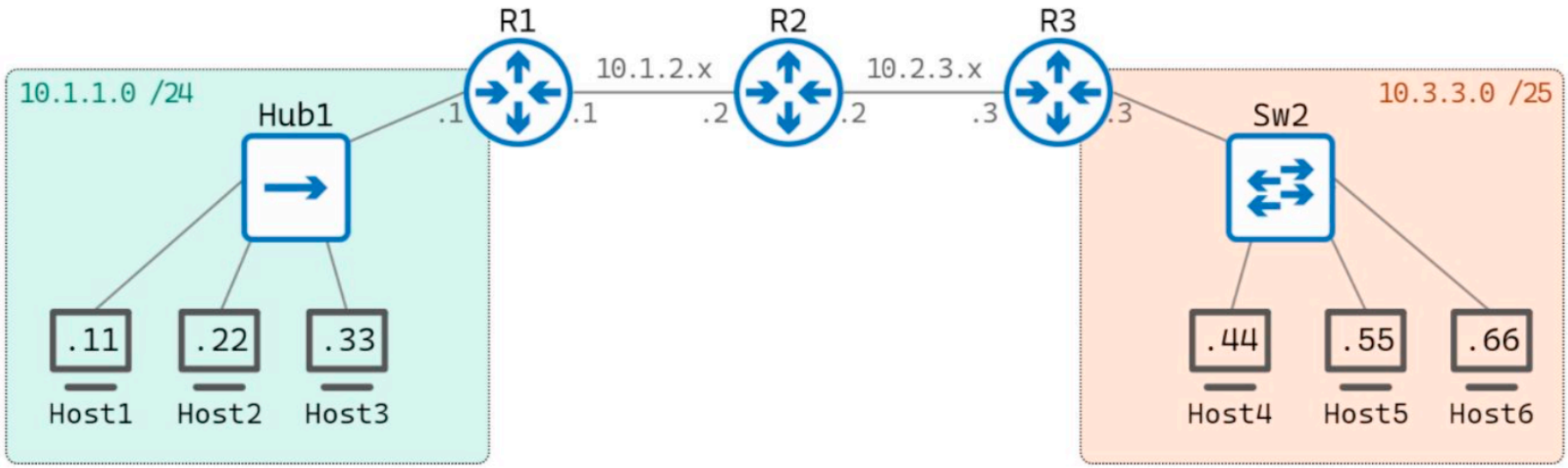
- Broadcast addresses
  - Directed broadcast (10.1.1.255)




```
> Ethernet II, Src: ee:ee:ee:11:11:11, Dst: ff:ff:ff:ff:ff:ff
> Internet Protocol Version 4, Src: 10.1.1.11, Dst: 10.1.1.255
v Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xb20d [correct]
    [Checksum Status: Good]
    Identifier (BE): 26881 (0x6901)
    Identifier (LE): 361 (0x0169)
    Sequence number (BE): 0 (0x0000)
    Sequence number (LE): 0 (0x0000)
  v [No response seen]
    > [Expert Info (Warning/Sequence): No response seen to ICMP request]
  > Data (56 bytes)
```

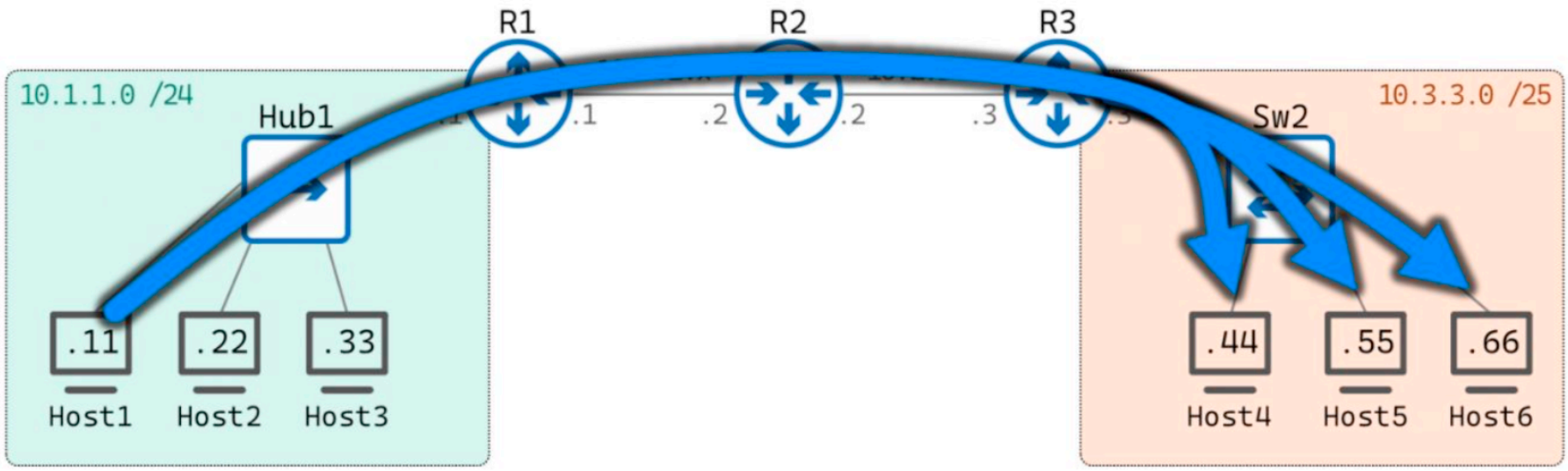
Network addresses

- Broadcast addresses
  - Directed broadcast to a foreign network



Network addresses

- Broadcast addresses
  - Directed broadcast to a foreign network



Network addresses

- Broadcast addresses
  - Directed broadcast to a foreign network

```
Host1# ping 10.3.3.127
PING 10.3.3.127 (10.3.3.127): 56 data bytes
64 bytes from 10.2.3.3: seq=0 ttl=253 time=1.171 ms
64 bytes from 10.3.3.66: seq=0 ttl=61 time=3.683 ms (DUP!)
64 bytes from 10.3.3.55: seq=0 ttl=61 time=7.340 ms (DUP!)
64 bytes from 10.3.3.44: seq=0 ttl=61 time=9.838 ms (DUP!)
^C
--- 10.3.3.127 ping statistics ---
1 packets transmitted, 1 packets received, 3 duplicates, 0% packet loss
round-trip min/avg/max = 1.171/5.508/9.838 ms
```

Protocol	Source	Destination	Info
ICMP	10.1.1.11	255.255.255.255	Echo (ping) request id=0x6b01, seq=0/0, ttl=61 (broadcast)
ICMP	10.3.3.66	10.1.1.11	Echo (ping) reply id=0x6b01, seq=0/0, ttl=64
ICMP	10.3.3.55	10.1.1.11	Echo (ping) reply id=0x6b01, seq=0/0, ttl=64
ICMP	10.3.3.44	10.1.1.11	Echo (ping) reply id=0x6b01, seq=0/0, ttl=64

```
> Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
> Ethernet II, Src: ee:ee:10:33:33:33, Dst: ff:ff:ff:ff:ff:ff
> Internet Protocol Version 4, Src: 10.1.1.11, Dst: 255.255.255.255
> Internet Control Message Protocol
```

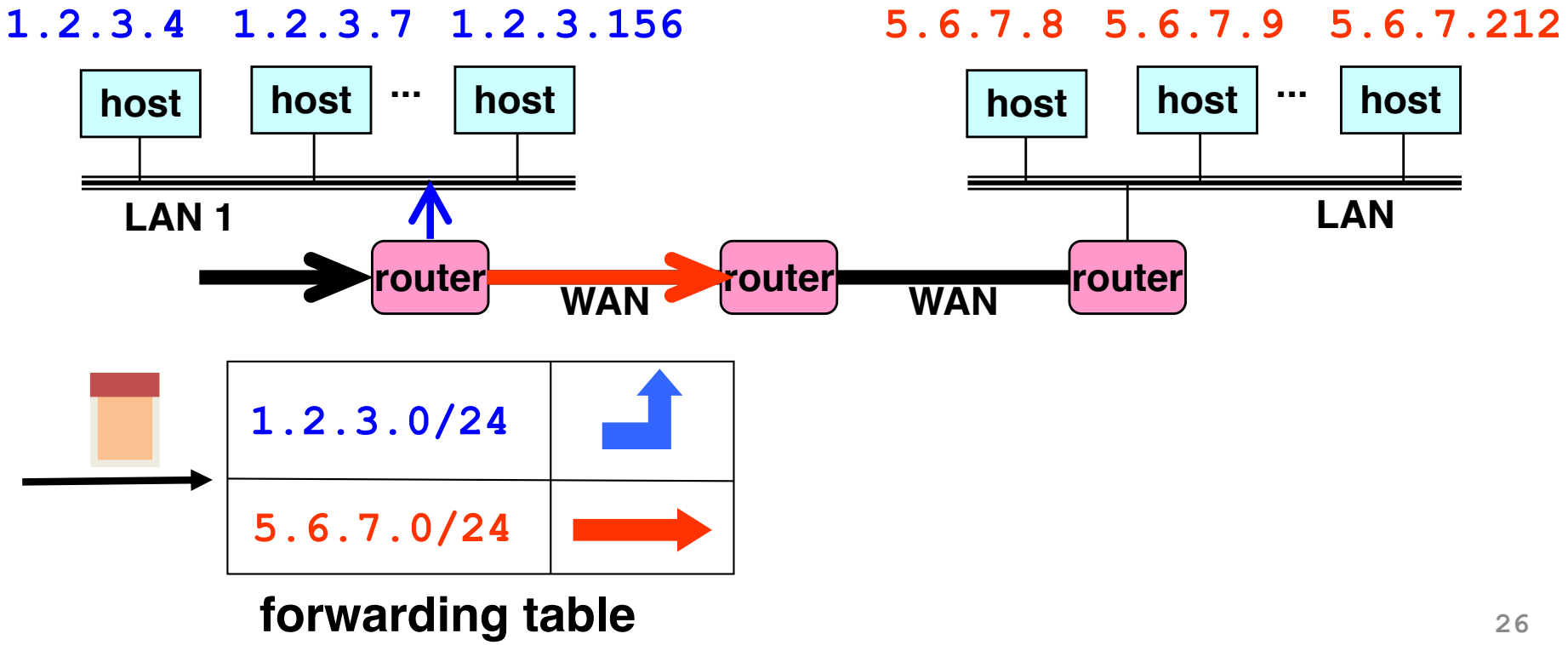
Packet Forwarding

Hop-by-Hop Packet Forwarding

- Each router has a forwarding table
  - Maps destination address to outgoing interface
- Upon receiving a packet
  - Inspect the destination address in the header
  - Index into the table
  - Determine the outgoing interface
  - Forward the packet out that interface
- Then, the next router in the path repeats

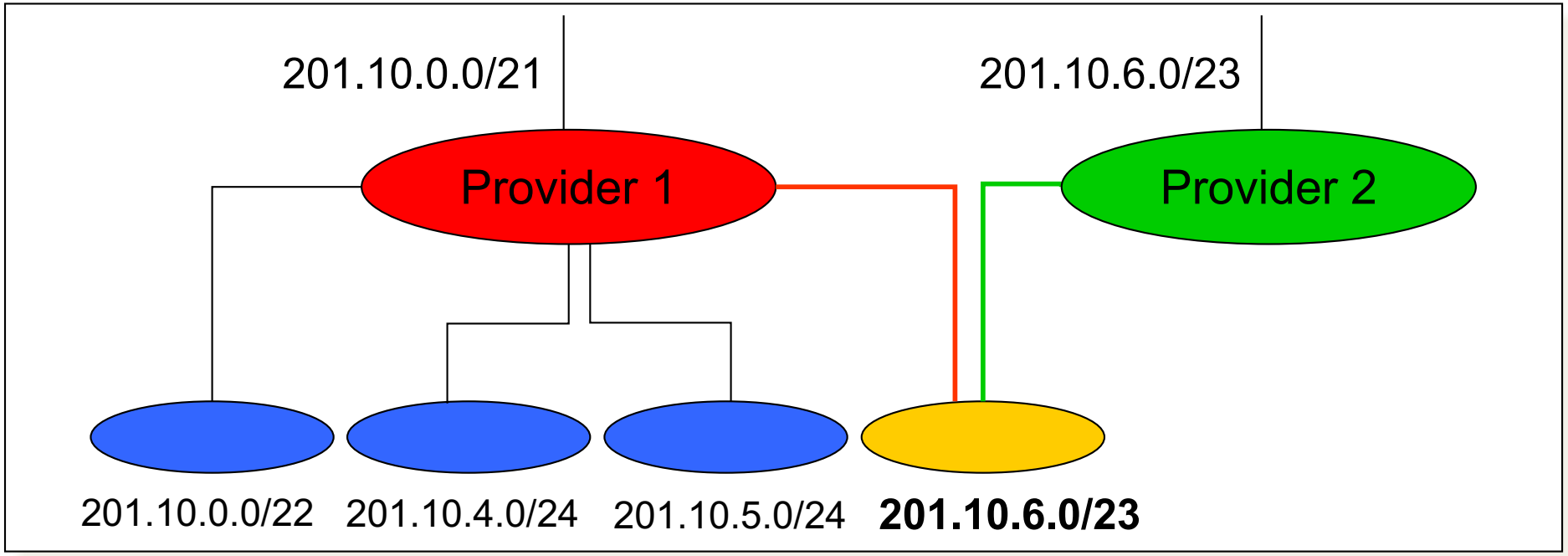
Separate Forwarding Entry Per Prefix

- Prefix-based forwarding
  - Map the destination address to matching prefix
  - Forward to the outgoing interface



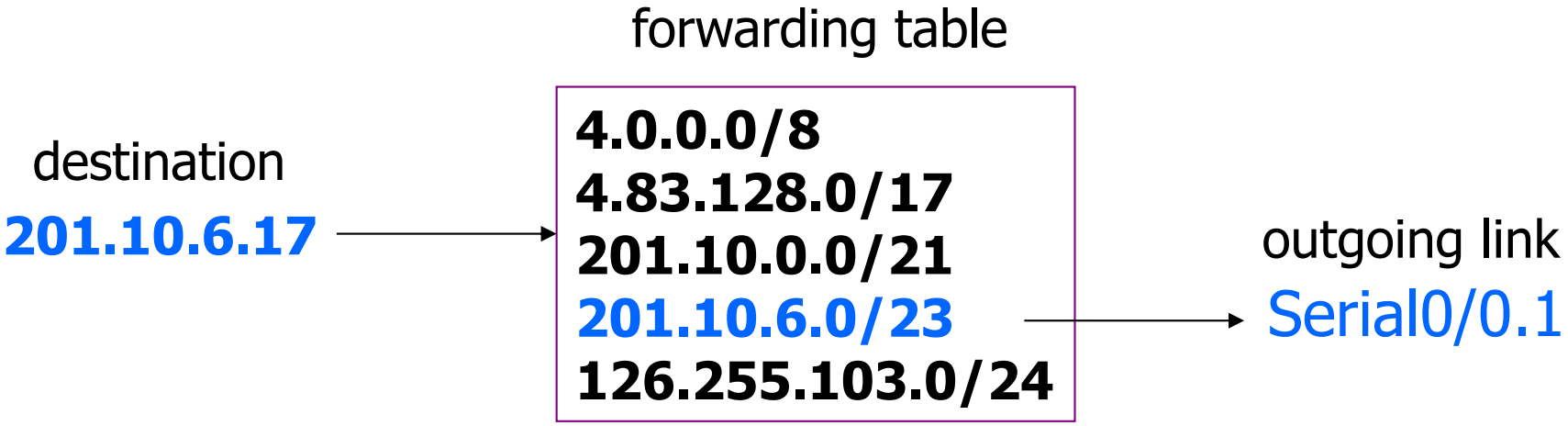
CIDR Makes Packet Forwarding Harder

- Forwarding table may have many matches
  - E.g., entries for 201.10.0.0/21 and 201.10.6.0/23
  - The IP address 201.10.6.17 would match both!



Longest Prefix Match Forwarding

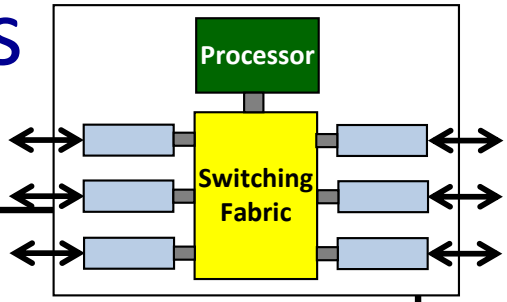
- Destination-based forwarding
  - Packet has a destination address
  - Router identifies longest-matching prefix
  - Cute algorithmic problem: very fast lookups



Creating a Forwarding Table

- Entries can be statically configured
  - E.g., “map 12.34.158.0/24 to Serial0/0.1”
- But, this doesn’t adapt
  - To failures
  - To new equipment
  - To the need to balance load
- That is where the *control plane* comes in
  - Routing protocols

Data, Control, & Management Planes



	Data	Control	Management
Time-scale	Packet (ns)	Event (10 ms to sec)	Human (min to hours)
Tasks	Forwarding, buffering, filtering, scheduling	Routing, signaling	Analysis, configuration
Location	Line-card hardware	Router software	Humans or scripts

Q's: MAC vs. IP Addressing

- Hierarchically allocated

  Y) MAC M) IP C) Both A) Neither

- Organized topologically

  Y) MAC M) IP C) Both A) Neither

- Forwarding via exact match on address

  Y) MAC M) IP C) Both A) Neither

- Automatically calculate forwarding by observing data

  Y) Ethernet switches M) IP routers C) Both A) Neither

- Per connection state in the network

  Y) MAC M) IP C) Both A) Neither

- Per host state in the network

  Y) MAC M) IP C) Both A) Neither

Q's: MAC vs. IP Addressing

- Hierarchically allocated

  Y) MAC M) IP **C) Both** A) Neither

- Organized topologically

  Y) MAC **M) IP** C) Both A) Neither

- Forwarding via exact match on address

  Y) MAC M) IP C) Both A) Neither

- Automatically calculate forwarding by observing data

  Y) Ethernet switches M) IP routers C) Both A) Neither

- Per connection state in the network

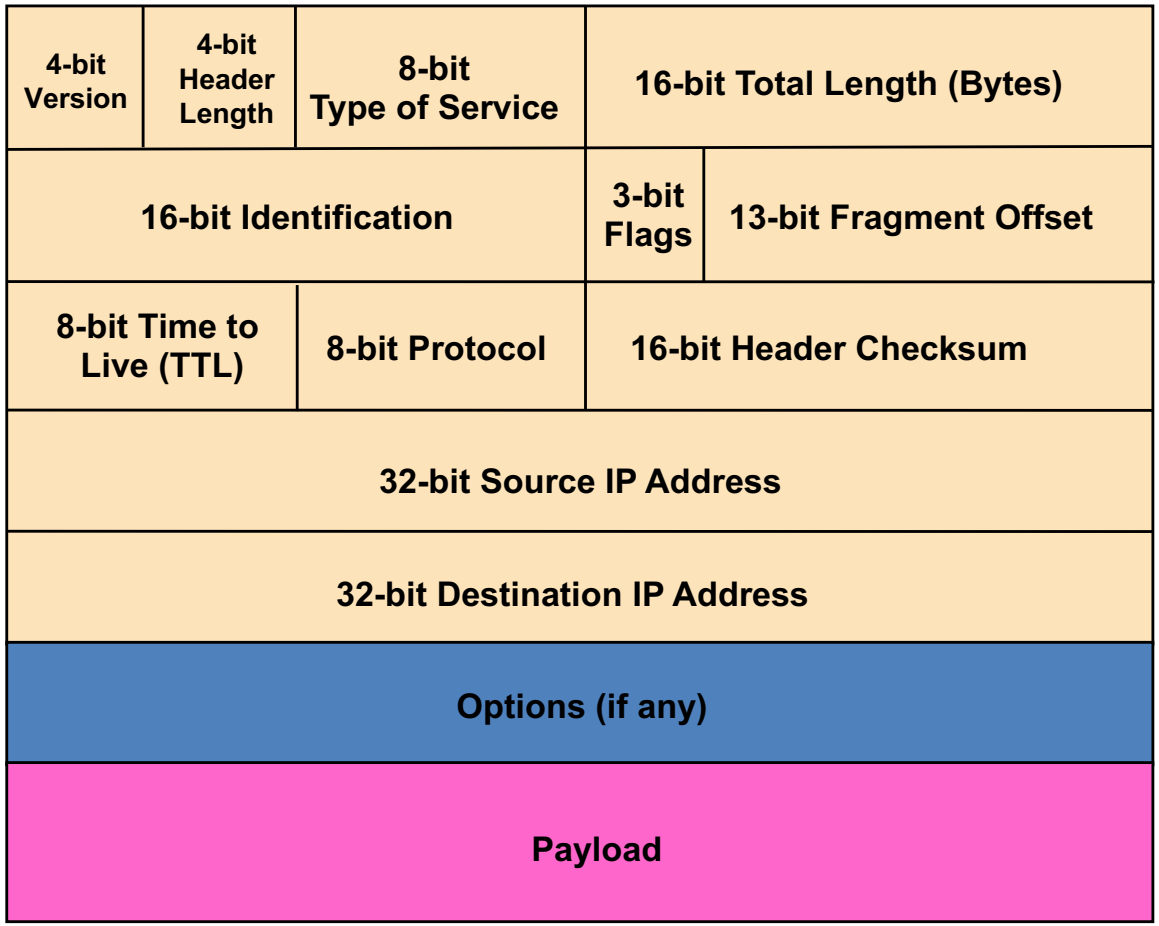
  Y) MAC M) IP C) Both **A) Neither**

- Per host state in the network

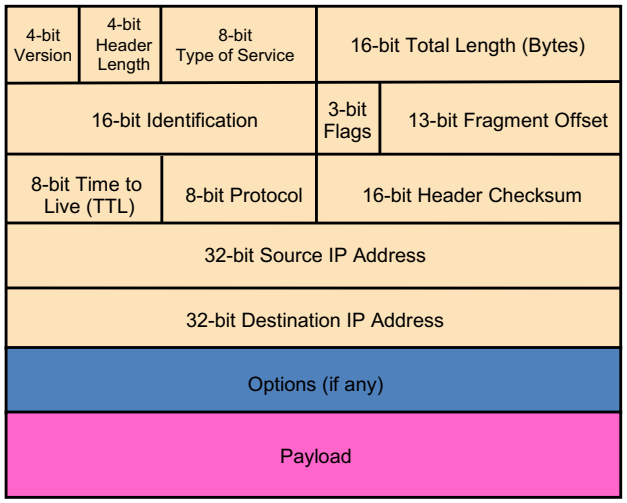
  Y) MAC M) IP C) Both A) Neither

IP Packet Format

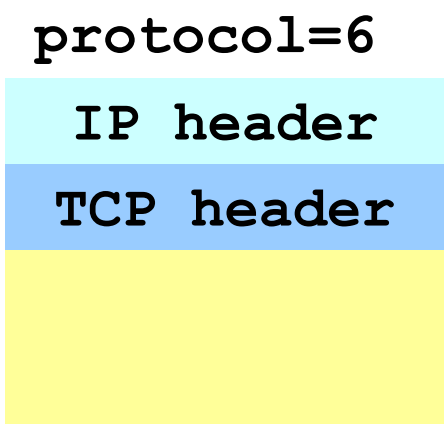
IP Packet Structure

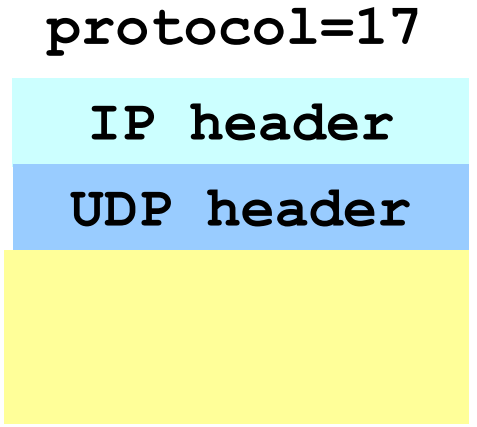


IP Header: Transport Protocol

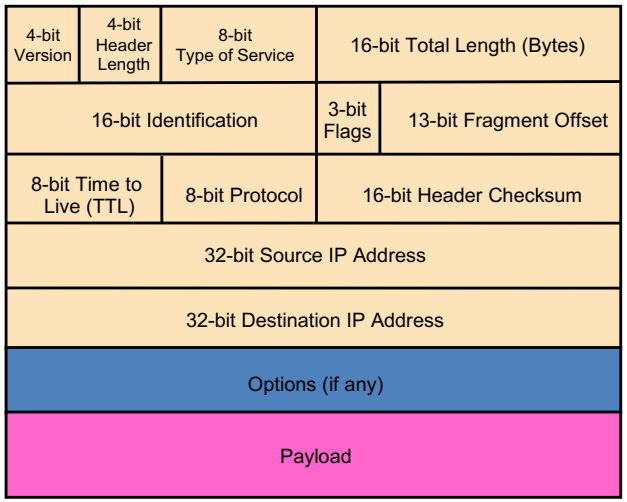


- Protocol (8 bits)
  - Identifies the higher-level protocol
    - E.g., "6" for the Transmission Control Protocol (TCP)
    - E.g., "17" for the User Datagram Protocol (UDP)
  - Important for demultiplexing at receiving host
    - Indicates what kind of header to expect next



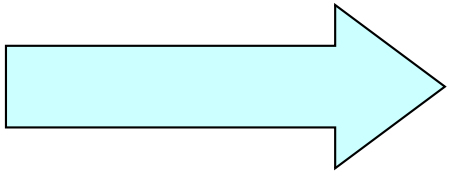


IP Header: Header Checksum



- Checksum (16 bits)
  - Sum of all 16-bit words in the header
  - If header bits are corrupted, checksum won't match
  - Receiving discards corrupted packets

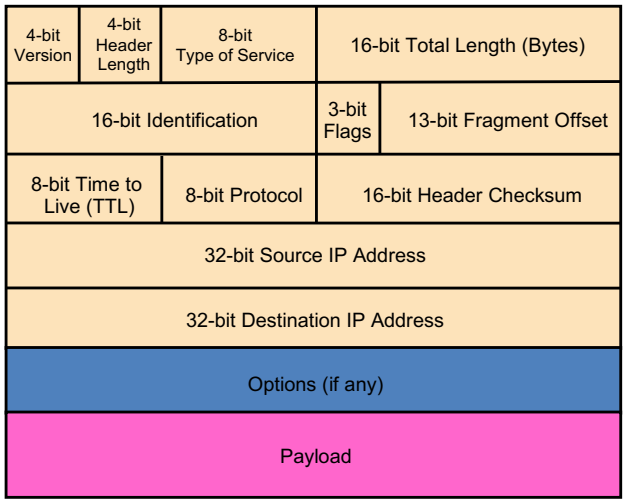
```
  134
+ 212
-----
= 346
```



Mismatch!

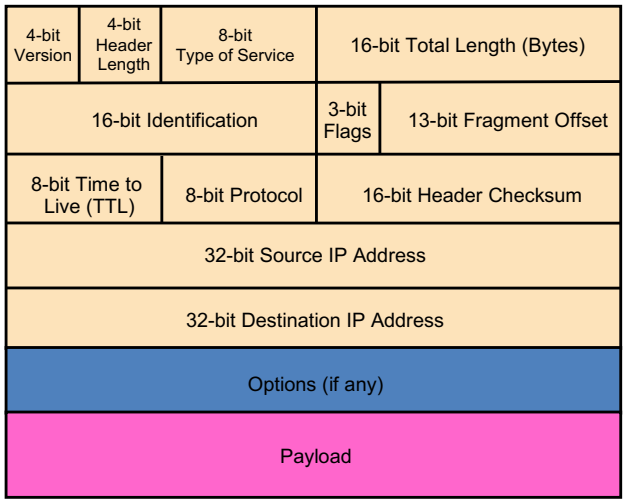
```
  134
+ 216
-----
= 350
```

IP Header: Version, Length, ToS



- Version number (4 bits)
  - Necessary to know what other fields to expect
  - Typically “4” (for IPv4), and sometimes “6” (for IPv6)
- Header length (4 bits)
  - Number of 32-bit words in the header
  - Typically “5” (for a 20-byte IPv4 header)
  - Can be more when “IP options” are used
- Type-of-Service (8 bits)
  - Allow different packets to be treated differently
  - Low delay for audio, high bandwidth for bulk transfer

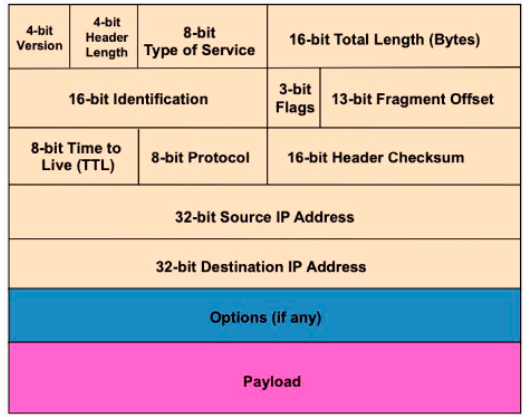
IP Header: Length, Fragments, TTL



- Total length (16 bits)
  - Number of bytes in the packet
  - Max size is 63,535 bytes (2^{16} -1)
  - … though most links impose smaller limits
- Time-To-Live (8 bits)
  - Used to identify packets stuck in forwarding loops
  - … and eventually discard them from the network
- Fragmentation information (32 bits)
  - Supports dividing a large IP packet into fragments
  - … in case a link cannot handle a large IP packet

IP packet format

- IP header: Fragmentation
  - E.g.,
    - 576 B MTU; 20 B header; 1420 B packet data



Fragment	Identification	Flags	Fragment Offset	Total Length
One	X=987	001	0	572
Two	X=987	001	69	572
Three	X=987	000	138	336

Conclusion

- Best-effort global packet delivery
  - Simple end-to-end abstraction
  - Enables higher-level abstractions on top
  - Doesn't rely on much from the links below
- IP addressing and forwarding
  - Hierarchy for scalability and decentralized control
  - Allocation of IP prefixes
  - Longest prefix match forwarding
- Next time: switches & routers