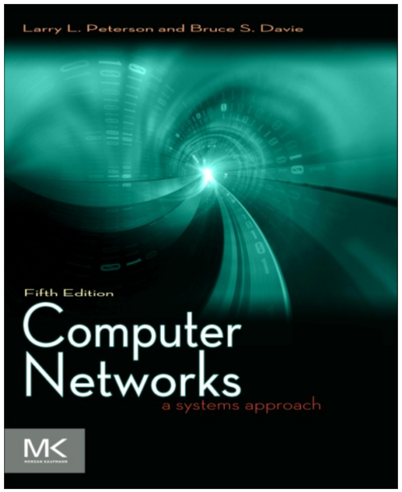


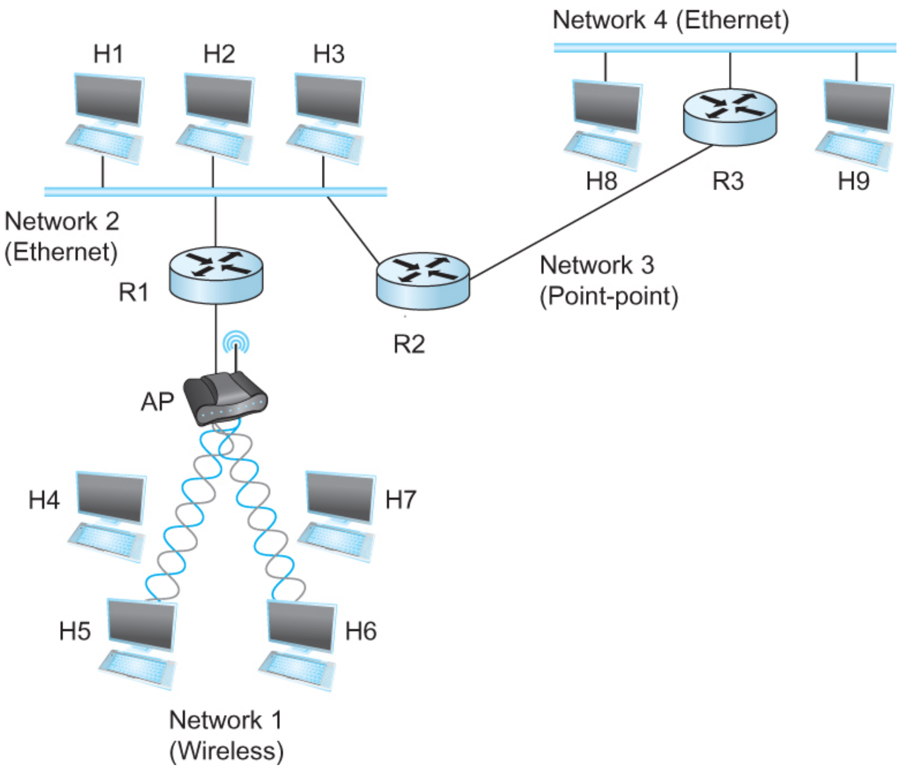
Internetworking: Addressing and Scalability

Errata from last week

- It is impossible for two hosts on the same Ethernet to transmit continuously at 10Mbps because they share the same transmission medium
- Every host on a switched network has its own link to the switch
  - So it may be entirely possible for many hosts to transmit at the full link speed (bandwidth) provided that the switch is designed with enough aggregate capacity
- Gigabit Ethernet is actually switched so not really in the CSMA/CD family

Internetworking

- An arbitrary collection of networks interconnected to provide some sort of host-host to packet delivery service

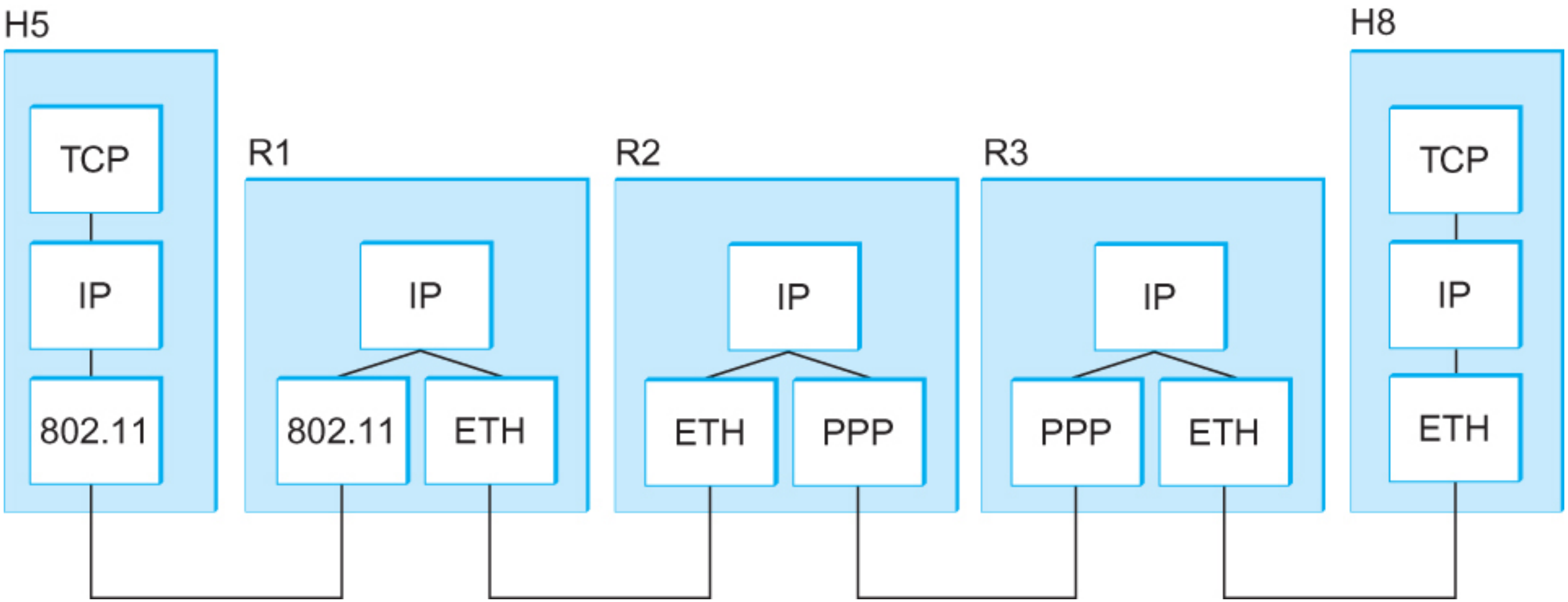


Datagram architecture

- Each host has a globally unique address
  - Every packet contains enough information to enable any switch to decide how to get it to destination
  - So, every packet contains the complete destination address
- Each packet is forwarded independently of previous packets – no hard forwarding state in routers
- Best-effort delivery means packets may be:
  - delayed or dropped
  - take different routes
  - delivered out of order, delivered multiple times

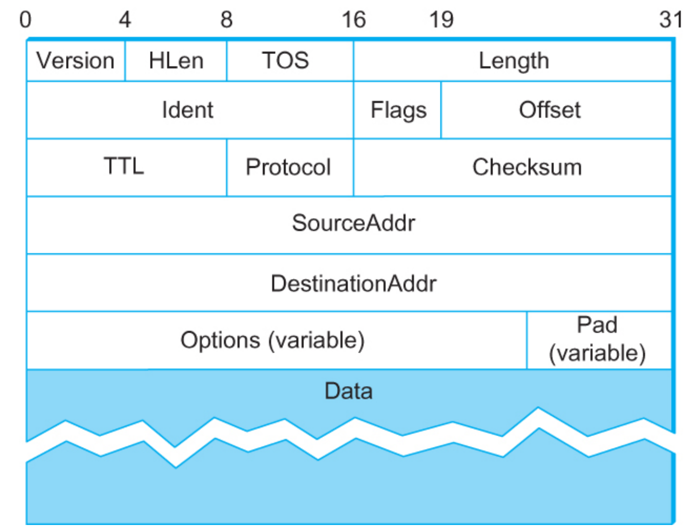
Internet Protocol (IP)

- Runs on all nodes, defines infrastructure that allows networks to function as a single logical internetwork
- IP Provides a way to identify, reach, all hosts in the network



IPv4 Packet Format

- Version (4): IPv4 or IPv6
- Hlen (4): number of 32-bit words in header
- TOS (8): type of service (not widely used)
- Length (16): number of **bytes** in this datagram
- Ident (16): used by fragmentation
- Flags/Offset (16): used by fragmentation
- TTL (8): number of hops this datagram has traveled
- Protocol (8): demux key (TCP=6, UDP=17)
- Checksum (16): **of the header only**
- DestAddr & SrcAddr (32 bits each IPv4)



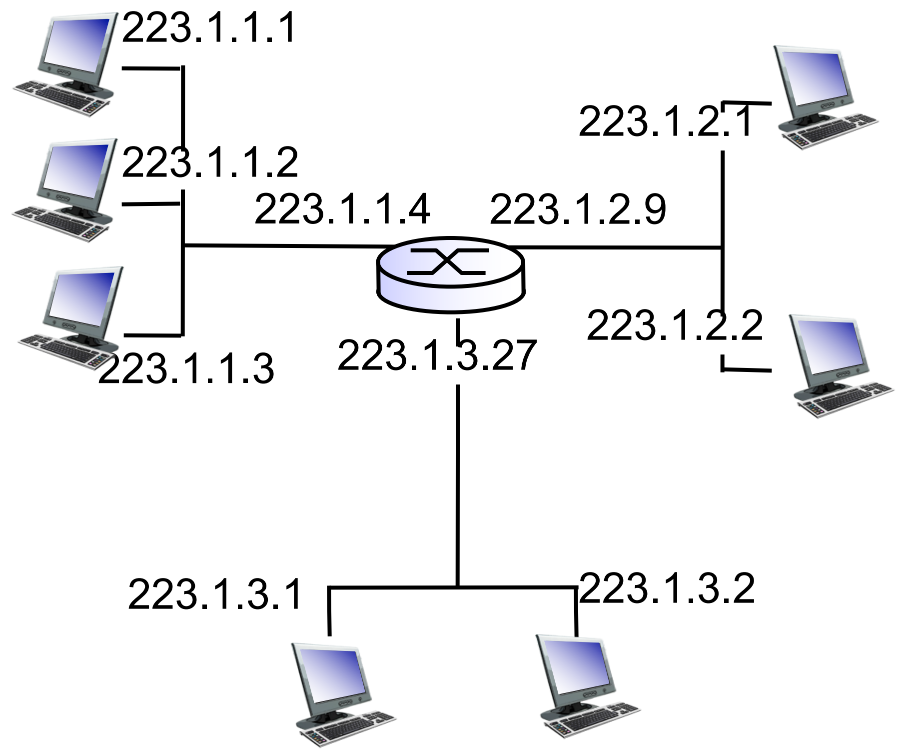
IP Datagram Forwarding

- every datagram contains destination's address
  - if directly connected to destination network, then forward to host
  - if not directly connected to destination network, then forward to some router
- forwarding table maps network number into next hop
- each host has a default router
- each router maintains a forwarding table

NetworkNum	NextHop
1	R1
2	Interface 1
3	Interface 0
4	R3

IP forwarding

- *IP address:* 32-bit identifier for host, router *interface*
- *interface:* connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one interface
  - IP addresses associated with each interface

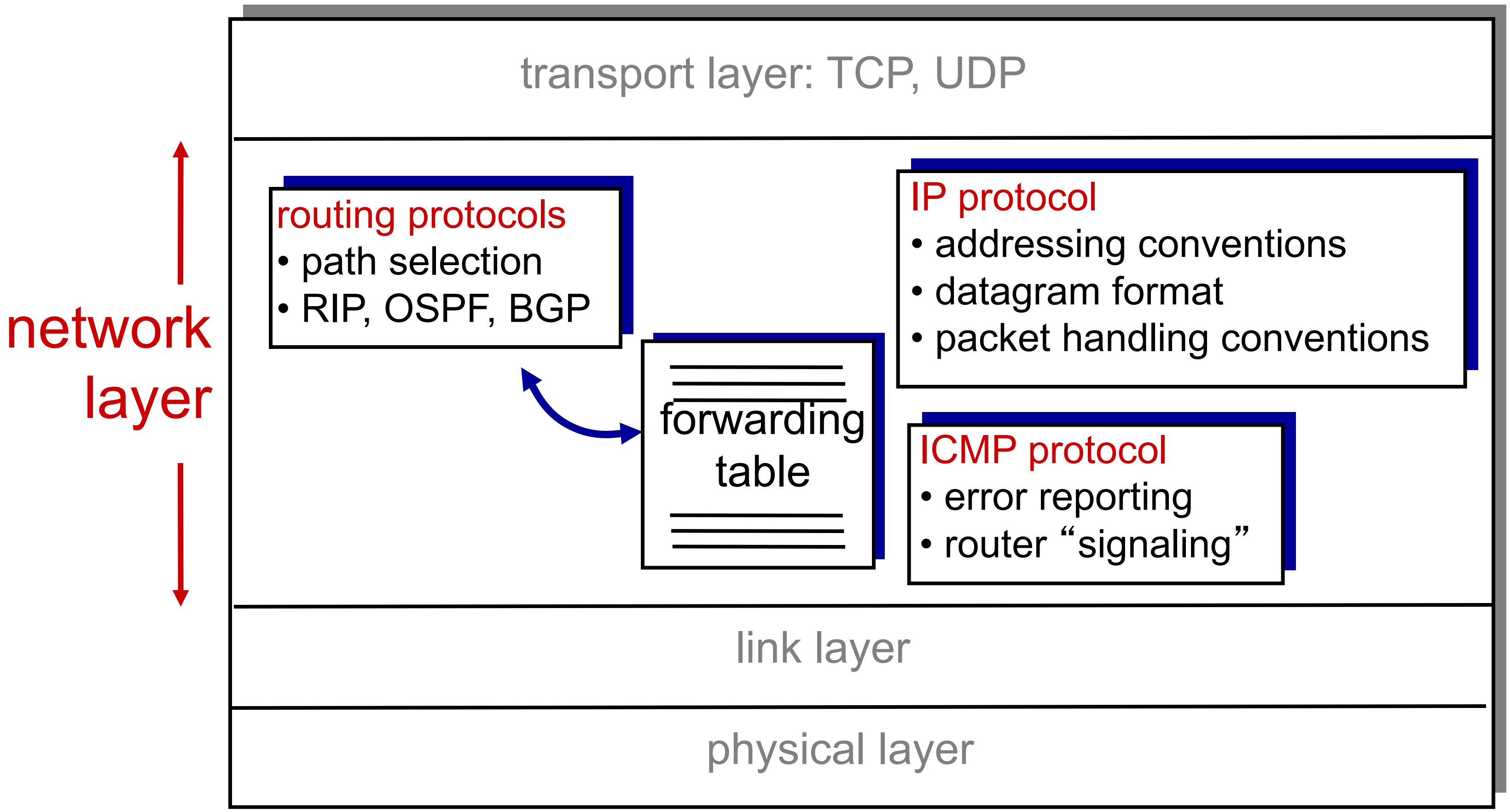


223.1.1.1 = 11011111 00000001 00000001 00000001

$\underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}$

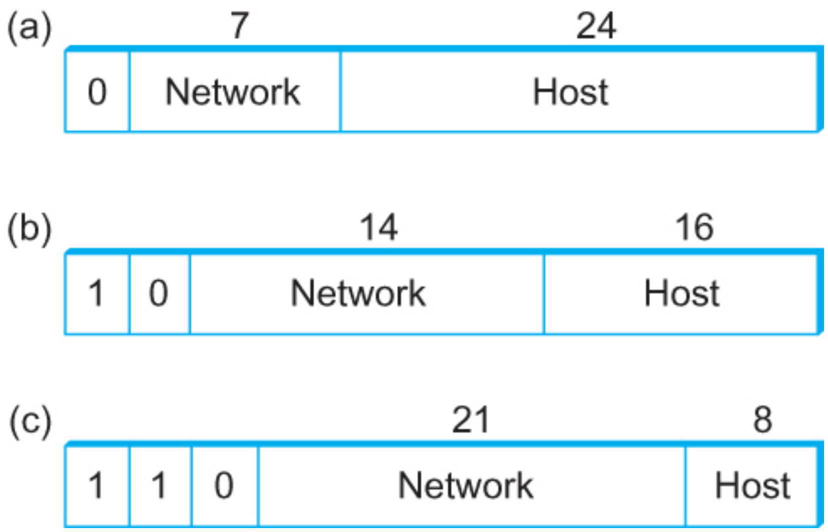
Scaling Challenges: Addressing and Routing

host, router network layer functions:



Global IPv4 Addresses

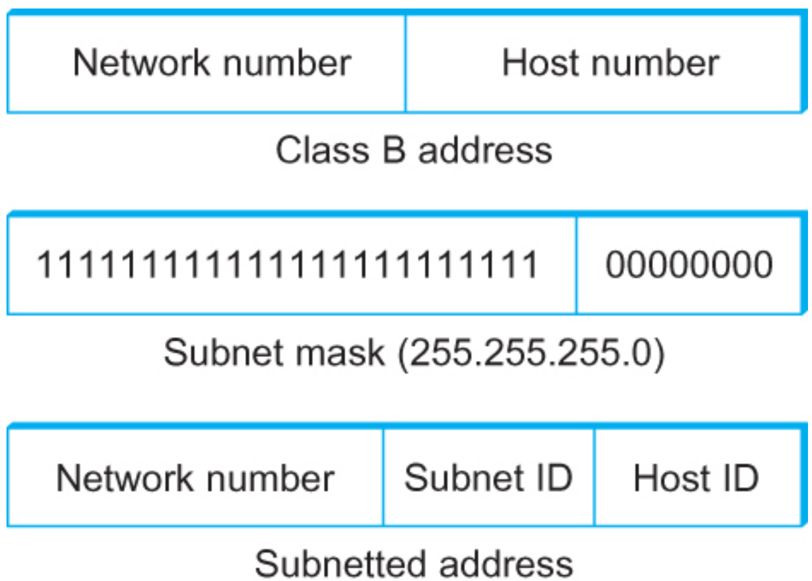
- Properties
  - globally unique
  - hierarchical: network + host – **Class based addressing**
  - 4 Billion IP addresses, 1/2 A type, 1/4 B type, and 1/8 C type
- Format



- Dot notation
  - 10.3.2.4
  - 128.96.33.81
  - 192.12.69.77

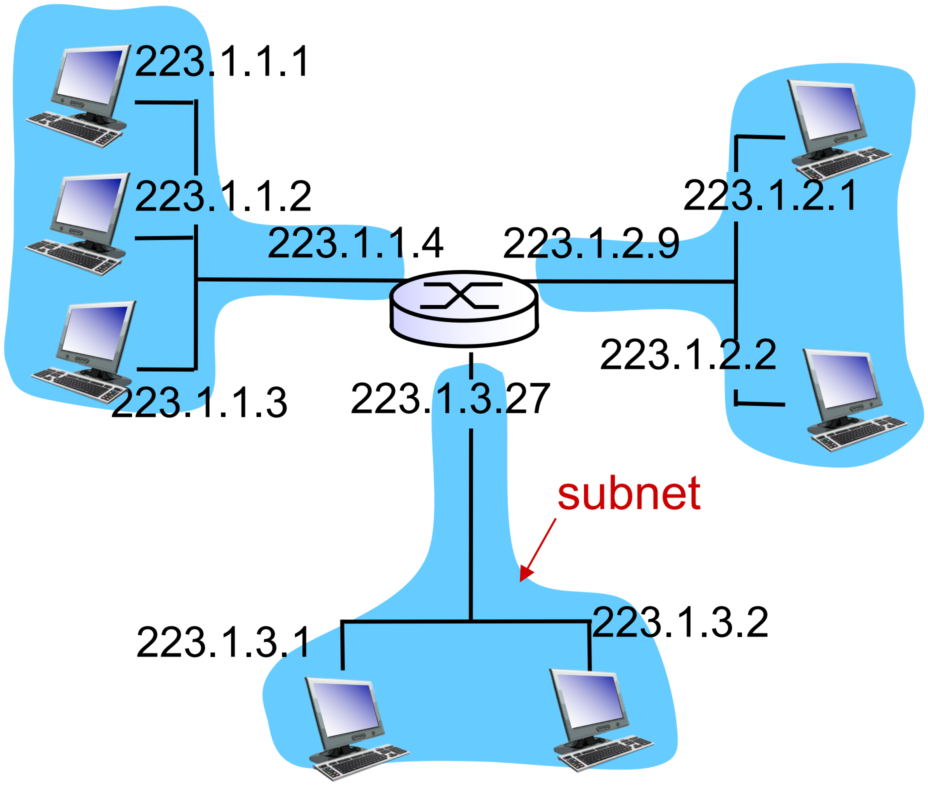
Subnetting for internal scalability

- Add another level to Intranet address/routing hierarchy: *subnet*
- *Subnet masks* define variable partition of host part of class A and B addresses since spaces are so BIG
- Subnets visible only within site – NOT rest of Internet
- Make internal network more efficient



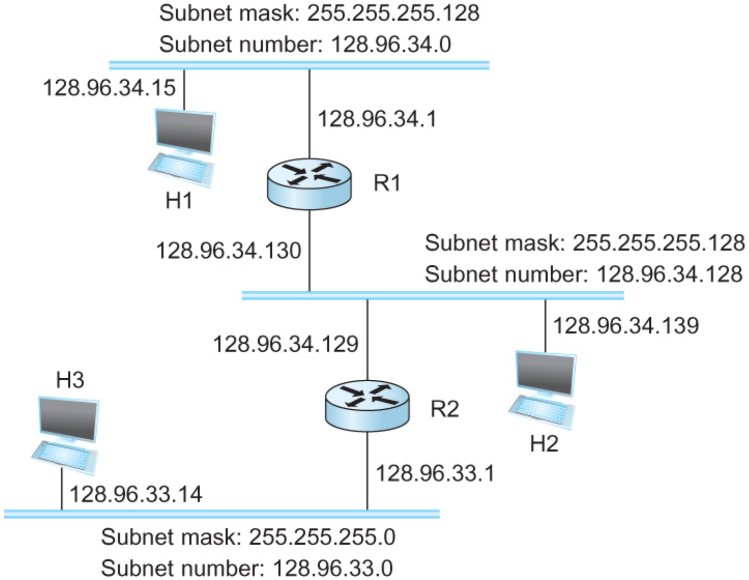
Subnet definition

- IP address:
  - subnet part - high order bits
  - host part - low order bits
- *what's a subnet ?*
  - device interfaces with same subnet part of IP address
  - can physically reach each other *without intervening router*



network consisting of 3 subnets

Subnetting example



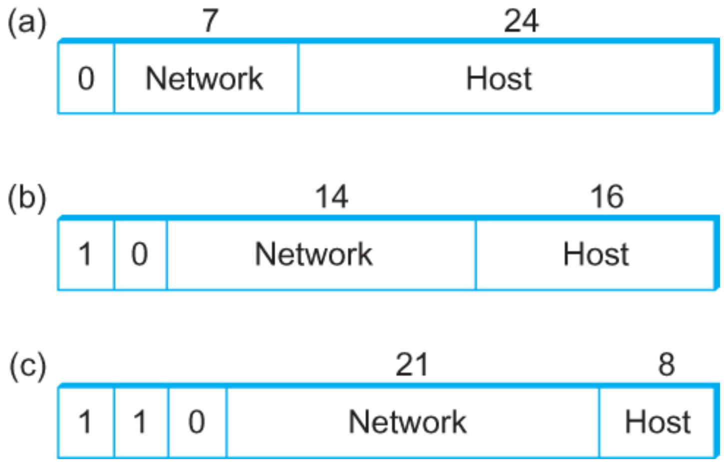
SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	Interface 0
128.96.34.128	255.255.255.128	Interface 1
128.96.33.0	255.255.255.0	R2

Forwarding Algorithm

```
D = destination IP address
for each entry < SubnetNum, SubnetMask, NextHop>
    D1 = SubnetMask & D
    if D1 = SubnetNum
        if NextHop is an interface
            deliver datagram directly to destination
        else
            deliver datagram to NextHop (a router)
```

Internet Addressing scaling issues

- Fixed bit-size address classes: A, B, C
- Class B address exhaustion concern began late ’80s



Addressing Routing Scaling Tradeoff

- Simple approach: Allocate multiple Class C addresses instead of Class B
- Overhead: Every router needs multiple entries to reach all hosts in a remote network that has multiple Class C's even when path to the destinations is the same
- **Classic tradeoff – Address space utilization vs. Routing table space**

CIDR balanced tradeoff

- Classless Inter-domain level routing: CIDR (1993)
- CIDR tries to balance the desire to minimize the number of routes that a router needs to know against the need to hand out addresses efficiently.
- CIDR uses aggregate routes
  - Uses a single entry in the forwarding table to tell the router how to reach a lot of different networks
  - Breaks the rigid boundaries between address classes
  - Variable #bits per aggregated range of addresses

Classless Address block management

- AS with 16 class C network numbers--Instead of handing out 16 addresses at random, hand out a block of **contiguous class C addresses**
  - E.g., class C network numbers from 192.4.16 through 192.4.31
  - top 20 bits of all the addresses in this range are the same (11000000 00000100 0001)
  - Implicitly created 20-bit network number (which is in between class B network number and class C number)
- Requires handing out blocks of class C addresses that share common prefix
- Prefix Convention: /X after prefix, prefix length in bits
  - 20-bit prefix for 192.4.16 through 192.4.31: 192.4.16/20
  - single class C network number, 24 bits long: 192.4.16/24

IP Forwarding w/ Longest match

- Router tables may have prefixes that overlap
  - Some addresses may match more than one prefix
  - both 171.69 (a 16 bit prefix) and 171.69.10 (a 24 bit prefix) in the forwarding table of a single router
  - packet destined to 171.69.10.5 clearly matches both prefixes.
- The rule is based on the principle of “longest match”
    - 171.69.10 in this case
- A packet destined to 171.69.20.5 would match to 171.69 and not 171.69.10

Longest prefix matching

longest prefix matching

when looking for forwarding table entry for given destination address, use longest address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

examples:

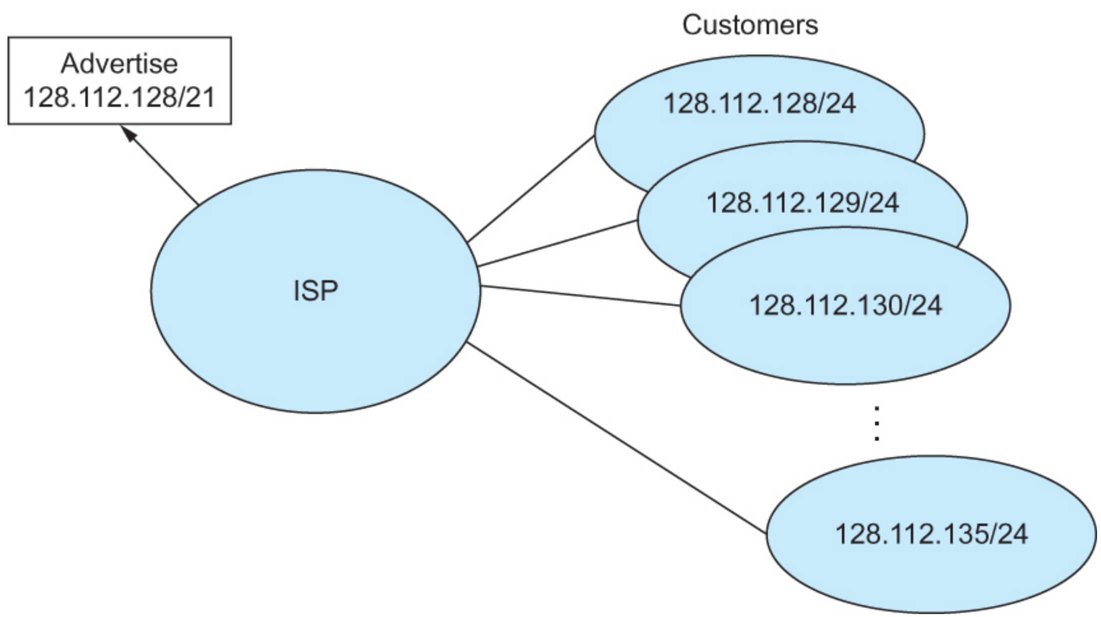
DA: 11001000 00010111 00010110 10100001 which interface?

DA: 11001000 00010111 00011000 10101010 which interface?

Classless Addressing

- network number may be of any length
- Represent network number with a single pair

  `<length, value>`
- All routers must understand CIDR addressing



IPv6: motivation

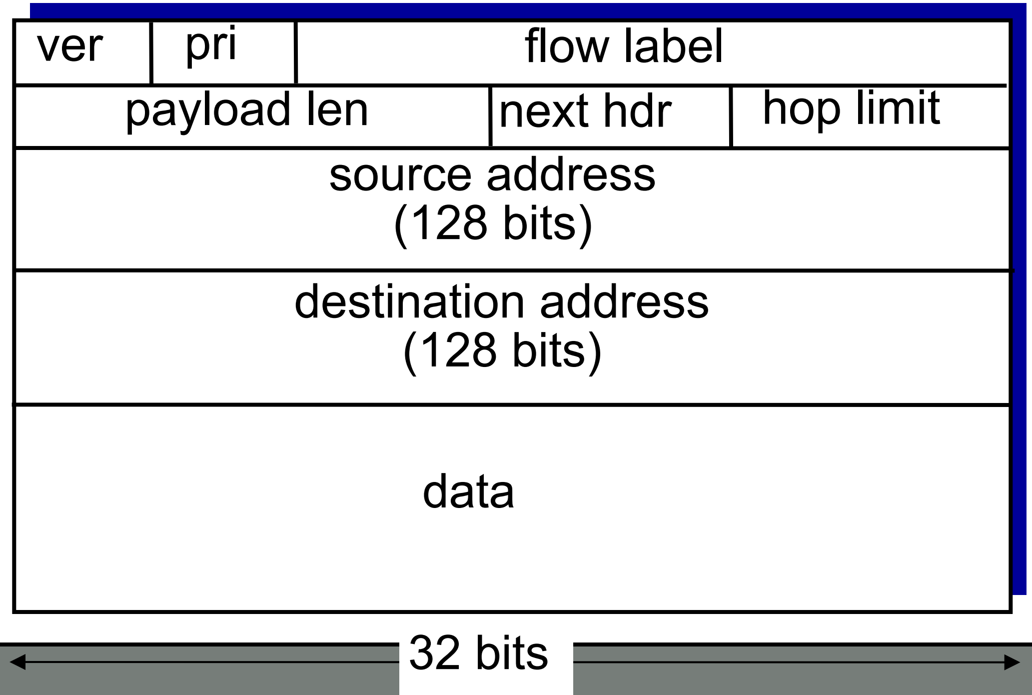
- initial motivation: 32-bit address space soon to be completely allocated.
- additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS

IPv6 datagram format:

  - fixed-length 40 byte header
  - 128 bit addresses
  - no fragmentation allowed

IPv6 datagram format

priority: identify priority among datagrams in flow
flow Label: identify datagrams in same “flow.”
(concept of“flow” not well defined).
next header: identify upper layer protocol for data

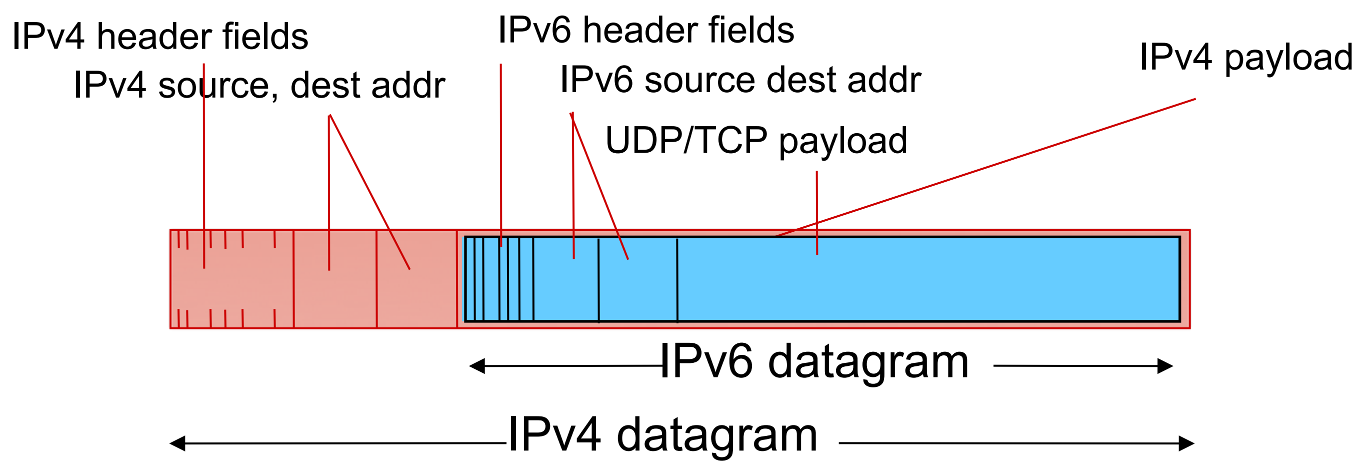


Other changes from IPv4

- *checksum*: removed entirely to reduce processing time at each hop
- *options:* allowed, but outside of header, indicated by “Next Header” field
- *ICMPv6:* new version of ICMP
  - additional message types, e.g. “Packet Too Big”
  - multicast group management functions

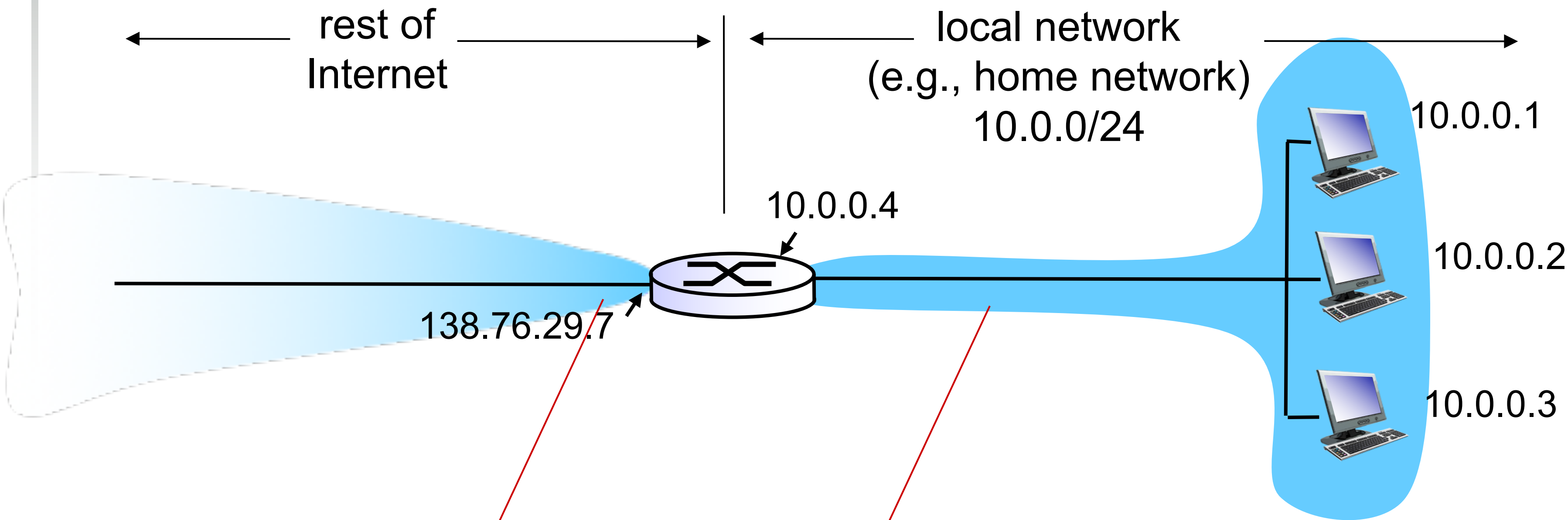
Transition from IPv4 to IPv6

- Impractical to upgrade all routers simultaneously:
  - no flag day
  - Incremental deployment w/mixed IPv4 and IPv6 internet
- *tunneling*: IPv6 datagram carried as *payload* in IPv4 datagram among IPv4 routers



Earlier work around: address translation

Network Address Translation: NAT

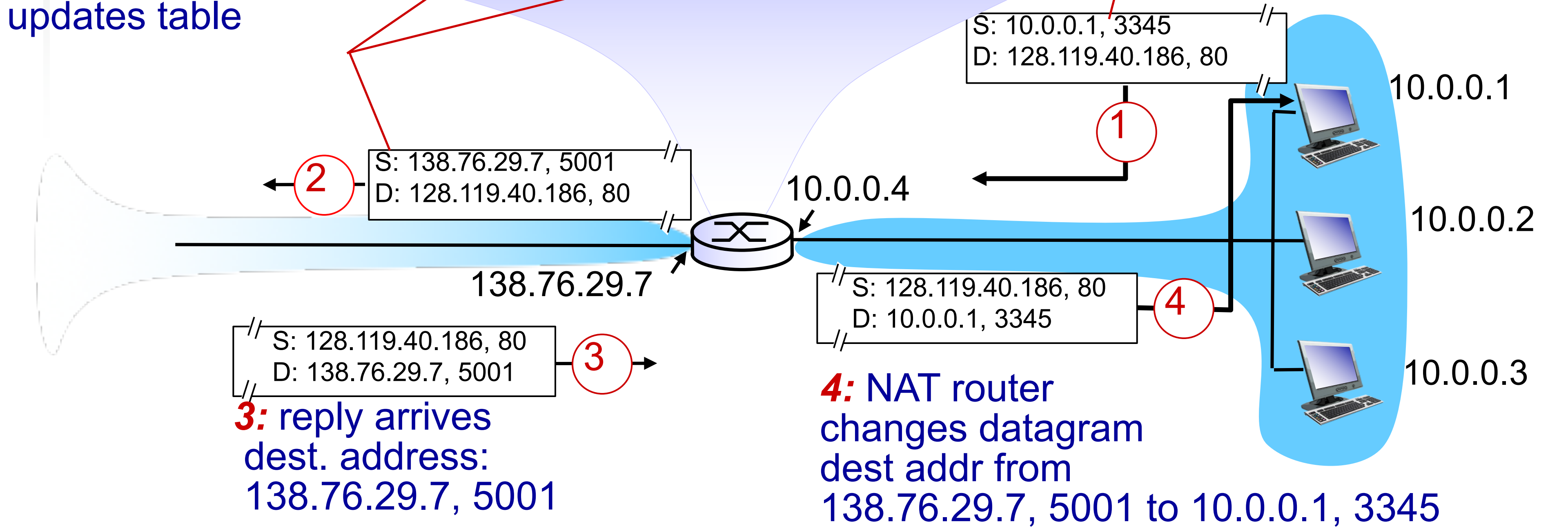


all datagrams *leaving* local network have same single source NAT IP address: 138.76.29.7,different source port numbers

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT: network address translation

NAT translation table	
WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
……	……

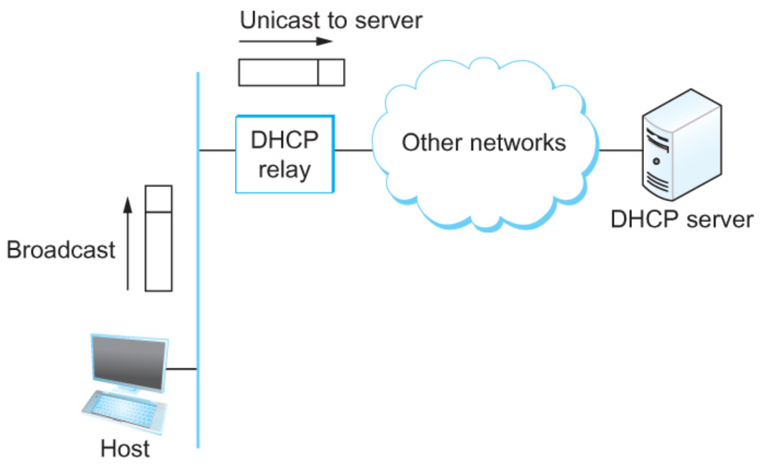


Host Configurations

- Ethernet addresses configured into network adapter by manufacturer -- unique
- IP addresses must be unique on given internetwork AND reflect structure of the internetwork for routing
- Automated Configuration Process to get IP address: Dynamic Host Configuration Protocol (DHCP)

Dynamic Host Configuration Protocol (DHCP)

- DHCP server provides configuration information to hosts
- At least one DHCP server for an administrative domain
- DHCP server maintains a pool of available addresses
- Newly booted/attached host sends DHCPDISCOVER message to special IP address (255.255.255.255)
- DHCP relay agent unicasts message to DHCP server; waits for response



Internet Control Message Protocol (ICMP)

- Defines a collection of error messages that are sent back to the source host whenever a router or host is unable to process an IP datagram successfully
  - Destination host unreachable due to link /node failure
  - Reassembly process failed
  - TTL had reached 0 (so datagrams don't cycle forever)
  - IP header checksum failed

- ICMP-Redirect
  - From router to a source host
  - With a better route information

Address Translation Protocol (ARP)

- Map IP addresses into physical addresses
  - destination host
  - next hop router
- ARP (Address Resolution Protocol)
  - table of IP to physical address bindings
  - broadcast request if IP address not in table
  - target machine responds with its physical address
  - table entries are discarded if not refreshed

ARP Packet Format

<table>
<tr><td>0</td><td>8</td><td>16</td><td>31</td></tr>
<tr><td colspan="2">Hardware type = 1</td><td colspan="2">ProtocolType = 0x0800</td></tr>
<tr><td>HLen = 48</td><td>PLen = 32</td><td colspan="2">Operation</td></tr>
<tr><td colspan="4">SourceHardwareAddr (bytes 0–3)</td></tr>
<tr><td colspan="2">SourceHardwareAddr (bytes 4–5)</td><td colspan="2">SourceProtocolAddr (bytes 0–1)</td></tr>
<tr><td colspan="2">SourceProtocolAddr (bytes 2–3)</td><td colspan="2">TargetHardwareAddr (bytes 0–1)</td></tr>
<tr><td colspan="4">TargetHardwareAddr (bytes 2–5)</td></tr>
<tr><td colspan="4">TargetProtocolAddr (bytes 0–3)</td></tr>
</table>

- HardwareType: type of physical network (e.g., Ethernet)
- ProtocolType: type of higher layer protocol (e.g., IP)
- HLEN & PLEN: length of physical and protocol addresses
- Operation: request or response
- Source/Target Physical/Protocol addresses