IPv4 Addressing: Basics

- Globally unique (for "public" IP addresses)
- IPv4 address: 32-bit identifier for host, router *interface*
- *Interface:* connection between host/router and physical link
  - router's typically have multiple interfaces
  - host may have multiple interfaces
  - IP addresses associated with each interface
- Dot notation (for ease of human reading)

223.1.1.1 = 11011111 00000001 00000001 00000001

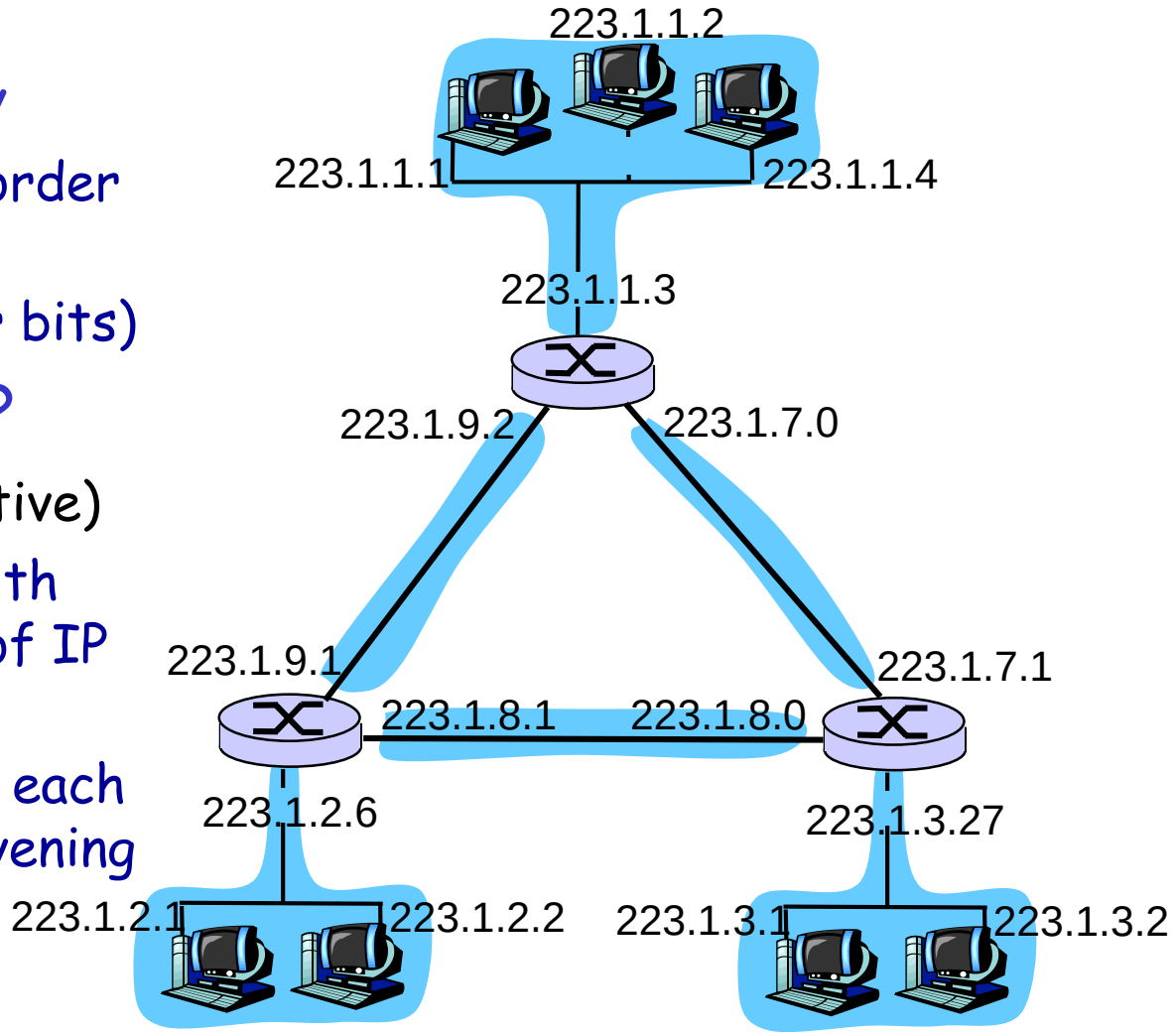
223 1 1 1

IP Addressing: Network vs. Host

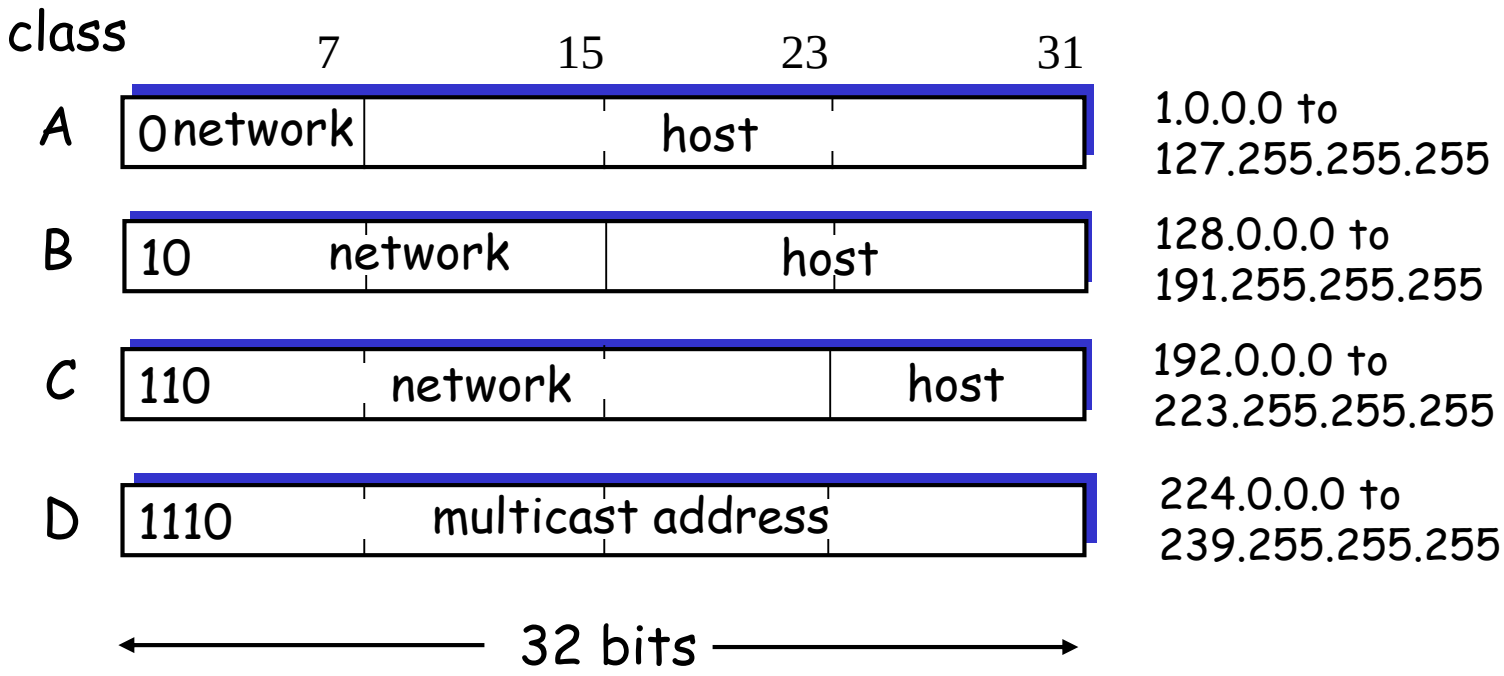
- Two-level hierarchy
  - network part (high order bits)
  - host part (low order bits)
- *What's a network ?*

(from IP address perspective)

  - device interfaces with same network part of IP address
  - can physically reach each other without intervening router



"Classful" IP Addressing

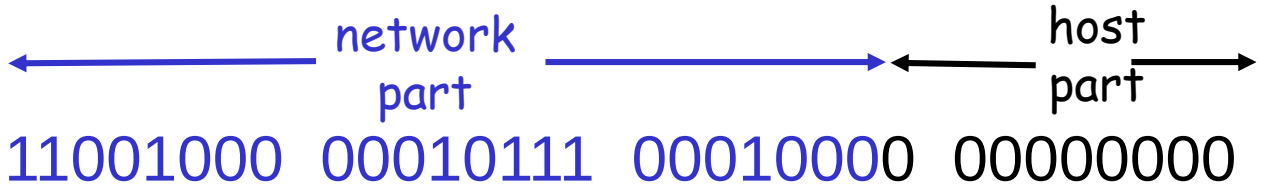


- Disadvantage: inefficient use of address space, address space exhaustion
- e.g., class B net allocated enough addresses for 65K hosts, even if only 2K hosts in that network

Classless Addressing: CIDR

CIDR: Classless InterDomain Routing

- Network portion of address is of arbitrary length
- Addresses allocated in contiguous blocks
  - Number of addresses assigned always power of 2
- Address format: a.b.c.d/x
  - x is number of bits in network portion of address



200.23.16.0/23

Special IP Addresses

- Network address: host id = all 0's
- Directed broadcast address: host id = all 1's
- Local broadcast address: all 1's
- Local host address (this computer): all 0's
- Loopback address
  - network id = 127, any host id (e.g. 127.0.0.1)

IP Addresses: How to Get One?

Q: How does *host* get IP address?

- "static" assigned: i.e., hard-coded in a file
  - Wintel: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- Dynamically assigned: using DHCP (Dynamic Host Configuration Protocol)
  - dynamically get address from as server
  - "plug-and-play"

DHCP: Dynamic Host Configuration Protocol

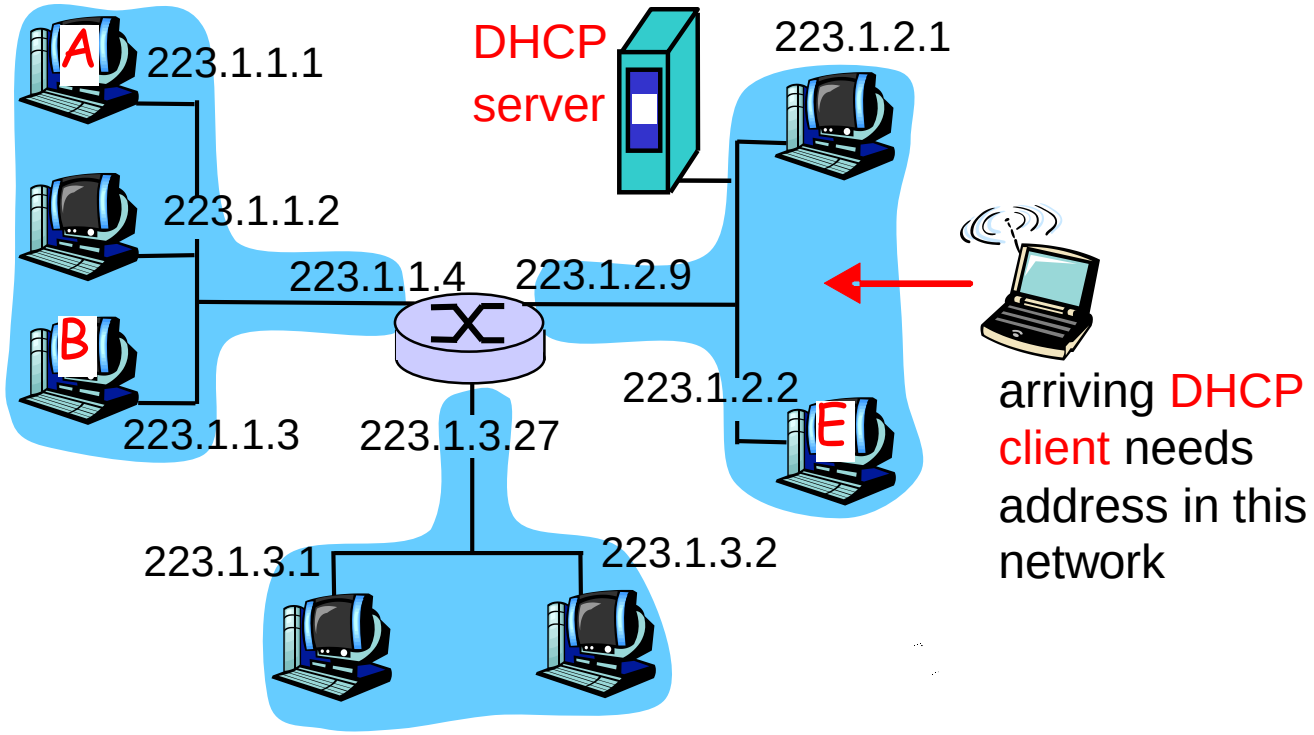
Goal: allow host to *dynamically* obtain its IP address from network server when it joins network

- Can renew its lease on address in use
- Allows reuse of addresses (only hold address while connected an "on"
- Support for mobile users who want to join network (more shortly)

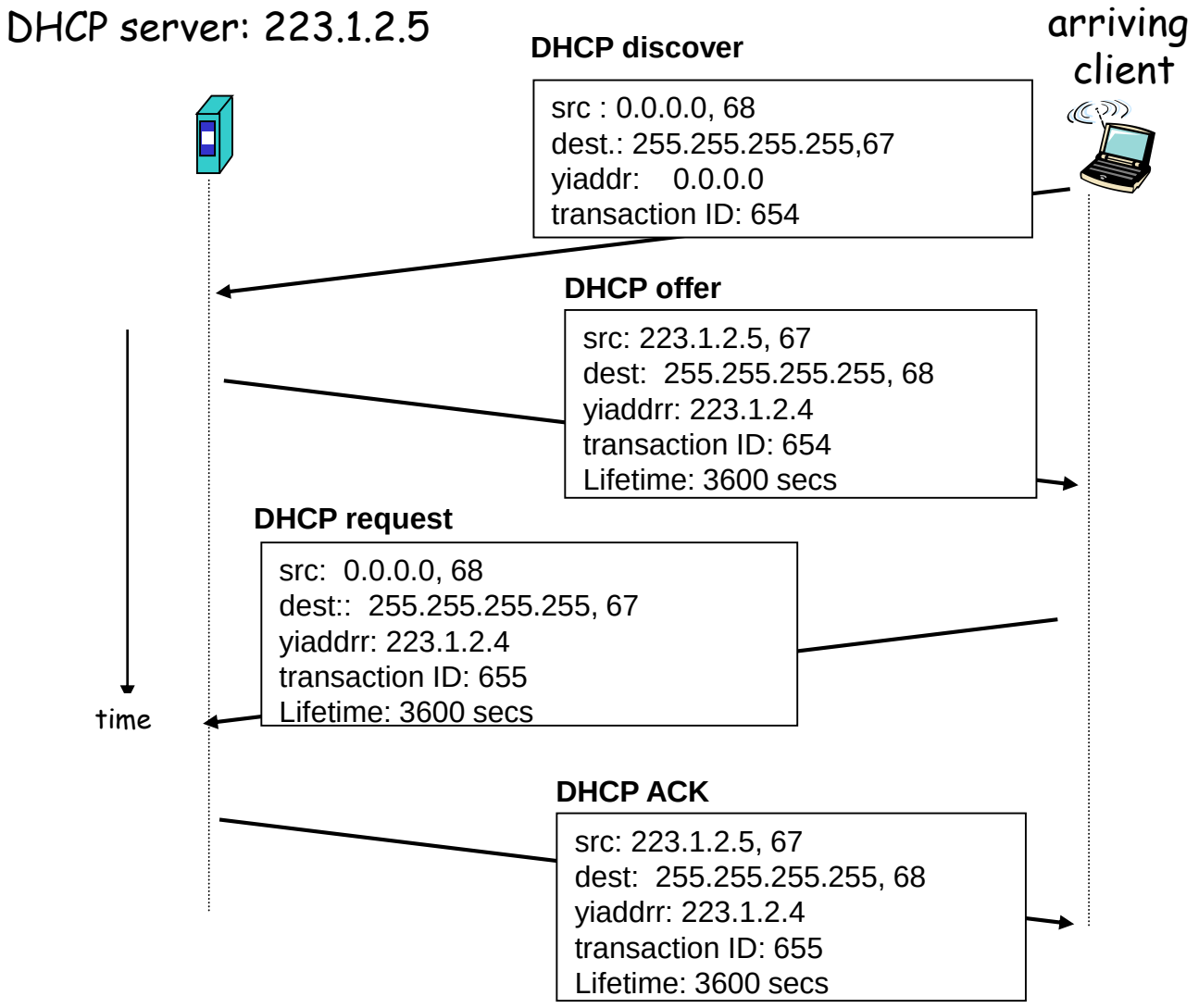
DHCP overview:

- 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

DHCP Client-Server Scenario



DHCP Client-Server Scenario



DHCP: More Than IP Addresses

DHCP can return more than just allocated IP address on subnet:

- address of first-hop router for client
- name and IP address of DNS sever
- network mask (indicating network versus host portion of address)

IP Addresses: How to Get One? …

Q: How does *network* get network part of IP addr?

A: gets allocated portion of its provider ISP's address space

ISP's block	11001000 00010111 00010000	00000000	200.23.16.0/20
Organization 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100	00000000	200.23.20.0/23
…	……	….	….
Organization 7	11001000 00010111 00011110	00000000	200.23.30.0/23

IP Addressing: the Last Word...

Q: How does an ISP get block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers

- allocates addresses
- manages DNS
- assigns domain names, resolves disputes

Network Layer
Connection vs. Connectionless Service

- datagram network provides network-layer connectionless service
- VC network provides network-layer connection service
- analogous to the transport-layer services, but:
  - service: host-to-host
  - generally no choice: network *typically* provides one or the other
  - implementation: in network core

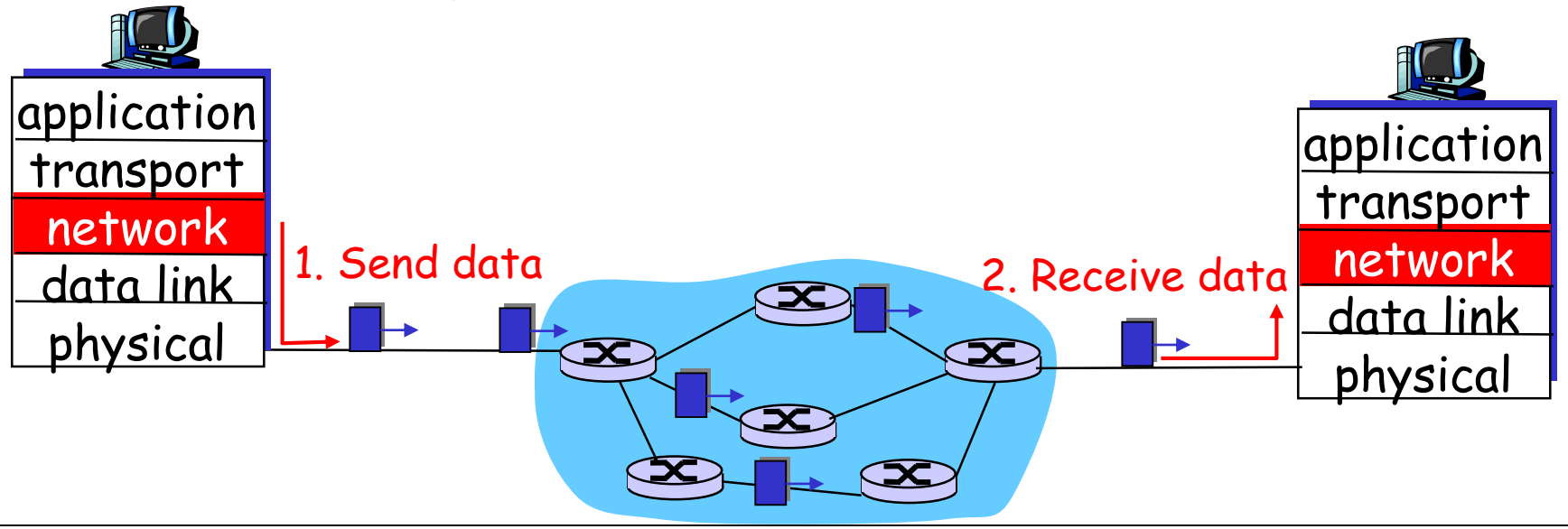
- network vs transport layer connection service:
  - network: between two hosts, in case of VCs, also involves intervening routers
  - transport: between two processes

Virtual Circuit vs. Datagram

- Objective of both: move packets through routers from source to destination
- Datagram Model:
  - *Routing:* determine next hop to each destination a priori
  - *Forwarding:* destination address in packet header, used at each hop to look up for next hop
    - routes may change during "session"
  - analogy: driving, asking directions at every gas station, or based on the road signs at every turn
- Virtual Circuit Model:
  - *Routing:* determine a path from source to each destination
  - *"Call" Set-up:* fixed path ("virtual circuit") set up at *"call" setup time*, remains fixed thru "call"
  - *Data Forwarding:* each packet carries "tag" or "label" (virtual circuit id, VCI), which determines next hop
  - *routers maintain "per-call" state*

Datagram Networks: the Internet model

- no call setup at network layer
- routers: no state about end-to-end connections
  - no network-level concept of "connection"
- packets forwarded using destination host address
  - packets between same source-dest pair may take different paths, when intermediate routes change!



Forwarding Table

4 billion possible entries

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

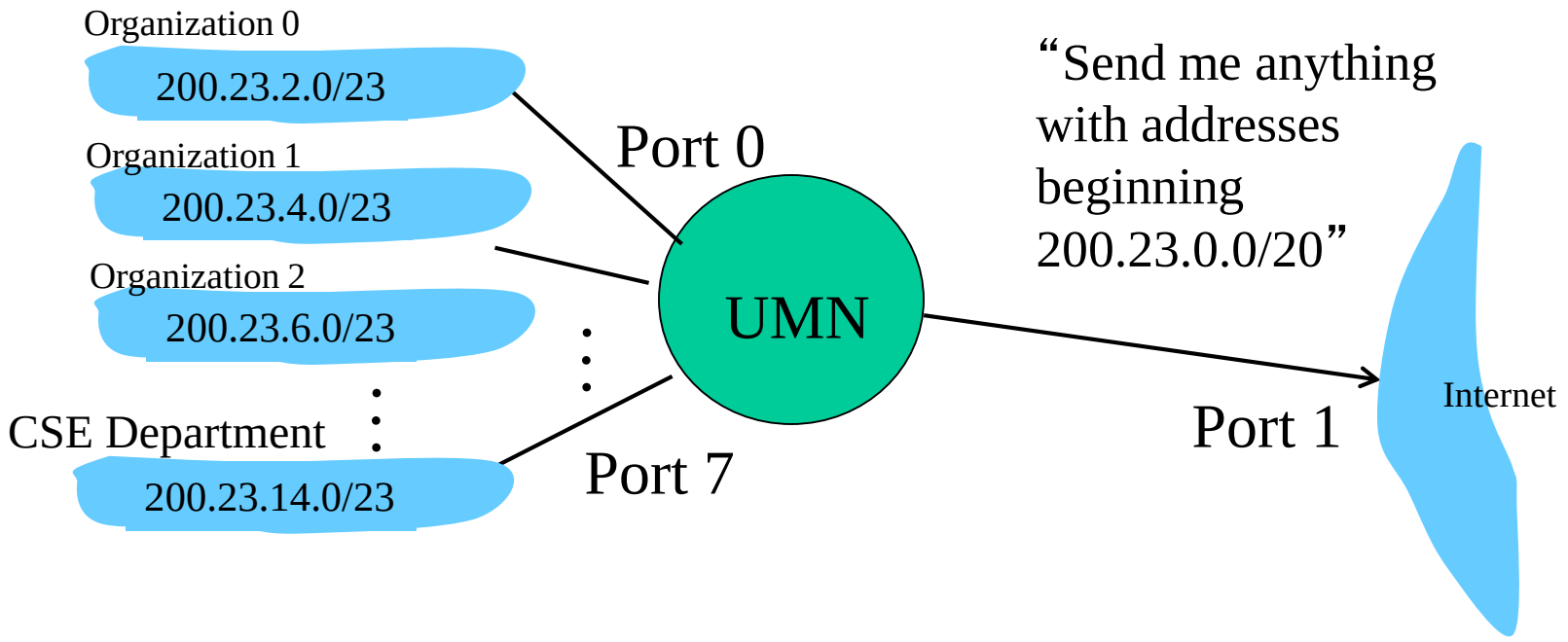
IP Forwarding Table

4 billion possible entries!
(in reality, far less, but can still have millions of "routes")

forwarding table entry format

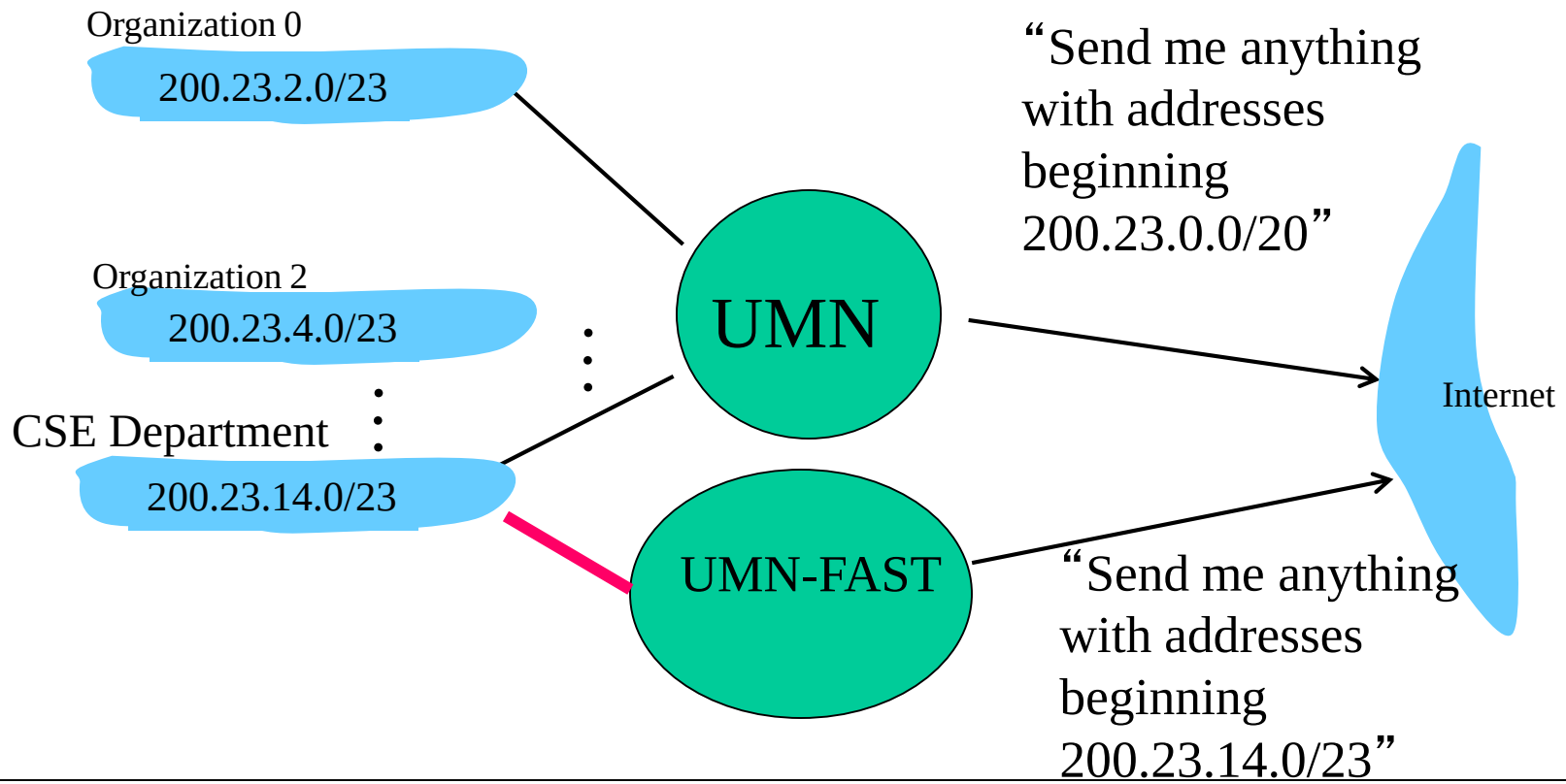
destination network (1st IP address , network mask)	next-hop (IP address)	link interface
11001000 00010111 00010000 00000000, 11111111 11111111 11111000 00000000	200.23.16.1	0
11001000 00010111 00011000 00000000, 11111111 11111111 11111111 00000000	- (direct)	1
11001000 00010111 00011001 00000000, 11111111 11111111 11111000 00000000	200.23.25.6	2
otherwise	128.30.0.1	3

Route aggregation: Shrinking the forwarding table



Route aggregation with more specific routes

UMN-FAST has a more specific route to CSE department



Longest Prefix Matching

longest prefix matching

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

Prefix Match	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

Examples

DA: 11001000 00010111 00010110 10100001 interface 0

DA: 11001000 00010111 00011000 10101010 interface 1

But *not* interface 2, the 3rd entry is also a match, but shorter!

IP Datagram Forwarding Model

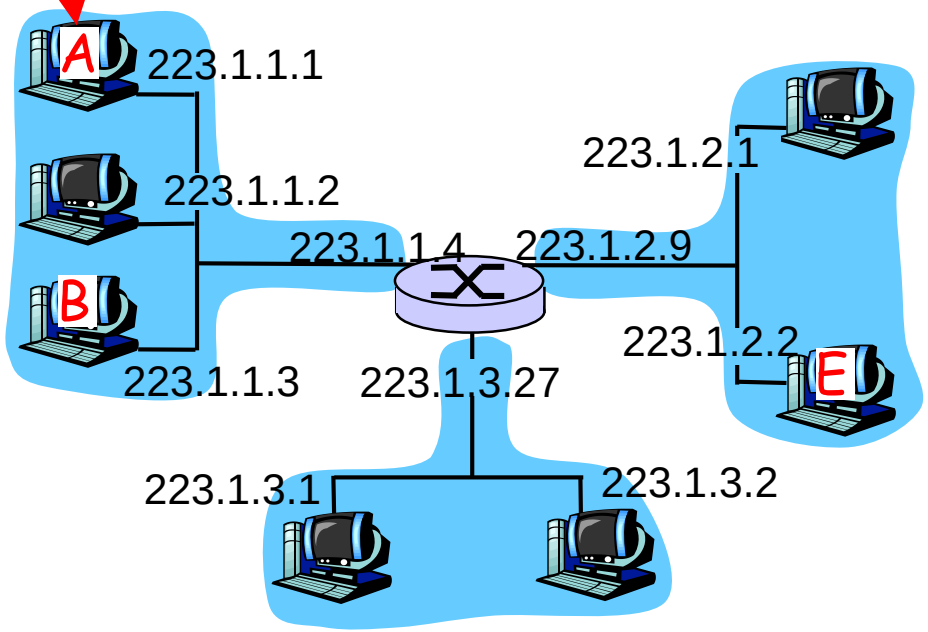
IP datagram:

misc fields	source IP addr	dest IP addr	data

- datagram remains unchanged, as it travels source to destination
- addr fields of interest here

forwarding table in A

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



IP Forwarding: Destination in Same Net

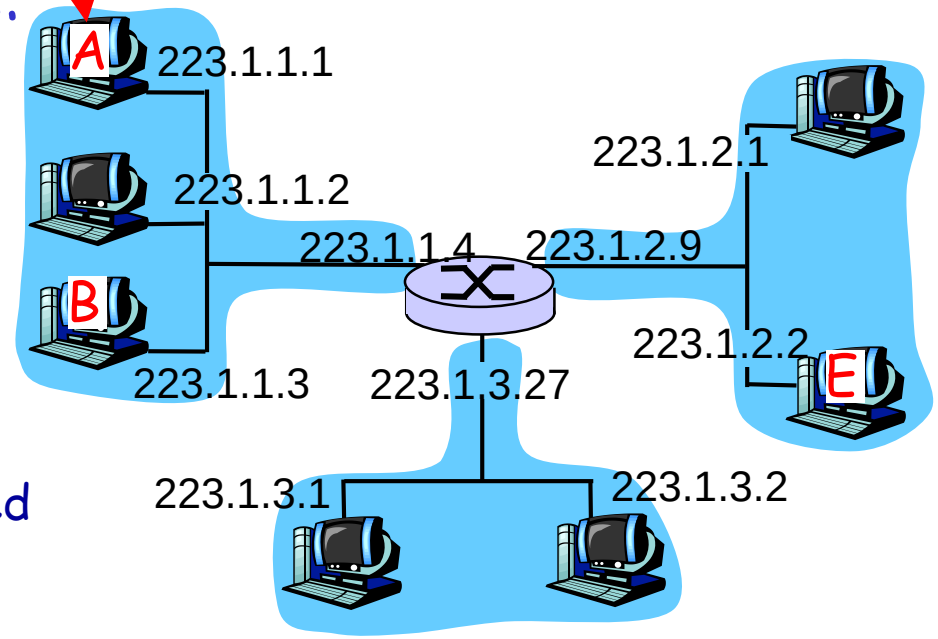
misc fields	223.1.1.1	223.1.1.3	data

forwarding table in A

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2

Starting at A, send IP datagram addressed to B:

- look up net. address of B in forwarding table
- find B is on same net. as A
- link layer will send datagram directly to B inside link-layer frame
  - B and A are directly connected



IP Forwarding: Destination in Diff. Net

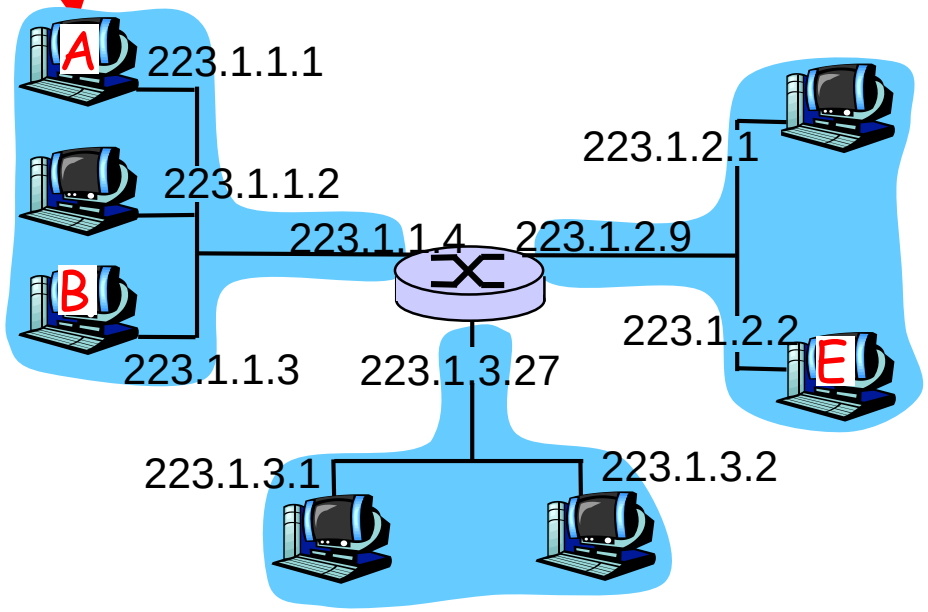
misc fields	223.1.1.1	223.1.2.3	data

Starting at A, dest. E:

- look up network address of E in forwarding table
- E on *different* network
  - A, E not directly attached
- routing table: next hop router to E is 223.1.1.4
- link layer sends datagram to router 223.1.1.4 inside link-layer frame
- datagram arrives at 223.1.1.4
- continued…..

forwarding table in A

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



IP Forwarding: Destination in Diff. Net …

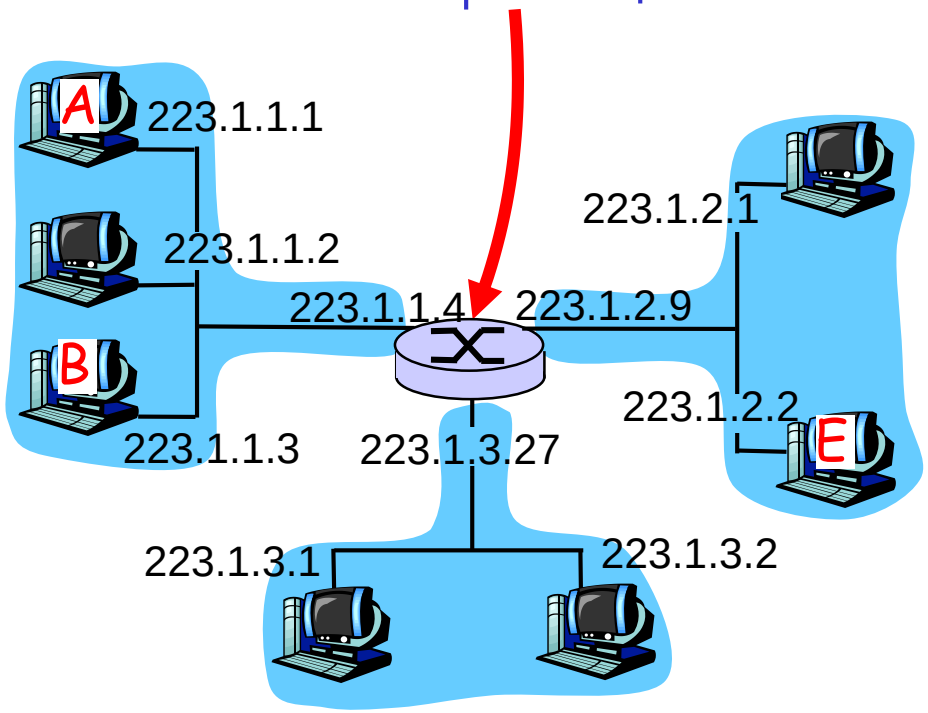
misc fields	223.1.1.1	223.1.2.3	data

forwarding table in router

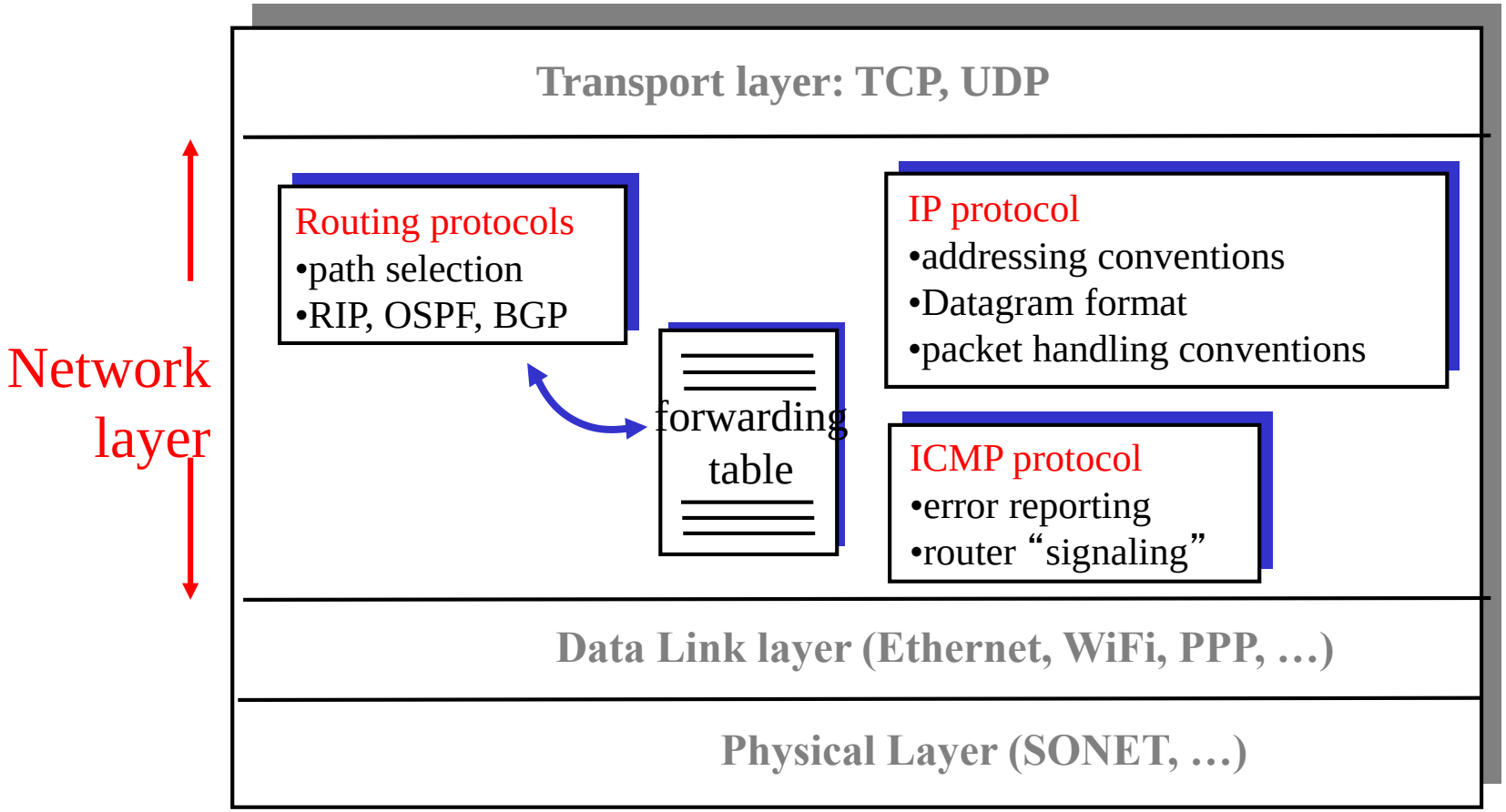
Dest. Net	router	Nhops	interface
223.1.1	-	1	223.1.1.4
223.1.2	-	1	223.1.2.9
223.1.3	-	1	223.1.3.27

Arriving at 223.1.4, destined for 223.1.2.2

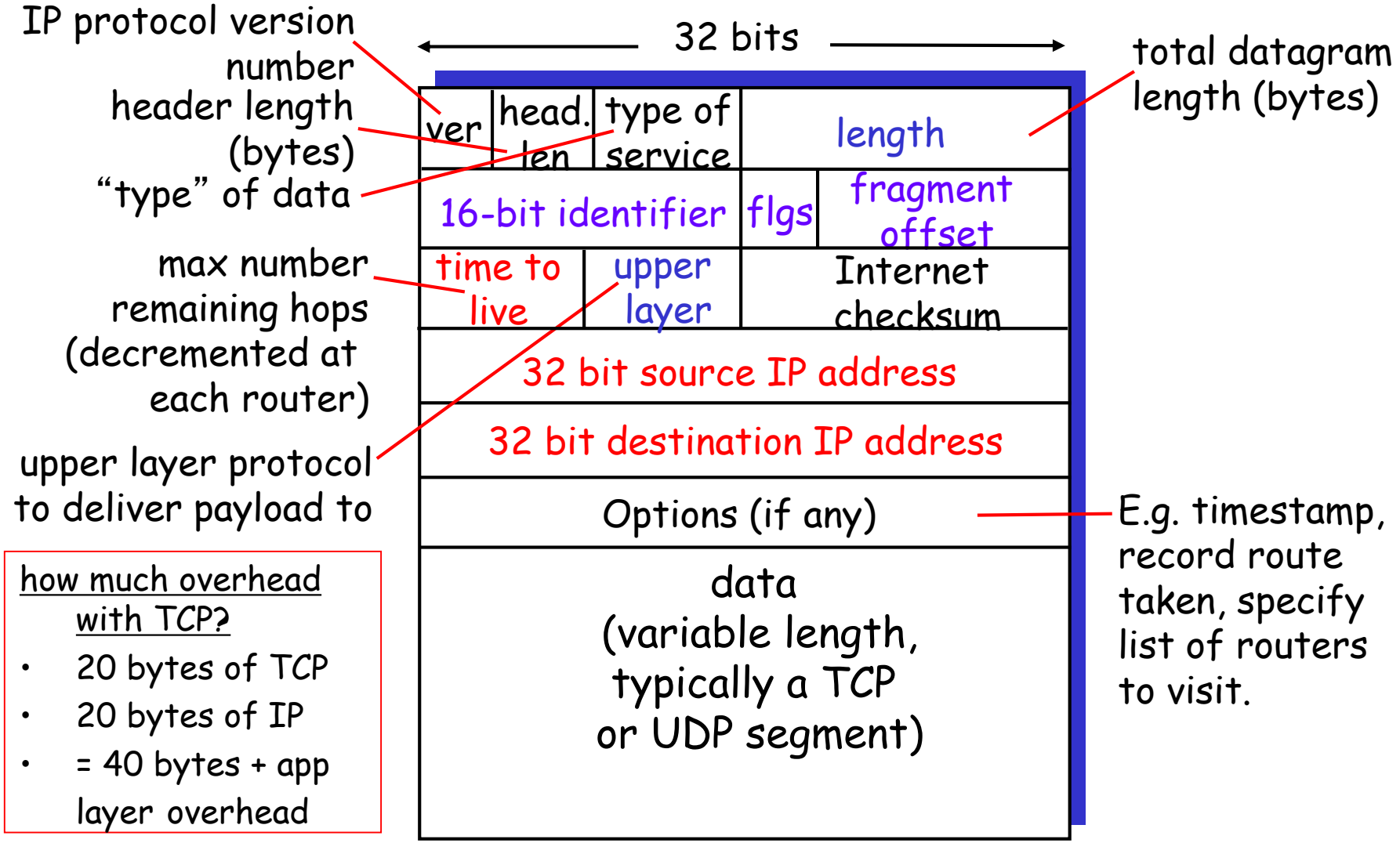
- look up network address of E in router's forwarding table
- E on *same* network as router's interface 223.1.2.9
  - router, E directly attached
- link layer sends datagram to 223.1.2.2 inside link-layer frame via interface 223.1.2.9
- datagram arrives at 223.1.2.2!!! (hooray!)



IP Forwarding & IP/ICMP Protocol



IP Datagram Format



how much overhead with TCP?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

Fields in IP Datagram

- IP protocol version: current version is 4, IPv4, new: IPv6
- Header length: number of 32-bit words in the header
- Type of Service:
  - 3-bit priority,e.g, delay, throughput, reliability bits, …
- Total length: including header (maximum 65535 bytes)
- Identification: all fragments of a packet have same identification
- Flags: don't fragment, more fragments
- Fragment offset: where in the original packet (count in 8 byte units)
- Time to live: maximum life time of a packet
- Protocol Type: e.g., ICMP, TCP, UDP etc
- IP Option: non-default processing, e.g., IP source routing option, etc.

Will come back to discuss more about some of these headers!