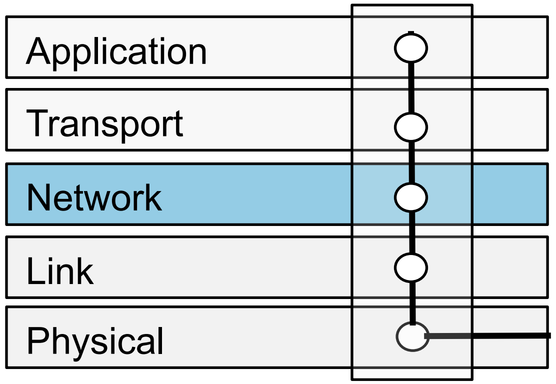
Internet Protocol

To do …

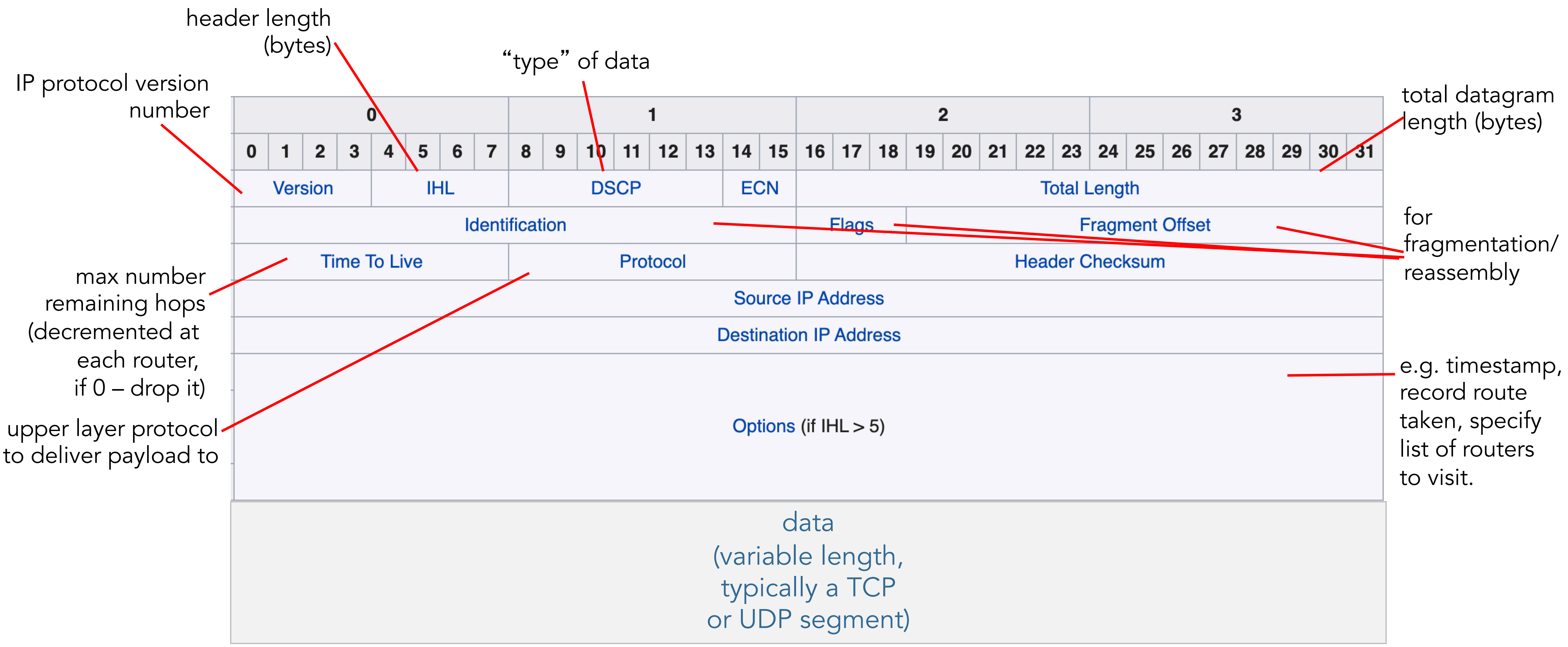
- The Internet protocol (IP) – IPv4, IPv6 …



Northwestern

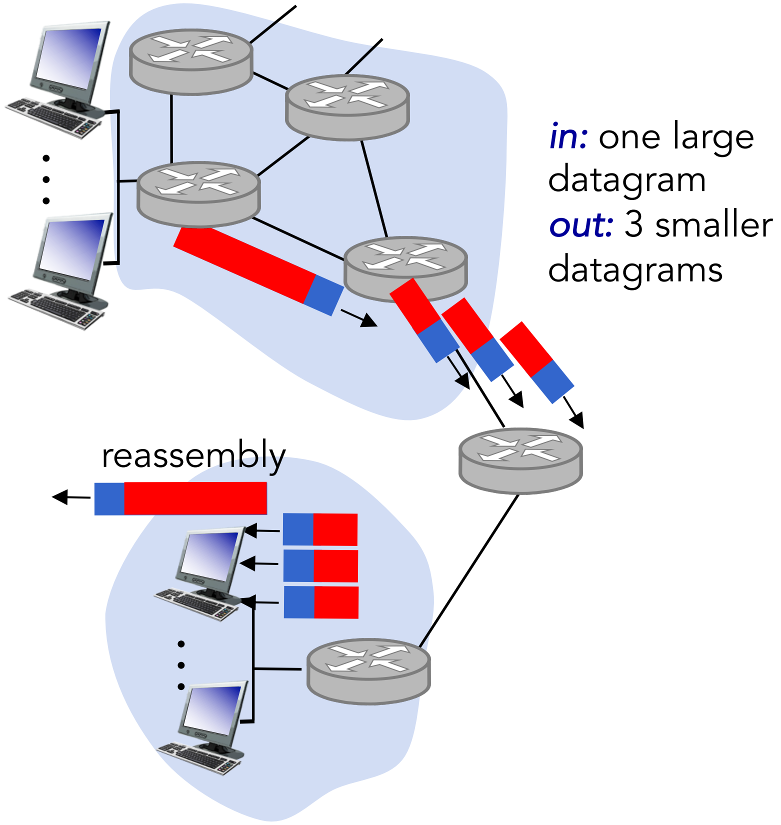
Slides partially based on J. Kurose and K. Ross, "Computer Networking: A Top Down Approach", 7Ed., 2016

IP datagram format



IP fragmentation, reassembly

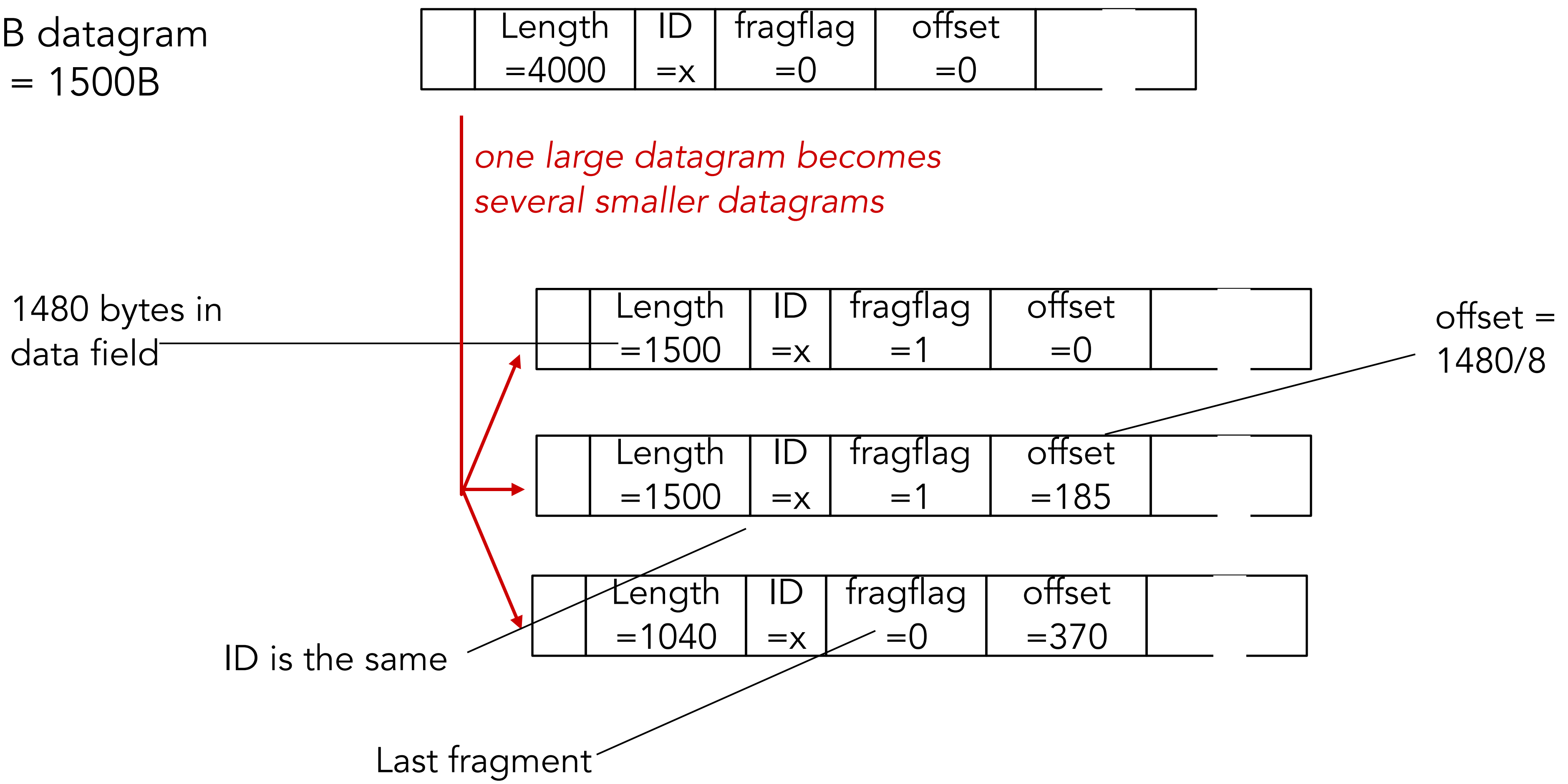
- Network links have some MTU (max. transfer size) – largest possible link-level frame
  - And different link types, different MTUs
- Large IP datagram divided ("fragmented") within net
  - One datagram becomes several datagrams
  - Reassembled only at final destination, on end systems (*why?*)
  - IP header bits used to identify, order related fragments



IP fragmentation, reassembly

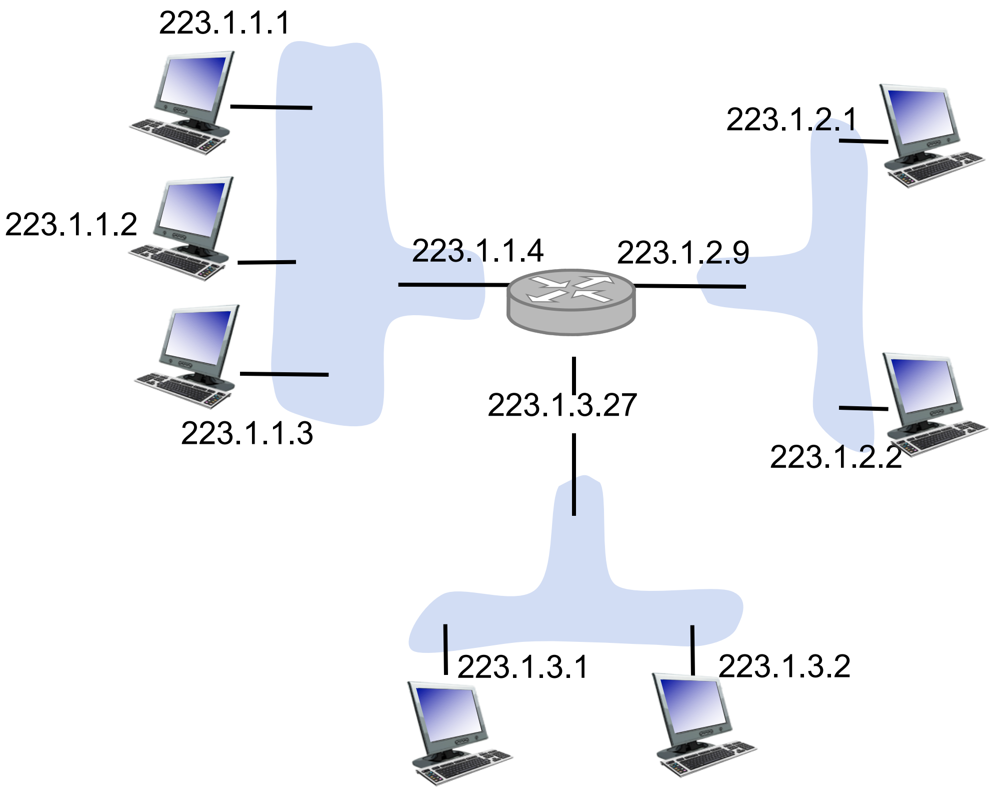
example:

- 4000B datagram
- MTU = 1500B



IPv4 addressing

- Interface: connection between host/router and physical link
  - Router’s typically have multiple interfaces
  - Hosts typically have one or two (e.g., wired Ethernet, wireless 802.11)
- IP addresses associated with each interface
- IP address: 32-bit identifier for host, router interface
  - Typically written in dotted-decimal notation (each byte in decimal and separated by a period)

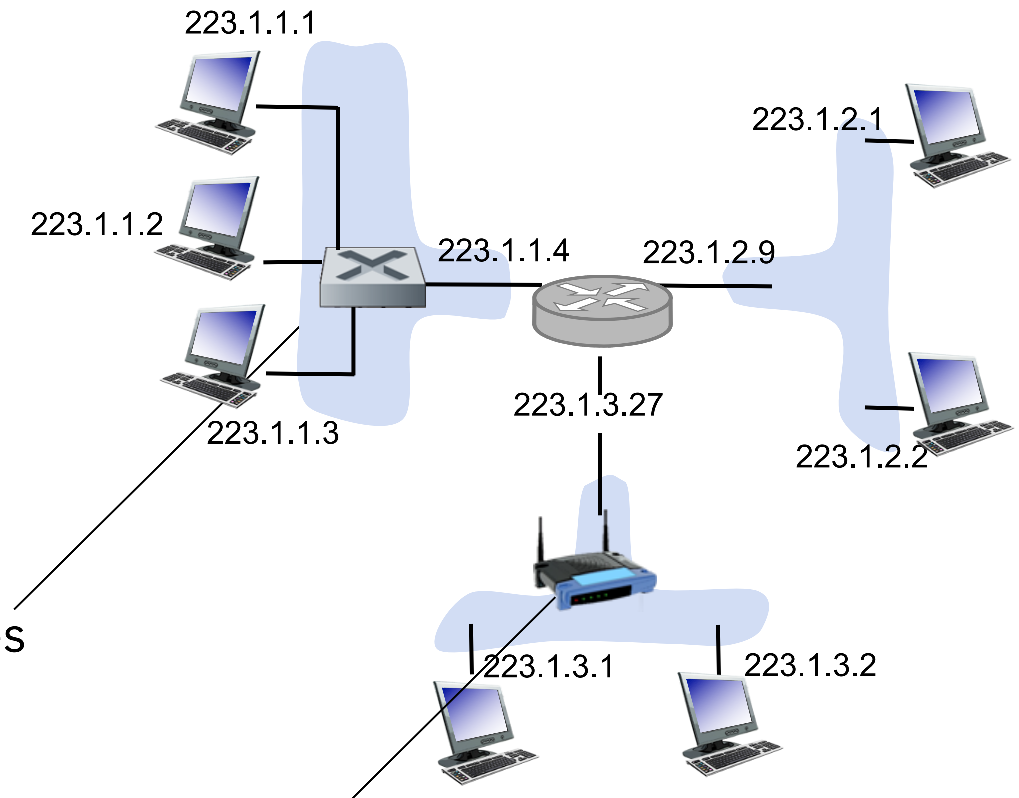


223.1.1.1 = 11011111|00000001|00000001|00000001

223 1 1 1

IP addressing: introduction

- How are interfaces actually connected?



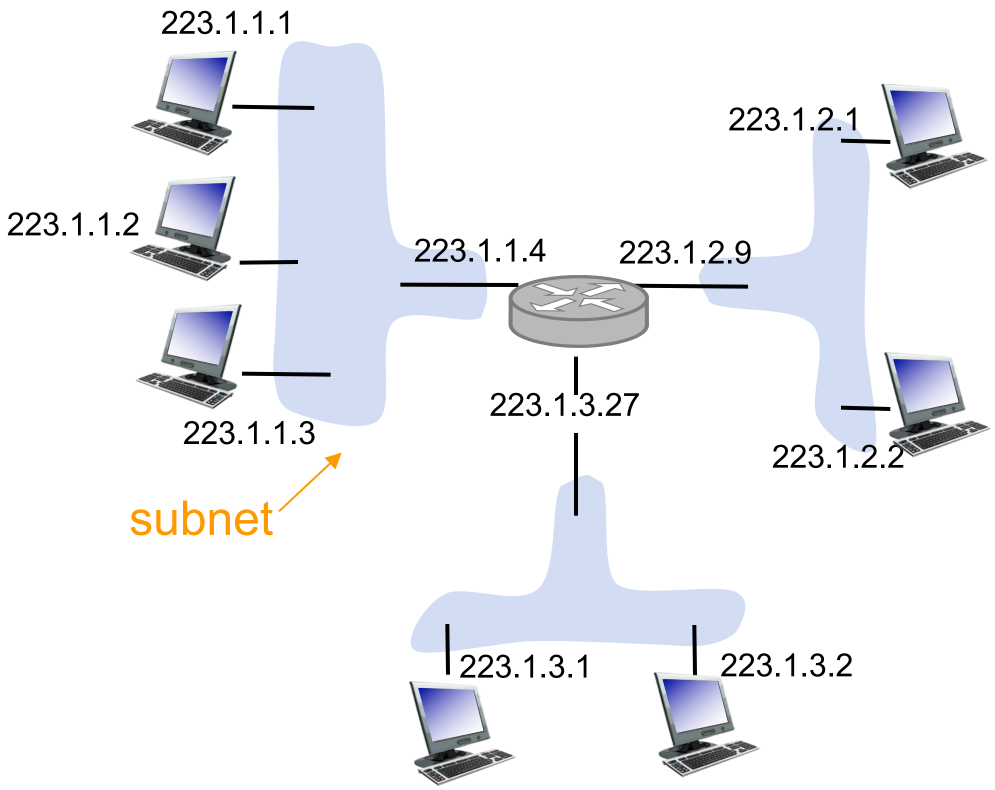
Wired Ethernet interfaces connected by Ethernet switches

For now: don't need to worry about how one interface is connected to another (with no intervening router)

Wireless WiFi interfaces connected by WiFi base station

Subnets

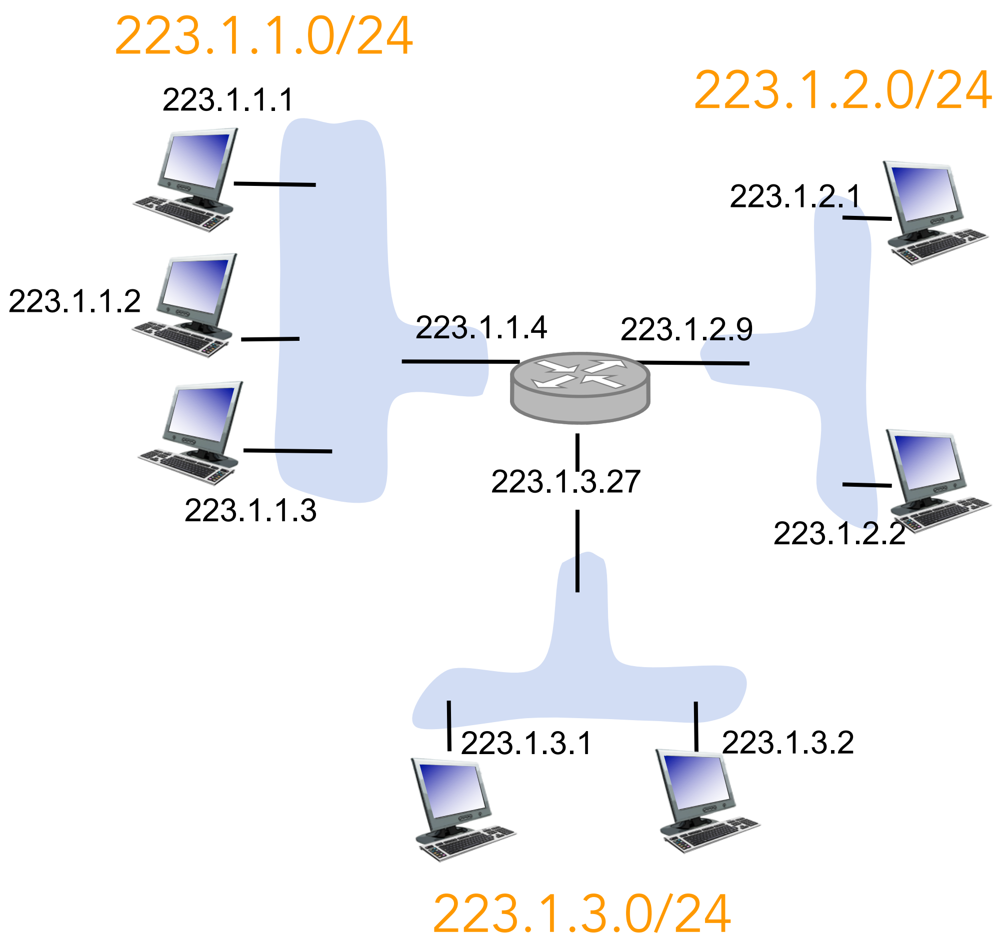
- IP address
  - Subnet part – high order bits
  - Host part – low order bits
- What's a subnet ?
  - Device interfaces with same subnet part of IP address
  - Can physically reach each other without intervening router



network consisting of 3 subnets

Subnets

- Recipe to determine the subnets
  - Detach each interface from its host or router, creating islands of isolated networks
  - Each isolated network is called a subnet



Subnet mask: /24

IP addressing: CIDR

- The Internet's address assignment strategy is called CIDR
- Classless InterDomain Routing
  - Subnet portion of address of arbitrary length
  - Address format: a.b.c.d/x, where x is # bits in subnet portion (most significant)

← subnet → ← host →

11001000 00010111 00010000 00000000

200.23.16.0/23

  - Using prefixes reduces the size of forwarding tables for routers outside the organization (a single entry for a.b.c.d/x is enough)
  - *Classful addressing* forced subnets portions to be 1 (A), 2 (B) or 3 (C) bytes, leading to wasted IPs – C too small (254), B (65,634) too large

IP addresses: how does a host get one?

- Hard-coded by system admin in a file
  - Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server
  - "plug-and-play"

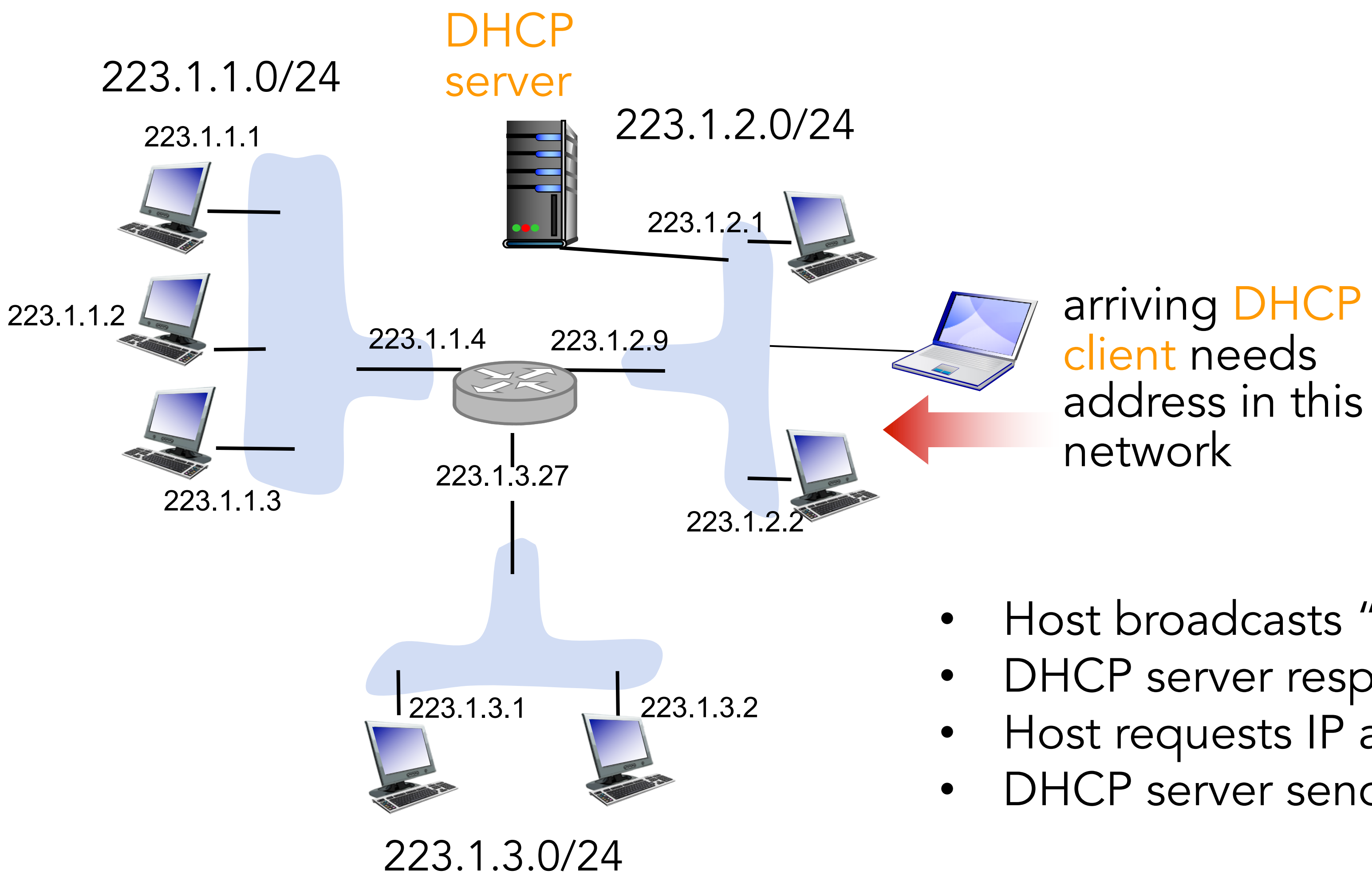
DHCP [RFC 2131]

- A host can dynamically obtain an IP from network server when joining a network
  - Host can renew its lease on address in use
  - Allows reuse of addresses (only hold address while connected/"on") or it can always assign the same
  - Support for mobile users who want to join network
- DHCP overview
  - Host broadcasts "DHCP discover" msg [optional]
  - DHCP server responds with "DHCP offer" msg [optional]
  - Host requests IP address: "DHCP request" msg
  - DHCP server sends address: "DHCP ack" msg
  - (DHCP request encapsulated in UDP)

DHCP [RFC 2131]

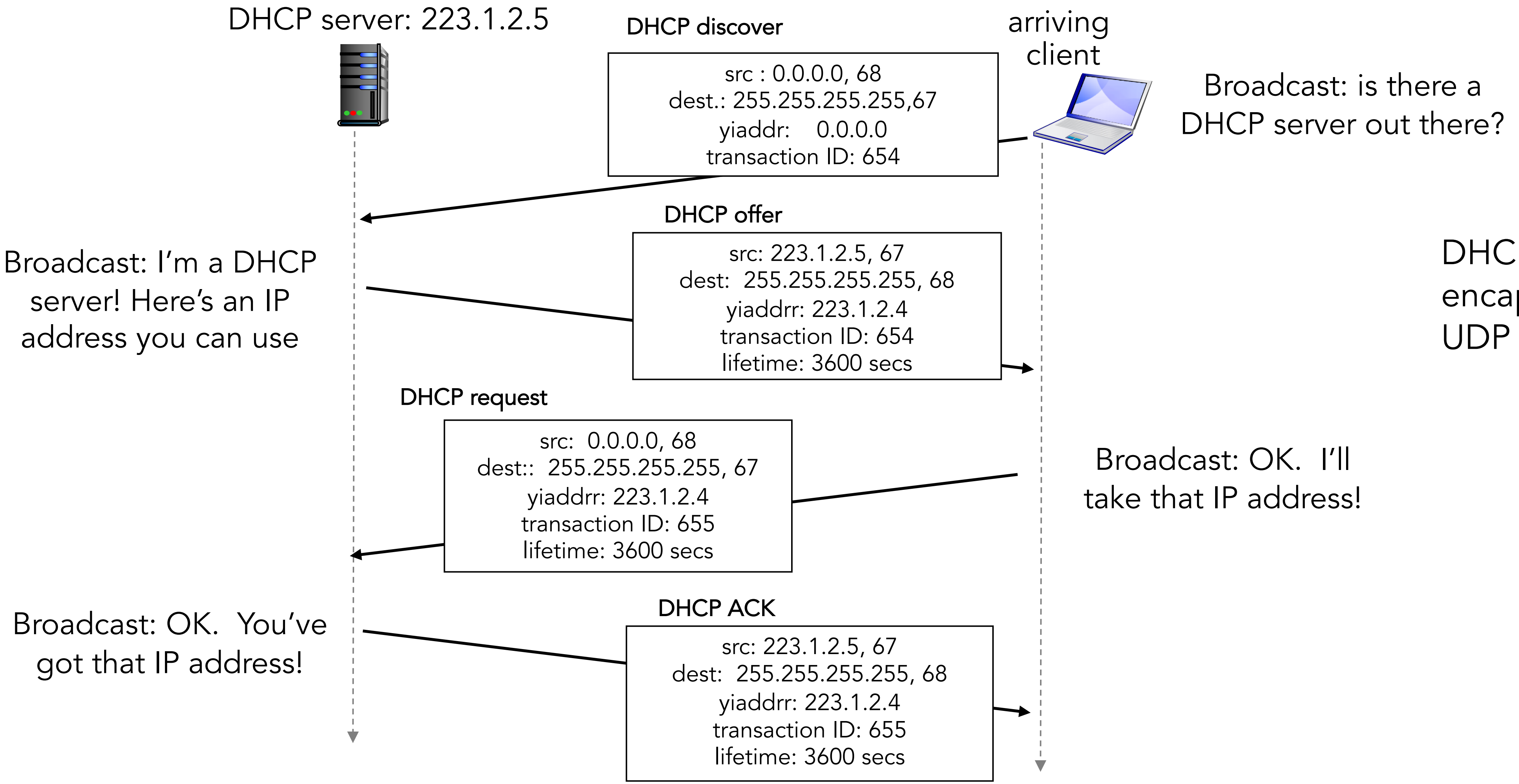
- A host can dynamically obtain an IP from network server when joining a network
  - Host can renew its lease on address in use
  - Allows reuse of addresses (only hold address while connected/"on") or it can always assign the same
  - Support for mobile users who want to join network
- DHCP can return more than just allocated IP address on subnet
  - Address of first-hop router for client, the default gateway
  - Name and IP address of DNS sever(s)
  - Network mask (indicating network versus host portion of address)

DHCP overview – a client-server protocol



- Host broadcasts "DHCP discover" msg [optional]
- DHCP server responds with "DHCP offer" msg [optional]
- Host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg

DHCP client-server scenario



DHCP request encapsulated in UDP

DHCP
UDP
IP
Eth
Phy

IP addresses: how to get one?

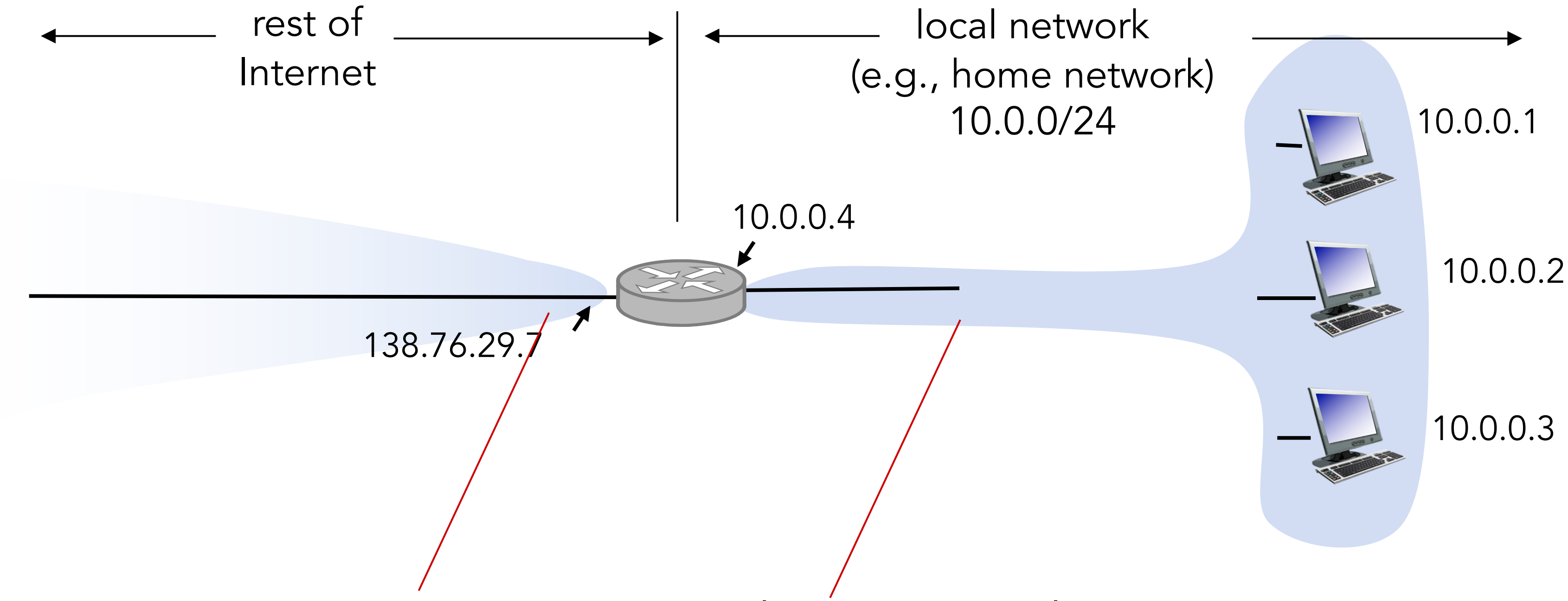
- How to get a subnet part of IP addr?
- Gets allocated portion of its provider ISP's address space
  - E.g., the ISP could have been allocated a /20, divide it into 8 parts and give you one

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...

- How does an ISP get block of addresses?
  - ICANN: Internet Corporation for Assigned
- Names and Numbers http://www.icann.org/
  - allocates addresses
  - manages DNS
  - assigns domain names, resolves disputes

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)

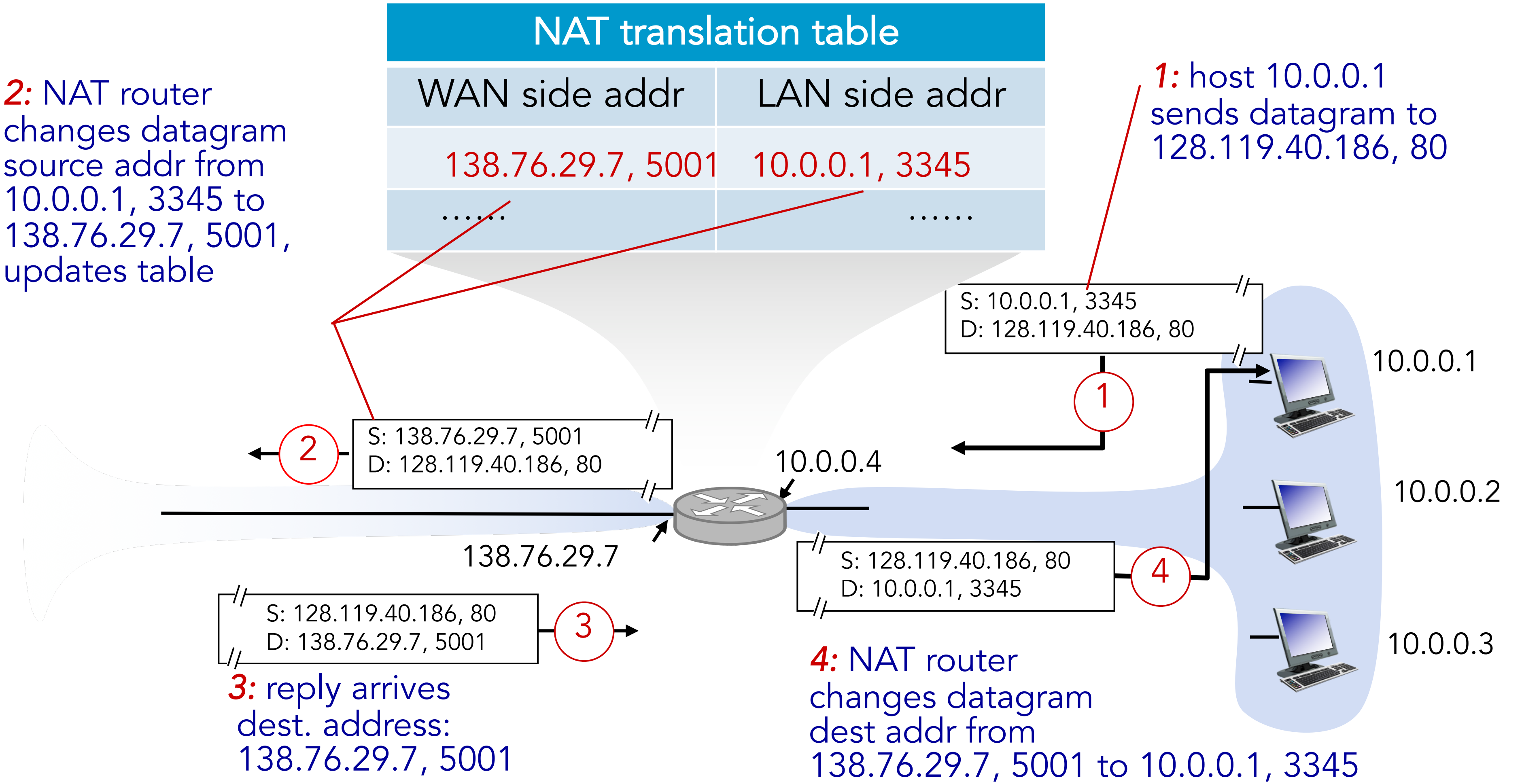
Why NATs?

- Local network uses just one IP address as seen from Range of addresses not needed from ISP: just one IP address for all devices
  - Can change addresses of devices in local network without notifying outside world
  - Can change ISP without changing addresses of devices in local network
  - Devices inside local net not explicitly addressable, visible by outside world (a security plus)

NAT implementation

- NAT router must
  - Outgoing datagrams: replace (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
    - . . . remote clients/servers will respond using (NAT IP address, new port #) as destination address
  - Remember (in NAT translation table) every (source IP address, port #) to (NAT IP address, new port #) translation pair
  - Incoming datagrams: replace (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT



* Check out the online interactive exercises for more examples:
http://gaia.cs.umass.edu/kurose_ross/interactive/

NAT

- 16-bit port-number field:
  - 60,000 simultaneous connections with a single LAN-side address!
- Some controversy
  - Routers should only process up to layer 3
  - Address shortage should be solved by IPv6
  - Violates end-to-end argument
    - NAT possibility must be taken into account by app designers, e.g., P2P applications
  - NAT traversal: what if client wants to connect to server behind NAT?

IPv6: motivation

- Initially, 32b 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
  - no fragmentation allowed

IPv6 datagram format

<table>
<tr><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="4">Version</td><td colspan="8">Traffic Class</td><td colspan="20">Flow Label</td></tr>
<tr><td colspan="16">Payload Length</td><td colspan="8">Next Header</td><td colspan="8">Hop Limit</td></tr>
<tr><td colspan="32">Source Address</td></tr>
<tr><td colspan="32">Destination Address</td></tr>
<tr><td colspan="32">data
(variable length,
typically a TCP
or UDP segment)</td></tr>
</table>

Identify datagrams in same "flow" (concept of"flow" not well defined).

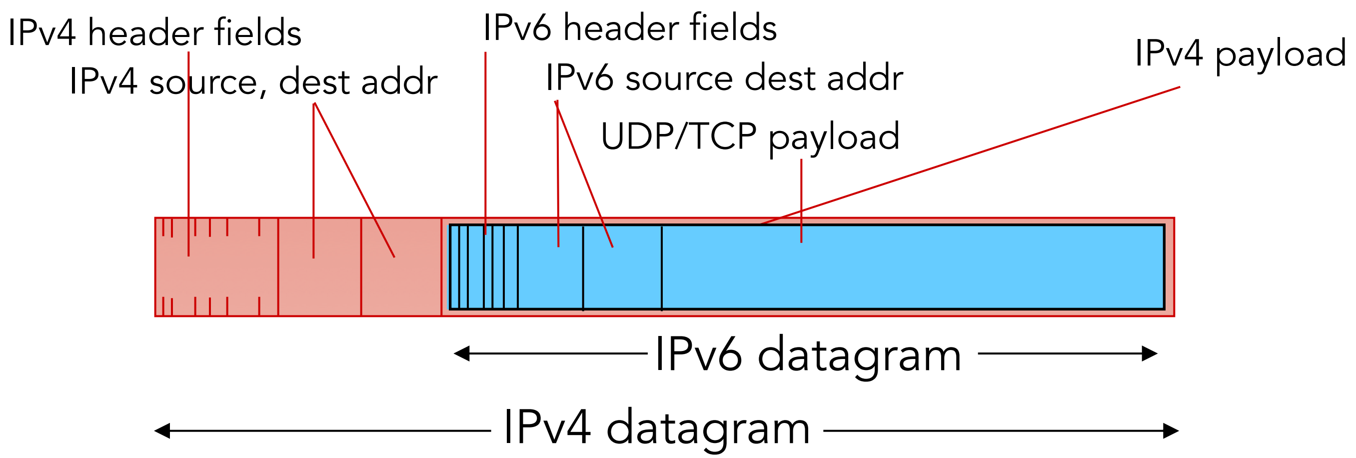
identify upper layer protocol for data

Other changes from IPv4

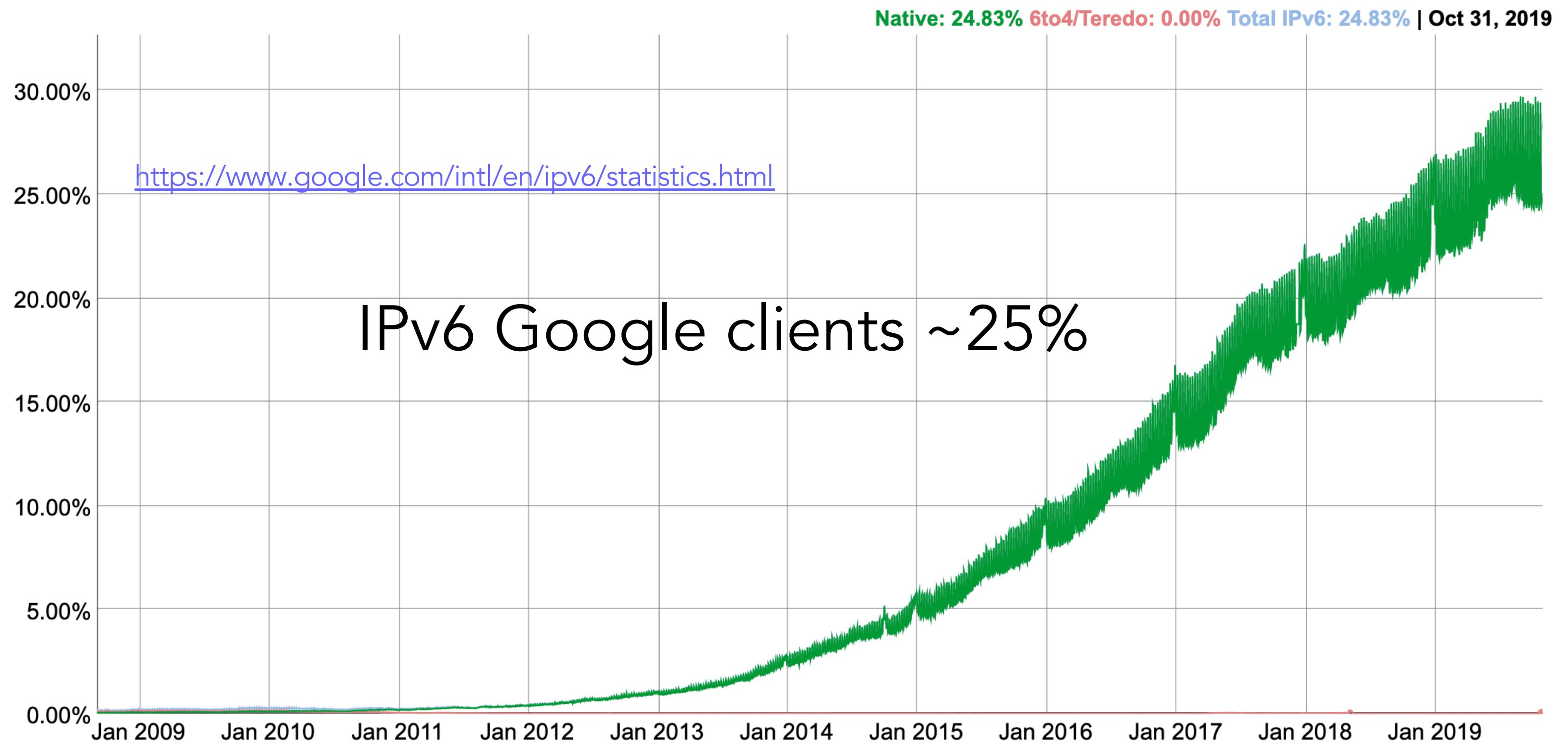
- checksum: removed entirely to reduce processing time at each hop
- options: allowed, but outside of header, indicated by "Next Header" field
- ICMPv6: new version of ICMP
  - additional message types, e.g. "Packet Too Big"
  - multicast group management functions

Transition from IPv4 to IPv6

- Not all routers can be upgraded simultaneously
  - no "flag days"
  - how will network operate with mixed IPv4 and IPv6 routers?
- Tunneling: IPv6 datagram carried as payload in IPv4 datagram among IPv4 routers



IPv6 adoption



- Long (long!) time for deployment, use
  - 20 years and counting!
  - think of application-level changes in last 20 years: WWW, Facebook, streaming media, Skype, …

Recap

- We covered the data plane function of the network layer
  - The per-router functions that rule how packets arriving on a router's input link are forwarded on an output link
- From routers' internals to addressing IPv4 and v6
- Next, the control plane - network-wide logic that control end-to-end routing and how services are configured and managed