Network Layer Protocols

IP

Kind Reminder

- Next week: week 11 Tues. 2-4pm EHB110A
  - Exam infor. + Conclusions (OS+Nets)
  - Q&A
- Exam infor.; Sample exam paper format, questions and solutions; Conclusion slides ALL on Learn

- Exam (in-person): double check your timetable

Quick review of last lecture (w9)

- Revisit Layers
- Transport layer
- Revisit Processes, Sockets & Ports
- UDP
  - Segment structure
- TCP
  - Segment structure
  - SQN
  - Connection setup
  - Connection clear down
  - Flow Control
  - Congestion control

Loughborough University

Today

Contents

Part 1:

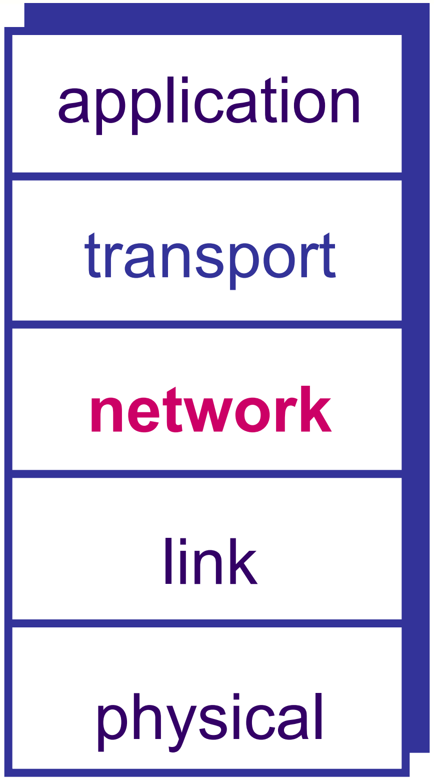
- Network Layer & IP Address Recap
- Dynamic Host Configuration Protocol (DHCP)
- Network Address Translation (NAT)

Part 2:

- IPv4
- IP Fragmentation
- Internet Control Message Protocol (ICMP)
- IPv6

Transport layer vs. Network layer

- Transport layer
  - Process-to-process communication
  - Transport protocols run in end systems
  - send side: breaks app messages into segments, passes to network layer
  - rcv side: reassembles segments into messages, passes to app layer
- Network layer
  - Host-to-host communication
  - Provide services to transport-layer & Using services from link-layer


application

transport

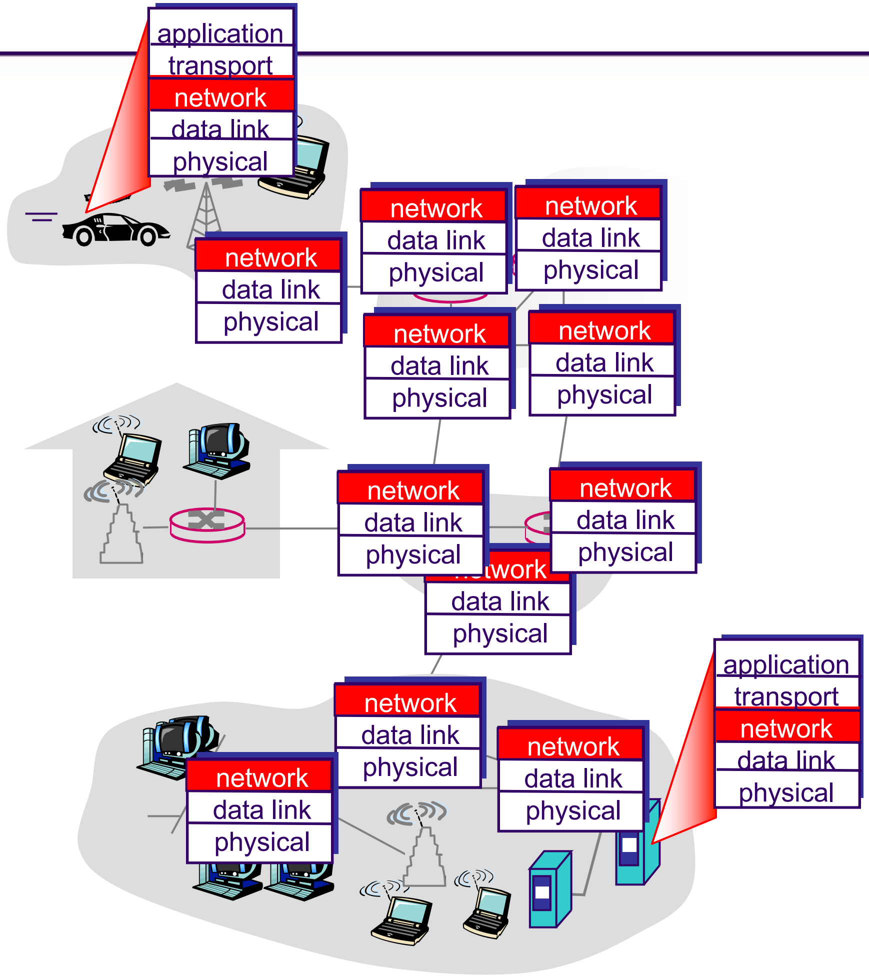
network

link

physical

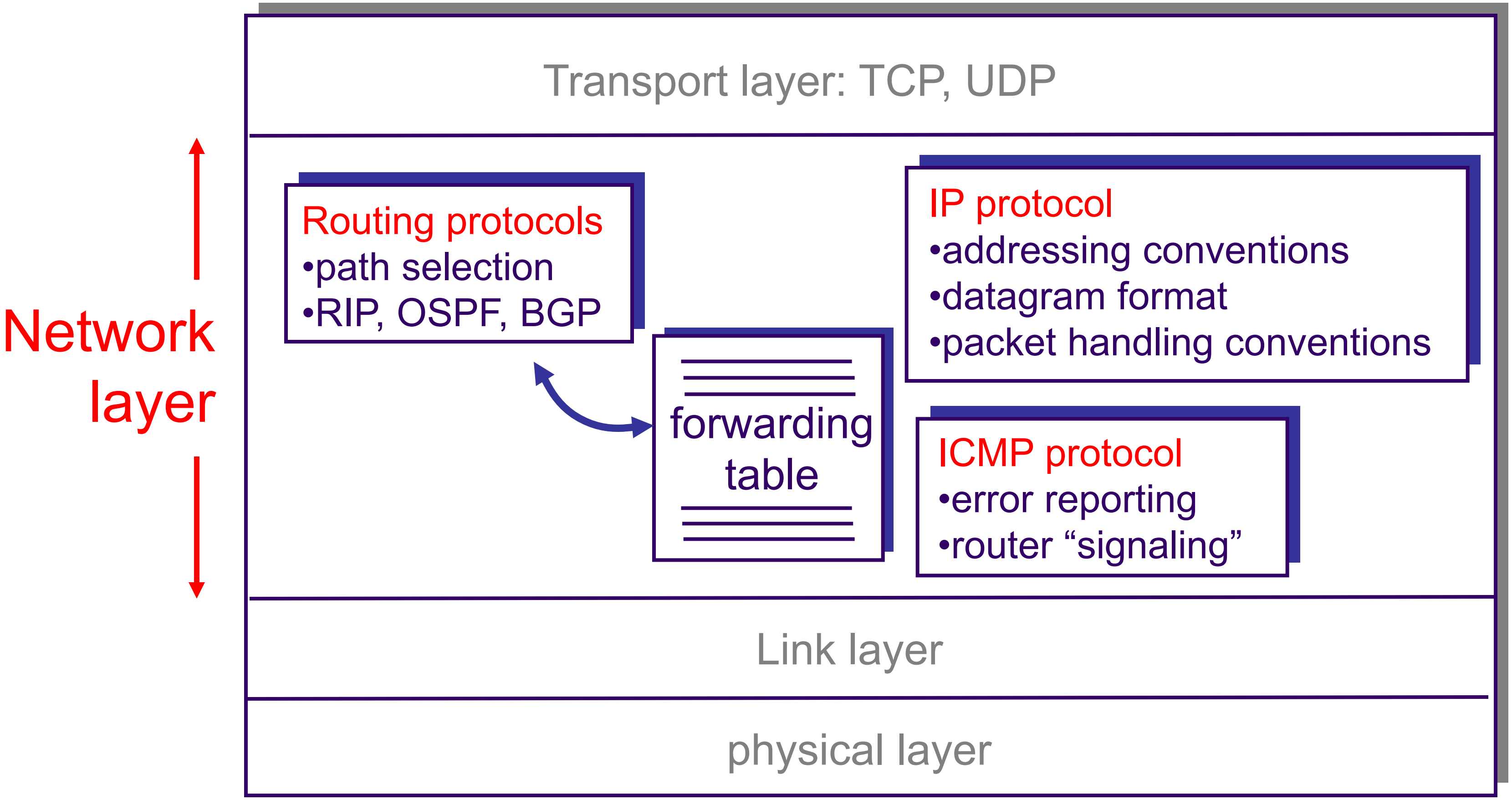
Network layer

- Transport segment from sending to receiving host
- On sending side encapsulates segments into datagrams
- On receiving side, delivers segments to transport layer
- Network layer protocols in *every* host, router
- Router examines header fields in all IP datagrams passing through it



The Internet Network Layer Components

Host, router network layer functions:



Transport layer: TCP, UDP

Network layer

Routing protocols
- path selection
- RIP, OSPF, BGP

forwarding table

IP protocol
- addressing conventions
- datagram format
- packet handling conventions

ICMP protocol
- error reporting
- router "signaling"

Link layer

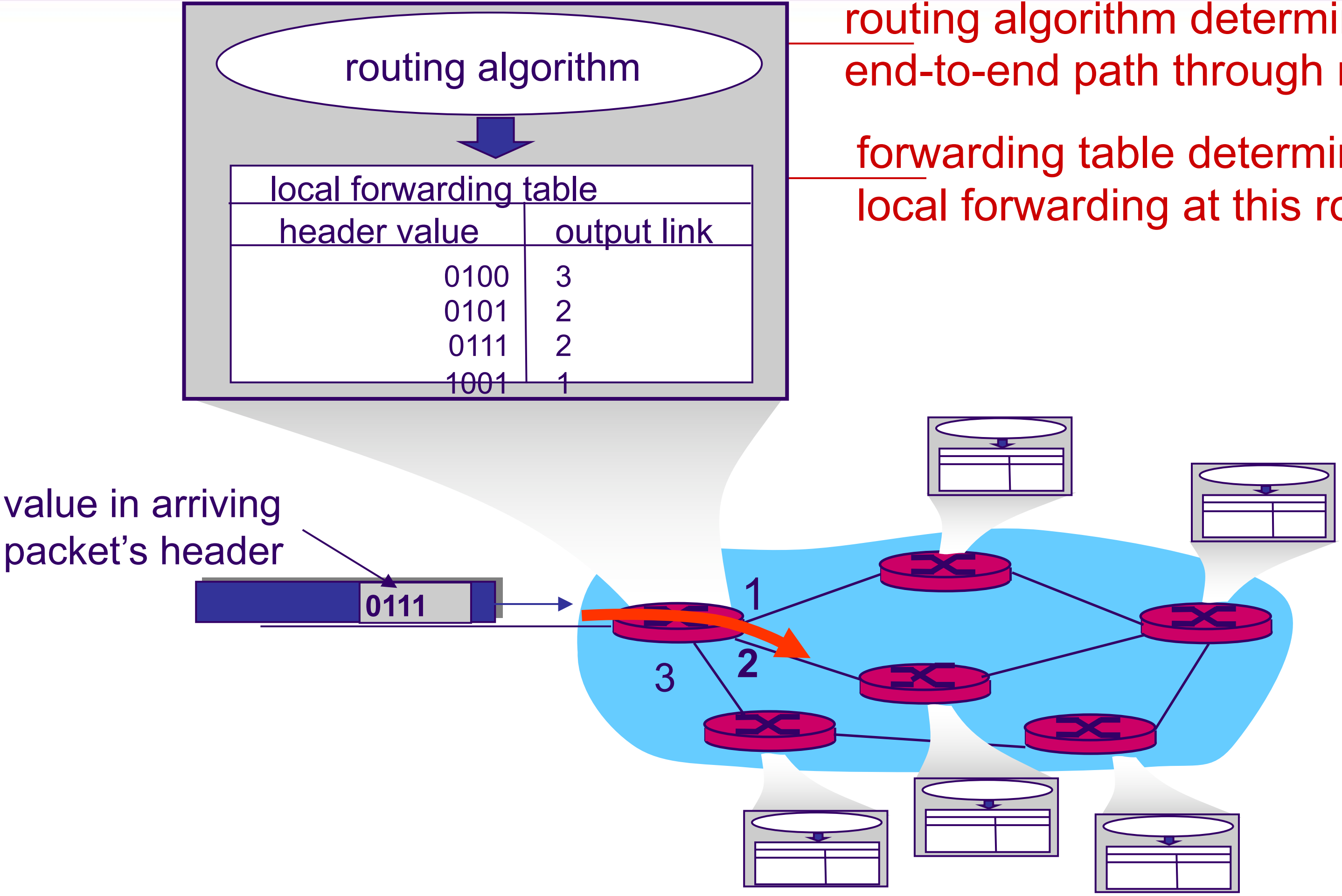
physical layer

Two Key Network Layer Functions

Two Key Network Layer Functions

- **Forwarding**: move packets from router’s input to appropriate router output
- **Routing**: determine the path taken by packets as they flow from a sender to a receiver
  - Routing algorithms – run at routers to determine “paths”;
  - Routers have a forwarding table
    - Destination address-based in Datagram networks

Routing and Forwarding



local forwarding table

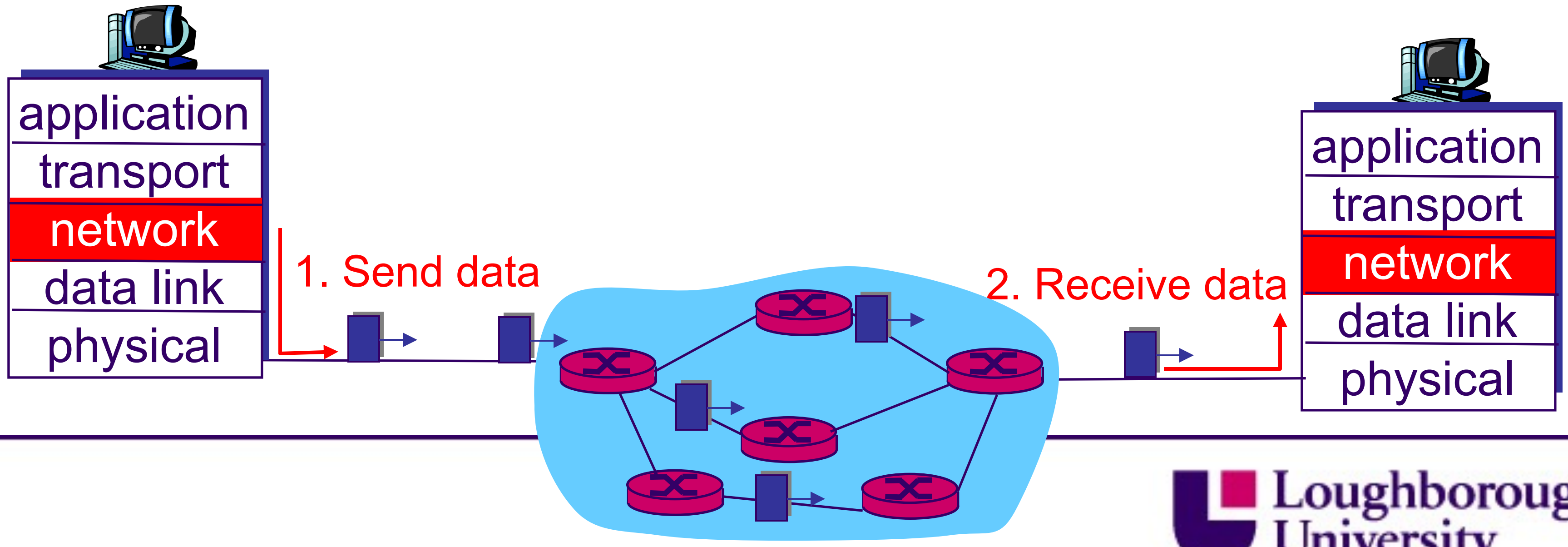
header value	output link
0100	3
0101	2
0111	2
1001	1

routing algorithm determines end-to-end path through network

forwarding table determines local forwarding at this router

Datagram networks

- 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



Network Service Model

- Internet *network-layer, however,* provides a single service (also known as "best-effort" service)

Network Architecture	Service Model	Bandwidth Guarantee	No-loss Guarantee	Ordering	Timing	Congestion Indication
Internet	Best-effort	None	None	Any order possible	Not maintained	None

Network Service Model

What are advantages to this approach?

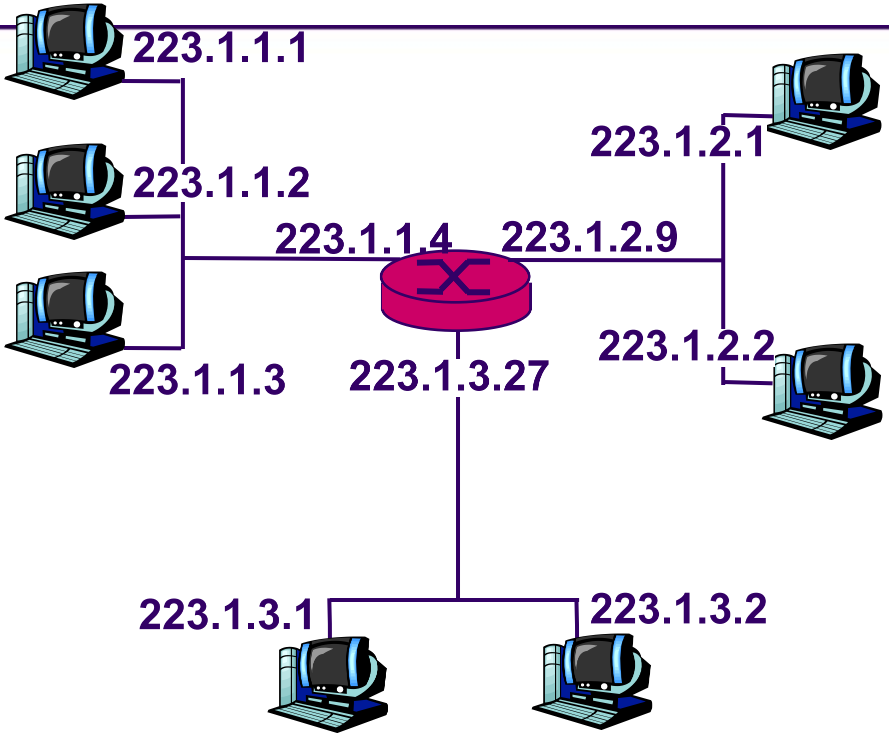
Network Service Model

There are advantages to this approach:

- Simplicity - different link-layer technologies can be interconnected more easily.
- New applications can be developed without the need to make any changes in lower layers or in the network itself.

IP Addresses

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

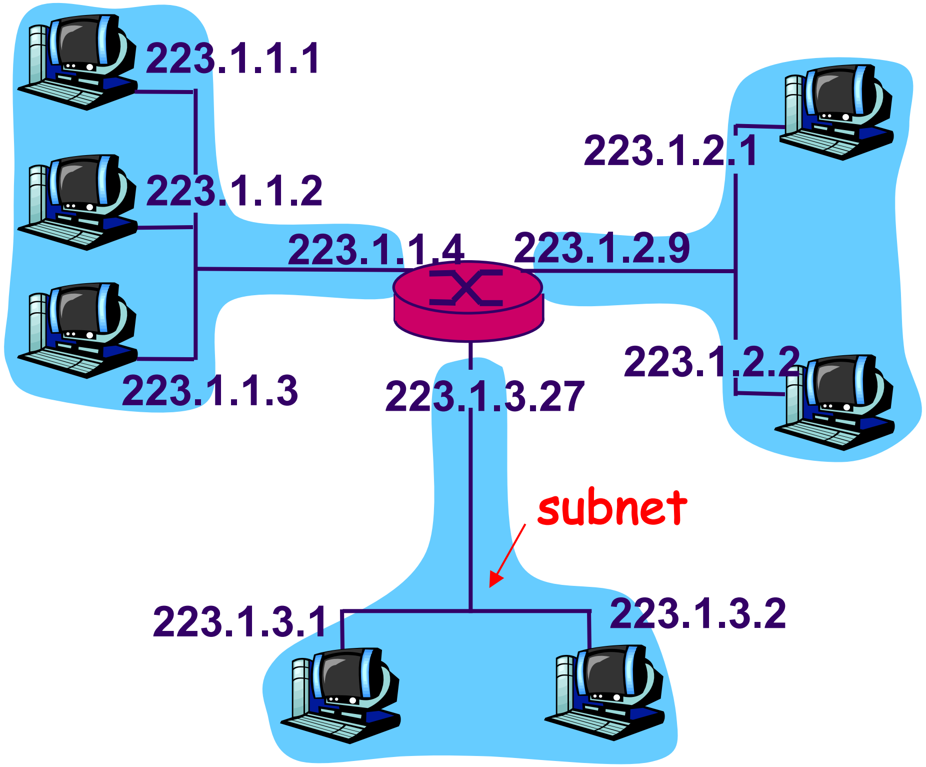


$$223.1.1.1 = \underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}$$

IP Addresses: Subnets

- IP address:
  - subnet part (high order bits)
  - host part (low order bits)
- *What's a **subnet** ?*

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

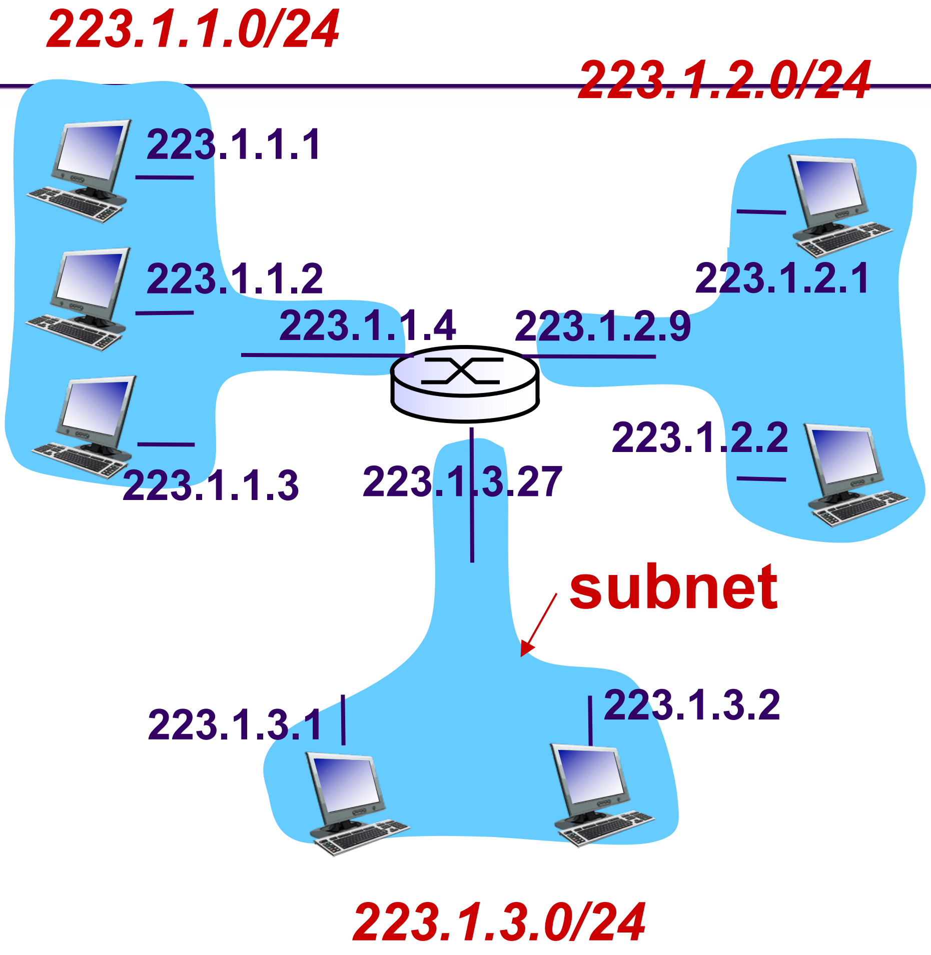


network consisting of 3 subnets

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*

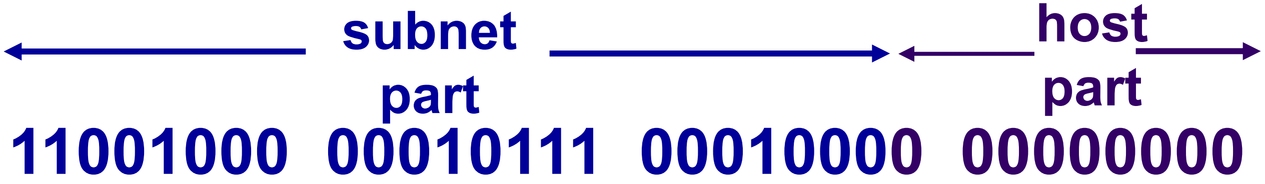


subnet mask: /24

IP addressing: CIDR

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 of address



subnet part

host part

11001000 00010111 00010000 00000000

200.23.16.0/23

Examples

- 16.2.4.6/20
  - 20 bits of subnet address
  - 12 bits (32-20=12) of host address
- 223.1.1.0/24
  - 24 bits of network address
  - 8 bits (32-24=8) of host address

*subnet and network are same meaning here!

IP Subnet Masking

- *Specifies the number of bits that make up the network-subnetwork part of the address*
- Sometimes you have a large number of IP addresses to manage
- By using subnet masking, you can break the host ID portion of the address into a subnet ID and host ID
- For example, the subnet mask 255.255.255.0 applied to a class B address will break the host ID (normally 16 bits) into an 8-bit subnet ID and an 8-bit host ID
    - 255.255.255.0=1111 1111, 1111 1111, 1111 1111, 0000 0000

IP Addresses: Subnets

- Hosts/routers in the same net/subnet have the same prefix in their IP address.
  - 128.6.101.0/24 (/24 is called a subnet mask) means all addresses which have the first 24 bits equal to 128.6.101, i.e. addresses 128.6.101.0 to 128.6.101.255. Also written 128.6.101.x or 128.6.101.xxx.

IP Addresses: Subnets

- Companies/organisations are allocated blocks of contiguous addresses with same prefix, e.g. 128.6.80.0/21.
- Originally the subnet mask for allocating addresses was restricted to 8,16,24 (whole bytes), which was wasteful
  - Companies could only obtain either 256, 65,536, or 16,777,216 addresses, when most needed a number in between.
- Now the subnet masks for allocating addresses can be arbitrary (hence less wasteful).
- Due to the rapid increase in demand for IP addresses, the IPv4 address space has been exhausted.

IP Addresses: How to get one?

Q: How does a *host* get an IP address?

IP Addresses: How to get one?

Q: How does a *host* get IP address?

- Network administrator can configure manually.
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from a server
  - “plug-and-play”

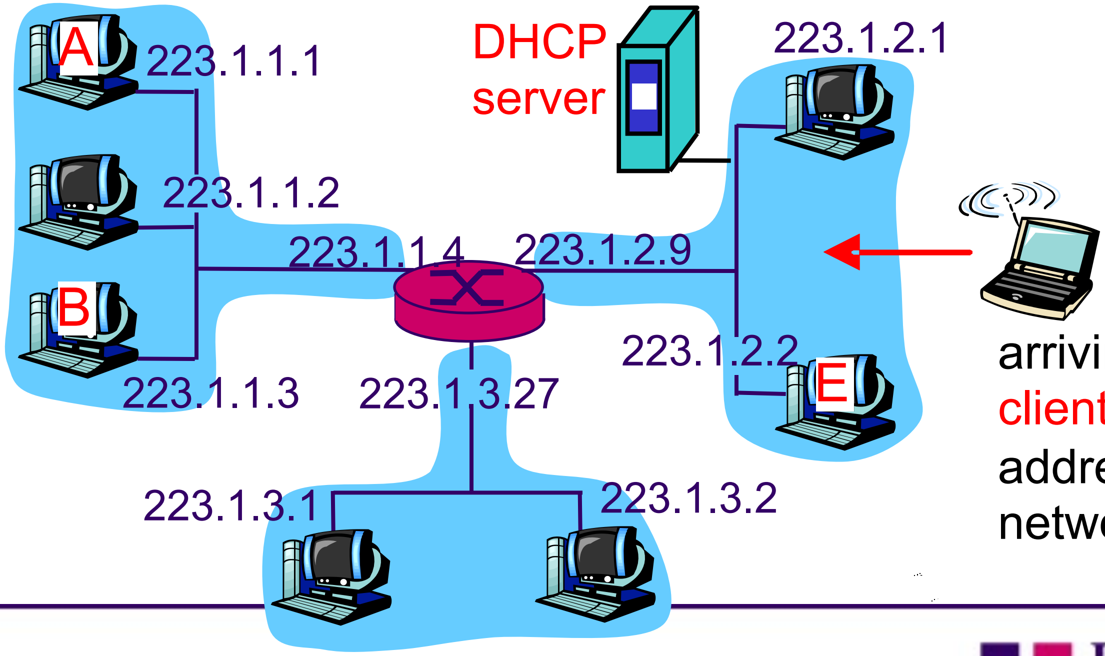
DHCP: Dynamic Host Configuration Protocol

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 and “on”).
- Support for mobile users who want to join network.

DHCP client-server scenario

- 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



arriving DHCP client needs address in this network

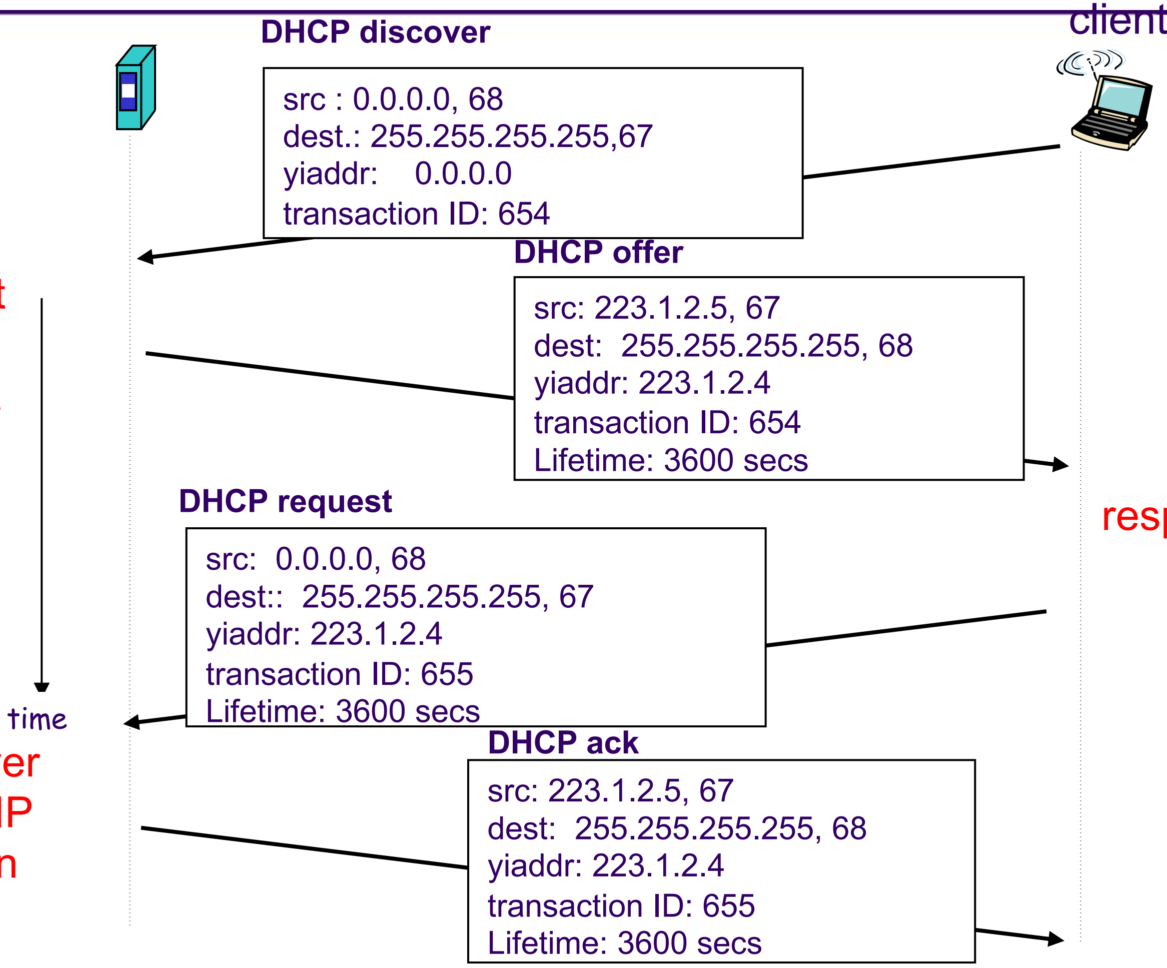
Further explanation of slide 26 figure

- DHCP is a client-server protocol
- A client is typically a newly arriving host wanting to obtain network configuration information, including an IP address for itself.
- In the simplest case, each subnet will have a DHCP server.
- If no server is present on the subnet, a DHCP relay agent (typically a router) that knows the address of a DHCP server for that network is needed.
- Slide 24 figure shows a DHCP sever attached to subnet 223.1.2/24, with the router serving as the relay agent for arriving clients attached to subnets 223.1.1/24 and 223.1.3/24.
- We assume that a DHCP server is available on the subnet in slide 26 figure.

DHCP client-server scenario

DHCP server: 223.1.2.5

arriving client



1. Client attempts to discover DHCP server

DHCP discover

src : 0.0.0.0, 68
dest.: 255.255.255.255,67
yiaddr: 0.0.0.0
transaction ID: 654

2. DHCP server responds to discovery request and reserves IP address for client (YIADDR)

DHCP offer

src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddr: 223.1.2.4
transaction ID: 654
Lifetime: 3600 secs

3. The client responds, requesting the offered IP address.

DHCP request

src: 0.0.0.0, 68
dest:: 255.255.255.255, 67
yiaddr: 223.1.2.4
transaction ID: 655
Lifetime: 3600 secs

4. The DHCP server confirms that the IP address has been reserved for the client.

DHCP ack

src: 223.1.2.5, 67
dest: 255.255.255.255, 68
yiaddr: 223.1.2.4
transaction ID: 655
Lifetime: 3600 secs

Further explanation of slide 28 figure (1)

- For a newly arriving host, the DHCP protocol is a four-step process, as shown in slide 26 figure, for the network setting shown in slide 24 figure.
- In slide 26 figure, yiaddr (as in “your Internet address”) indicates the address being allocated to the newly arriving clients.

The four steps are:

- 1) DHCP server discovery: The first task of a newly arriving host is to find a DHCP server with which to interact. This is done using a **DHCP discovery message**, which a client sends within a UDP packet to port 67.

- The UDP packet is encapsulated in an IP datagram. But to whom should this datagram be sent? The host doesn’t even know the IP address of the network to which it is attaching, much less the address of a DHCP server for this network.

- Given this, the DHCP client **broadcast** destination IP address of 255.255.255.255 and a “this host” source IP address of 0.0.0.0.

- The DHCP client passes the IP datagram to the link layer, which then broadcasts this frame to all nodes attached to the subnet.

Further explanation of slide 28 figure (2)

- 2) DHCP server offer(s): A DHCP server receiving a DHCP discovery message responds to the client with a **DHCP offer message** that is broadcast to all nodes on the subnet, again using the IP broadcast address of 255.255.255.255.
- You might think about want to think about why this server reply must also be broadcast?
- Since server DHCP servers can be present on the subnet, the client may find itself in the enviable position of being able to choose from among several offers.
- Each server offer message contains the transaction ID of the received discover message, the proposed IP address for the client, the network mask and an **IP address lease time** – the amount of time for which the IP address will be valid.
- It is common for the server to set the lease time to several hours or days

Further explanation of slide 28 figure (3)

- 3) DHCP request: The newly arriving client will choose from among one or more server offers and respond to its selected offer with a **DHCP request message**, echoing back the configuration parameters.

- 4) DHCP ACK: The server responds to the DHCP request message with a **DHCP ACK message**, confirming the requested parameters.

DHCP: more than IP address

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

- address of first-hop router (often called the default gateway) for client
- name and IP address of local DNS sever
- network mask (indicating network versus host portion of address)

IP addresses: how to get one?

Q: How does a *network* get a subnet part of IP address?

A: It gets allocated a 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 a block of addresses?

Loughborough University

IP Addressing: the last word...

Q: How does an ISP get a block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers http://www.icann.org/

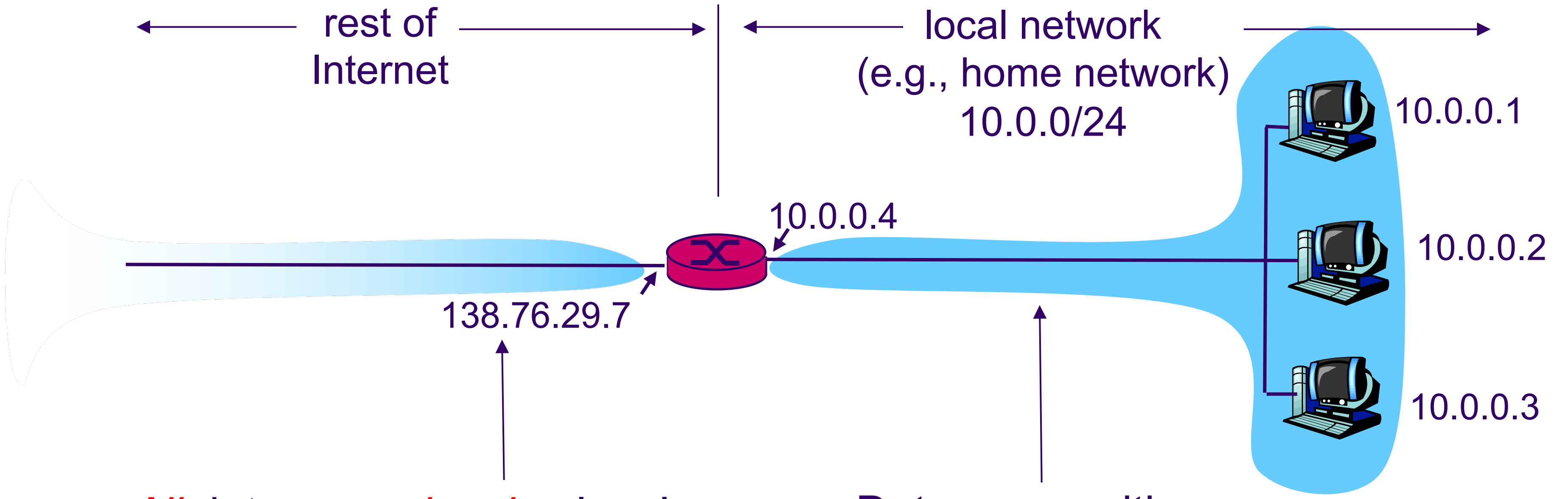
- allocates addresses
- manages DNS
- assigns domain names, resolves disputes

- Next, we will study NAT: Network Address Translation.

NAT: Network Address Translation

- Alleviates the shortage of IPv4 addresses
- Security measure – help to prevent unexpected messages reaching a device
- Several devices on a local network (e.g. home network) share one IP address
- The NAT-enabled router appears as one single host to the outside world.
- IP addresses of the form 10.0.0.0/8 , 172.16.0.0/12, 192.168.0.0/16 are reserved for private networks; hosts in the private network are assigned addresses in this domain and the NAT router translates them, using port numbers to distinguish them.

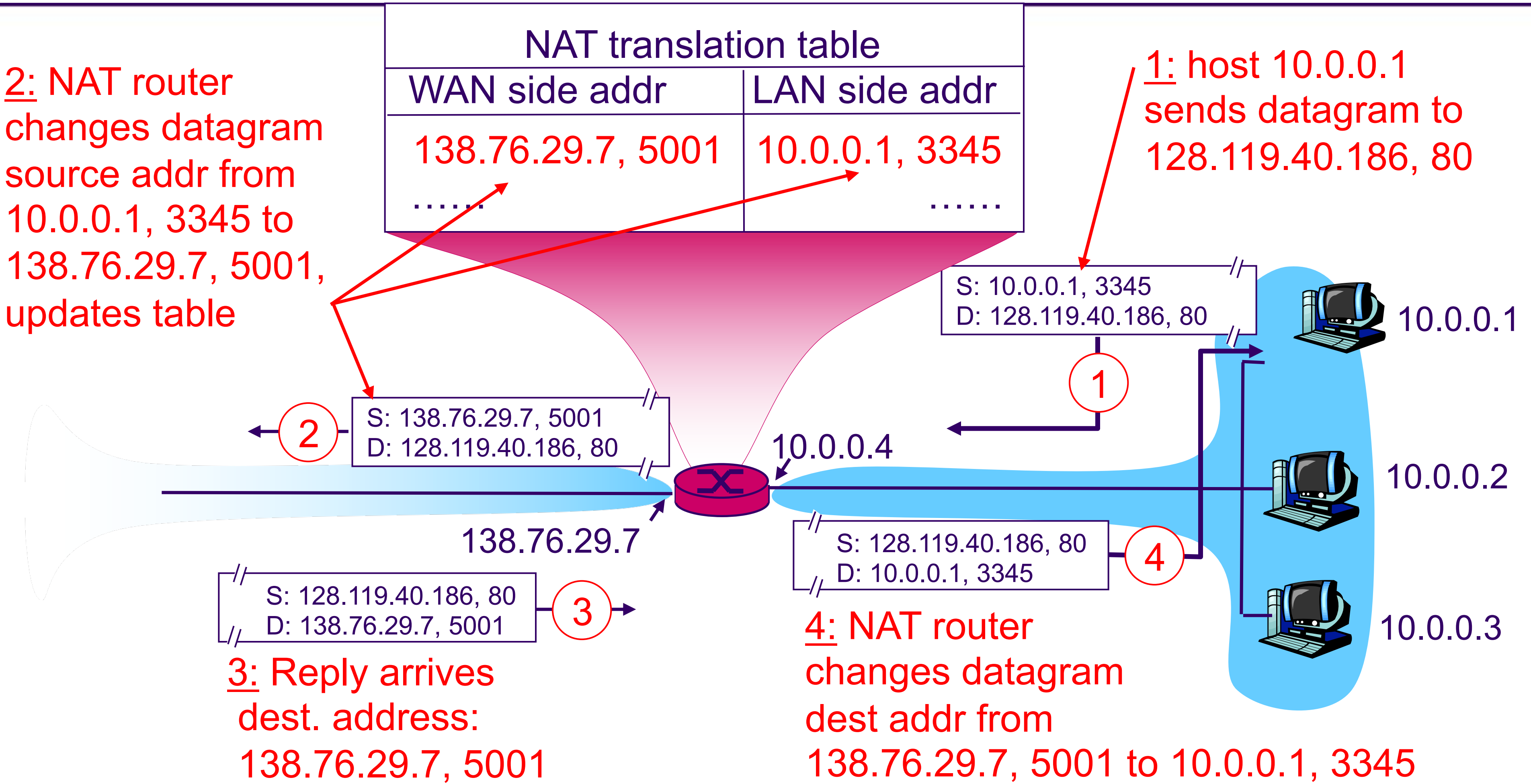
NAT: Network Address Translation



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
……	……

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80

2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

3: Reply arrives dest. address: 138.76.29.7, 5001

4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

Further explanation of slide 39 figure

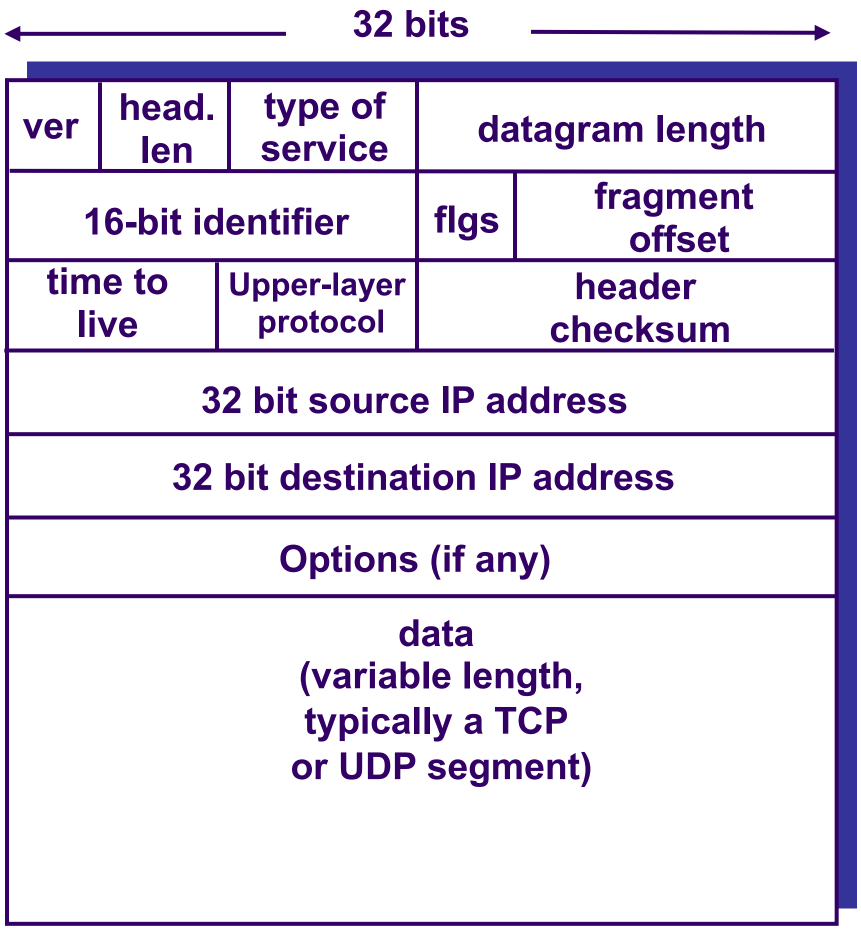
Consider the example in slide 39 figure:

- Suppose a user sitting in a home network behind host 10.0.0.1 request a Web page on some Web server (port 80) with IP address 128.119.40.186.
- The host 10.0.0.1 assigns the source port no. 3345 and sends the datagram into the LAN.
- The NAT router receives the datagram, generates a new source port no. 5001 for the datagram, replaces the source IP address with its WAN-side IP address 138.76.29.7, and replaces the original source port no. 3345 with the new source port no. 5001.
- When generating a new source port no., the NAT router can select any source port no. that is not currently in the NAT translation table.
- NAT in the router also adds an entry to its NAT translation table.
- The Web server blissfully unaware that the arriving datagram containing the HTTP request has been manipulated by the NAT router, responds with a datagram whose destination address is the IP address of the NAT router, and whose destination port no. is 5001.
- When this datagram arrives at the NAT router, the router indexes the NAT translation table using the destination IP address and destination port no. to obtain the appropriate IP address (10.0.0.1) and destination port no. (3345) for the browser in the home network.
- The router then rewrites the datagram’s destination address and destination port. no., and forwards the datagram into the home network.

NAT: Network Address Translation

- 16-bit port-number field:
  - NAT can support over 60,000 simultaneous connections with a single WAN-side IP address for the router !
- NAT is controversial:
  - routers should only process up to layer 3
  - violates end-to-end argument
    - NAT possibility must be taken into account by application designers, e.g, P2P applications
  - address shortage should instead be solved by IPv6

IP Datagram Structure (IPv4)



32 bits

ver

head. len

type of service

datagram length

16-bit identifier

flgs

fragment offset

time to live

Upper-layer protocol

header checksum

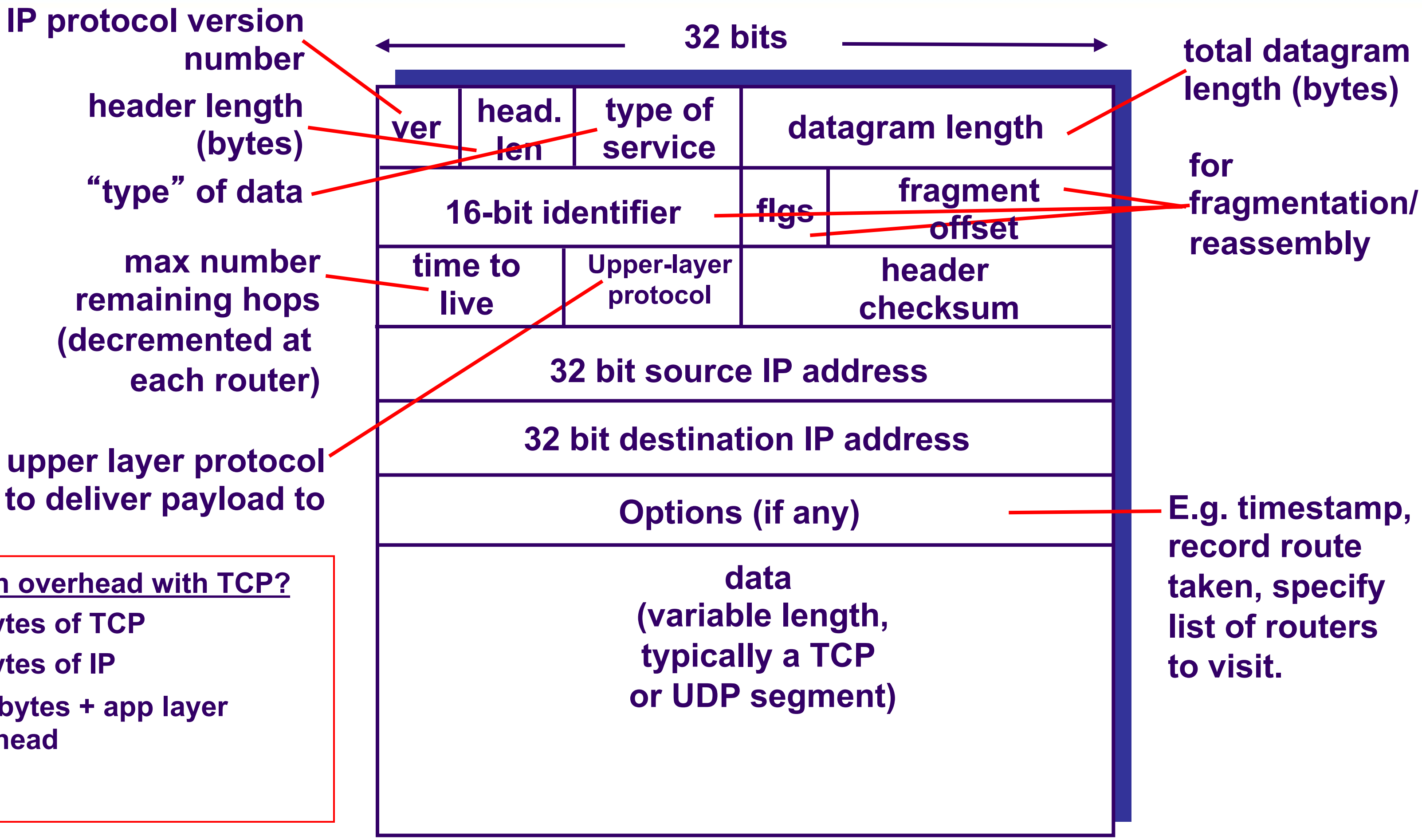
32 bit source IP address

32 bit destination IP address

Options (if any)

data
(variable length,
typically a TCP
or UDP segment)

IP Datagram Structure (IPv4)



32 bits

ver	head. len	type of service	datagram length
16-bit identifier		flgs	fragment offset
time to live	Upper-layer protocol	header checksum	
32 bit source IP address			
32 bit destination IP address			
Options (if any)			
data (variable length, typically a TCP or UDP segment)			

IP protocol version number

header length (bytes)

"type" of data

max number remaining hops (decremented at each router)

upper layer protocol to deliver payload to

total datagram length (bytes)

for fragmentation/ reassembly

E.g. timestamp, record route taken, specify list of routers to visit.

how much overhead with TCP?

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

IP datagram structure (IPv4)

- **Version (4 bits)** = 4 for IPv4
- **Header length (4 bits)**: length of header, in multiples of 32-bit words; typically header length = 5 (i.e. 20bytes).
- **Type of service**: for differentiated service,
  - It might be useful to distinguish real-time datagrams (e.g. IP telephony application) from non-real-time traffic (e.g. FTP)
- **Datagram length (16 bits)**: length of datagram, in bytes; usually up to 1500B.
- **Identifier, flags, fragmentation offset**: to be used if datagram needs to be fragmented at a router due to link layer protocol unable to transport a large enough frame.

IP datagram structure (IPv4)

- **Time-to-live**: number of hops that the datagram is allowed to travel (decremented by 1 at each router; when it reaches 0, it is dropped). Makes sure datagrams do not survive indefinitely, e.g. due to loops in routing.
- **Upper layer protocol**: 6 for TCP, 17 for UDP, 1 for ICMP; used only at destination to pass it to the correct upper layer protocol.
- **Header checksum**: The header checksum aids a router by detecting bit errors in the received IP datagram.
  - Computed by treating each 2 bytes in the header as a number and summing these using 1s complement arithmetic. If errors detected, datagram is dropped.
  - Checksum has to be recomputed at each router as header changes (e.g. TTL decrement).

IP datagram structure (IPv4)

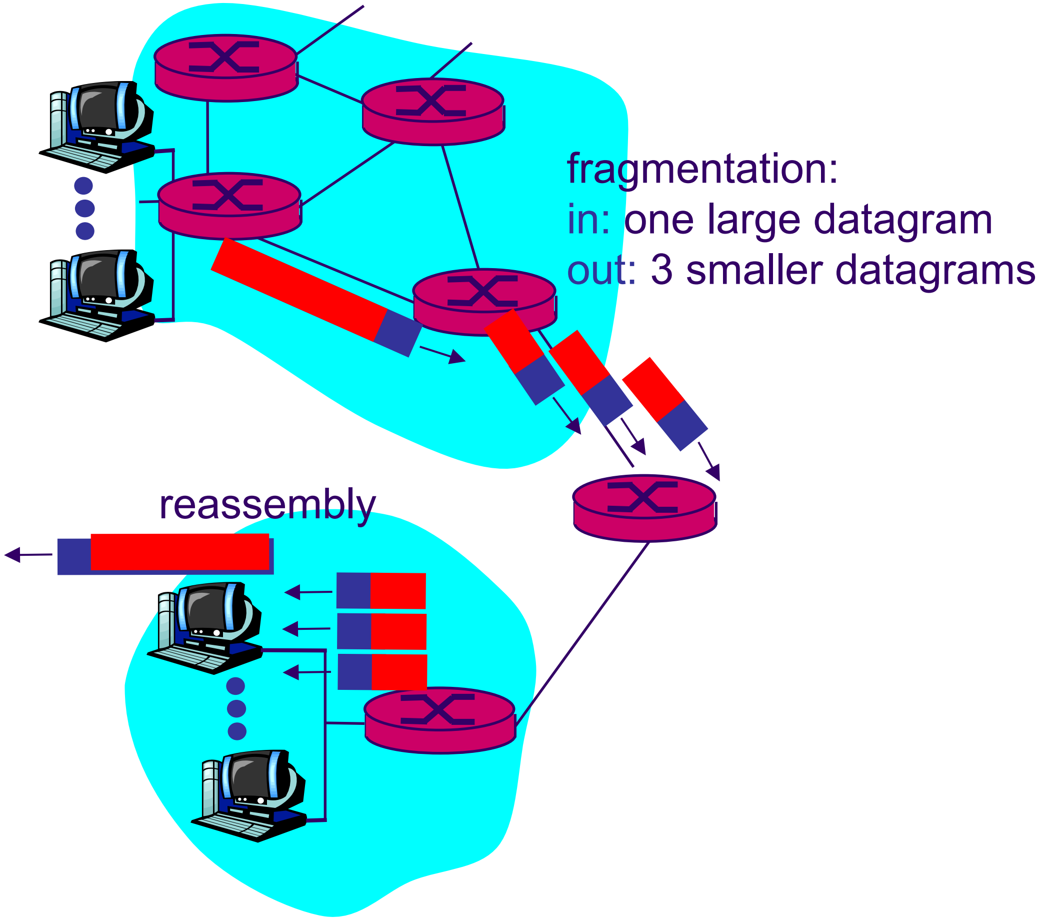
- **Source and destination IP address**: 32-bits each in IPv4; used to identify the hosts.
  - 4,294,967,296 (2^{32}) addresses.
- **Options**
  - timestamp, record route taken, specify list of routers to visit.
- **Data**: can carry TCP segment, UDP datagram, ICMP information.

IPv4: MTU & MSS

- **Maximum size of IP datagram:** theoretically 65535B; In practice, it needs to fit in the link layer frame.
- **Maximum Transmission Unit (MTU):** maximum size of data (payload) that can fit in the link layer frame. For Ethernet 1500B, so IP datagram length at most 1500.
  - If an IP datagram does not fit in the link-layer frame of a particular link, it needs to be fragmented.
  - Even if it fits in the frame on the first link, it may not fit on some other link on the route; routers perform segmentation; a flag DF (don't fragment) prevents fragmenting if set.
- **Maximum Segment Size (MSS):** Determined by the length of the link-layer frame, minus the TCP/IP header length.
  - IP header: 20B, TCP header: 20B
  - MSS for TCP: 1460B = (1500-20-20)

IP Fragmentation & Reassembly

- Network links have MTU (max. transfer size) - largest possible link-level frame.
  - different link types, different MTUs
- large IP datagram divided (“fragmented”) within net
  - one datagram becomes several datagrams
  - “reassembled” only at final destination
  - IP header bits used to identify, order related fragments



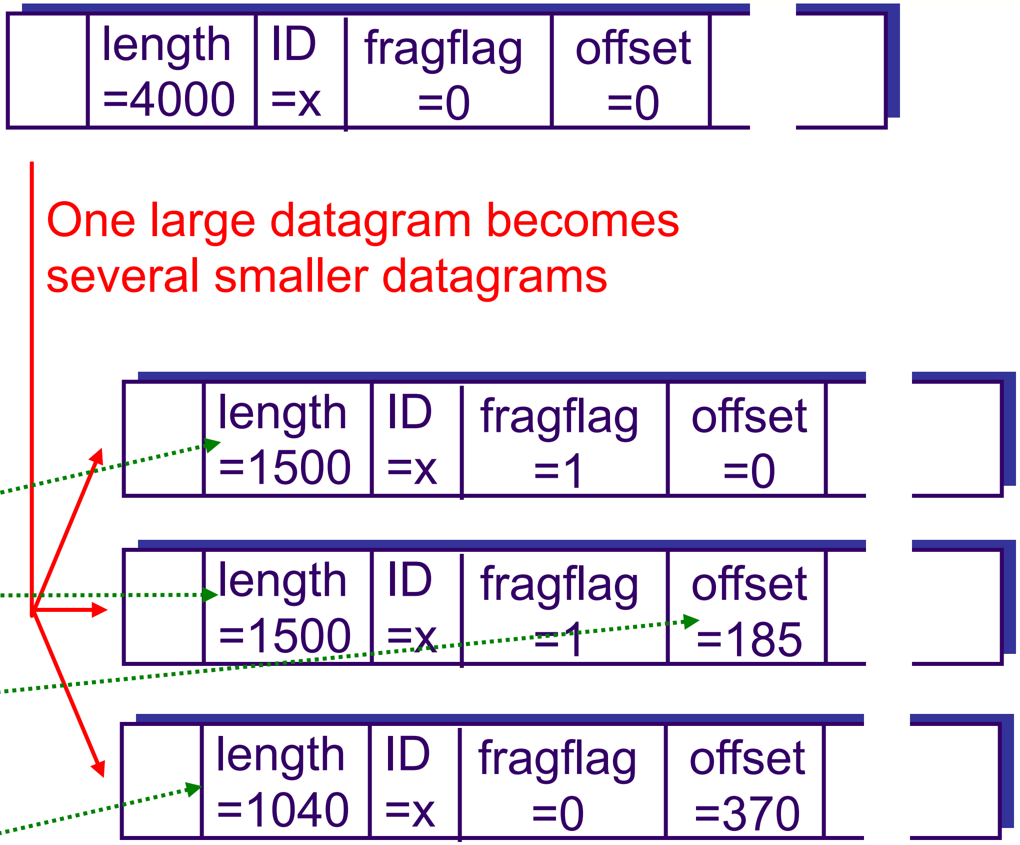
fragmentation:
in: one large datagram
out: 3 smaller datagrams

reassembly

IP Fragmentation and Reassembly

Example

- 4000 bytes datagram (20 bytes IP header + 3980 bytes IP payload)
- MTU = 1500 bytes



length	ID	fragflag	offset
=4000	=x	=0	=0

One large datagram becomes several smaller datagrams

length	ID	fragflag	offset
=1500	=x	=1	=0
=1500	=x	=1	=185
=1040	=x	=0	=370

1480 bytes in data field

offset = 185 meaning the data should be inserted beginning at byte 1480 (note 185*8=1,480)

1020 bytes in data field (=3980-1480-1480)

*No requirement to do calculation, understand fragmentation idea

Loughborough University

ICMP: Internet Control Message Protocol

Internet Control Message Protocol (RFC 792)

- Part of the Internet Protocol Suite.
- Primarily used to report errors.
- ICMP information is transported in IP datagrams, identified by protocol value = 1.

ICMP: Internet Control Message Protocol

Typical messages:

- Host unreachable (e.g. router interface is down)
- Port unreachable (e.g. port 80 but no webserver running)
- TTL expired (used in traceroute)
- Echo request/reply (ping)
- Fragmentation needed but “Don’t Fragment” flag was set.

ICMP message also contains the header and first 8 bytes (source and destination port number for TCP or UDP) of the IP packet that caused the problem.

ICMP message types

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Example usage of ICMP: Traceroute

- Source sends series of UDP segments to destination, incrementing the TTL each time.
  - First has TTL=1, Second has TTL=2, etc.
  - Unlikely port number.
- When nth datagram arrives to nth router:
  - Router discards datagram
  - Router sends an ICMP message to source (type 11, code 0) – TTL expired
  - Message includes name of router & IP address

- When ICMP message arrives at the source it calculates RTT
- Traceroute does this 3 times for each TTL.

Stopping criterion

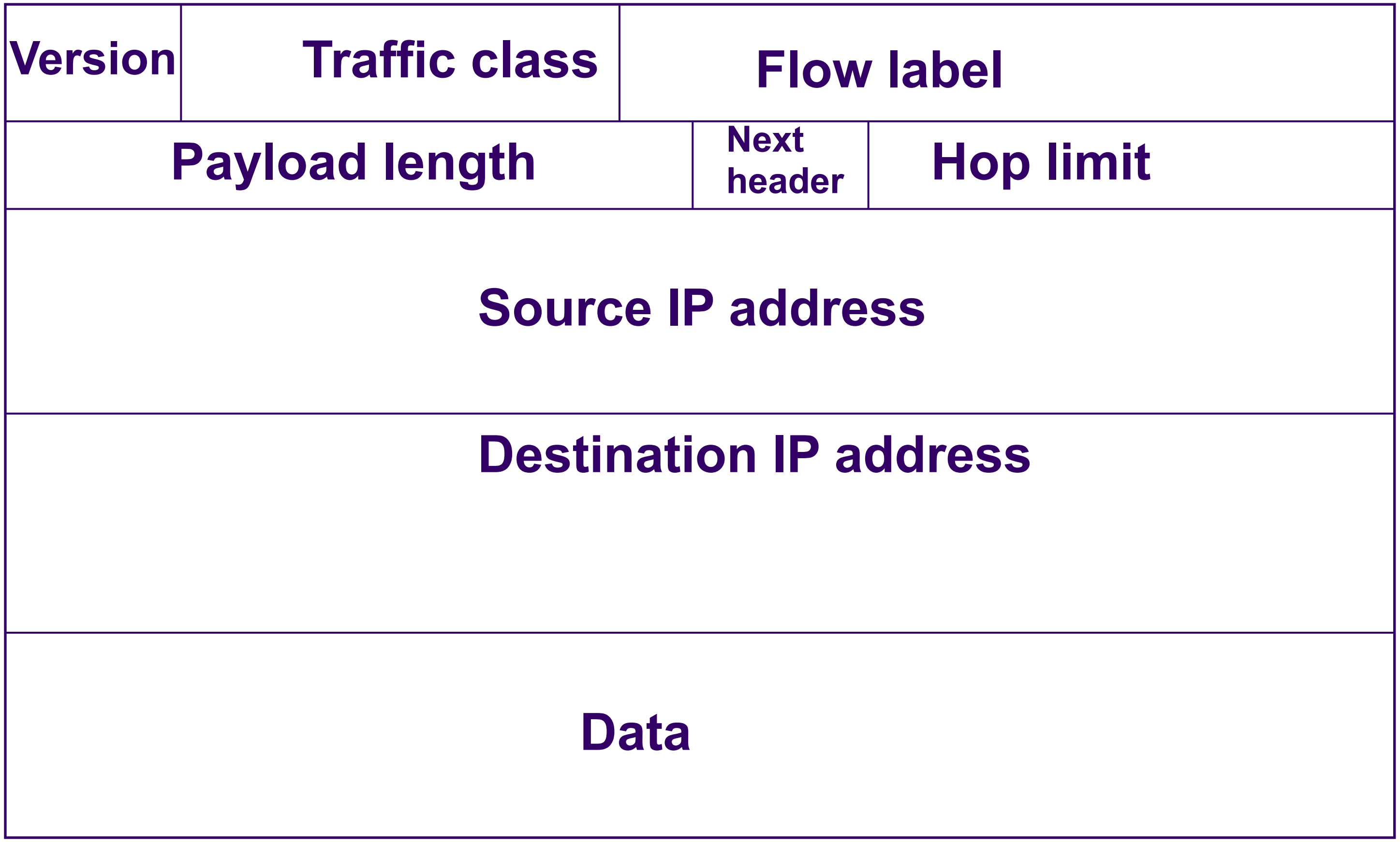
- UDP segment eventually arrives at destination host
- Destination returns ICMP “port unreachable” packet (type 3, code 3)
- When source gets this ICMP, stops.

*** Self-interested study slide**

IPv6

- Initial motivation: 32-bit IPv4 address exhaustion.
- Additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS
    - Quality of service (QoS) is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow.

IP datagram structure (IPv6)



Version

Traffic class

Flow label

Payload length

Next header

Hop limit

Source IP address

Destination IP address

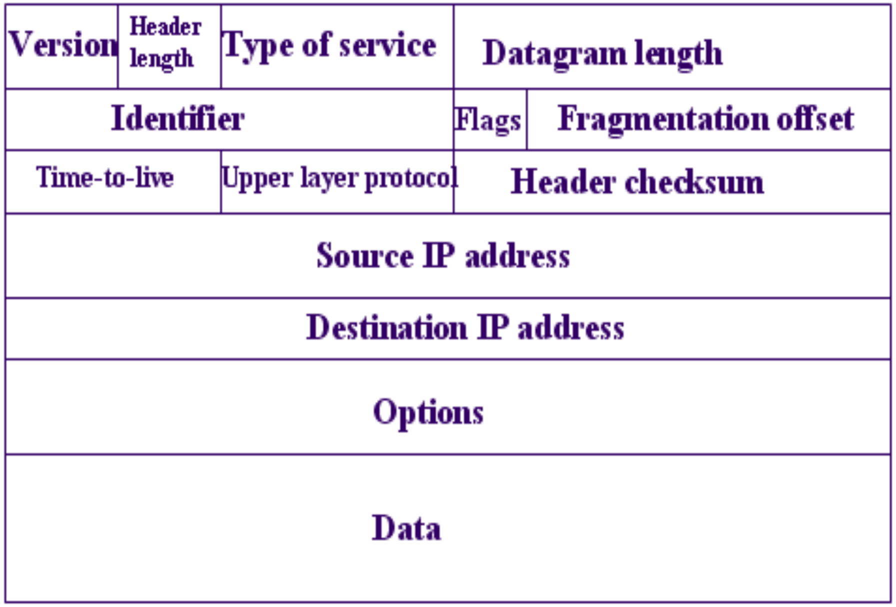
Data

IP datagram structure (IPv6)

- **Version** (4 bits) = 6 for IPv6
- **Traffic class:** similar to “Type of service” in IPv4
  - It can be used to give priority to certain within a flow.
- **Flow label:** labelling flow from certain applications for differentiated handling
  - E.g. audio and video transmission might likely be treated as a flow.
- **Payload length** (16 bits)
- **Next header:** same as “Upper layer protocol” in IPv4; can also be used for signalling the presence of the equivalent of the IPv4 “Options” part of the header after the usual IPv6 header.
- **Hop limit:** same as “time-to-live” in IPv4
- **Source and destination IP address:** 128 bits each – maximum of approximately 3.4x10^{38} addresses.
  - A new type of address, **Anycast address** which allows a datagram to be delivered to any one of a group of hosts.
  - E.g. To send an HTTP GET to the nearest of a number of mirror sites that contain a given document.

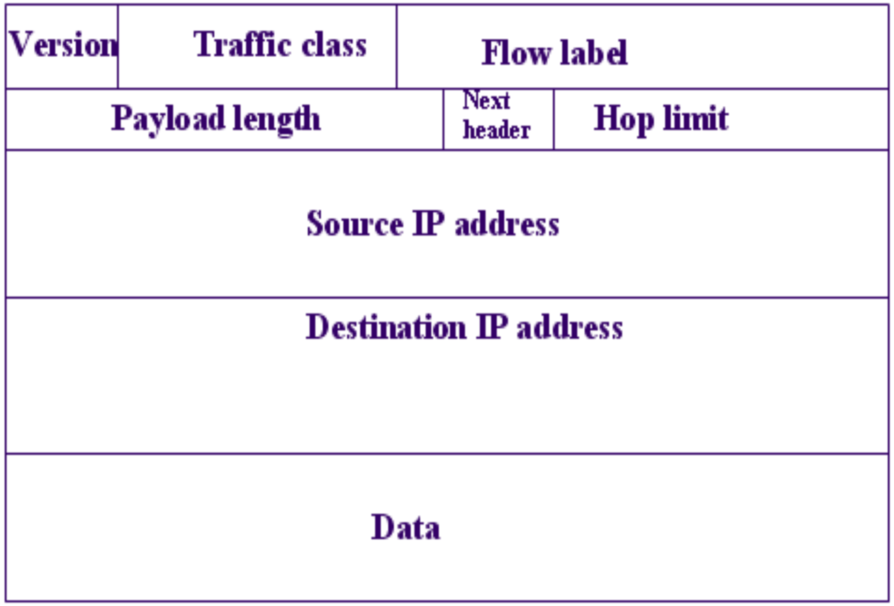
Spot the difference

IPv4



<table>
<tr><td>Version</td><td>Header length</td><td>Type of service</td><td colspan="2">Datagram length</td></tr>
<tr><td colspan="3">Identifier</td><td>Flags</td><td>Fragmentation offset</td></tr>
<tr><td colspan="2">Time-to-live</td><td>Upper layer protocol</td><td colspan="2">Header checksum</td></tr>
<tr><td colspan="5">Source IP address</td></tr>
<tr><td colspan="5">Destination IP address</td></tr>
<tr><td colspan="5">Options</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>

IPv6



<table>
<tr><td>Version</td><td>Traffic class</td><td colspan="3">Flow label</td></tr>
<tr><td colspan="3">Payload length</td><td>Next header</td><td>Hop limit</td></tr>
<tr><td colspan="5">Source IP address</td></tr>
<tr><td colspan="5">Destination IP address</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>

IPv6 vs. IPv4

- **Main changes** in IPv6 compared to IPv4:
  - IP address has 128 bits (the 32-bit addresses exhaustion)
  - Fixed length header: 40 bytes (however options can be included using a next header field)
  - No fragmentation allowed: if datagram is too big for a link, it is dropped and an ICMP message sent to sender (saves time at routers)
  - No header checksum (saves time at routers)

IPv6 vs. IPv4

- *ICMPv6:* new version of ICMP
  - additional message types, e.g. "Packet Too Big"
  - multicast group management functions

Transition From IPv4 To IPv6

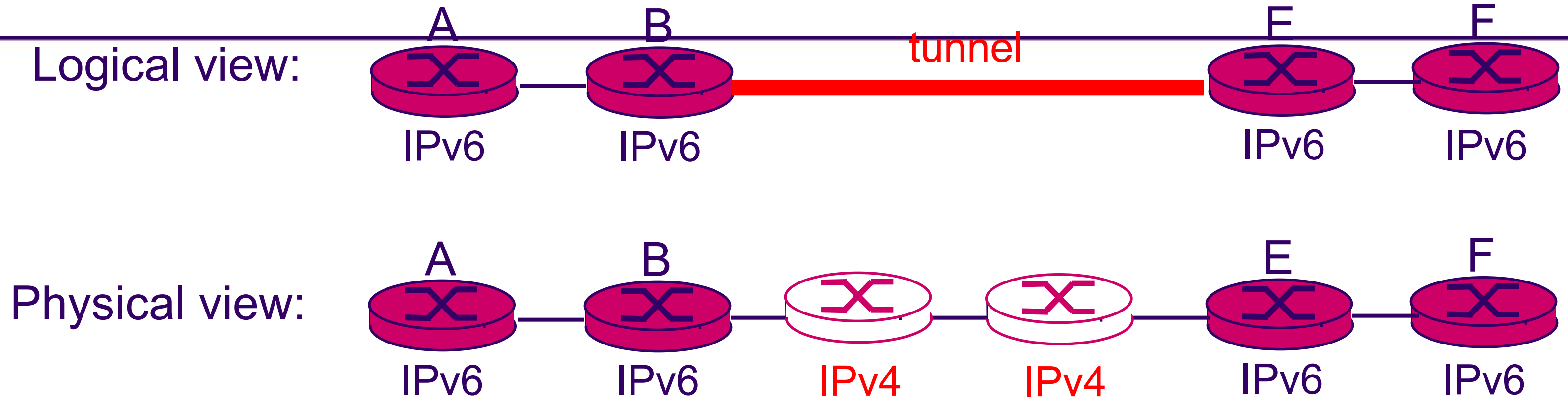
- Not all routers can be upgraded simultaneously
  - [Q] How will the network operate with mixed IPv4 and IPv6 routers?

Transition From IPv4 To IPv6

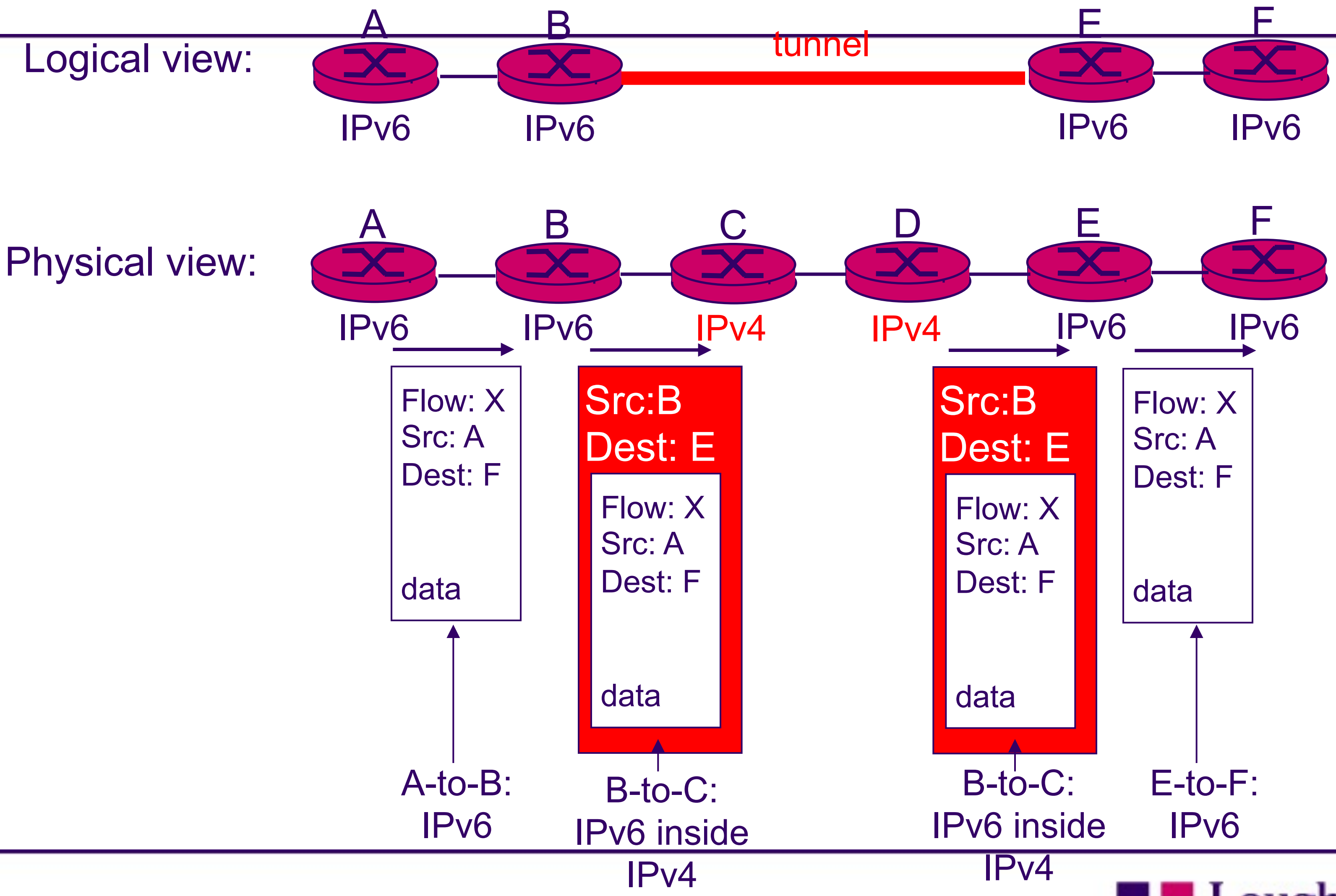
- Not all routers can be upgraded simultaneously
  - [Q] How will the network operate with mixed IPv4 and IPv6 routers?

- *Tunneling:* IPv6 carried as payload in IPv4 datagram among IPv4 routers

Tunneling


Logical view:
A
B
tunnel
E
F
IPv6
IPv6
IPv6
IPv6
Physical view:
A
B
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6

Tunneling



Logical view:

A B E F

tunnel

IPv6 IPv6 IPv6 IPv6

Physical view:

A B C D E F

IPv6 IPv6 IPv4 IPv4 IPv6 IPv6

Flow: X
Src: A
Dest: F

data

Src:B
Dest: E

Flow: X
Src: A
Dest: F

data

Src:B
Dest: E

Flow: X
Src: A
Dest: F

data

Flow: X
Src: A
Dest: F

data

A-to-B:
IPv6

B-to-C:
IPv6 inside
IPv4

B-to-C:
IPv6 inside
IPv4

E-to-F:
IPv6

Further explanation of slide 63 figure

We refer to the intervening set of IPv4 routers between two IPv6 routers as **a tunnel**:

- With tunneling, **the IPv6 node on the sending side of the tunnel (in this example, B) takes the entire IPv6 datagram and puts it in the data (payload) field of an IPv4 datagram.**
- This IPv4 datagram is then addressed to the IPv6 node on the receiving side of the tunnel (in this example, E) and sent to the first node in the tunnel (in this example, C).
- The intervening IPv4 router in the tunnel route this IPv4 datagram among themselves, just as they would any other datagram, blissfully unaware that the IPv4 datagram itself contains a complete IPv6 datagram.
- The IPv6 node on the receiving side of the tunnel eventually receives the IPv4 datagram determines that the IPv4 datagram contains an IPv6 datagram, extracts the IPv6 datagram, and then routes the IPv6 datagram exactly as it would if it had received the IPv6 datagram from a directly connected IPv6 neighbor.

Transition From IPv4 To IPv6

- Some initiatives trying to test the water for the introduction of IPv6. For example “World IPv6 Day” (2011) & “World IPv6 Launch Day” (2012) :
  - On 8 June, 2011, websites and Internet service providers, including Google, Facebook, Yahoo!, participated a global trial of IPv6.
  - The event provided an opportunity for participants to test their IPv6 readiness.
- Still a lot of work to do.

Question

- What percentage of Internet traffic is IPv6?

Question

What percentage of Internet traffic is IPv6?

- Google's statistics March 2022: the IPv6 adoption rate globally is around 34%. Adoption is uneven across countries and Internet service providers.

- Worldwide IPv6 Adoption Visualization

https://www.akamai.com/visualizations/state-of-the-internet-report/ipv6-adoption-visualization

Statistics

- Google collects statistics about IPv6 adoption in the Internet on an ongoing basis.
  - IPv6 Adoption
  - Per-Country IPv6 adoption

http://www.google.com/intl/en/ipv6/statistics.html

- *Video: Vint Cerf explains why moving to IPv6 is so important.*
- https://www.zdnet.com/article/googles-vint-cerf-quarter-of-internet-is-ipv6-but-heres-why-thats-not-enough/

Other IP Versions

- An often-asked question is what happened to IPv5?

- In short, IPv5 never became an official protocol. Many years ago, Internet Stream Protocol (ST) was considered IP version five by industry researchers, but ST was abandoned before ever becoming a standard or widely known as IPv5.

Summary

- **IPv4 vs. IPv6**
  - Removals and additions to IP header.
  - Increases address space.
- **DHCP**
  - Used to assign IP addresses to hosts.
- **ICMP**
  - Used primarily to transmit network error messages.
- **NAT**
  - Used to translate a single IP address into many local IP addresses.

Key points this week

- IPv4 vs. IPv6 (see slides 57, 58, no need to remember IPv4 and IPv6 structure)
- DHCP (understand slides 26, 28)
- NAT (understand slides 38, 39)
- ICMP - general understanding