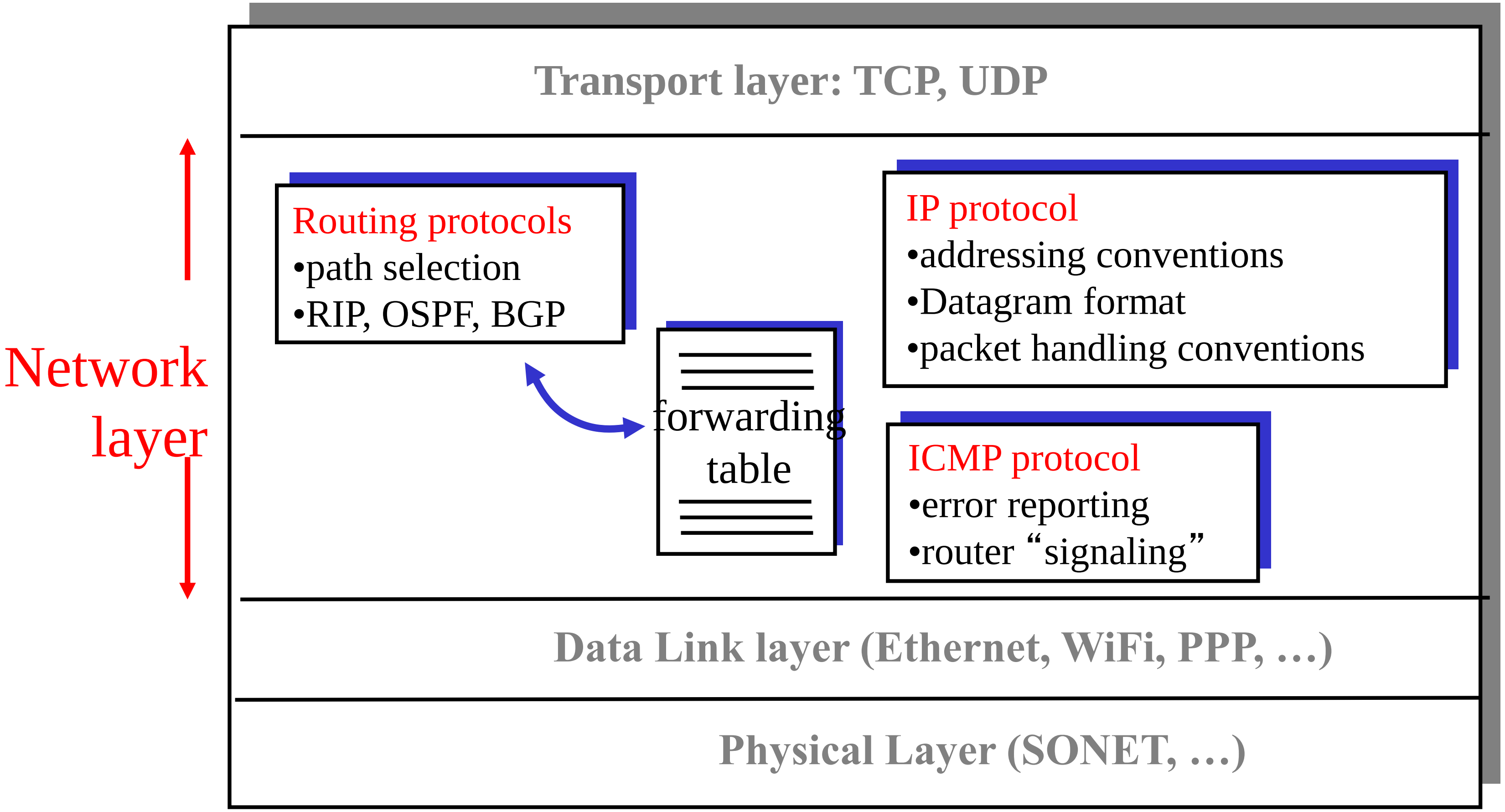
IP Forwarding & IP/ICMP Protocol



Network layer

Transport layer: TCP, UDP

Routing protocols
- path selection
- RIP, OSPF, BGP

forwarding table

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

ICMP protocol
- error reporting
- router "signaling"

Data Link layer (Ethernet, WiFi, PPP, …)

Physical Layer (SONET, …)

IP Datagram Format

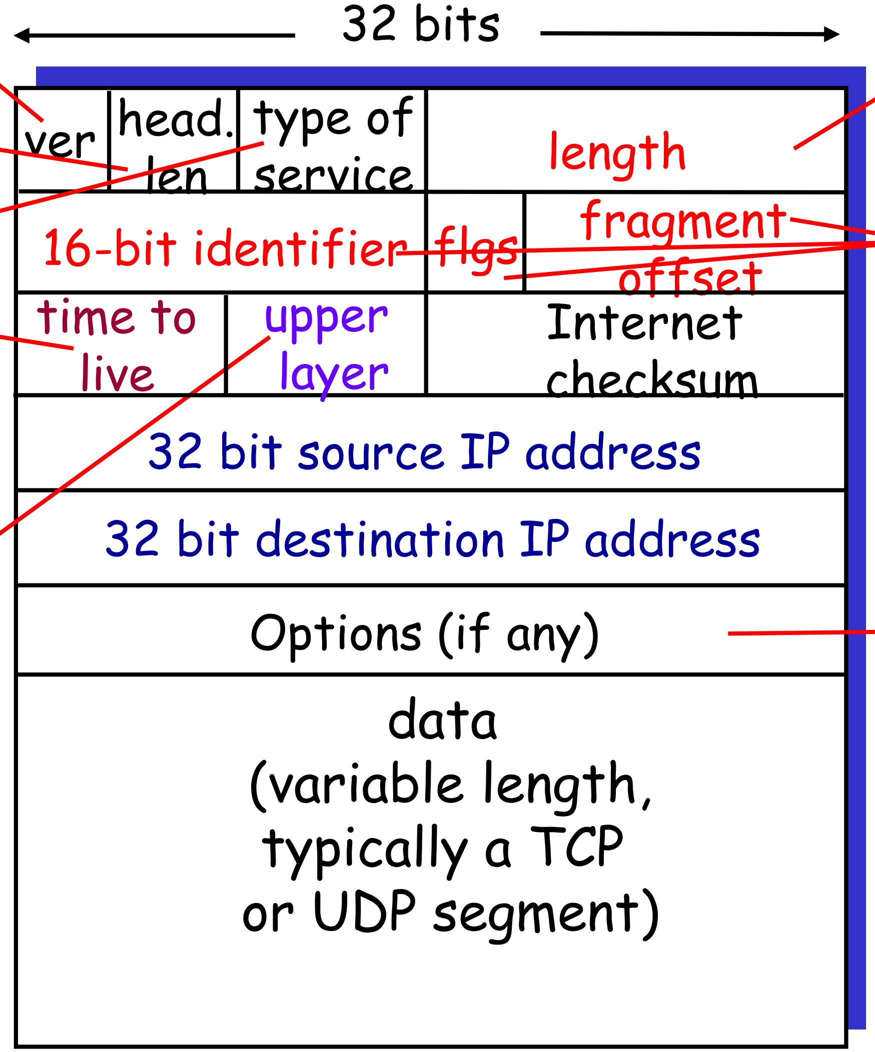
IP protocol version number

header length (bytes)

"type" of data

max number remaining hops (decremented at each router)

upper layer protocol to deliver payload to



32 bits

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

total datagram length (bytes)

for fragmentation/ reassembly

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

<u>how much overhead with TCP?</u>

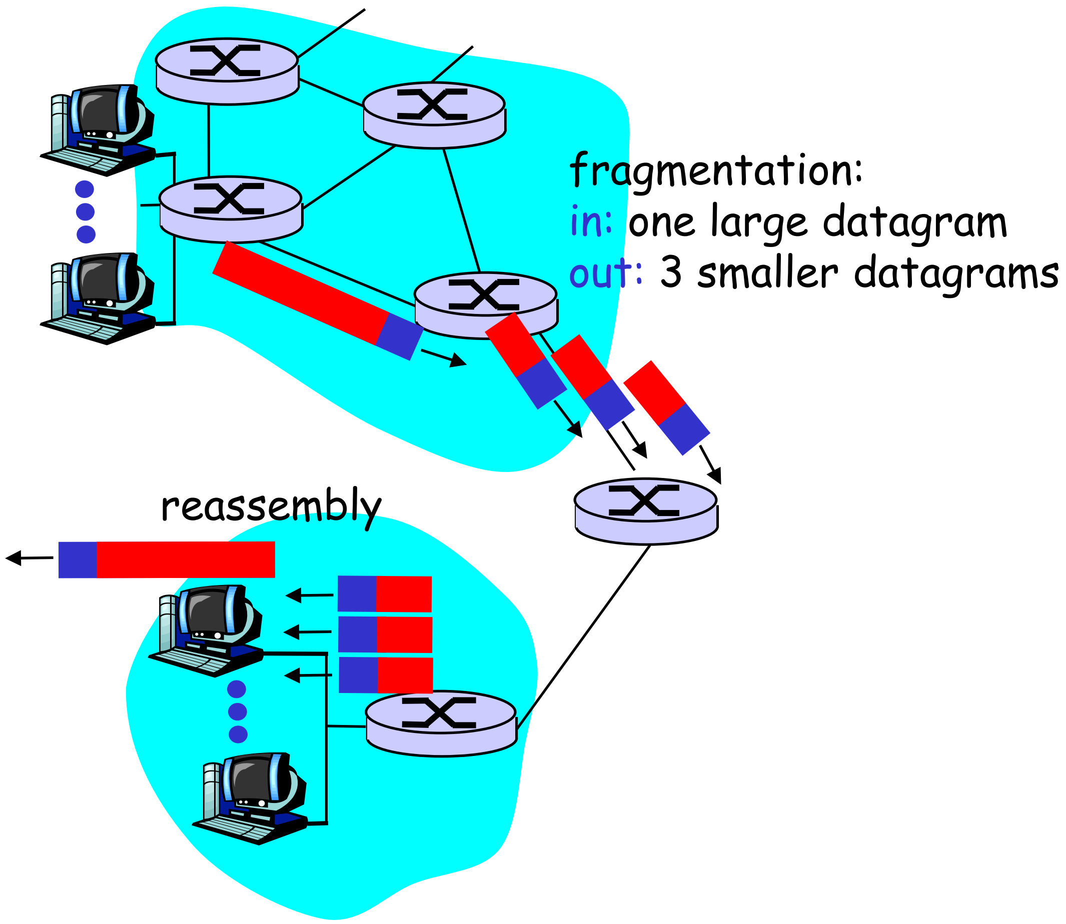
- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

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.

IP Fragmentation & Reassembly: Why

- 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



IP Fragmentation & Reassembly: How

- An IP datagram is chopped by a router into smaller pieces if
  - datagram size is greater than network MTU
  - Don't fragment option is not set
- Each datagram has unique datagram identification
  - Generated by source hosts
  - All fragments of a packet carry original datagram id
- All fragments except the last have more flag set
  - Fragment offset and Length fields are modified appropriately
- Fragments of IP packet can be further fragmented by other routers along the way to destination !
- Reassembly only done at destination host (why?)
  - Use IP datagram id, fragment offset, fragment flags. Length
  - A timer is set when first fragment is received (why?)

IP Fragmentation and Reassembly: Exp

Example

- 4000 byte datagram
- MTU = 1500 bytes

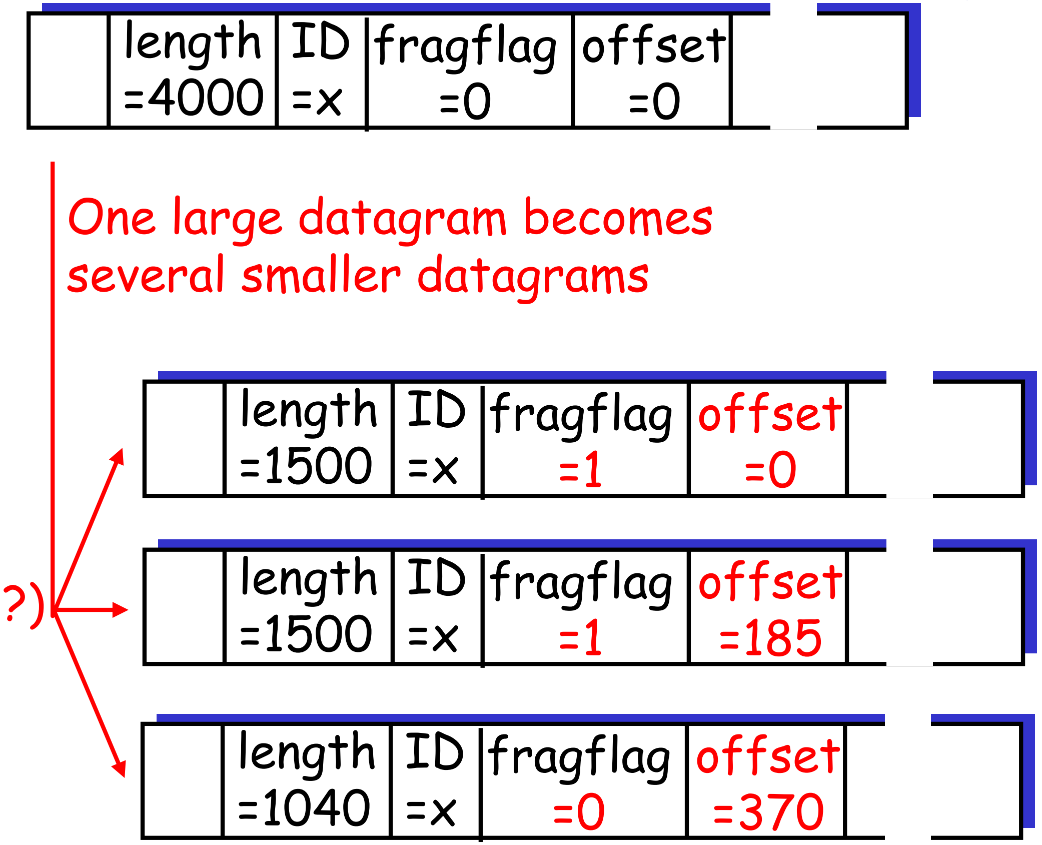
- offset in the second fragment:

  185x8=1480

(why not 1500 bytes =length?)

- offset in the third fragment:

  370x8=2960



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

One large datagram becomes several smaller datagrams

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

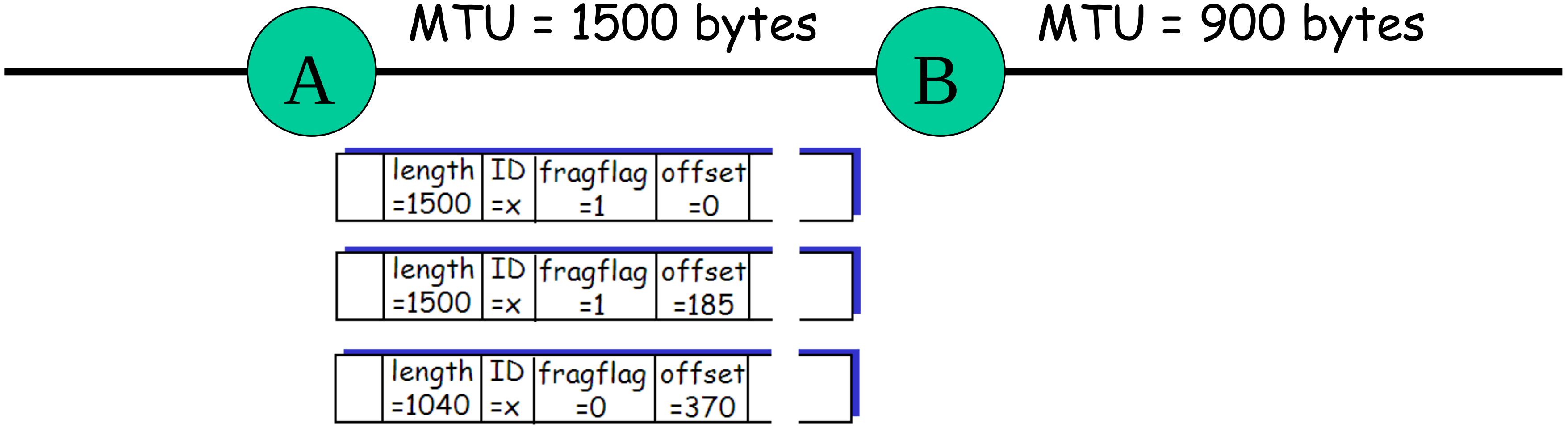
Except for last fragment, IP fragment payload size (i.e., excluding IP header) must be multiple of 8!

Quiz: Calculating length & Offset

Example

- 4000 byte datagram
- MTU = 1500 bytes

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

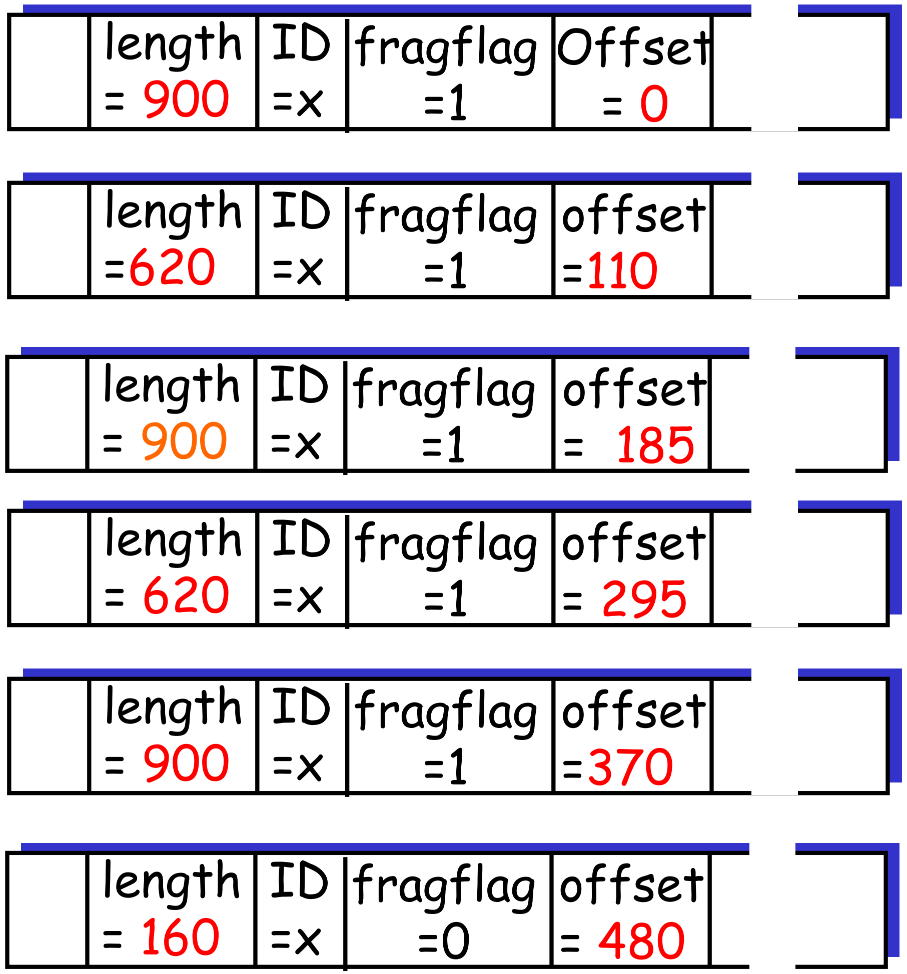


length	ID	fragflag	offset
=1500	=x	=1	=0

length	ID	fragflag	offset
=1500	=x	=1	=185

length	ID	fragflag	offset
=1040	=x	=0	=370

Answer



length = 900	ID =x	fragflag =1	Offset = 0
length =620	ID =x	fragflag =1	offset =110
length = 900	ID =x	fragflag =1	offset = 185
length = 620	ID =x	fragflag =1	offset = 295
length = 900	ID =x	fragflag =1	offset =370
length = 160	ID =x	fragflag =0	offset = 480

ICMP: Internet Control Message Protocol

- used by hosts, routers, gateways to communication network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer “above” IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

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	4	datagram too big
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
5	0,1	*redirect for network/host*
8	0	echo request (ping)
9	0	route advertisement
10	0	router solicitation
11	0	TTL expired
12	0	bad IP header

ICMP Message Transport & Usage

- ICMP messages carried in IP datagrams
- Treated like any other datagrams
  - But no error message sent if ICMP message causes error
- Message sent to the source
  - 8 bytes of the original header included
- ICMP Usage (non-error, informational): Examples
  - Testing reachability: ICMP echo request/reply
    - ping
  - Tracing route to a destination: Time-to-live field
    - traceroute
  - Path MTU discovery (see next slide for more details)
    - Don't fragment bit
  - IP redirect (for hosts only): inform hosts of better routes

ICMP and Path MTU (RFC 1191)

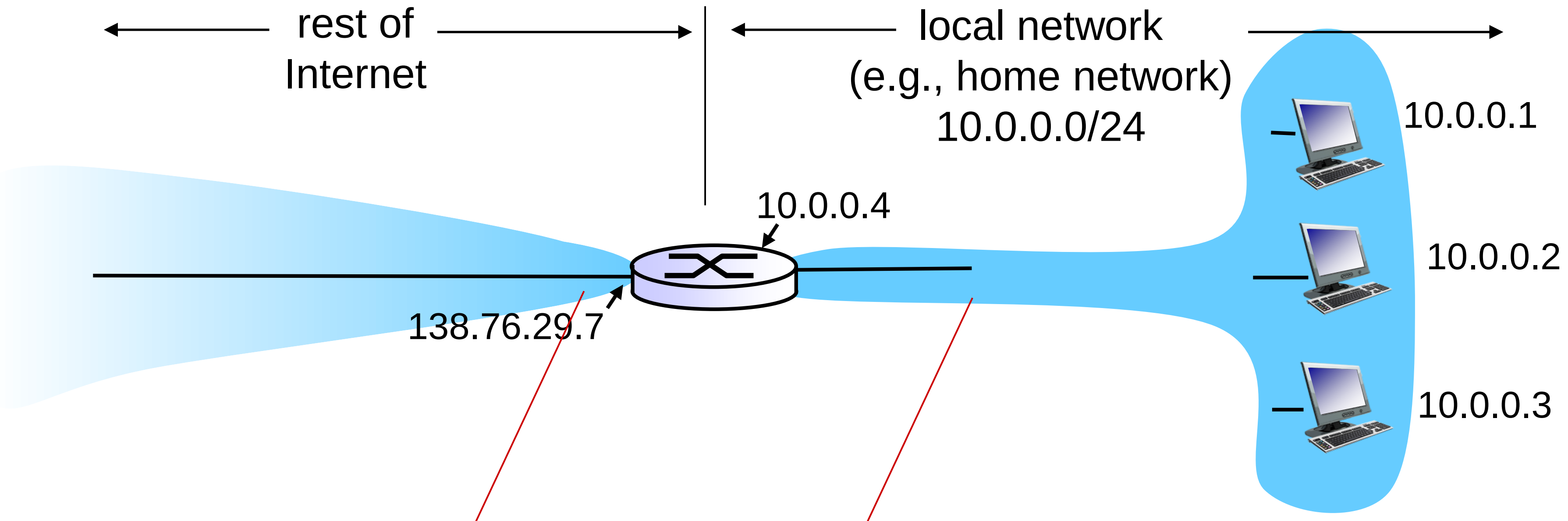
When a router is unable to forward a datagram, because it exceeds the MTU of the next-hop network *and* its “Don't Fragment” bit is set, the router is required to

- return an ICMP “Destination Unreachable” message (type 3) to the source of the datagram, with code 4, indicating ”Fragmentation required and DF flag set".

To support Path MTU Discovery, the router MUST

- include the MTU of that next-hop network in the low-order 16 bits of the ICMP header field that is labelled "unused" in the ICMP specification.
- The high-order 16 bits remain unused, and MUST be set to zero.

NAT (Network Address Translation) A fix to limited IPv4 address space:



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

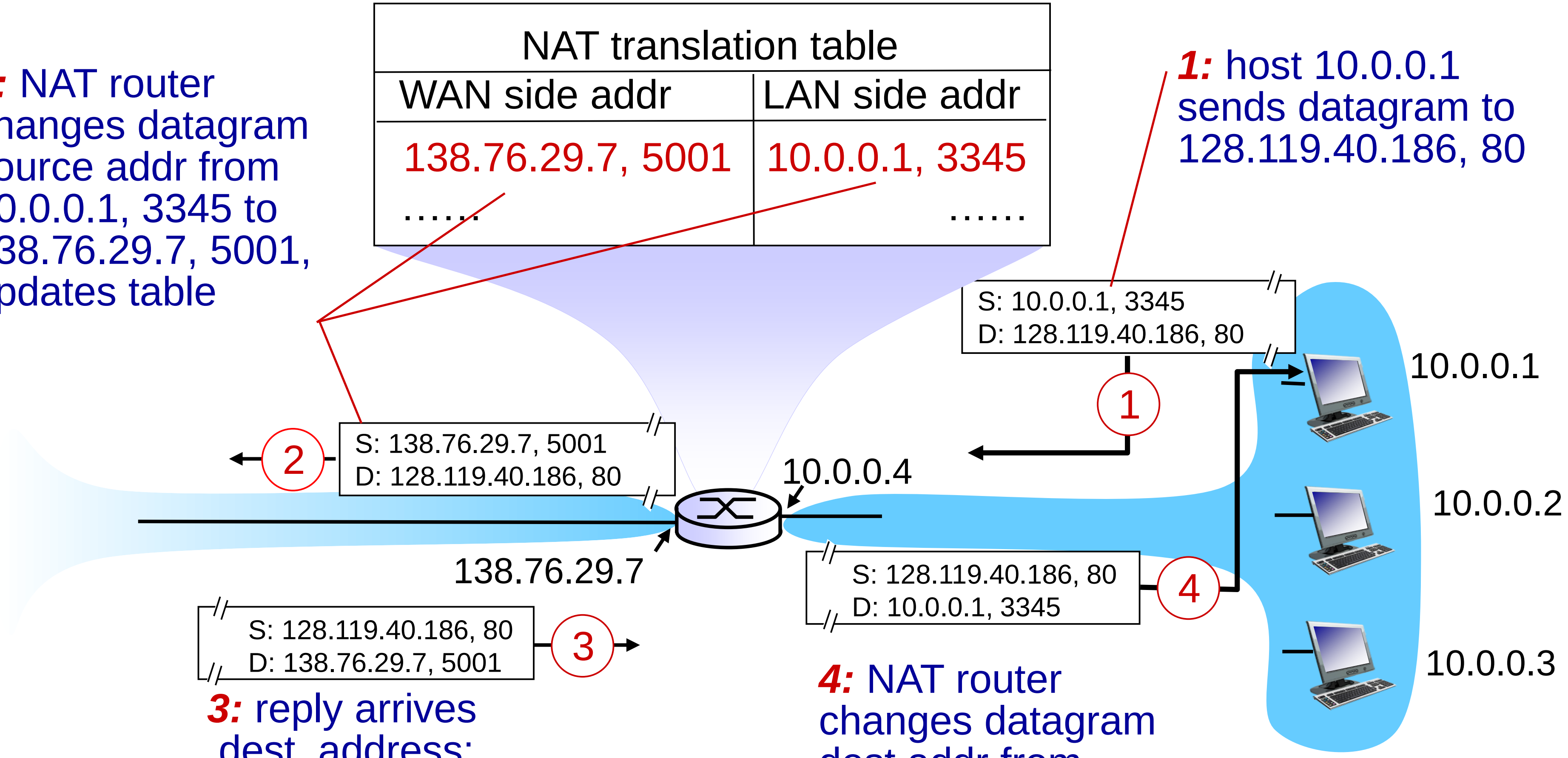
datagrams with source or destination in this network have 10.0.0.0/24 address for source, destination (as usual)

NAT (Network Address Translation)

motivation: local network uses just one IP address as far as outside world is concerned:

- 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 (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