Network Layer: DHCP, ICMP, NAT, IPv6

CS 352, Lecture 11

http://www.cs.rutgers.edu/~sn624/352-S19

Srinivas Narayana

(heavily adapted from slides by Prof. Badri Nath and the textbook authors)

RUTGERS
UNIVERSITY | NEW BRUNSWICK

IP addresses: how to get one?

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

- Hard-coded by system admin in a file
  - UNIX: /etc/network/interfaces
  - Windows: controlpanel -> network -> configuration -> tcp/ip -> properties
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server
  - “plug-and-play”

Similar bootstrapping problems

- How does a host get its IP address?
- How does a host know its local DNS server?
- How does a host know its subnet mask?
- How does a host know which router is its “gateway” to other networks?

DHCP: Dynamic Host Configuration Protocol

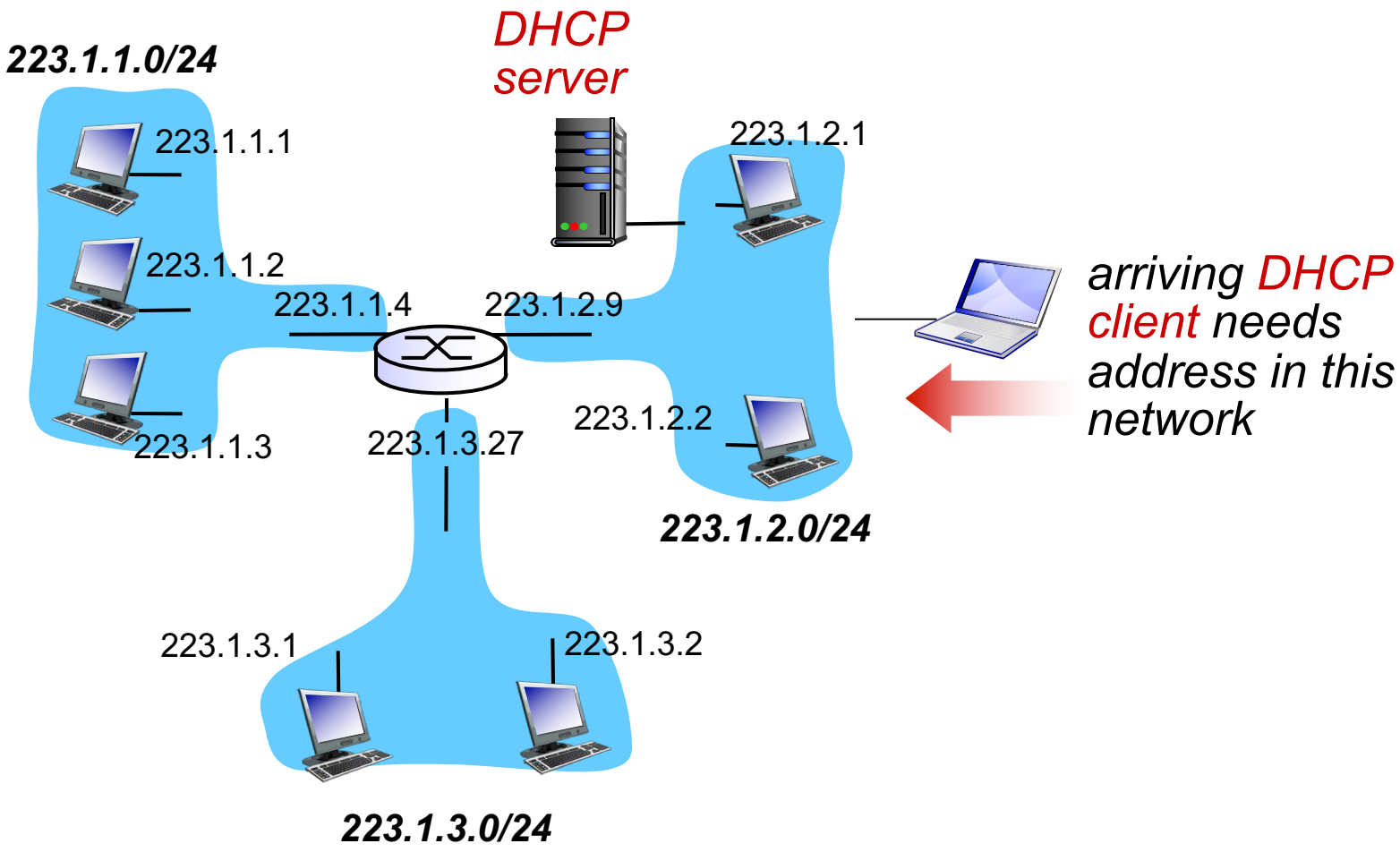
goal: allow host to *dynamically* obtain its IP address from network server when it joins network

- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected/"on")
- support for mobile users who want to join network

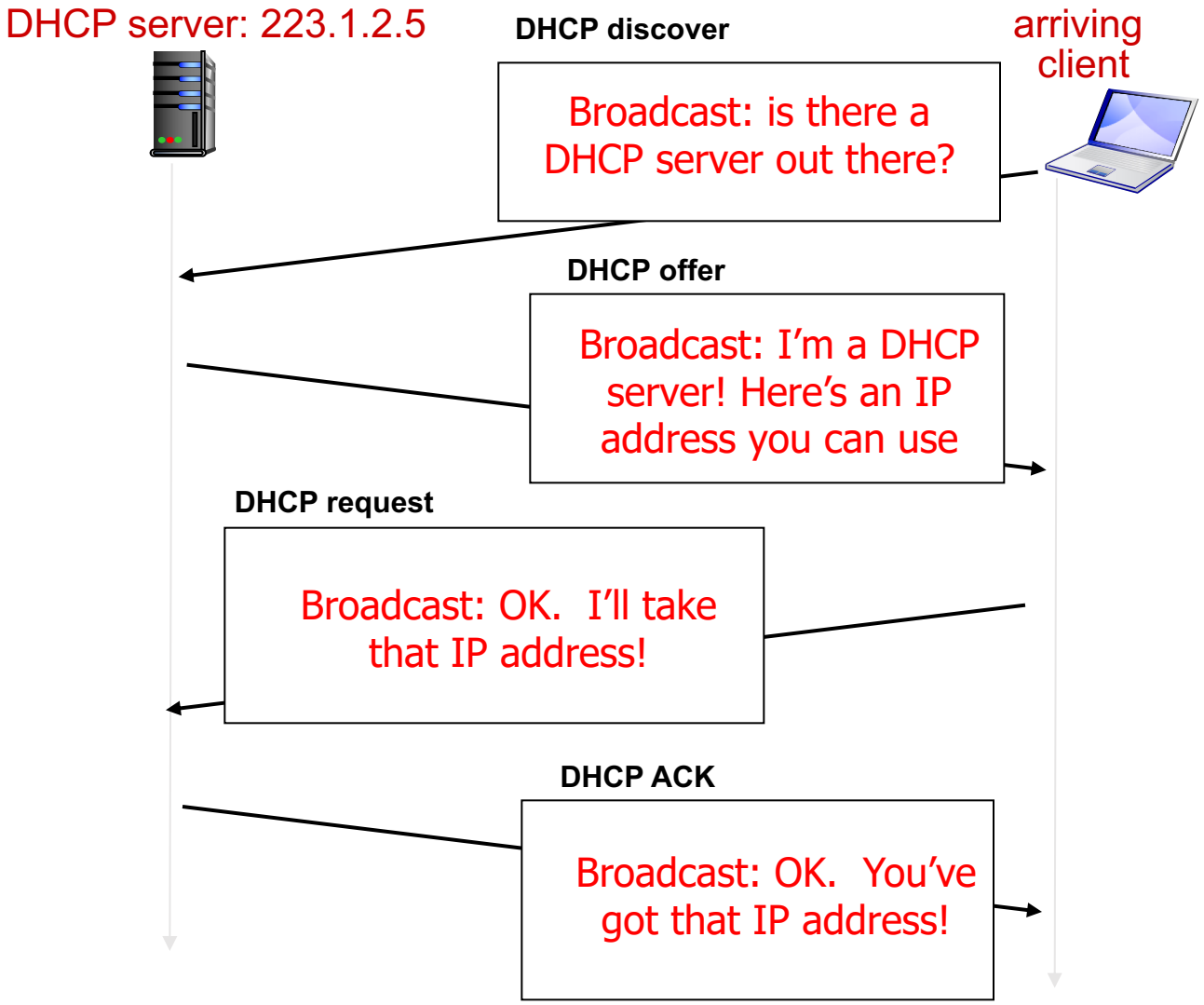
DHCP overview:

- 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

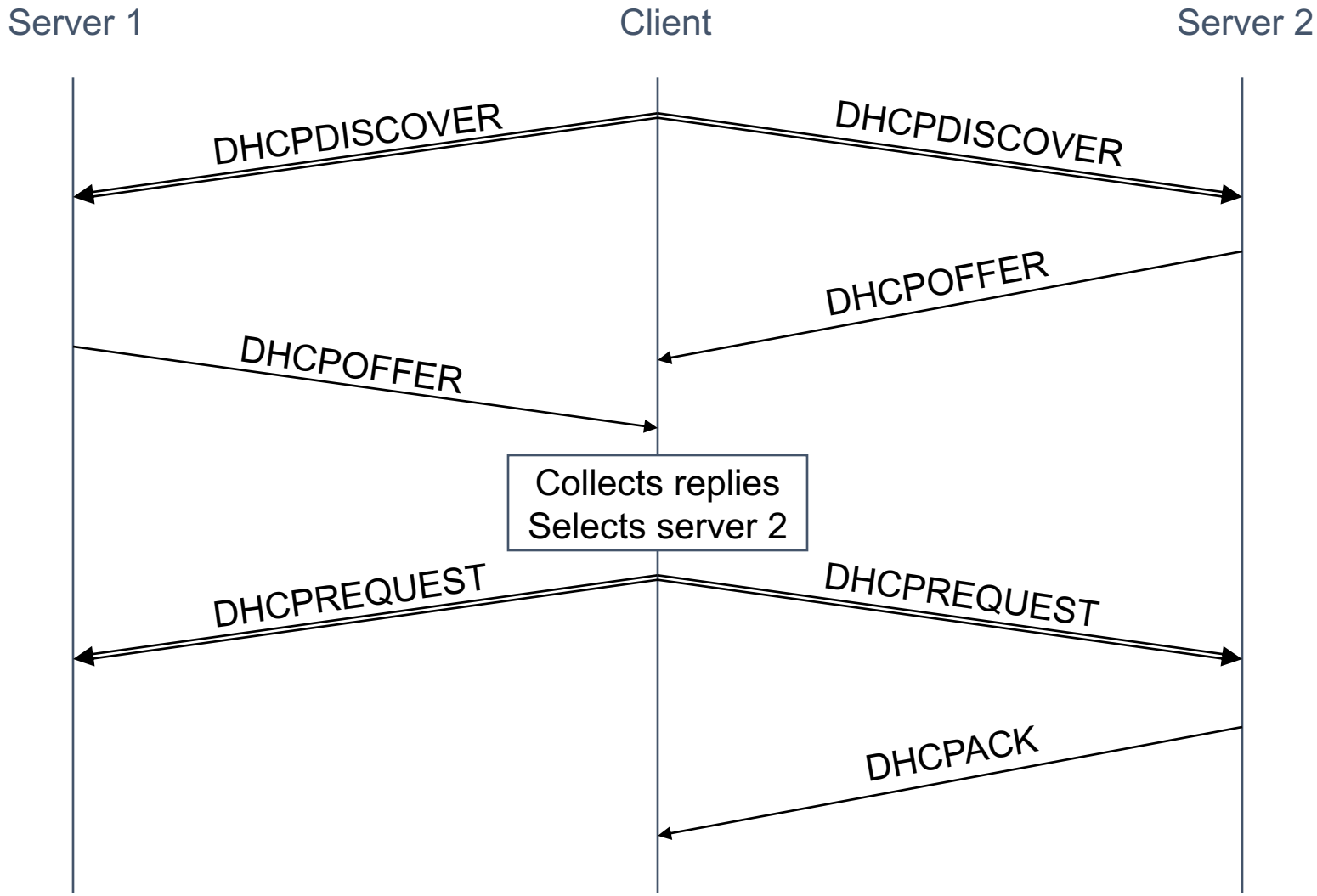
DHCP client-server scenario



DHCP client-server scenario



DHCP Protocol



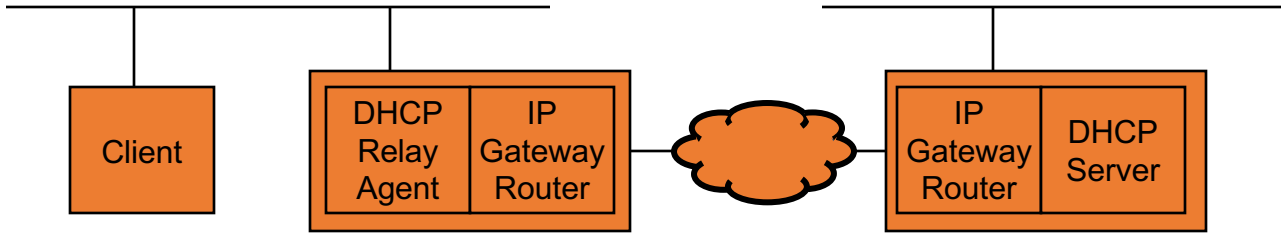
DHCP: more than IP addresses

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

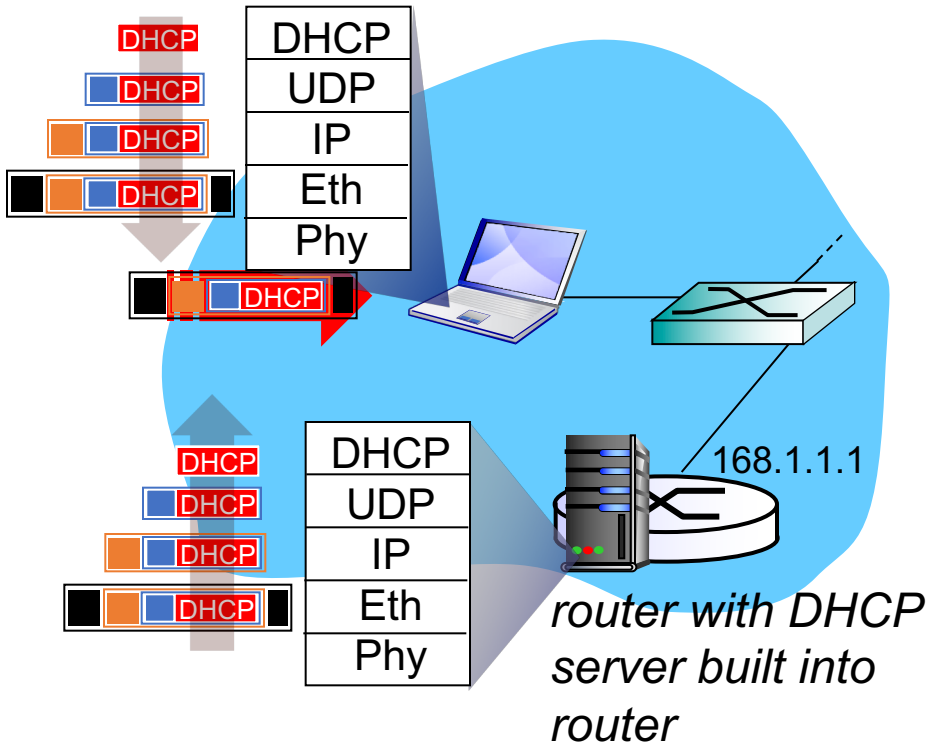
- address of first-hop router for client to reach other subnets
- name and IP address of the local DNS sever
- subnet mask

DHCP Relay Agents

- DHCP relay agents allow DHCP servers to handle requests from other subnets

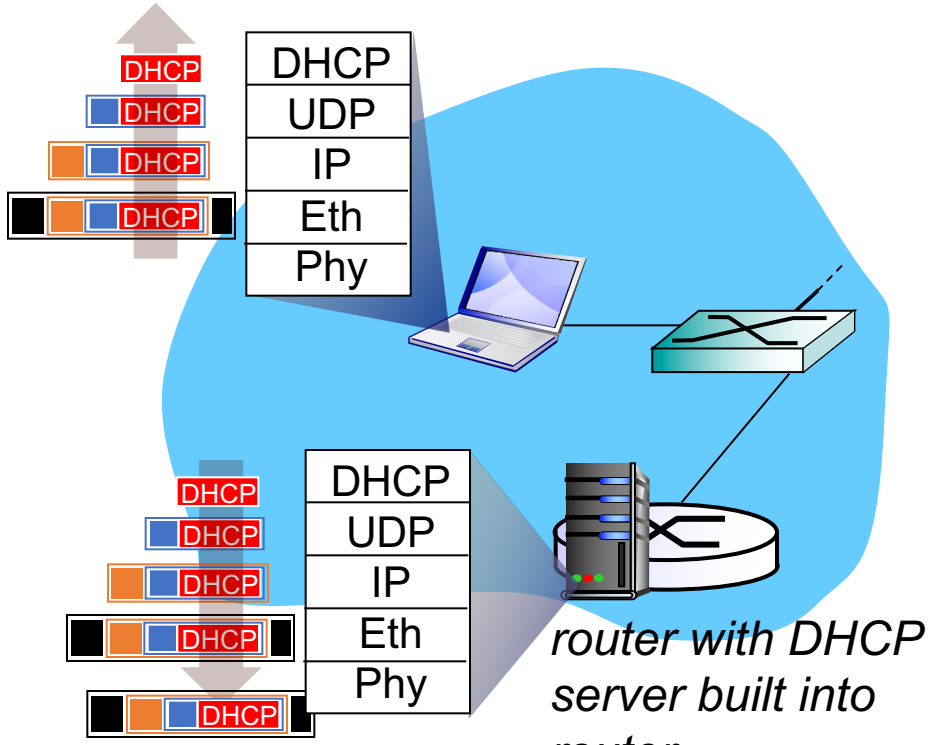


DHCP: example



- connecting laptop needs its IP address, addr of first-hop router, addr of DNS server: use DHCP
- DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.1 Ethernet
- Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
- Ethernet demuxed to IP demuxed, UDP demuxed to DHCP

DHCP: example



- DCP server formulates DHCP ACK containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server
- encapsulation of DHCP server, frame forwarded to client, demuxing up to DHCP at client
- client now knows its IP address, name and IP address of DNS server, IP address of its first-hop router

Summary

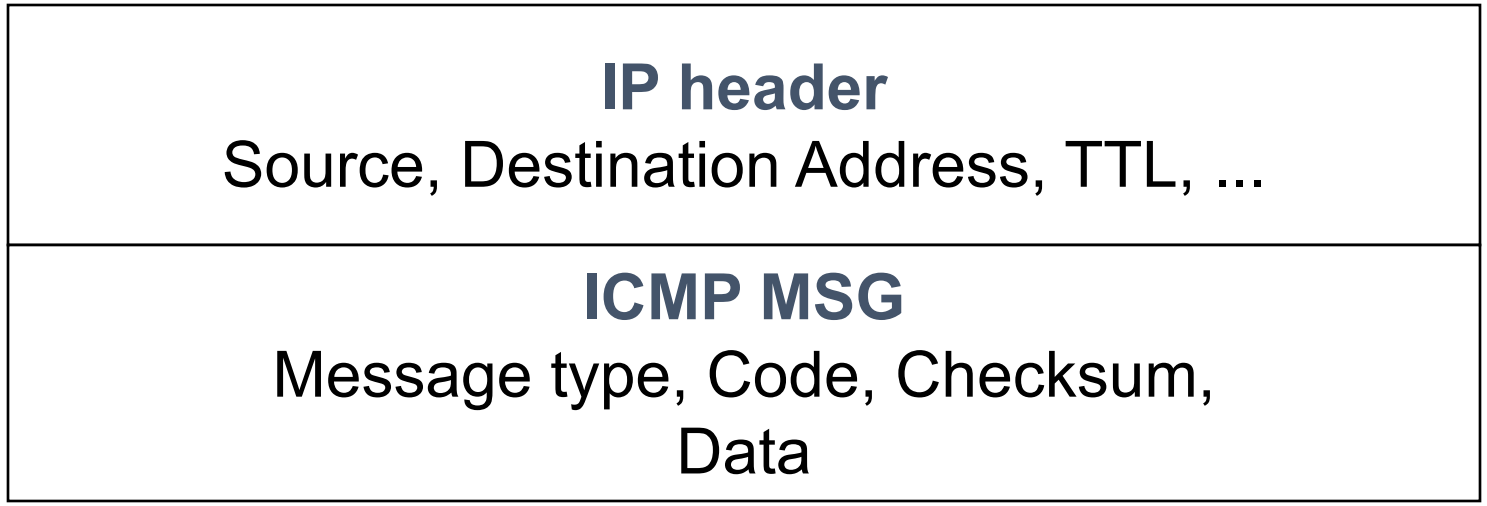
- IP addresses don’t have to be manually configured into hosts
- DHCP allows “ignorant” hosts to receive IP addresses (and more) at start-up time
- DHCP solves important bootstrapping problems in attaching new hosts to a network

Internet Control Message Protocol (ICMP)

ICMP

- Protocol for error detection and reporting
    - tightly coupled with IP, unreliable
- ICMP messages delivered in IP packets
- ICMP functions:
    - Announce reachability and network errors
    - Announce “time exceeded” errors for IP packets
    - Announce network congestion
- ICMP assists network troubleshooting in general

ICMP message



ICMP: Internet Control Message Protocol

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

Time for an activity!

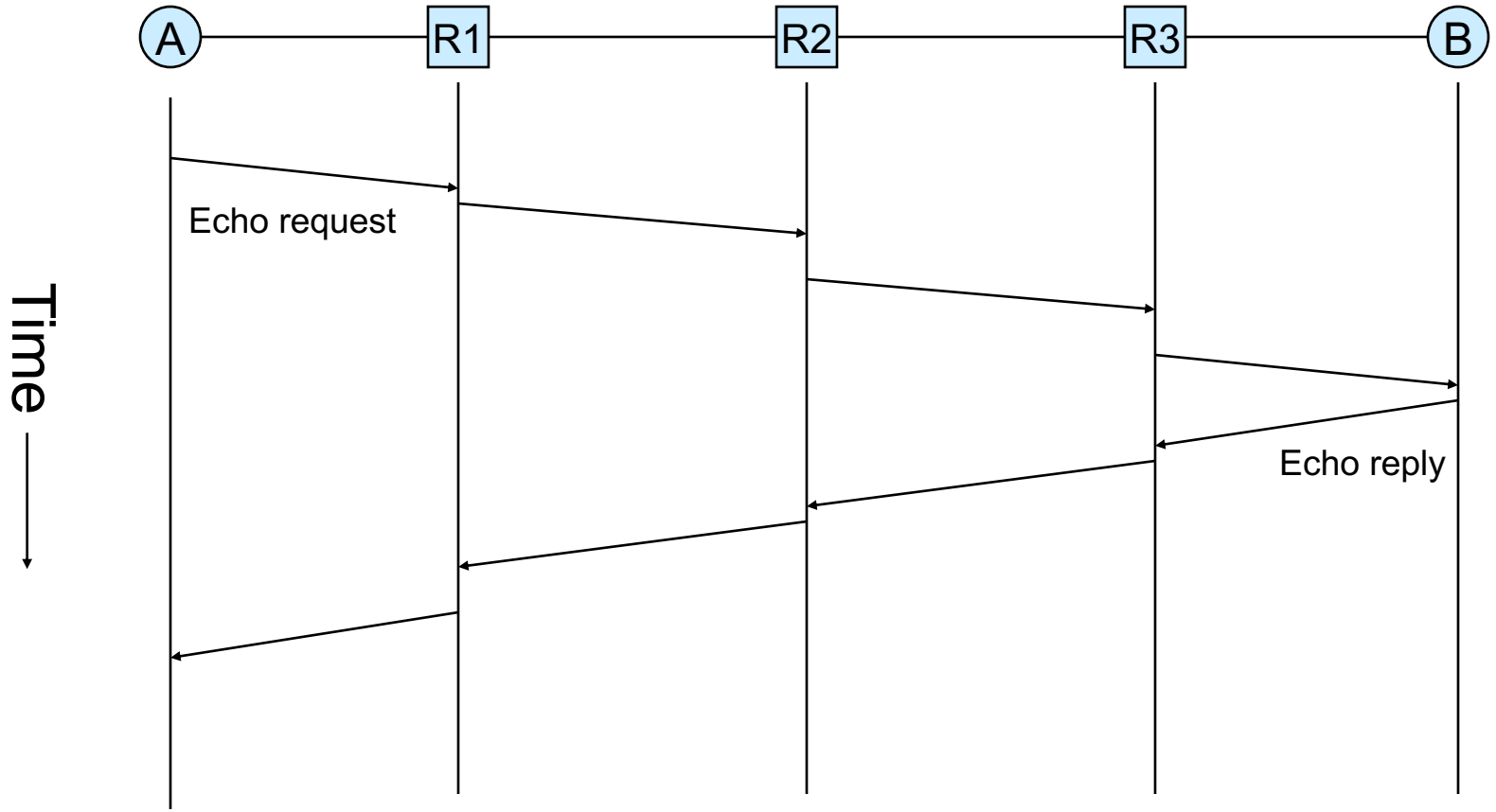
Specific uses of ICMP

- Echo request reply
  - Can be used to check if a host is alive
- Destination unreachable
  - Invalid address and/or port
- TTL expired
  - Routing loops, or too far away

Ping

- Uses ICMP echo request/reply
- Source sends ICMP echo request message to the destination address
- Destination replies with an ICMP echo reply message containing the data in the original echo request message
- Source can calculate round trip time (RTT) of packets
- If no echo reply comes back then the destination is unreachable

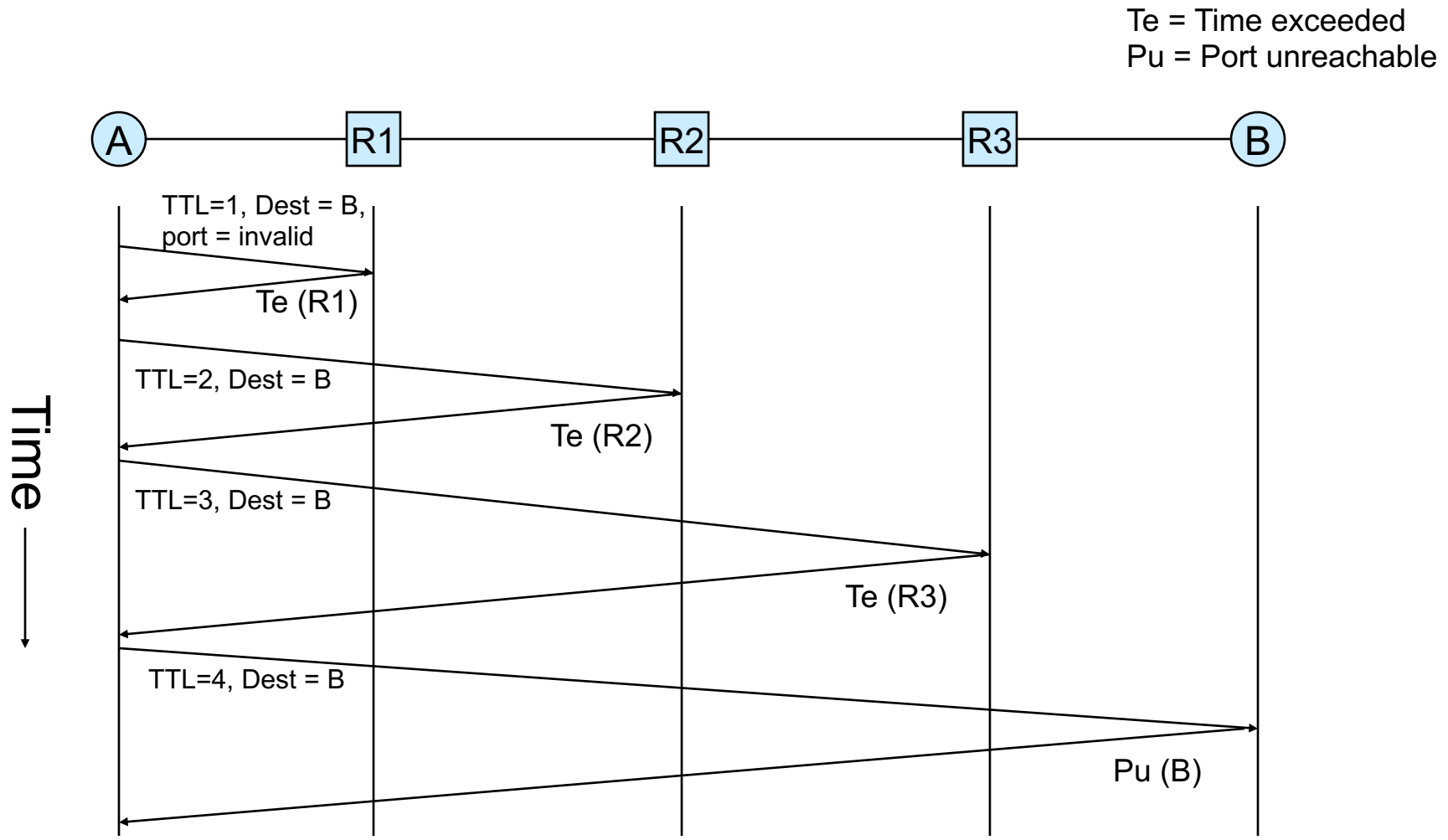
Ping *(cont'd)*



Traceroute

- Traceroute records the route that packets take
- A clever use of the TTL field
- When a router receives a packet, it decrements TTL
- If TTL=0, it sends an ICMP time exceeded message back to the sender
- To determine the route, progressively increase TTL
  - Every time an ICMP time exceeded message is received, record the sender's (router's) address
  - Repeat until the destination host is reached or an error message occurs
- If packet reaches the destination, the dest host usually sends an ICMP port unreachable

Traceroute *(cont'd)*

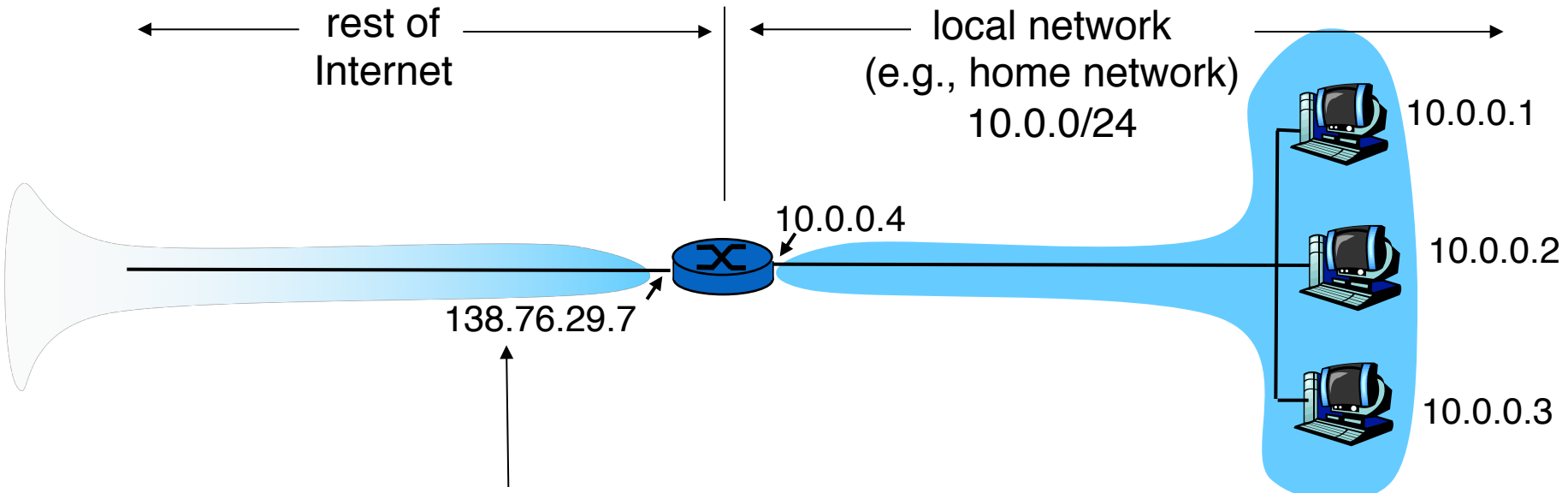


Traceroute example

```
[flow:352]$  traceroute google.com
traceroute to google.com (172.217.10.238), 64 hops max, 52 byte packets
 1  vlan451-sr03-hill-nbp.runet.rutgers.net (172.25.112.1)  9.278 ms  3.210 ms  3.124 ms
 2  xe-1-3-0-0-cr02-hill-nbp.runet.rutgers.net (172.29.6.65)  37.125 ms  2.912 ms  2.899 ms
 3  ae1-2000-cr10-hill-nbp.runet.rutgers.net (172.29.6.42)  3.078 ms  3.086 ms  3.016 ms
 4  ae5-2000-cr02-halsey-nwk.runet.rutgers.net (172.29.6.55)  3.693 ms  3.707 ms  3.793 ms
 5  ae2-0-er10-halsey-ext.runet.rutgers.net (172.29.8.6)  3.699 ms  3.693 ms  3.766 ms
 6  ae1-0-fw01-halsey-nwk.runet.rutgers.net (172.29.8.41)  4.019 ms  3.909 ms  3.750 ms
 7  et-2-2-0-0-er10-halsey-ext.runet.rutgers.net (172.29.8.46)  4.310 ms  4.181 ms  3.948 ms
 8  gateway-pni.google.com (128.6.1.114)  4.426 ms  3.901 ms  4.161 ms
 9  108.170.248.65 (108.170.248.65)  5.024 ms
    108.170.248.1 (108.170.248.1)  6.147 ms  6.165 ms
10  72.14.233.201 (72.14.233.201)  5.316 ms  5.426 ms  5.359 ms
11  lga25s59-in-f14.1e100.net (172.217.10.238)  5.410 ms  5.156 ms  5.135 ms
[flow:352]$
```

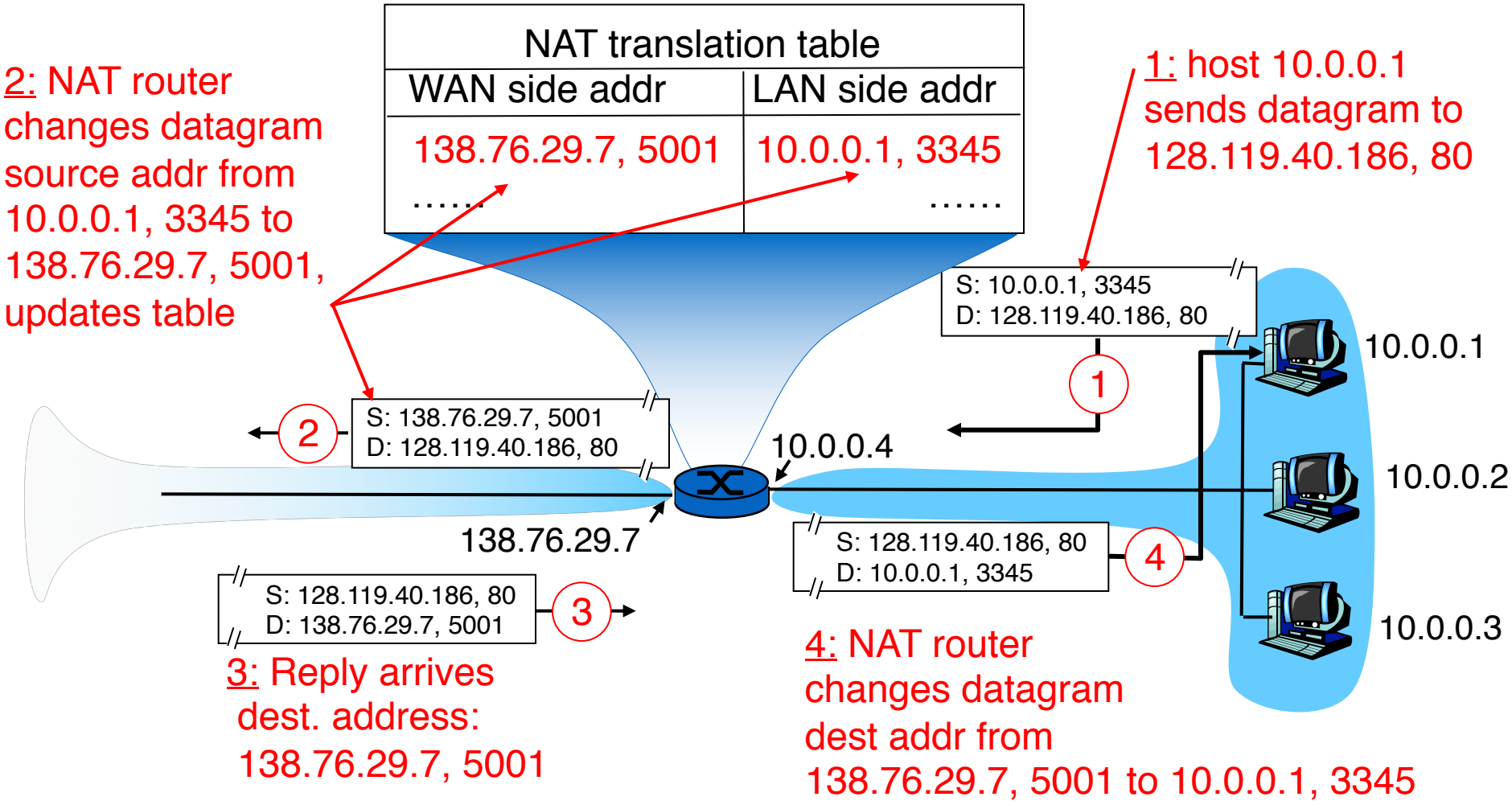
Network Address Translation (NAT)

NAT: Network Address Translation



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

NAT: Network Address Translation



NAT: Network Address Translation

- Features: 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).

Think about…

- How do the hosts inside the home network get their IP addresses?

- How does your home router get its externally visible IP address?

NAT: Network Address Translation

- 16-bit port-number field:
  - 60,000 simultaneous connections with a single LAN-side address!
- NAT is controversial:
  - Routers should only work upto the network layer, not transport ports!
  - violates “end-to-end argument”
    - NAT must be taken into account by app designers
    - e.g., P2P applications like skype
  - Purists: address shortage should instead be solved by IPv6

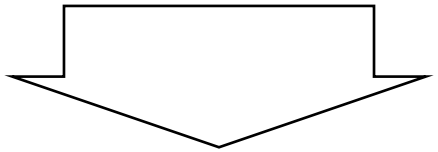
Internet Protocol v6 (IPv6)

Recent Developments: IPv6

- IPv4 has limited address space (32 bits) and is running out of addresses. 32 bits are not enough!

- More devices: phones, watches, your refrigerator(!), …

- Real-time traffic and mobile users are also becoming more common



IP version 6

IPv6: Main changes from IPv4

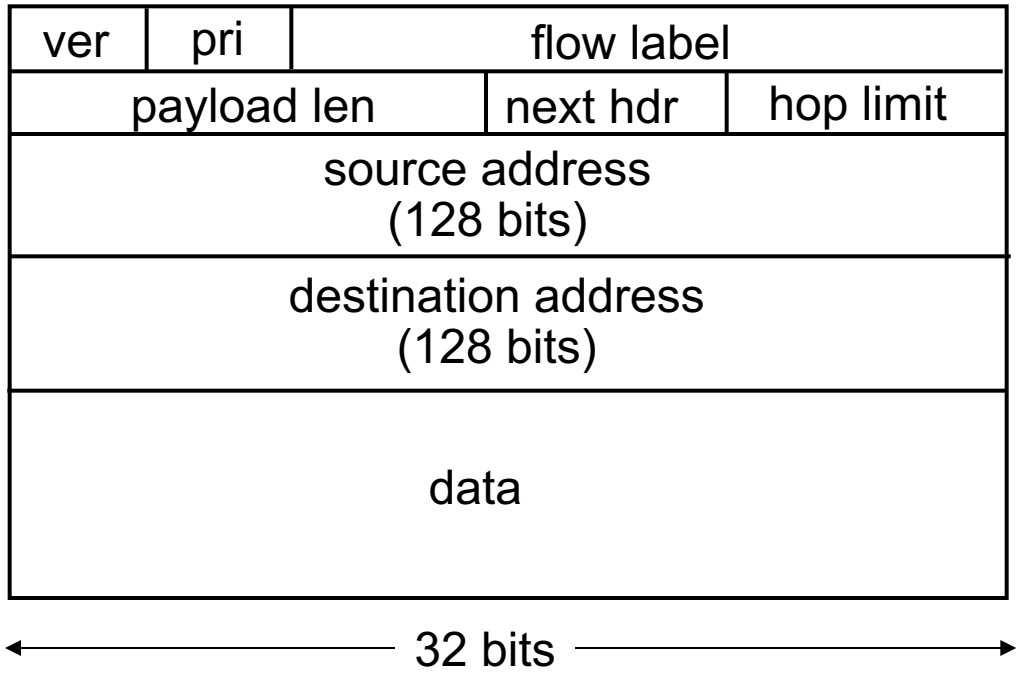
- Large address space:
  - 128-bit addresses (16 bytes)
  - Allows up to 340,282,366,920,938,463,463,374,607,431,768,211,456 unique addresses (3.4×10^{38})
- Fixed length headers (40 bytes)
  - Improves the speed of packet processing in routers
- IPv6 “options” processing happens through a separate mechanism

IPv6 datagram format

priority: identify priority among datagrams in flow

flow Label: identify datagrams in same "flow"
(concept of"flow" left undefined)

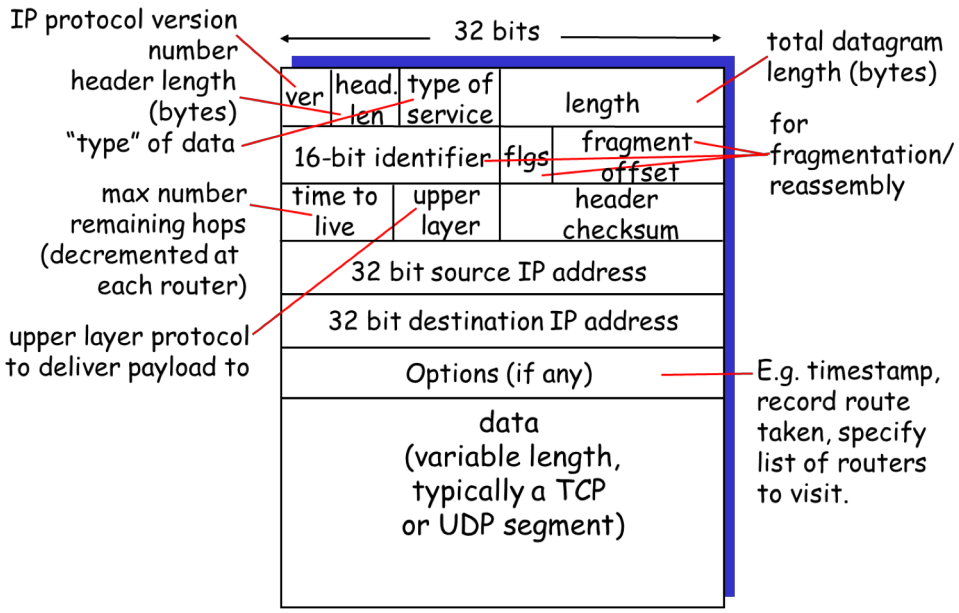
next header: identify upper layer protocol for data

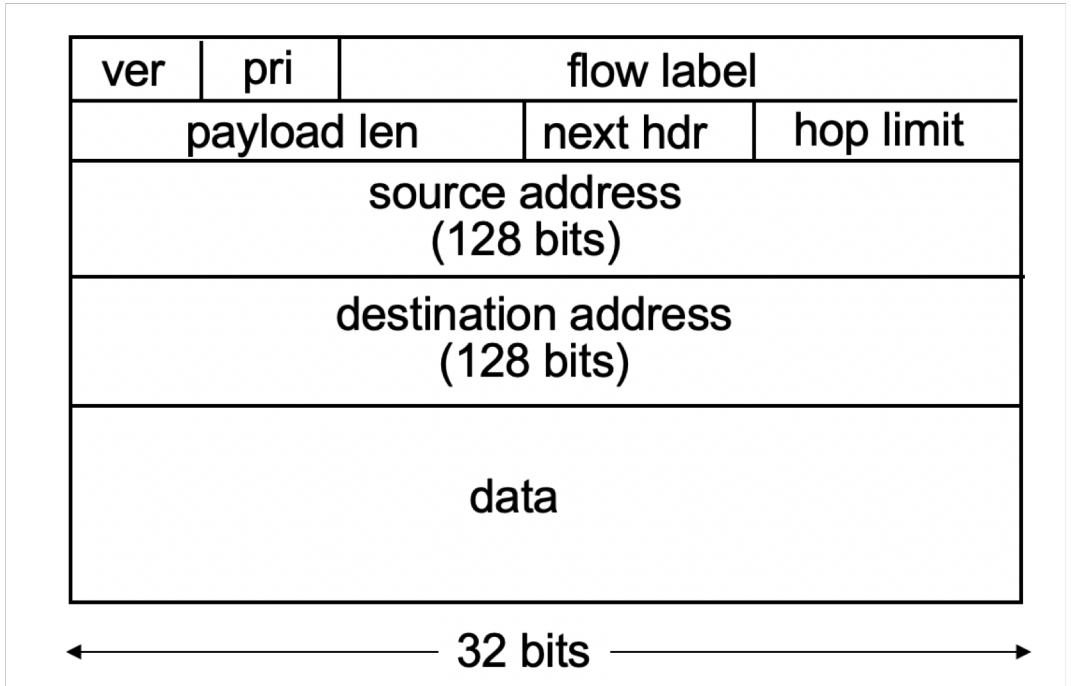


Other changes from IPv4

- *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

IPv4 vs IPv6: Can you tell the differences?





IPv6 Flows

- Support for “flows”
  - Flows help support real-time service in the Internet
  - A “flow” is a number in the IPv6 header that can be used by routers to see which packets belong to the same stream
  - Guarantees can then be assigned to certain flows
  - Example:
    - Packets from flow 10 should receive rapid delivery
    - Packets from flow 12 should receive reliable delivery

IPv6 Addresses

- Classless addressing/routing (similar to CIDR)

- Notation: xx:xx:xx:xx:xx:xx:xx:xx
  - x = 4-bit hex number
  - contiguous 0s are compressed: 47CD::A456:0124
  - IPv6 compatible IPv4 address: ::128.64.18.87
    - First 96 bits are 0
  - Global unicast addresses start with 001….
  - 2000::/3 prefix

IPv6: Adoption

- Google: 8% of clients access services via IPv6
- NIST: 1/3 of all US government domains are IPv6 capable

- *Long (long!) time for deployment, use*
  - 20 years and counting!

- Think of application-level changes in last 20 years: WWW, Facebook, streaming media, Skype, …
- *Why?*

Traceroute Examle

```
 1  lcsr-gw (128.6.13.21)  1.206 ms  0.973 ms  0.782 ms
 2  rucs-gw (165.230.212.129)  0.697 ms  0.569 ms  0.571 ms
 3  transition2-gw (165.230.12.145)  2.786 ms  0.994 ms  0.769 ms
 4  rutgers-gw.Rutgers.EDU (198.151.130.209)  1.726 ms  2.048 ms Vl1000-sr02-hil
1012-svcs.Rutgers.EDU (198.151.130.14)  1.278 ms
 5  rutgers-gw.Rutgers.EDU (198.151.130.209)  1.755 ms  1.241 ms  1.828 ms
 6  198.151.130.226 (198.151.130.226)  2.748 ms  3.070 ms  2.640 ms
 7  clev-nycm.abilene.ucaid.edu (198.32.8.29)  15.162 ms  14.619 ms  14.663 ms
 8  ipls-clev.abilene.ucaid.edu (198.32.8.25)  21.220 ms  22.497 ms  21.450 ms
 9  kscy-ipls.abilene.ucaid.edu (198.32.8.5)  30.257 ms  30.604 ms  30.969 ms
10  dnvr-kscy.abilene.ucaid.edu (198.32.8.13)  40.823 ms  41.181 ms  41.076 ms
11  snva-dnvr.abilene.ucaid.edu (198.32.8.1)  65.436 ms  66.068 ms  65.569 ms
12  198.32.249.161 (198.32.249.161)  65.673 ms  65.771 ms  66.006 ms
13  BERK--SUNV.POS.calren2.net (198.32.249.13)  67.183 ms  67.131 ms  66.858 ms
14  pos1-0.inr-000-eva.Berkeley.EDU (128.32.0.89)  67.192 ms  66.749 ms  67.720
ms
15  vlan198.inr-201-eva.Berkeley.EDU (128.32.0.194)  67.373 ms  67.067 ms  67.82
1 ms
16  fast8-0-0.inr-210-cory.Berkeley.EDU (128.32.255.122)  67.634 ms  68.735 ms
68.413 ms
17  GE.cory-gw.EECS.Berkeley.EDU (169.229.1.46)  67.575 ms  68.222 ms  67.772 ms
18  gig8-1.snr1.CS.Berkeley.EDU (169.229.3.66)  67.454 ms  67.988 ms  67.177 ms
19  now.CS.Berkeley.EDU (128.32.44.96)  67.892 ms *  67.818 ms
```