CS 356: Computer Network Architectures

Lecture 11: IP Fragmentation, ARP, and ICMP

Xiaowei Yang

xwy@cs.duke.edu

Overview

- Wrapping up from last leture
- IP fragmentation
- ARP
- ICMP

The longest prefix matching algorithm

1. Search for a match on all 32 bits
2. Search for a match for 31 bits

    …..

32. Search for a match on 0 bits

Host route, loopback entry

→ 32-bit prefix match

Default route is represented as 0.0.0.0/0

→ 0-bit prefix match

Why longest prefix match?

- Longest → smallest network
- Network prefixes may be aggregated

Example

128.143.71.21

↓

Destination address	Next hop
10.0.0.0/8	eth0
128.143.0.0/16	R2
128.143.64.0/20	R3
128.143.192.0/20	R3
128.143.71.0/24	R4
128.143.71.55/32	R3
0.0.0.0/0 (default)	R5

↓

The longest prefix match for 128.143.71.21 is for 24 bits with entry 128.143.71.0/24

Datagram will be sent to R4

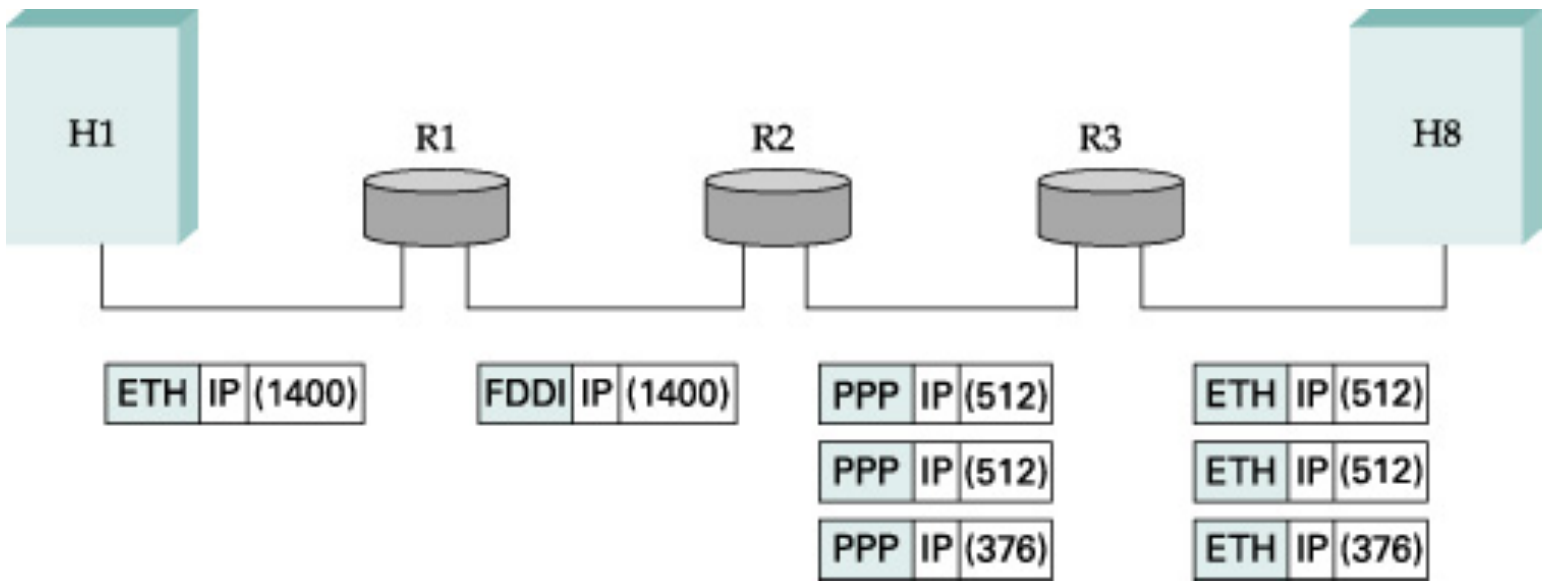
Example: address allocation

- Duke network operators receive a /16 address prefix 152.3.0.0/16 from ARIN
- Allocate address prefixes to three departmental networks
  - ME must have at least 50 hosts
  - ECE and CS must have at least 100 hosts
- Smallest address prefix to each deparment?

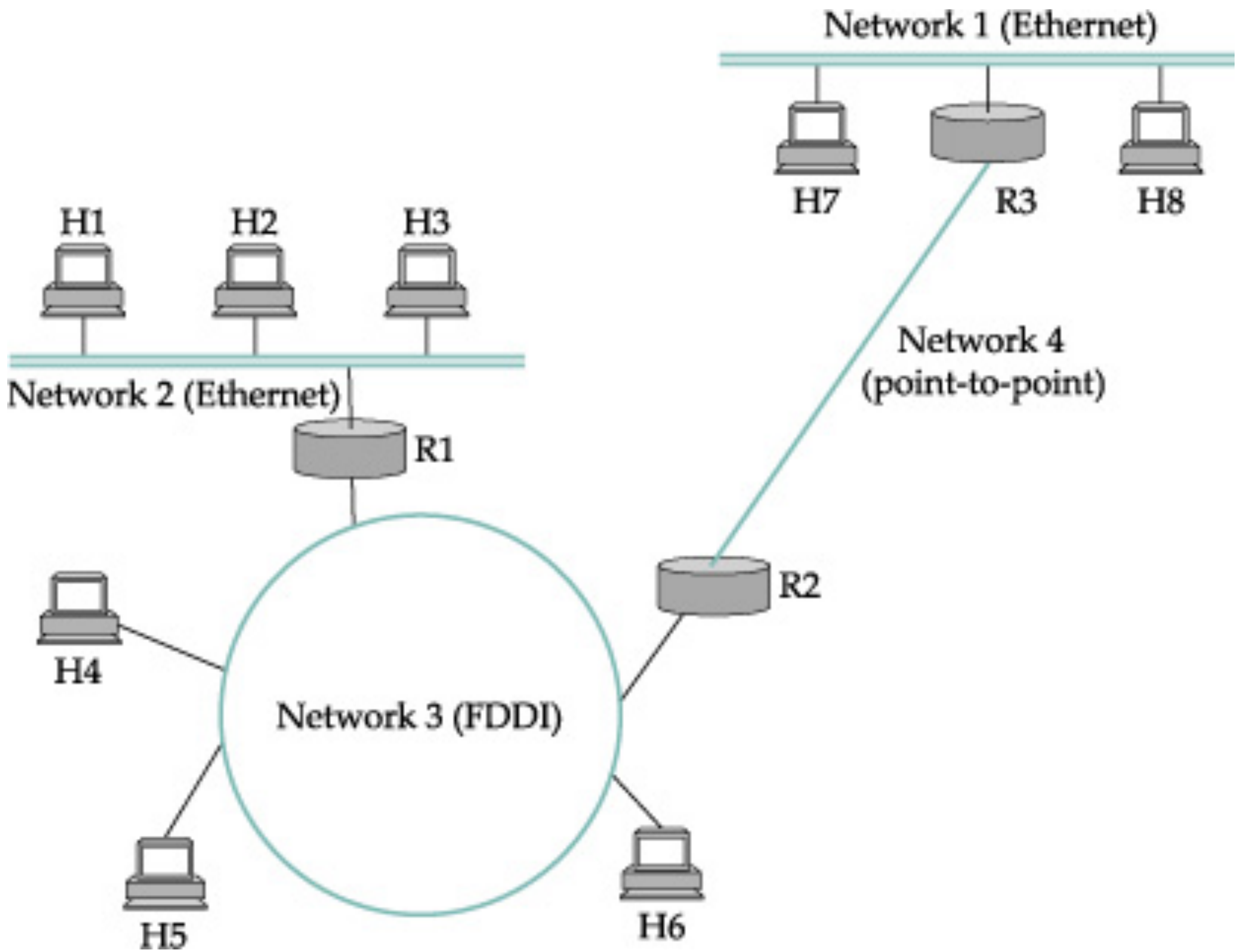
Fragmentation and Reassembly

(not required for Lab 2)

Different networks have different Maximum Transmission Units (MTUs)



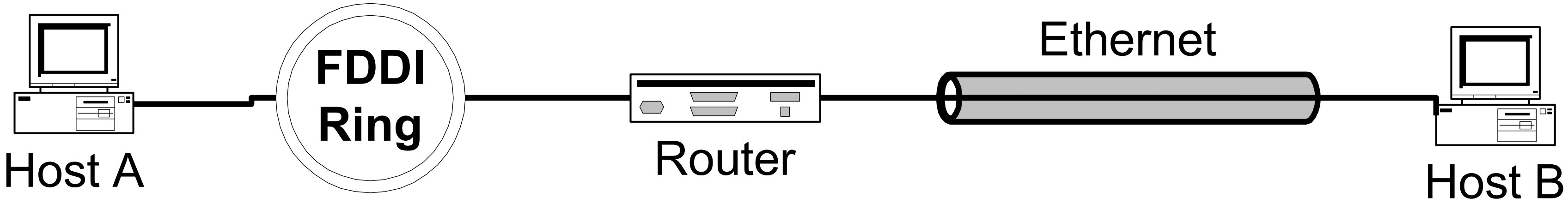
Packets may traverse different types of links



IP Fragmentation and Reassembly

- What if the size of an IP datagram exceeds the MTU?
  IP datagram is fragmented into smaller units.

- What if the route contains networks with different MTUs?

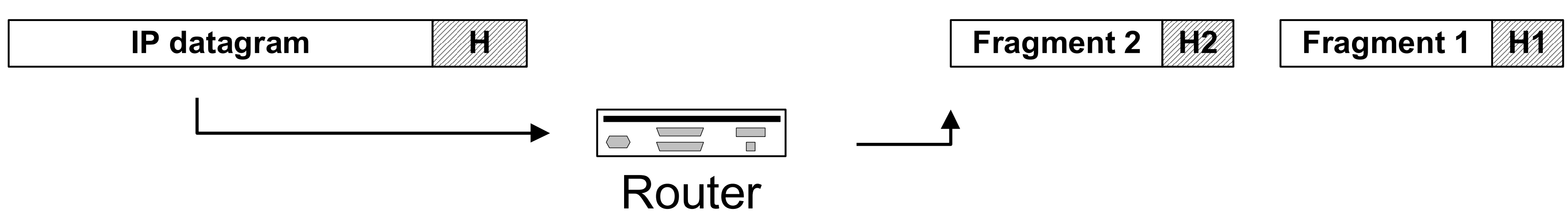


MTUs: FDDI: 4352 Ethernet: 1500

- **Fragmentation**:
  - IP router splits the datagram into several datagrams

Design question: Where is Fragmentation/reassembly done?

- In IPv4, Fragmentation can be done at the sender or at intermediate routers
- The same datagram can be fragmented several times.
- Reassembly of original datagram is only done at destination hosts !! (why?)



What' s involved in Fragmentation?

- The following fields in the IP header are involved:

version	header length	DS	ECN	total length (in bytes)		
Identification			0	DF	MF	Fragment offset
time-to-live (TTL)	protocol		header checksum			

- Identification
  – When a datagram is fragmented, the identification is the same in all fragments
  – Used to reassemble the original packet
- Flags
  – DF bit is set: datagram cannot be fragmented and must be discarded if MTU is too small
    - ICMP sent
  – MF bit:
    - 1: this is not the last fragment
    - 0: last fragment

What's involved in Fragmentation?

- The following fields in the IP header are involved:

version	header length	DS	ECN	total length (in bytes)			
Identification				0	DF	MF	Fragment offset (13-bit)
time-to-live (TTL)	protocol			header checksum			

- *Fragment offset*
  - Offset of the payload of the current fragment in the original datagram in units of 8 bytes
    - Why?
    - Because the field is only 13 bits long, while the total length is 16 bits.
- Total length
  - Total length of the current fragment

Example of Fragmentation

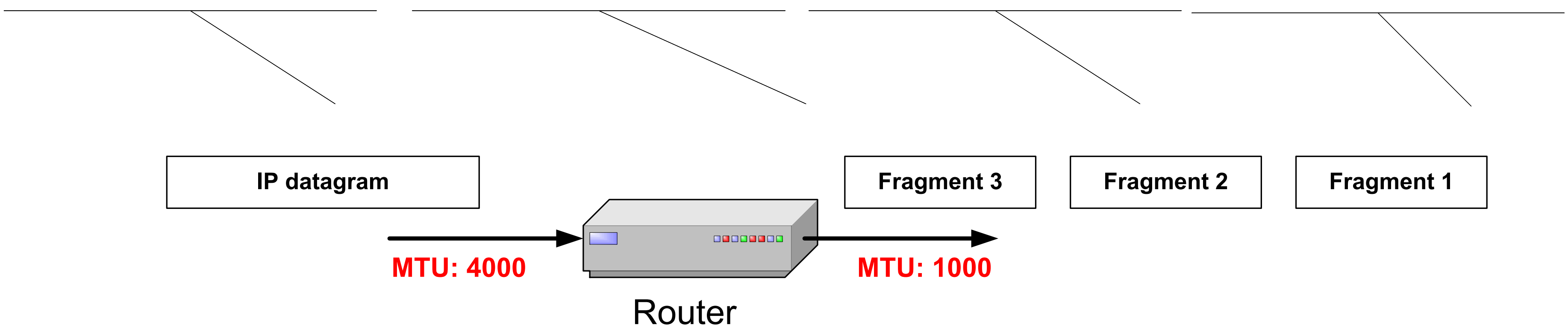
- A datagram with size 2400 bytes must be fragmented according to an MTU limit of 1000 bytes

Header length: 20
Total length: 2400
Identification: 0xa428
DF flag: 0
MF flag: 0
Fragment offset: 0

Header length: 20
Total length: 448
Identification: 0xa428
DF flag: 0
MF flag: 0
Fragment offset: 244

Header length: 20
Total length: 996
Identification: 0xa428
DF flag: 0
MF flag: 1
Fragment offset: 122

Header length: 20
Total length: 996
Identification: 0xa428
DF flag: 0
MF flag: 1
fragment offset: 0



Determining the length of fragments

- Maximum payload length = 1000 – 20 = 980 bytes
- Offset specifies the bytes in multiple of 8 bytes. So the payload must be a multiple of 8 bytes.
- 980 - 980 % 8 = 976 (the largest number that is less than 980 and divisible by 8)
- The payload for the first fragment is 976 and has bytes 0 ~ 975 of the original IP datagram. The offset is 0.
- The payload for the second fragment is 976 and has bytes 976 ~ 1951 of the original IP datagram. The offset is 976 / 8 = 122.
- The pay load of the last fragment is 2400 – 976 * 2 = 428 bytes and has bytes 1952 ~ 2400 of the original IP datagram. The offset is 244.
- Total length of three fragments: 996 + 996 + 448 = 2440 > 2400
  - Why?
  - Two additional IP headers.

Path MTU discovery

- Fragmentation slows down the router
- → should be done by end hosts

- How does a sender know the MTU of a path?
  - A host only knows the MTU of its links

- Solution
  - Sends large packets with DF set
  - If receives ICMP Fragmentation needed messages, reduces maximum segment size

Overview

- IP fragmentation
- ARP
- ICMP

Longest prefix match

- **Longest Prefix Match:** Search for the forwarding table entry that has the longest match with the prefix of the destination IP address

1. Search for a match on all 32 bits
2. Search for a match for 31 bits

   …..

32. Search for a match on 0 bits

Host route, loopback entry
→ 32-bit prefix match
Default route is represented as 0.0.0.0/0
→ 0-bit prefix match

128.143.71.21

↓

Destination address	Next hop
10.0.0.0/8	eth0
128.143.0.0/16	R2
128.143.64.0/20	R3
128.143.192.0/20	R3
128.143.71.0/24	R4
128.143.71.55/32	R3
0.0.0.0/0 (default)	R5

↓

The longest prefix match for 128.143.71.21 is for 24 bits with entry 128.143.71.0/24

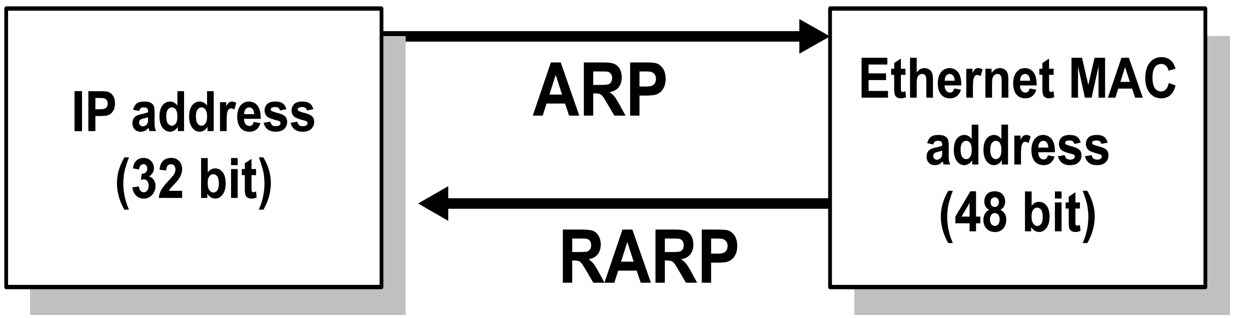
Datagram will be sent to R4

How to find out a host's Ethernet address after knowing its IP address?

→ Address Resolution Protocol

ARP and RARP

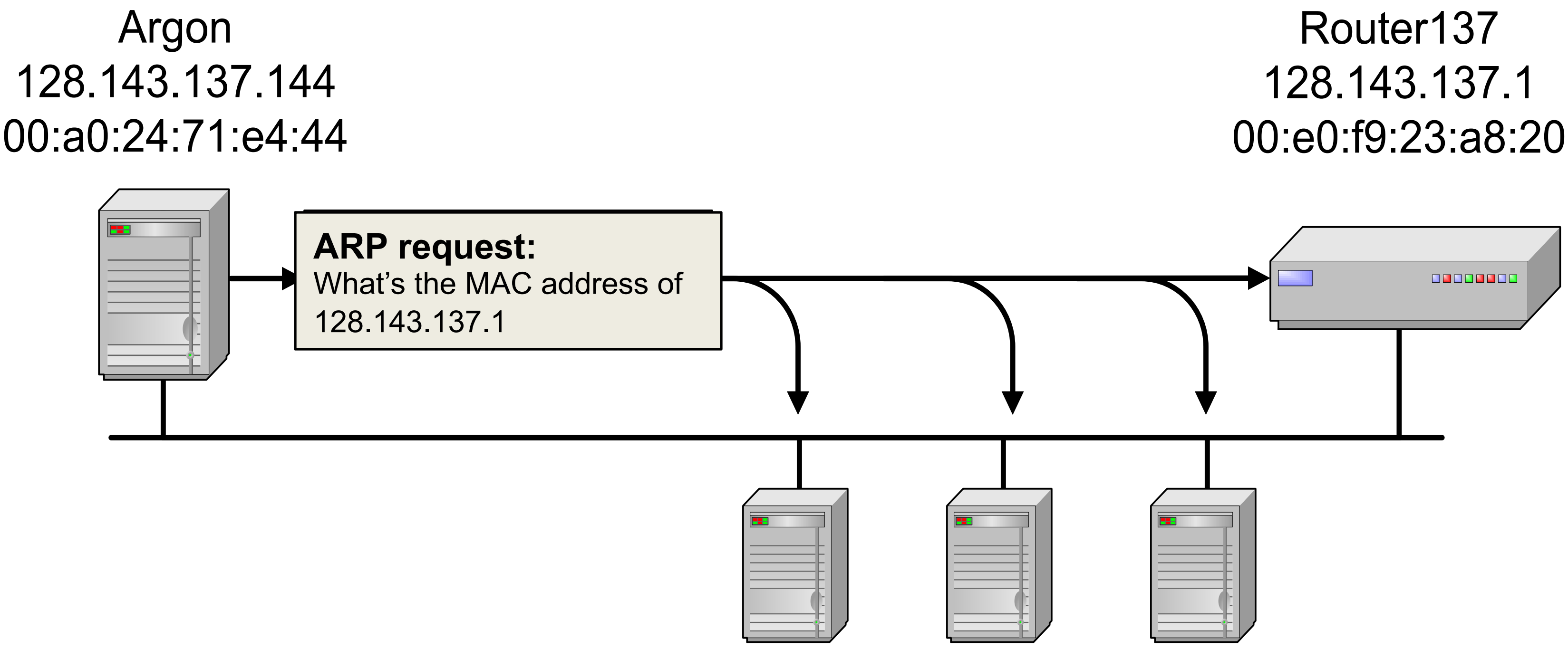
- The Internet is based on IP addresses
- Data link protocols (Ethernet, FDDI, ATM) may have different (MAC) addresses
- The ARP and RARP protocols perform the translation between IP addresses and MAC layer addresses
- We will discuss ARP for broadcast LANs, particularly Ethernet LANs
  - RFC 826
- RARP obsolete



Address Translation with ARP

ARP Request:

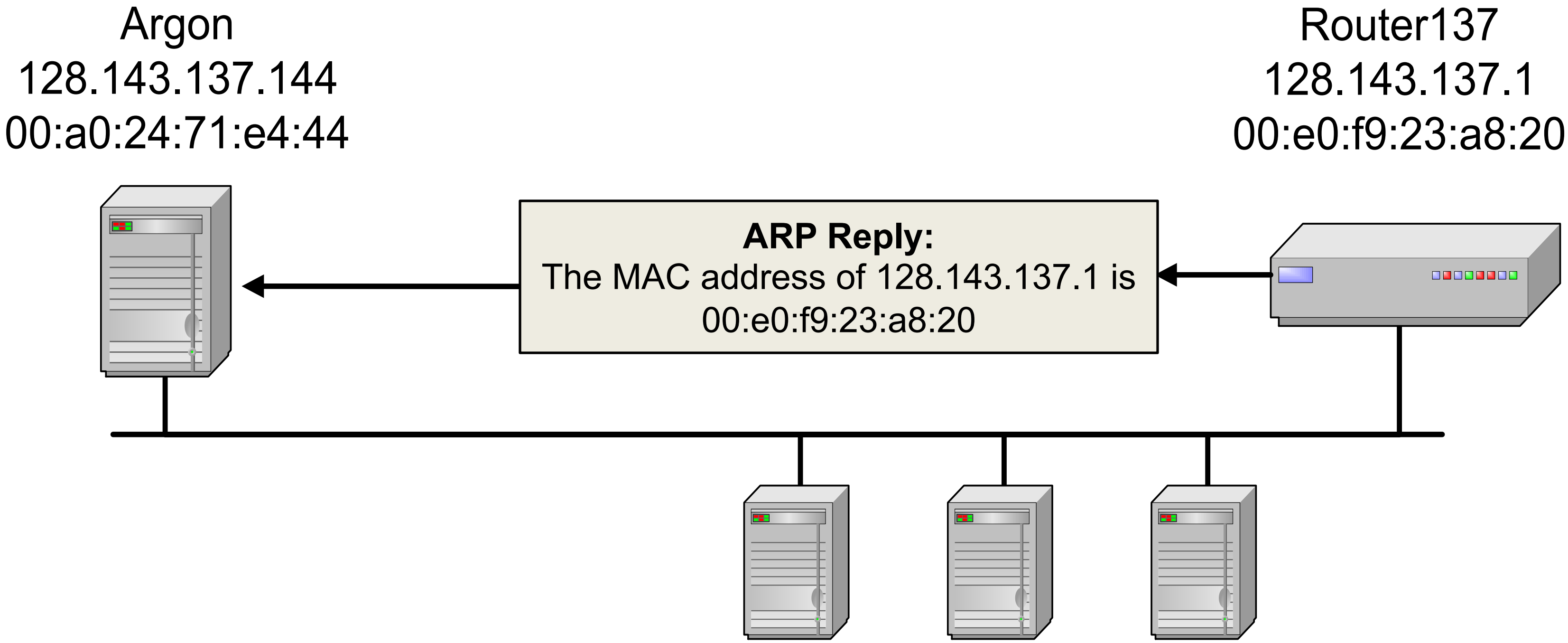
Argon broadcasts an ARP request to all stations on the network: **"What is the hardware address of 128.143.137.1?"**



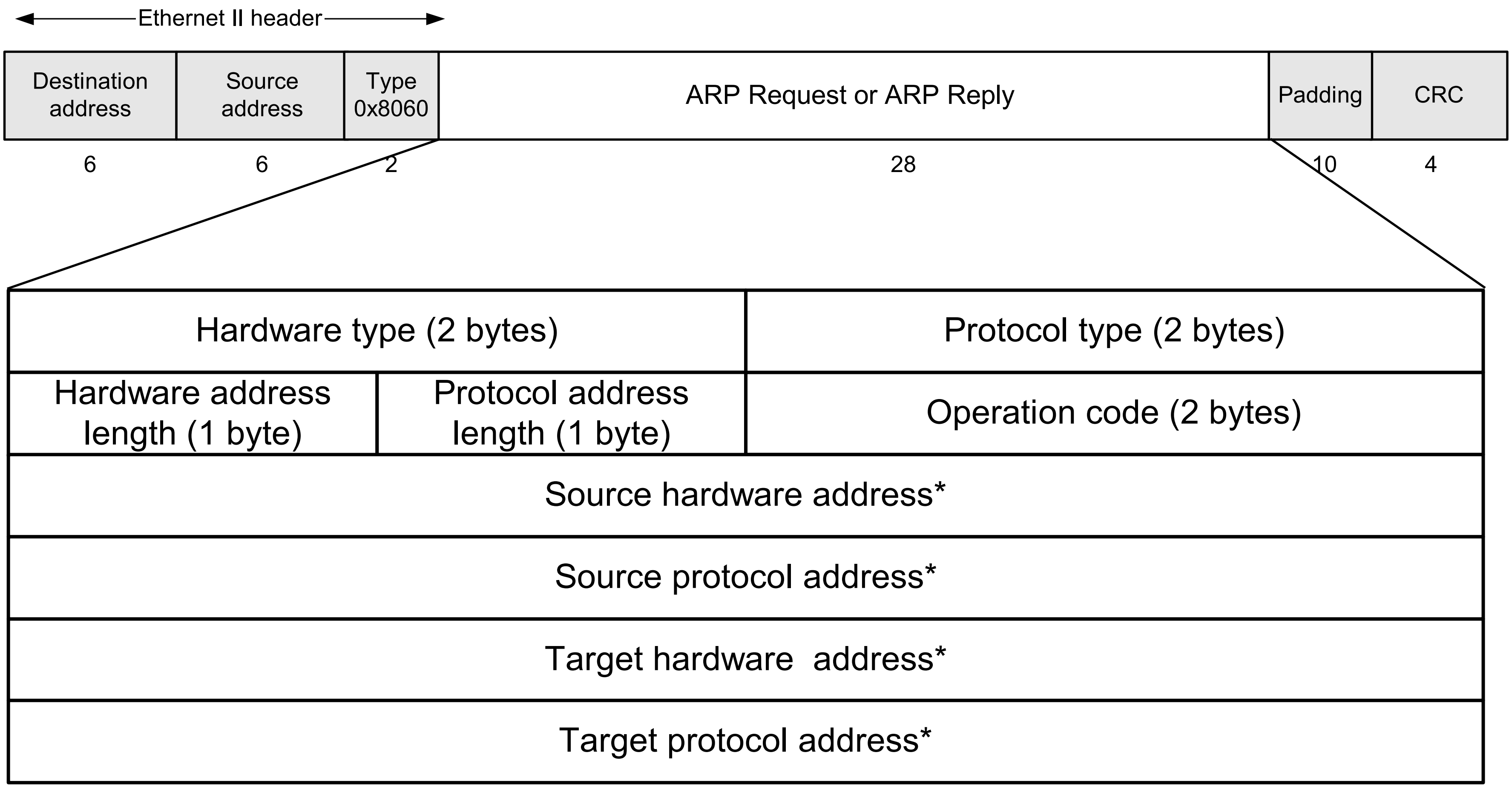
Address Translation with ARP

ARP Reply:

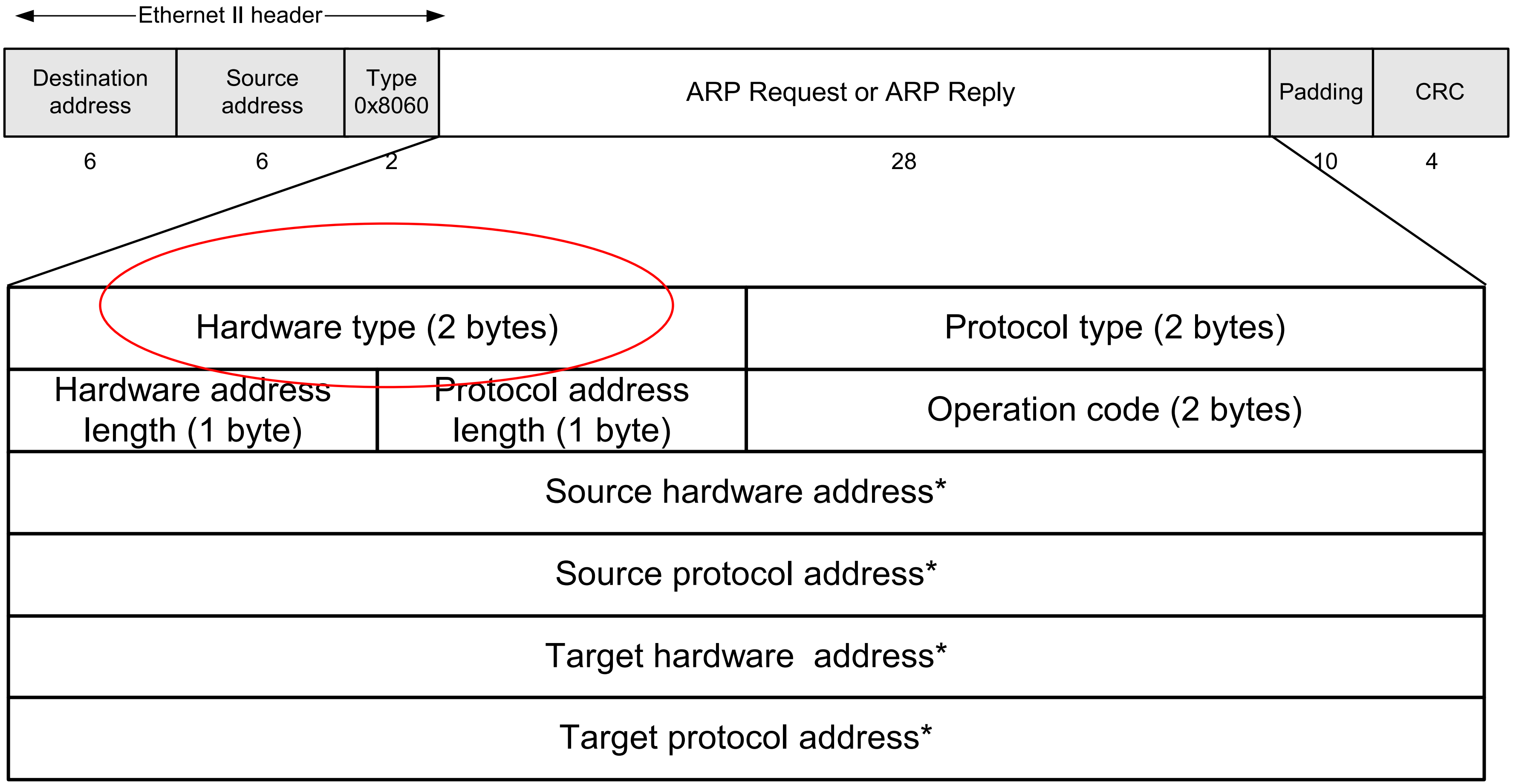
Router 137 responds with an ARP Reply which contains the hardware address



ARP Packet Format

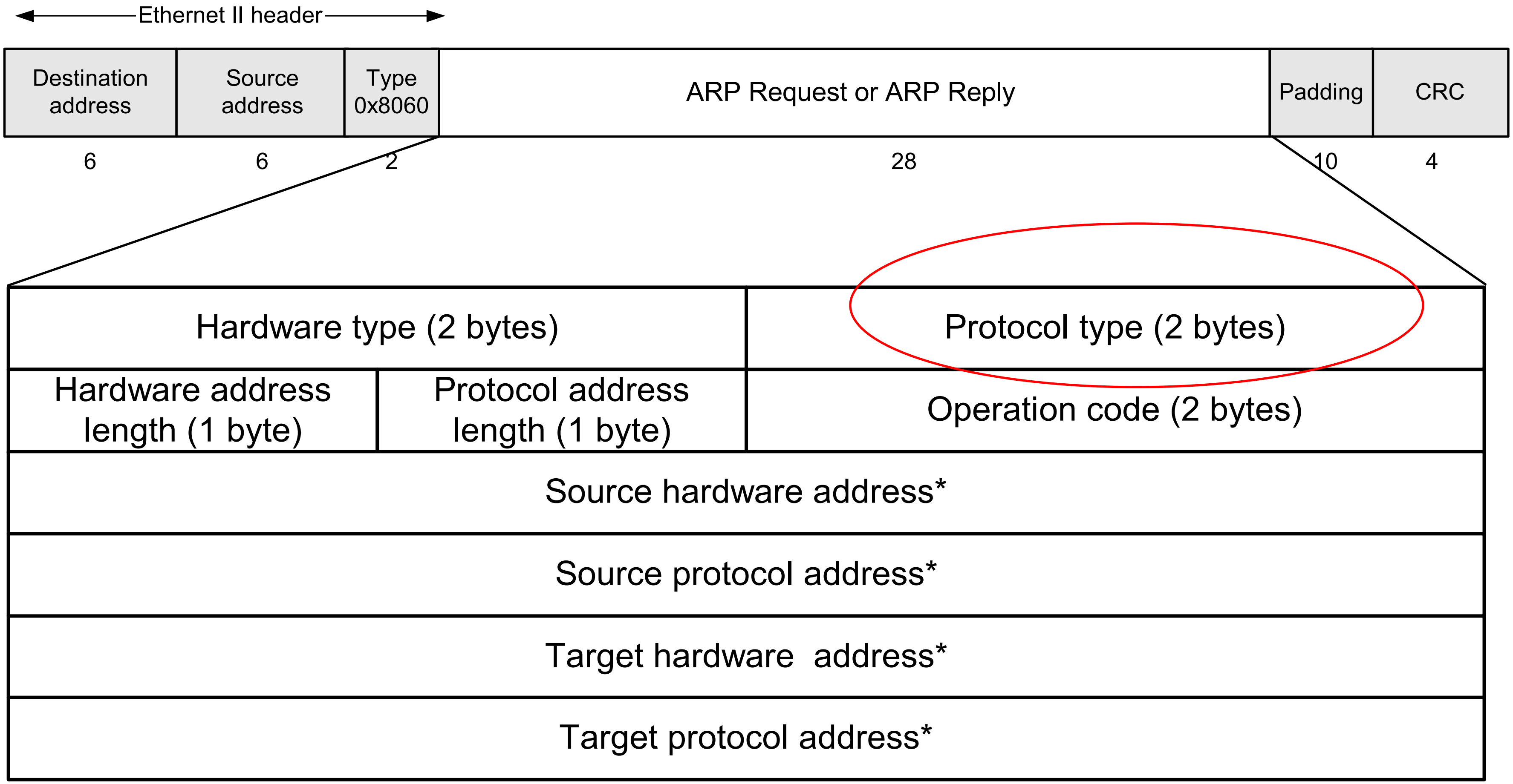


* Note: The length of the address fields is determined by the corresponding address length fields



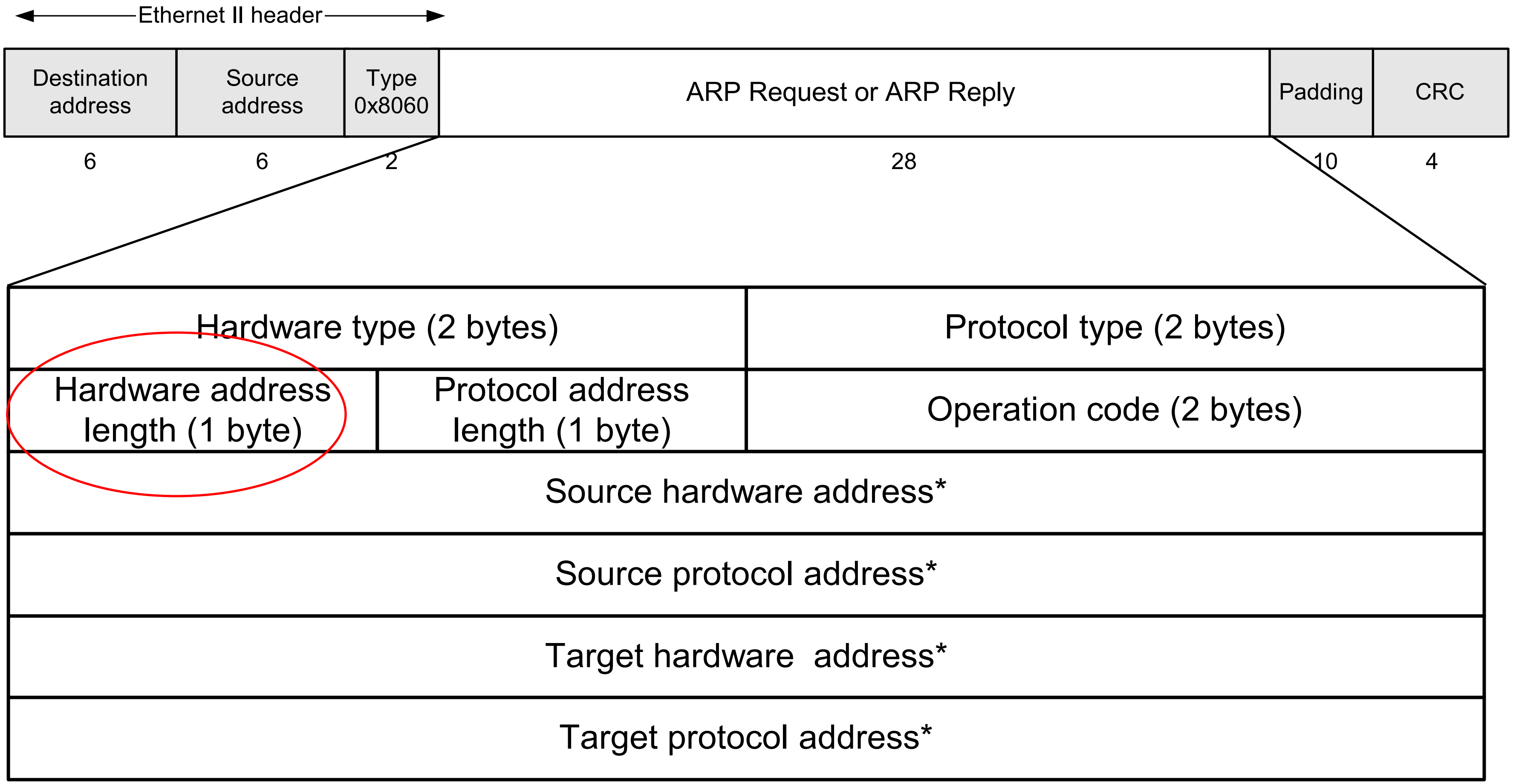
* Note: The length of the address fields is determined by the corresponding address length fields

- Hardware type: ether (1)



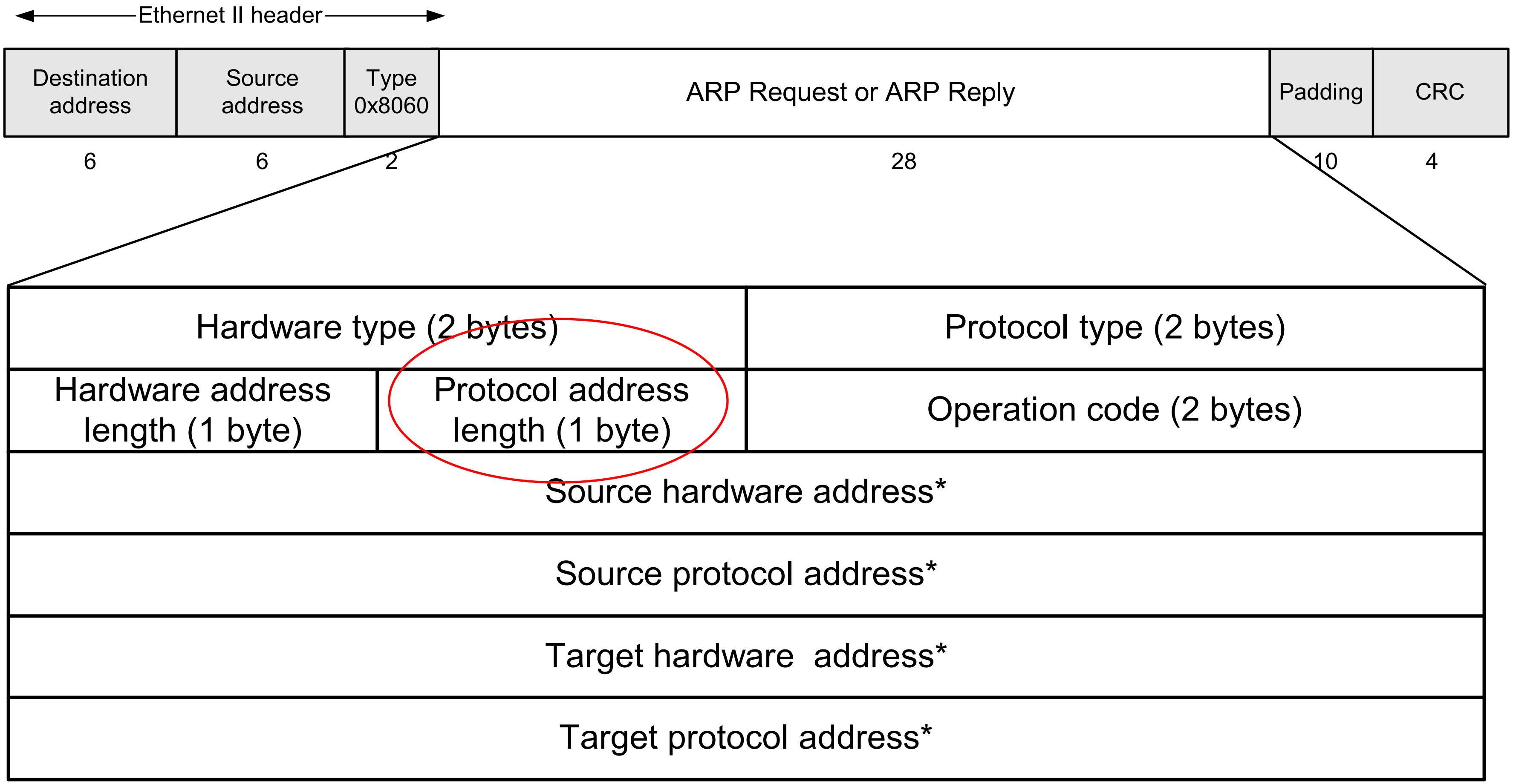
* Note: The length of the address fields is determined by the corresponding address length fields

- Prototype: taken from the set ether_type
  - IP: 0x0800



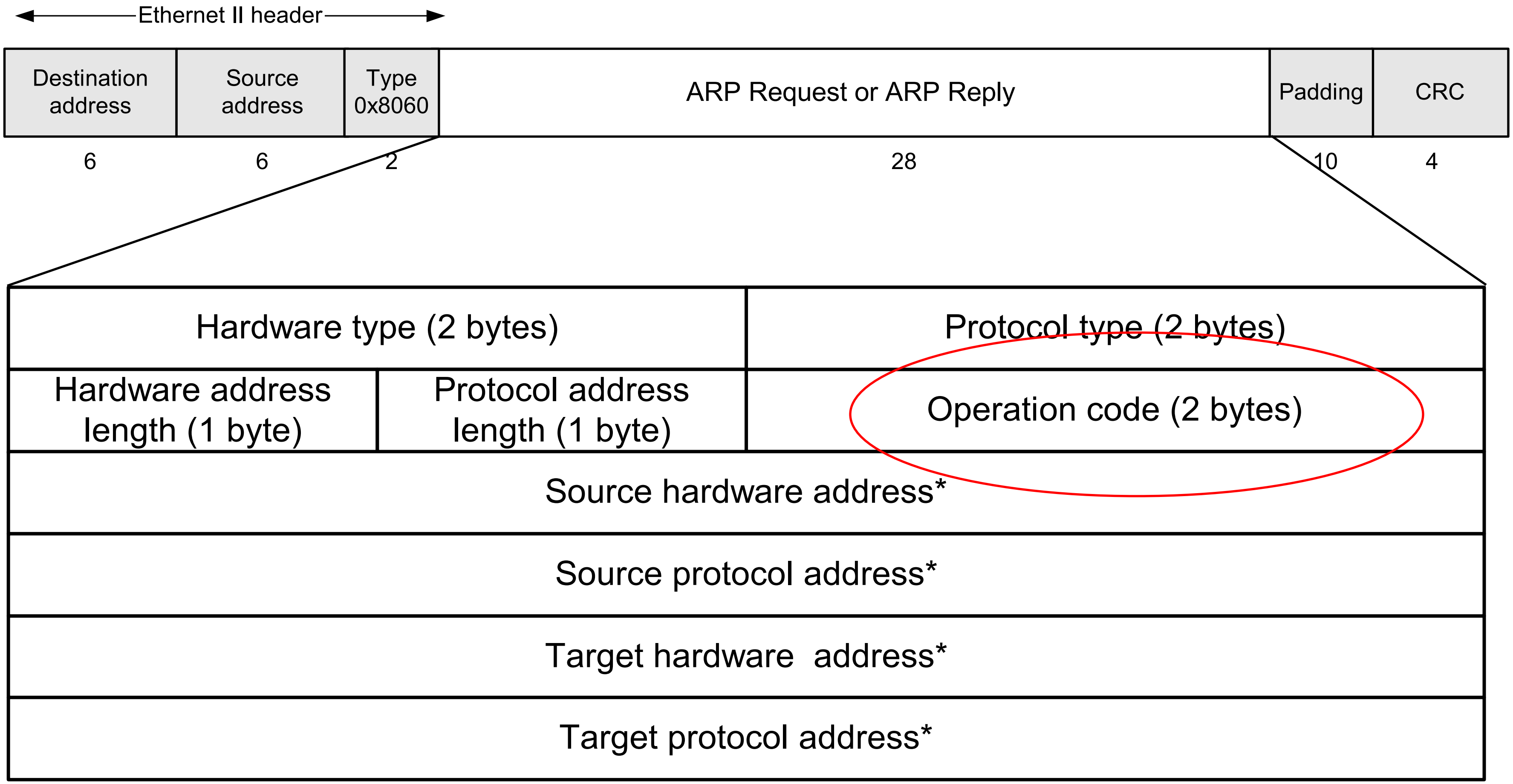
* Note: The length of the address fields is determined by the corresponding address length fields

- Hardware address length
  - Length of an Ethernet address



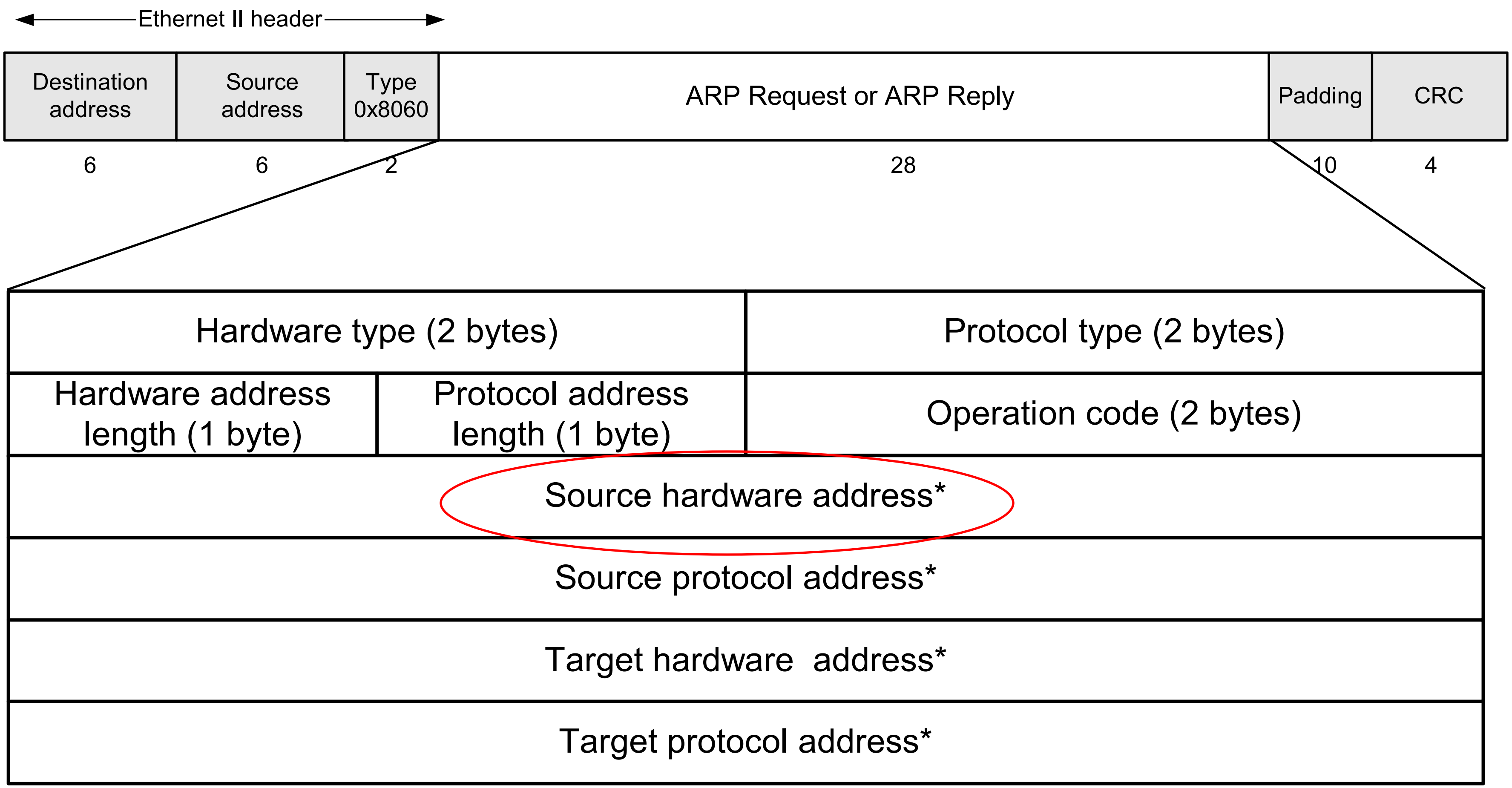
* Note: The length of the address fields is determined by the corresponding address length fields

- Protocol address length
  - Length of an IP address



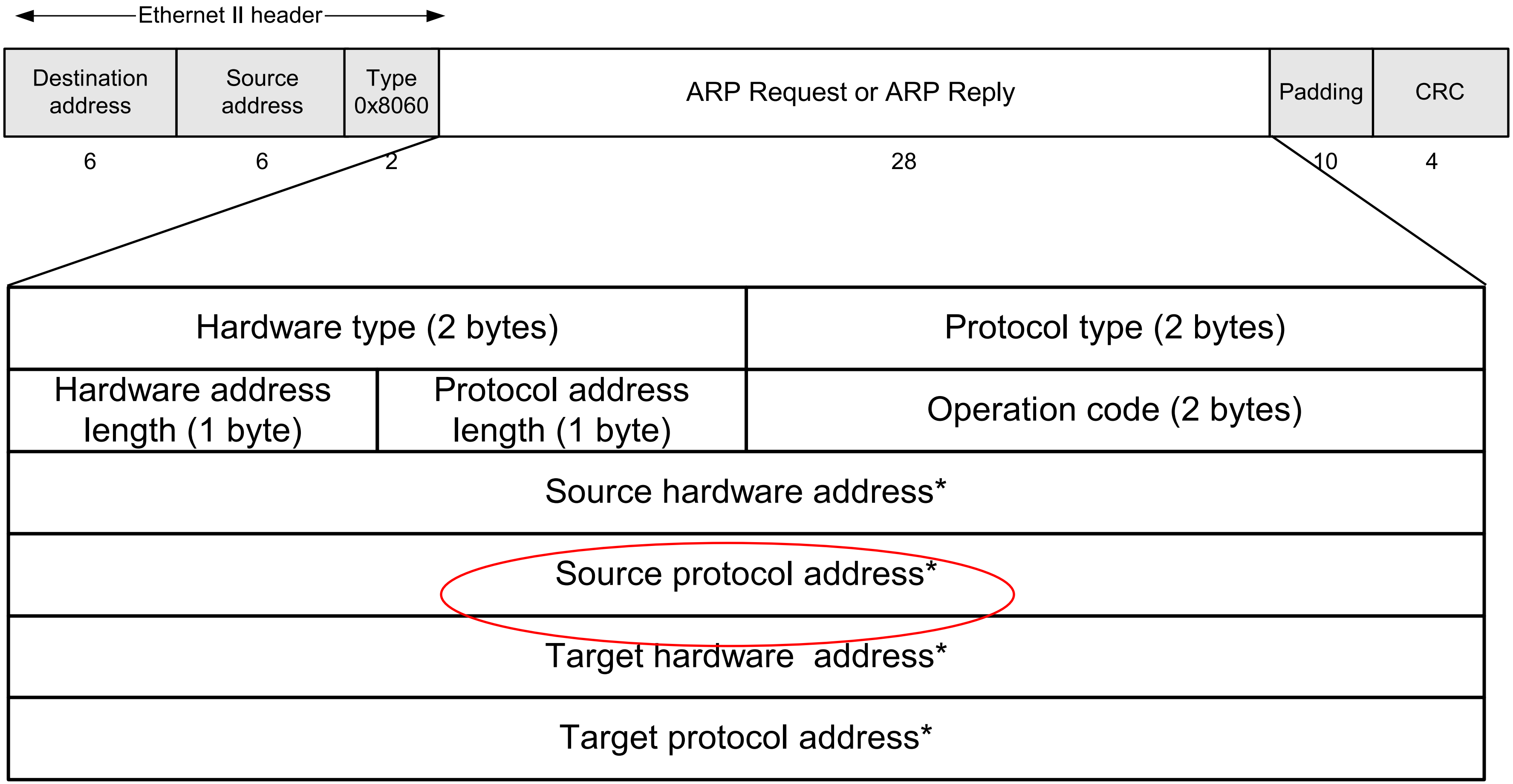
* Note: The length of the address fields is determined by the corresponding address length fields

- Opcode
  - ARP request: 1
  - ARP reply: 2



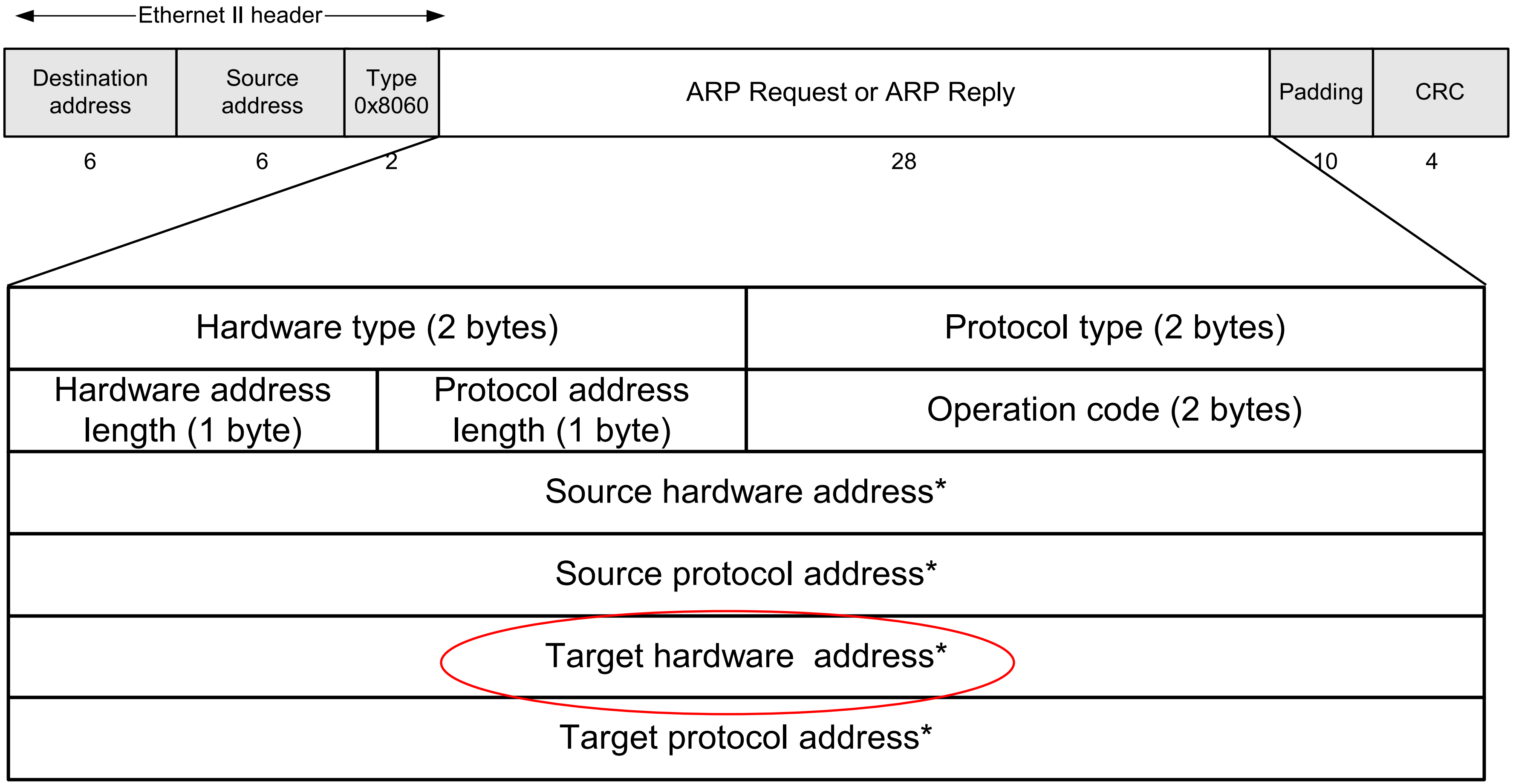
* Note: The length of the address fields is determined by the corresponding address length fields

- Source hardware address
  - Sender's Ethernet address



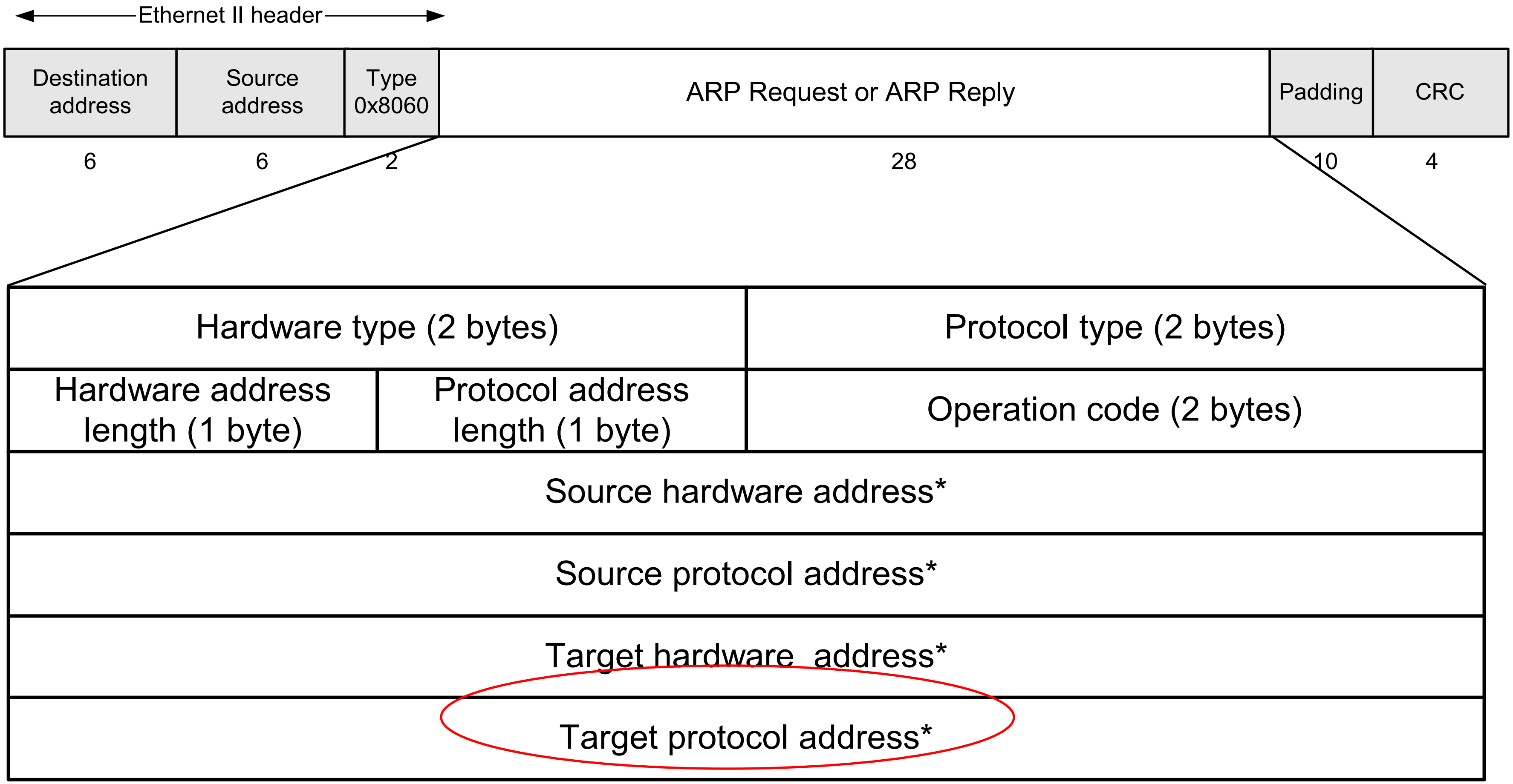
* Note: The length of the address fields is determined by the corresponding address length fields

- Source protocol address
  - Sender's protocol (IP) address



* Note: The length of the address fields is determined by the corresponding address length fields

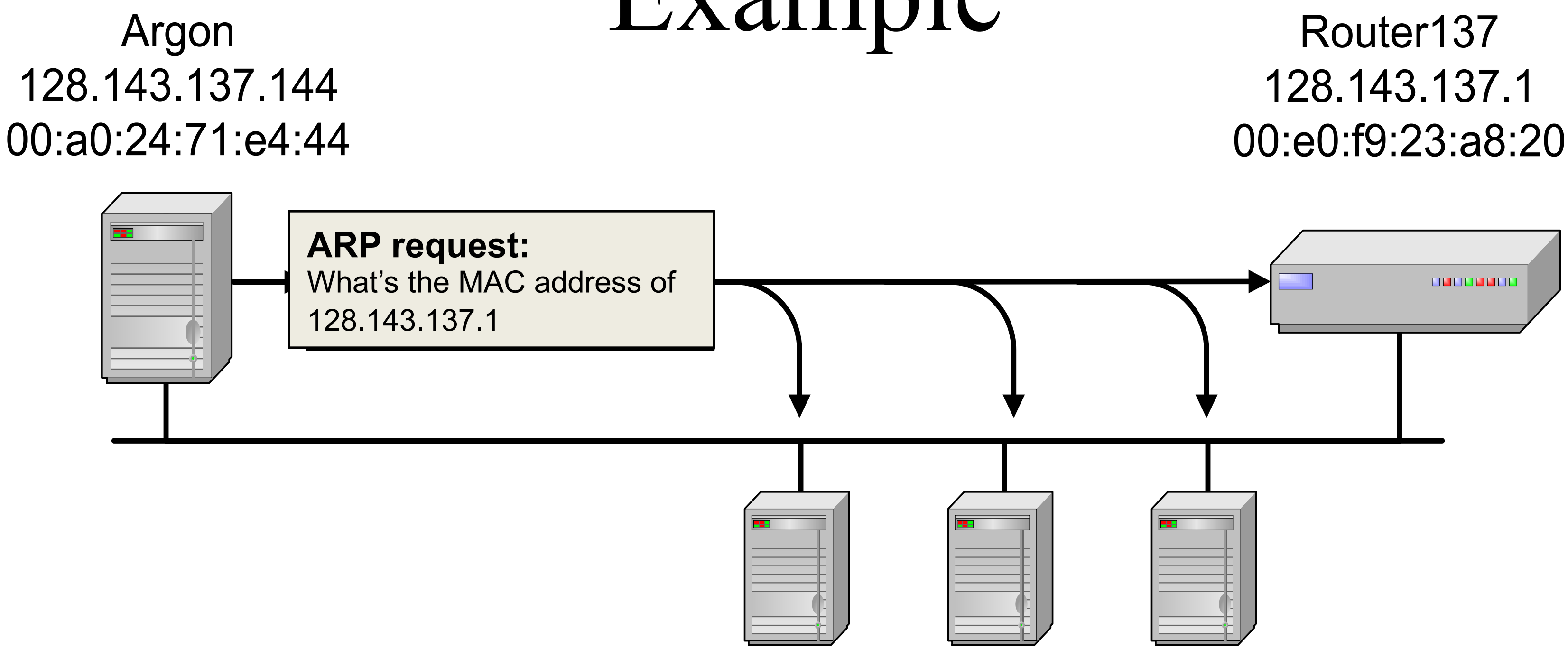
- Target hardware address
  - Request: empty
  - Reply: the target's Ethernet address



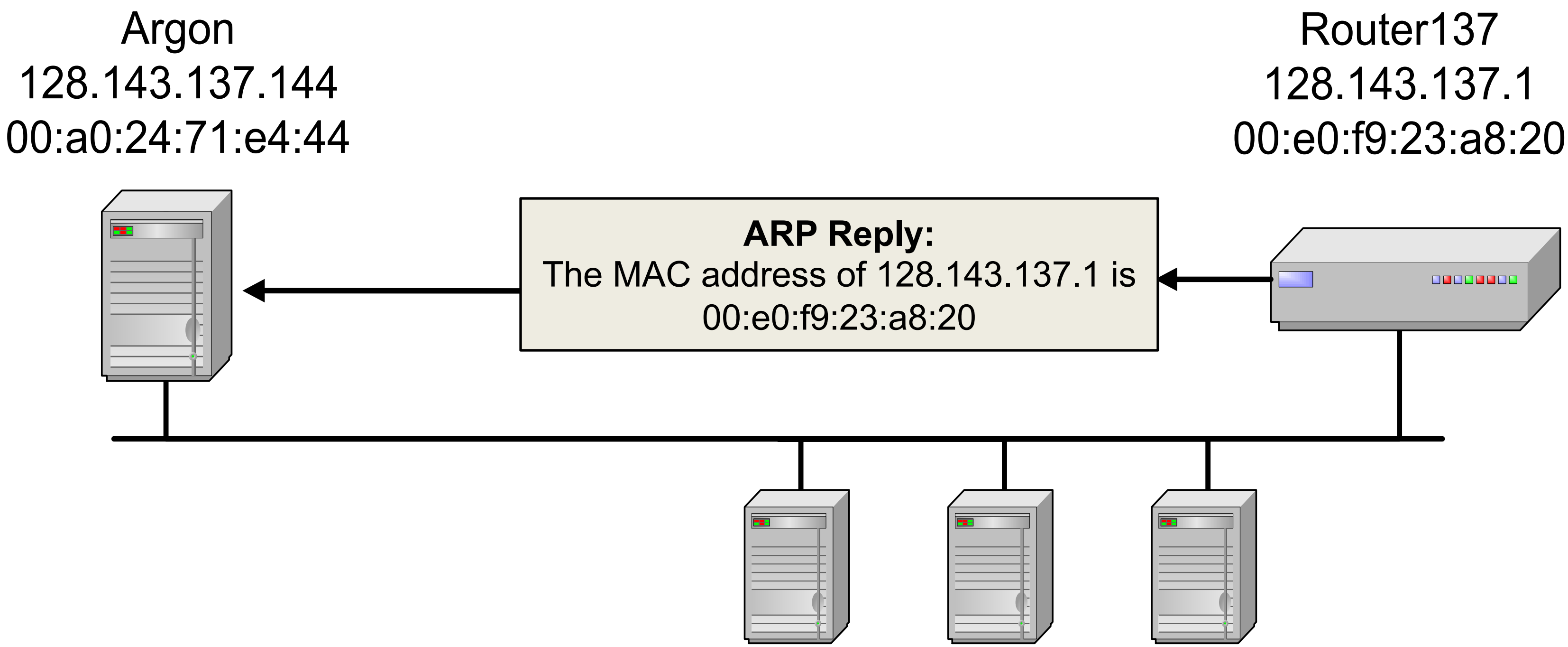
* Note: The length of the address fields is determined by the corresponding address length fields

- Target protocol address
  - Request: target IP address
  - Reply: destination IP address

Example



- *ARP Request from Argon is broadcasted:*
  - Source addr in Ethernet header: 00:a0:24:71:e4:44
  - Destination addr in Ethernet header: FF:FF:FF:FF:FF:FF
- Source hardware address: 00:a0:24:71:e4:44
- Source protocol address: 128.143.137.144
- Target hardware address: 00:00:00:00:00:00
- Target protocol address: 128.143.137.1



- *ARP Reply from Router137 is unicasted:*
  - Source addr: 00:e0:f9:23:a8:20
  - Dst addr: 00:a0:24:71:e4:44
- Source hardware address: 00:e0:f9:23:a8:20
- Source protocol address: 128.143.137.1
- Target hardware address: 00:a0:24:71:e4:44
- Target protocol address: 128.143.137.144

Comments

- ARP requests: broadcast
  - Other hosts learn the source IP/MAC mapping

- ARP replies: unicast

ARP Cache

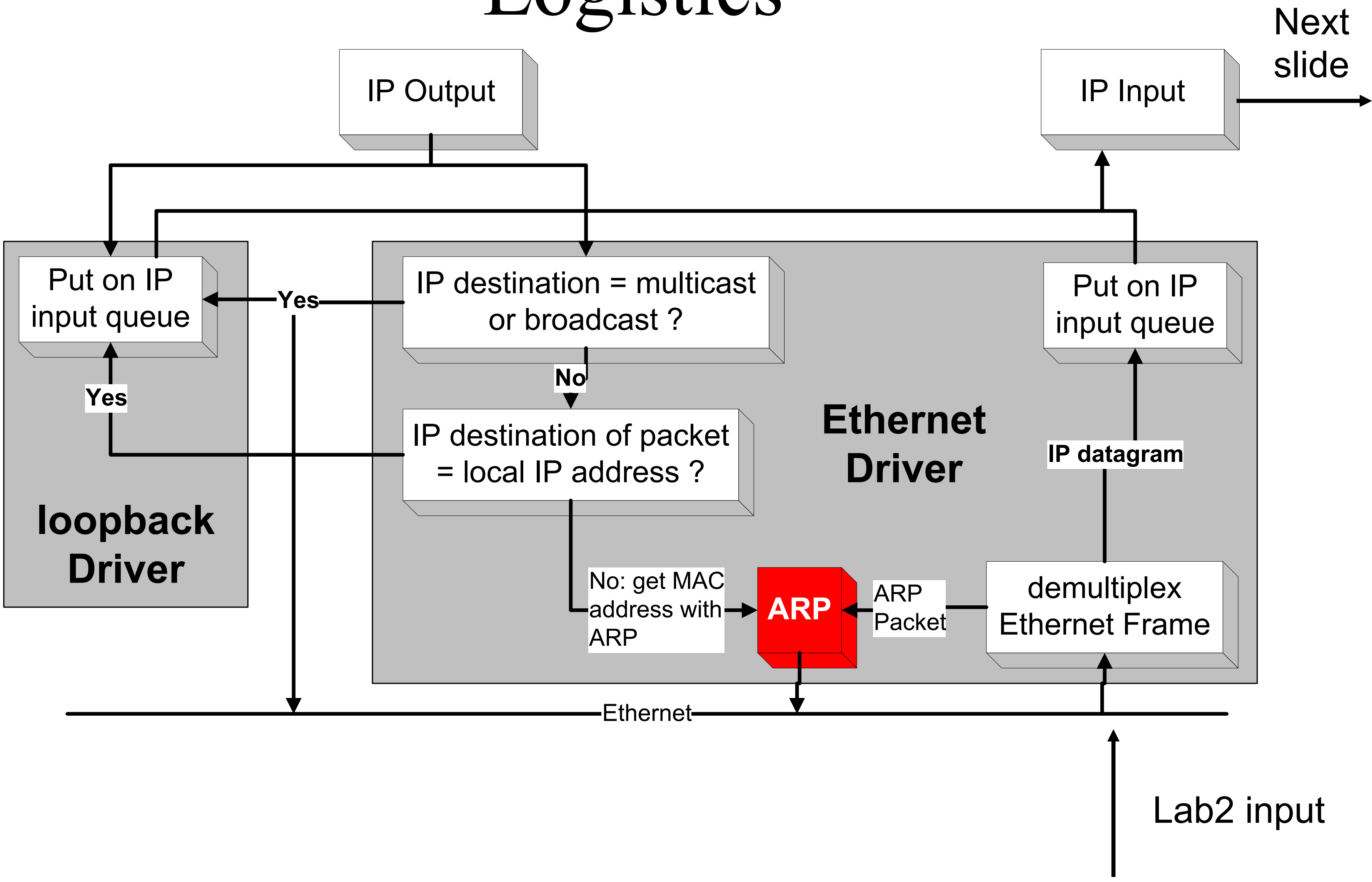
- Since sending an ARP request/reply for each IP datagram is inefficient, hosts maintain a cache (ARP Cache) of current entries. The entries expire after a time interval.

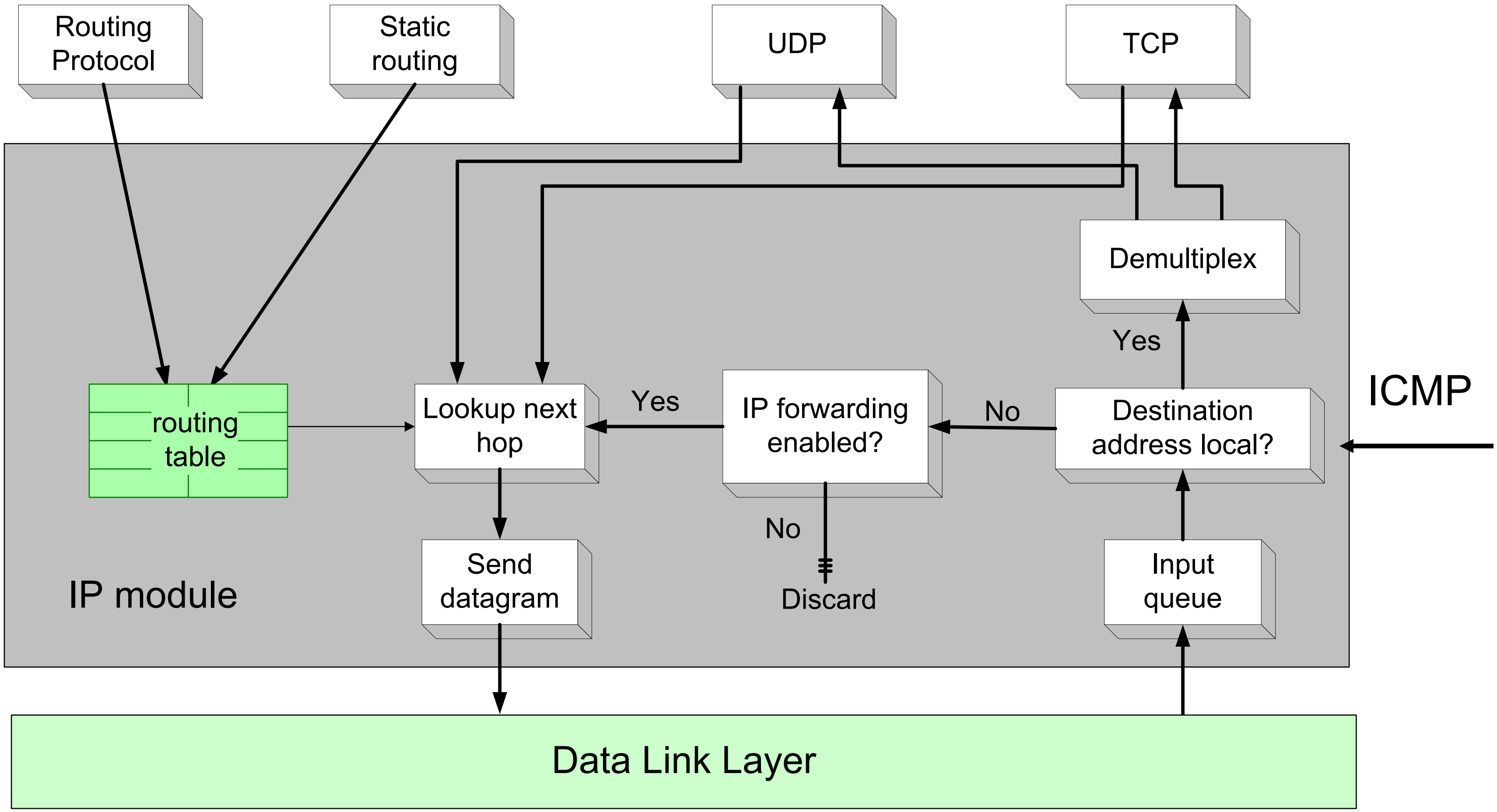
- Linux: arp -a

- Contents of the ARP Cache:
  (128.143.71.37) at 00:10:4B:C5:D1:15 [ether] on eth0
  (128.143.71.36) at 00:B0:D0:E1:17:D5 [ether] on eth0
  (128.143.71.35) at 00:B0:D0:DE:70:E6 [ether] on eth0
  (128.143.136.90) at 00:05:3C:06:27:35 [ether] on eth1
  (128.143.71.34) at 00:B0:D0:E1:17:DB [ether] on eth0
  (128.143.71.33) at 00:B0:D0:E1:17:DF [ether] on eth0

Putting it together

IP Forwarding Implementation Logistics




Routing Protocol
Static routing
UDP
TCP
Demultiplex
Yes
ICMP
routing table
Lookup next hop
Yes
IP forwarding enabled?
No
Destination address local?
No
Discard
Send datagram
Input queue
IP module
Data Link Layer

IP Forwarding Logistics (Lab 2)

1. Sanity-check
   - Meets minimum length and has correct checksum

2. Update header
   - Decrement the TTL by 1, and compute the packet checksum over the modified header.

3. Next hop IP lookup
   - Find out which entry in the routing table has the longest prefix match with the destination IP address.

4. Next hop MAC lookup
   - Check the ARP cache for the next-hop MAC address corresponding to the next-hop IP. If it's there, send it. Otherwise, send an ARP request for the next-hop IP (if one hasn't been sent within the last second), and add the packet to the queue of packets waiting on this ARP request.

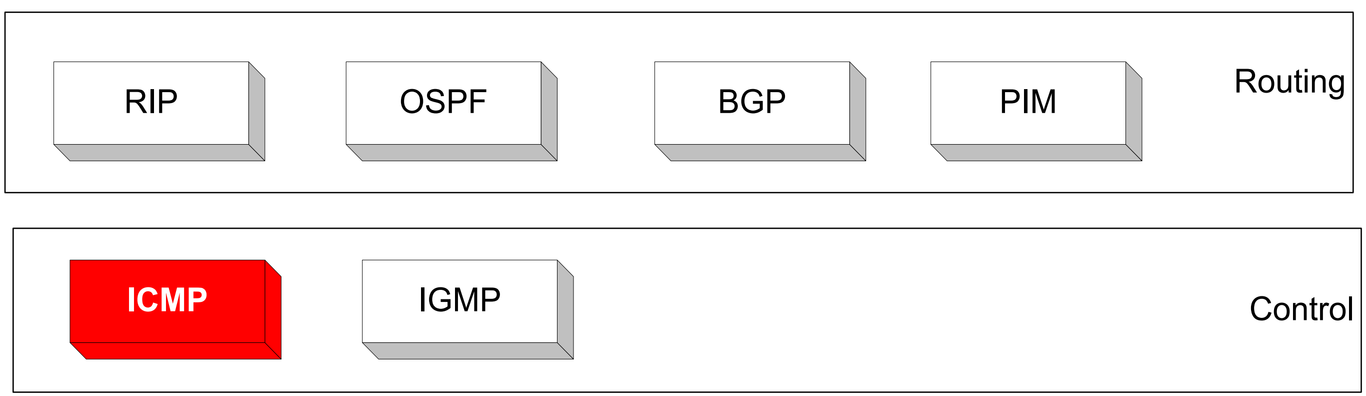
5. Error reporting

Error reporting

- Internet Control Message Protocol (ICMP)
  - Ill-formatted packets
  - TTL == 0
  - ARP receives no reply
  - No protocol or application running at the destination
  - No routing table match
  - …

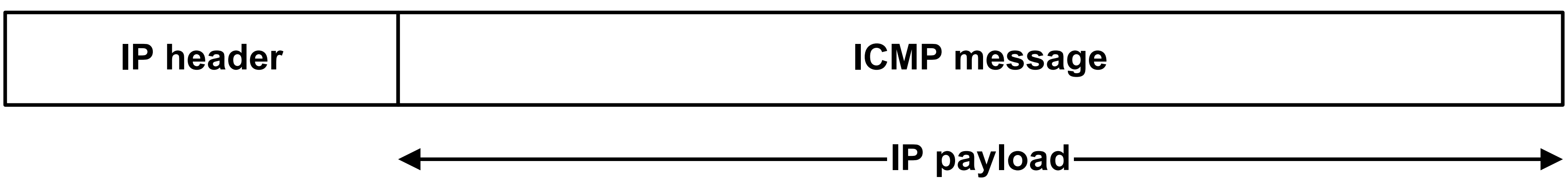
Location in the protocol stack

- The IP (Internet Protocol) relies on several other protocols to perform necessary control and routing functions:
  - Control functions (ICMP)
  - Multicast signaling (IGMP)
  - Setting up forwarding tables (RIP, OSPF, BGP, PIM, …)

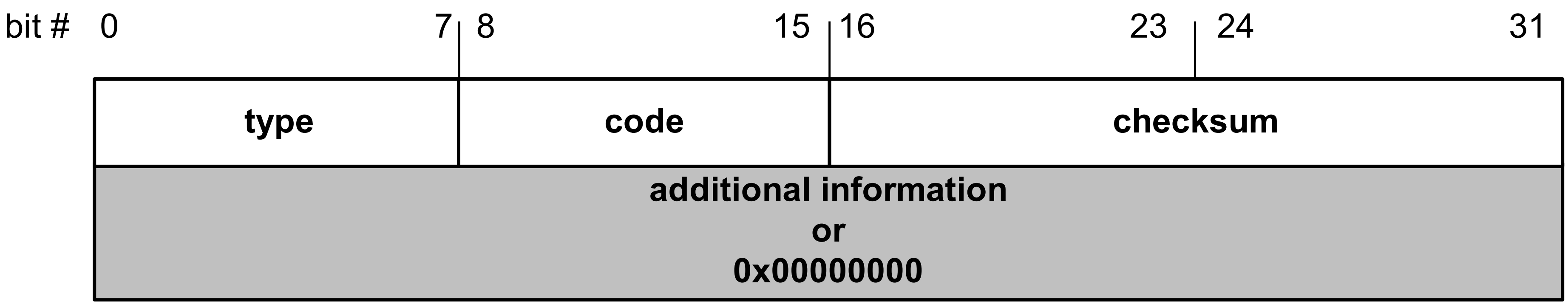


Overview

- The **Internet Control Message Protocol (ICMP)** is a helper protocol that supports IP with facility for
  - Error reporting
  - Simple queries
  - ICMP messages are encapsulated as IP datagrams



ICMP message format



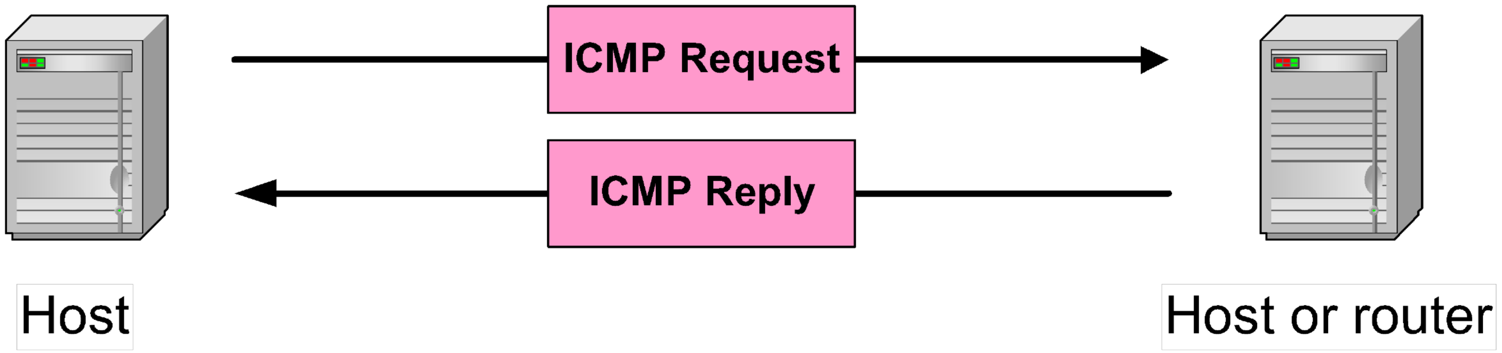
4 byte header:

- Type (1 byte): type of ICMP message
- Code (1 byte): subtype of ICMP message
- Checksum (2 bytes): similar to IP header checksum. Checksum is calculated over the entire ICMP message

If there is no additional data, there are 4 bytes set to zero.

→ each ICMP message is at least 8 bytes long

ICMP Query message



ICMP query:

- Request sent by host to a router or host
- Reply sent back to querying host

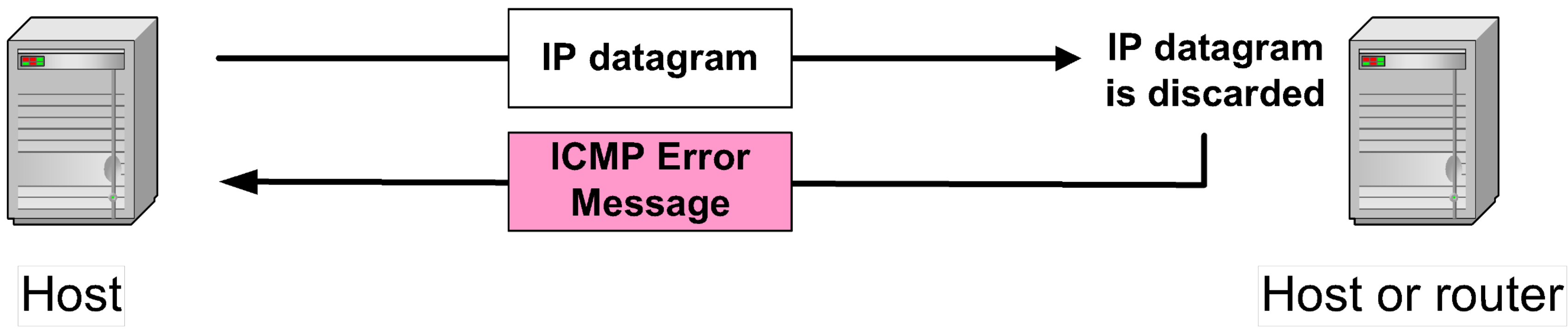
Example of ICMP Queries

Type/Code:	Description	
8/0	Echo Request	The ping command uses Echo Request/ Echo Reply
0/0	Echo Reply	
13/0	Timestamp Request	
14/0	Timestamp Reply	

Extension (RFC 1256):

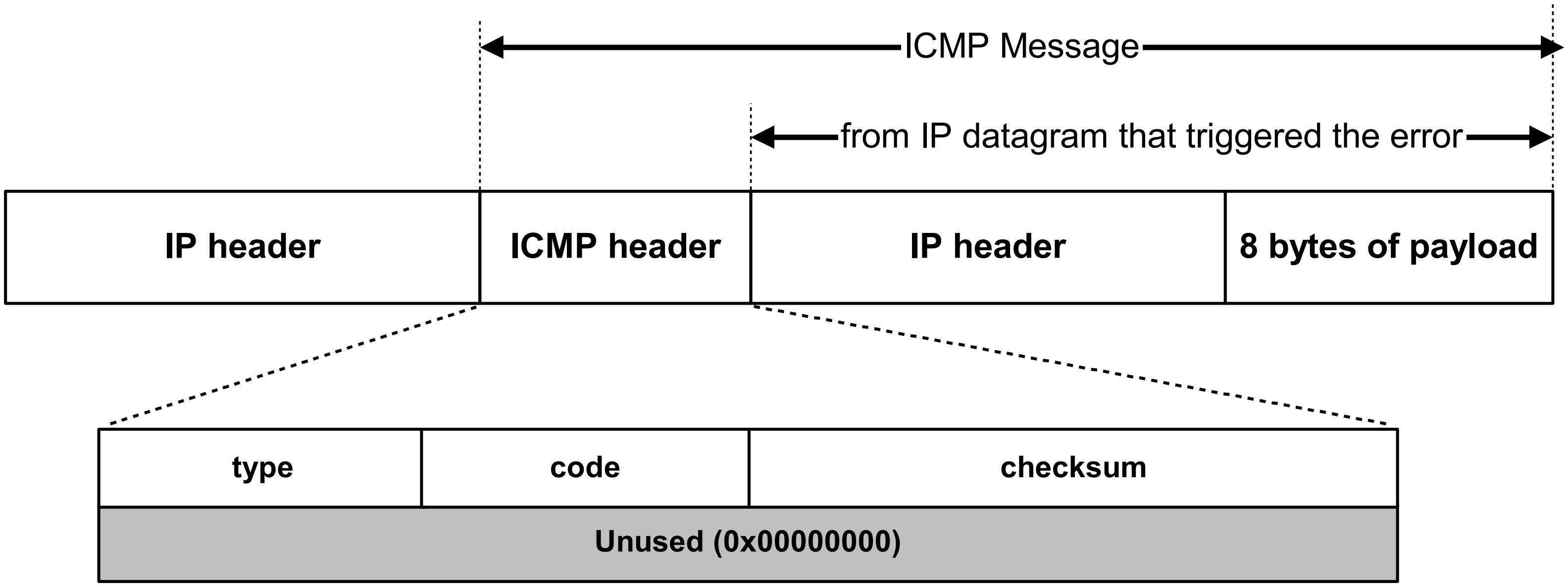
10/0	Router Solicitation
9/0	Router Advertisement

ICMP Error message



- ICMP error messages report error conditions
- Typically sent when a datagram is discarded
- Error message is often passed from ICMP to the application program

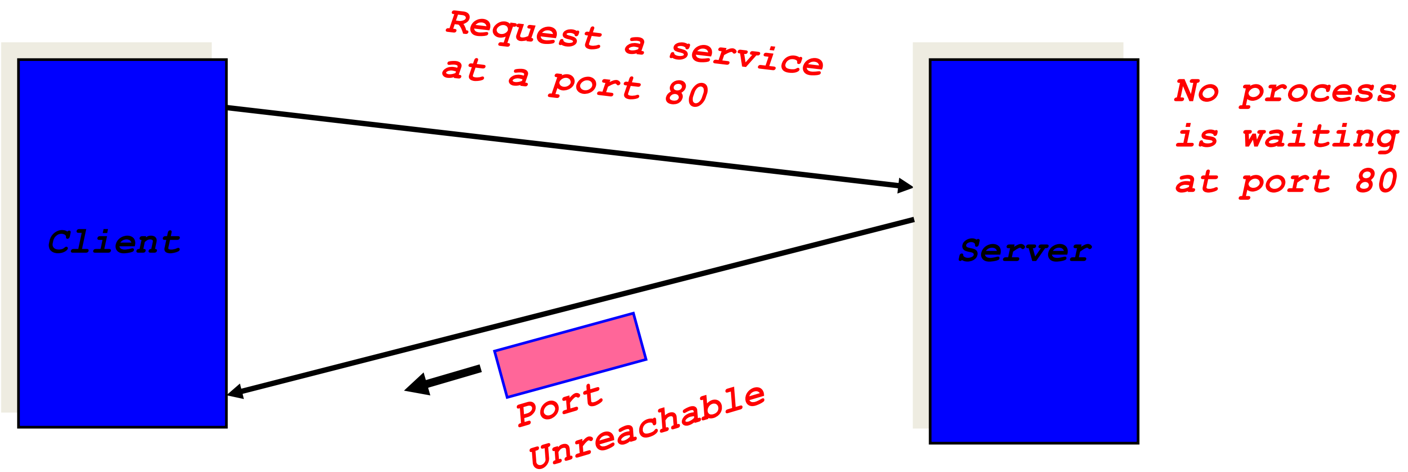
ICMP Error message



- ICMP error messages include the complete IP header and the first 8 bytes of the payload (typically: UDP, TCP)

Example: ICMP Port Unreachable

- RFC 792: If, in the destination host, the IP module cannot deliver the datagram because the indicated protocol module or process port is not active, the destination host may send a destination unreachable message to the source host.



Common ICMP Error messages

Type	Code	Description	
3	0–5	Destination unreachable	Notification that an IP datagram could not be forwarded and was dropped. The code field contains an explanation. (traceroute)

Some subtypes of the "Destination Unreachable"

Code	Description	Reason for Sending
0	Network Unreachable	No routing table entry is available for the destination network.

Summary

- IP fragmentation
- ARP
- ICMP