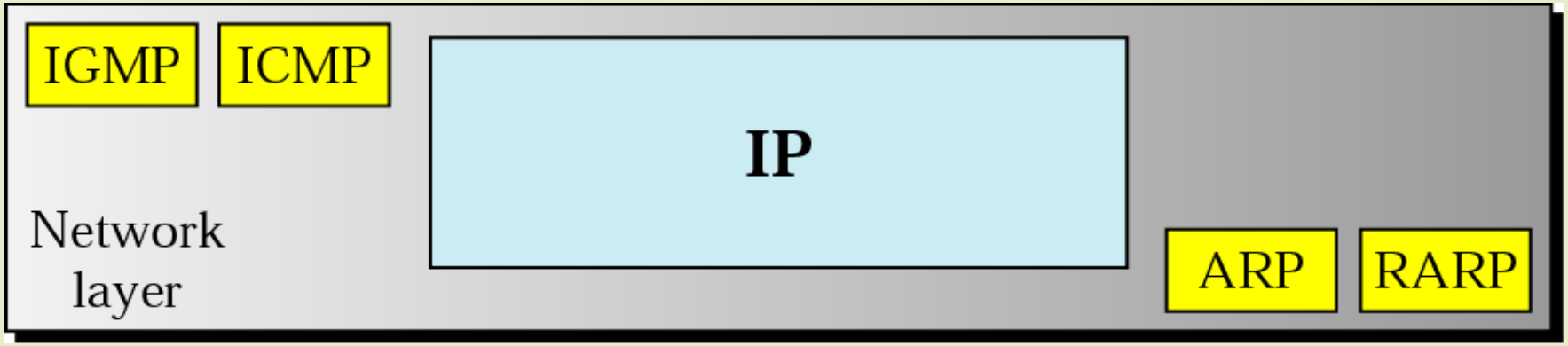
数据通信基础
第20章 (网络层协议)
Network Layer Protocols

康教授
北京理工大学信息与电子学院

Figure 20.1 **Protocols at network layer**



Address Resolution Protocol (ARP)

Problem – if a host wants to contact a known destination IP address, what MAC address should it put in the Ethernet frame ?

Solution – it should use ARP to broadcast a message to all hosts on the network asking for the MAC address of the given IP address

(it must be a 'broadcast' – why?)

Figure 20.2 ARP operation

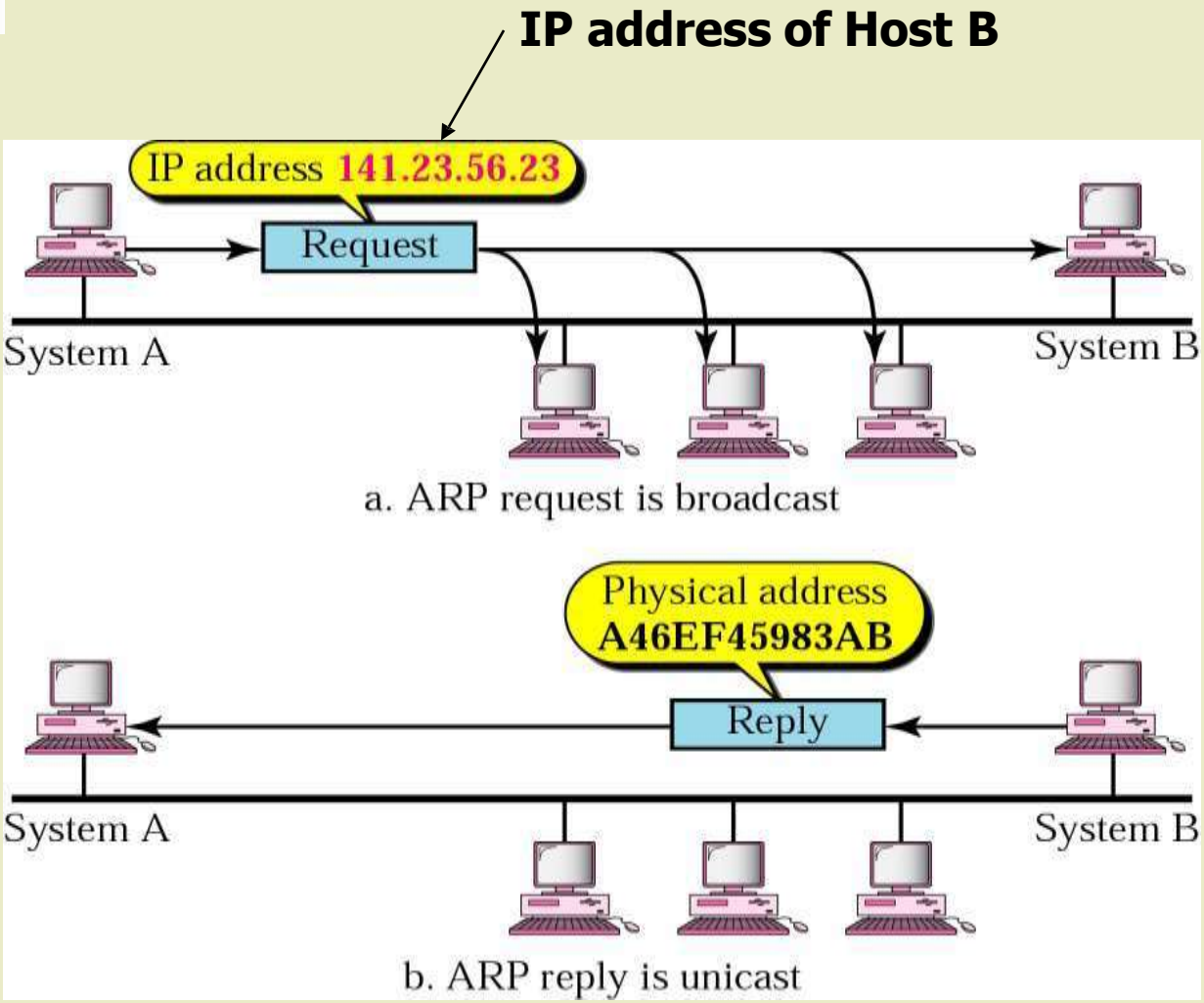
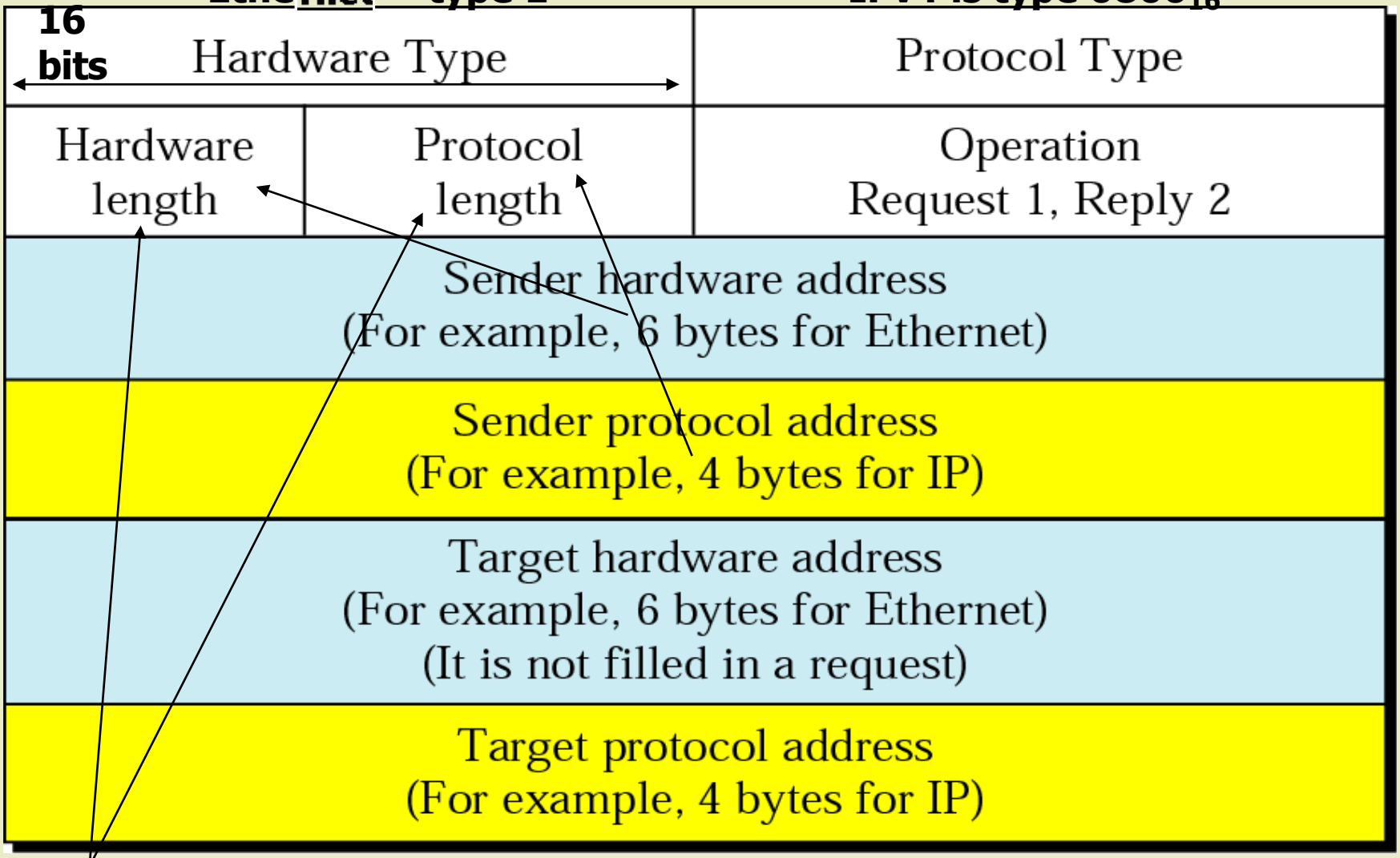


Figure 20.3 ARP packet

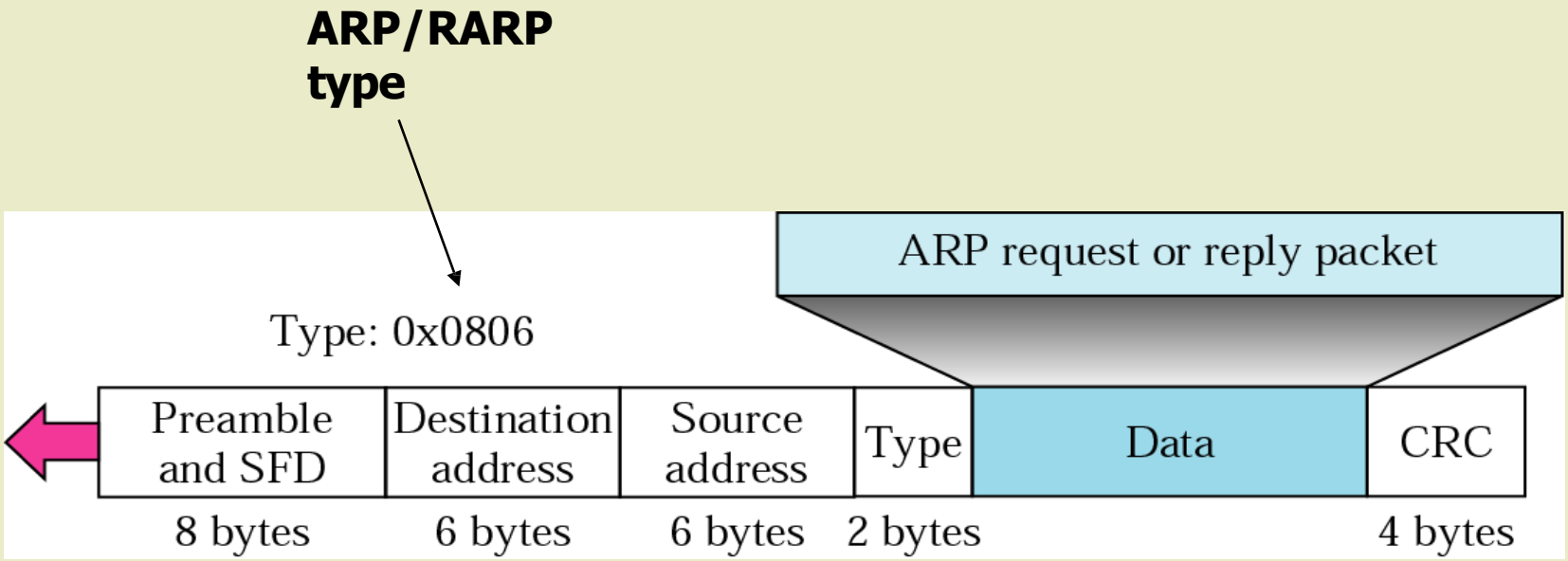
Note: 4 – byte boundaries are used on this type of diagram

Ethernet = type 1 **IPv4 is type 0800_{16}**



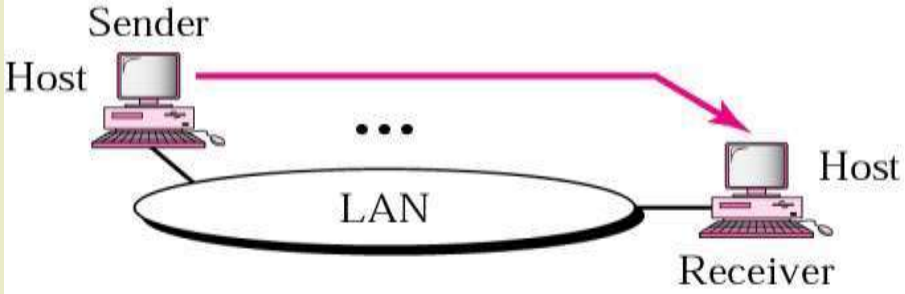
These two 'lengths' define the size of the rest of the packet

Figure 20.4 Encapsulation of ARP packet

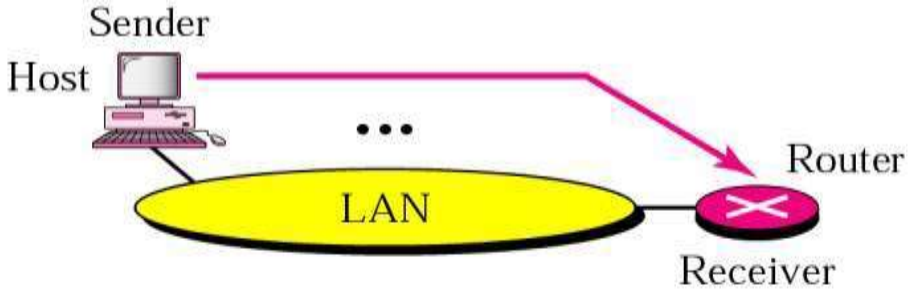


Note that the ARP packet is the 'data' for an Ethernet frame – this is 'Encapsulation'

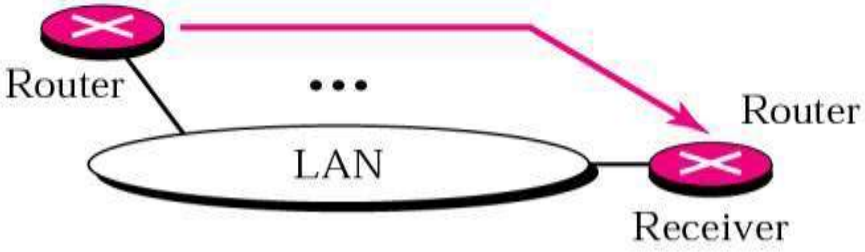
Figure 20.5 Four cases using ARP



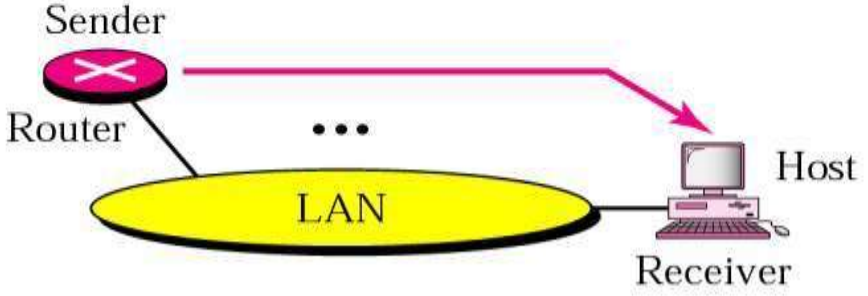
Case 1. A host has a packet to send to another host on the same network.



Case 2. A host wants to send a packet to another host on another network.
It must first be delivered to the appropriate router.



Case 3. A router receives a packet to be sent to a host on another network.
It must first be delivered to the appropriate router.



Case 4. A router receives a packet to be sent to a host on the same network.

All the above require ARP to discover the MAC address to be inserted into the Ethernet frame

Note:

An ARP request is broadcast – it must be, since it does not know the MAC address yet – the broadcast MAC address is 111111111111_{16} (48 bits)

An ARP reply is unicast – it replies to the address that made the request

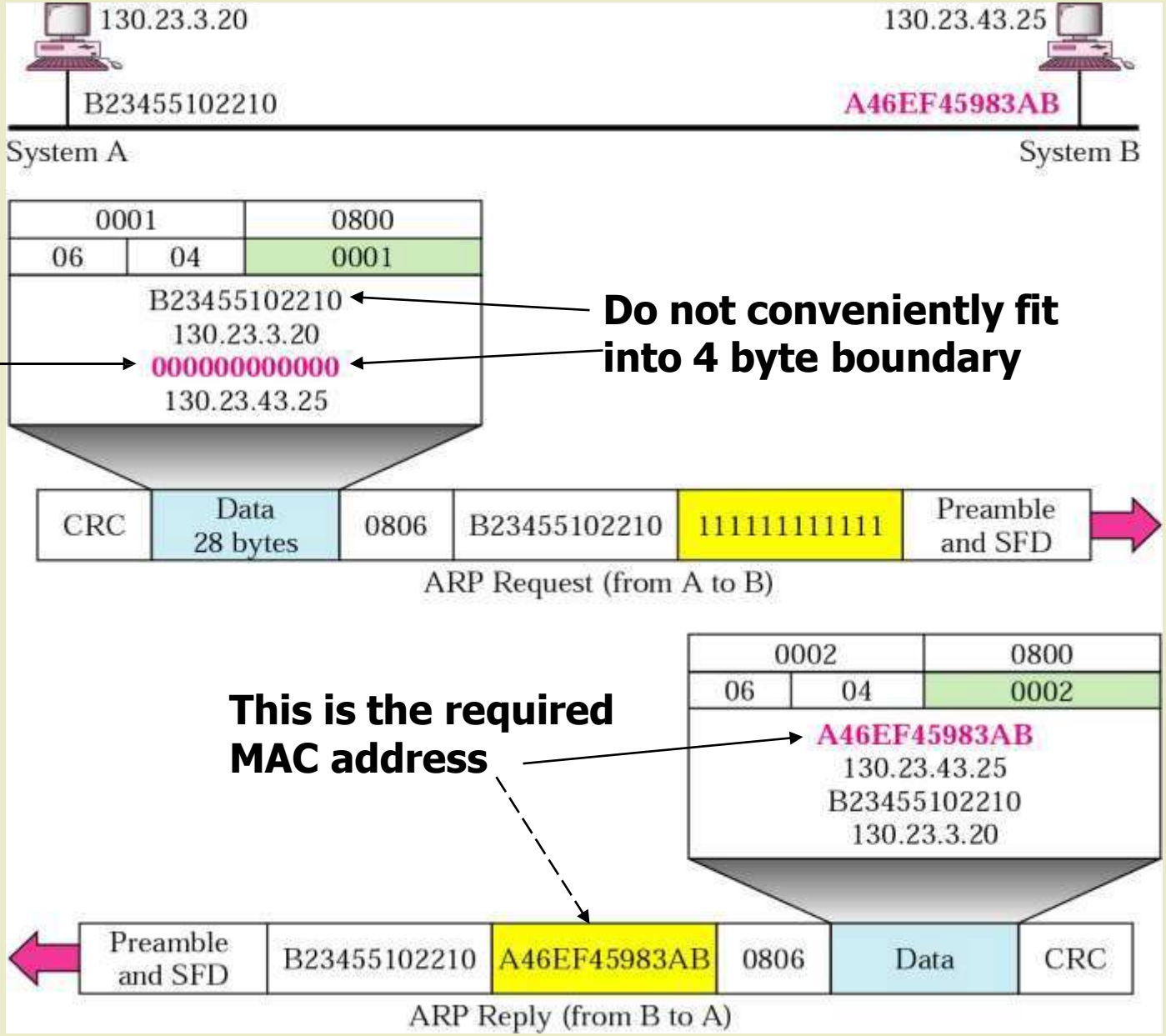
Example 1

A host with IP address 130.23.3.20 and physical address B23455102210 has a packet to send to another host with IP address 130.23.43.25 and physical address A46EF45983AB. The two hosts are on the same Ethernet network. Show the ARP request and reply packets encapsulated in Ethernet frames.

Solution

Figure 20.6 shows the ARP request and reply packets. Note that the ARP data field in this case is 28 bytes, and that the individual addresses do not fit in the 4-byte boundary. That is why we do not show the regular 4-byte boundaries for these addresses. Note that we use hexadecimal for every field except the IP addresses.

Figure 20.6 Example 1

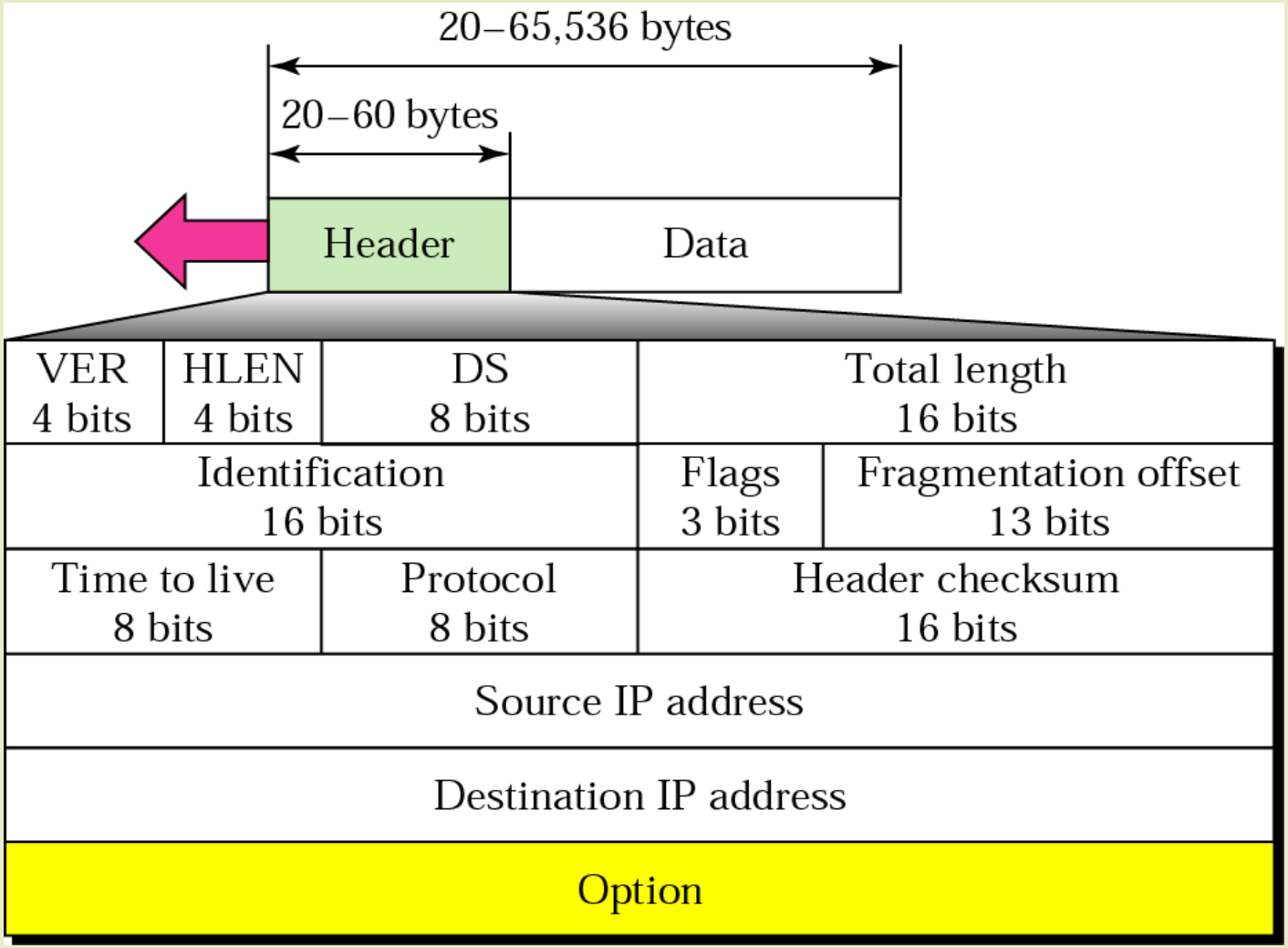


20.2 IPv4 protocol

Datagram – the equivalent of an Ethernet 'frame' is called a datagram or a packet at layer 3

Fragmentation – the process of breaking up large datagrams so that they can be transferred over a given network link

Figure 20.7 **IP datagram**



VER – IP version, currently 4

HLEN – header length expressed in 4 byte blocks note that 'options' make the header length variable

DS – defines 'quality of service (QoS)

Total length field defines the total length of the datagram including the header

Time to live (TTL) – decremented after each 'hop' – avoids packets circulating endlessly

Protocol being carried– e.g. TCP, UDP, ICMP, etc.

Figure 20.8 **Multiplexing**

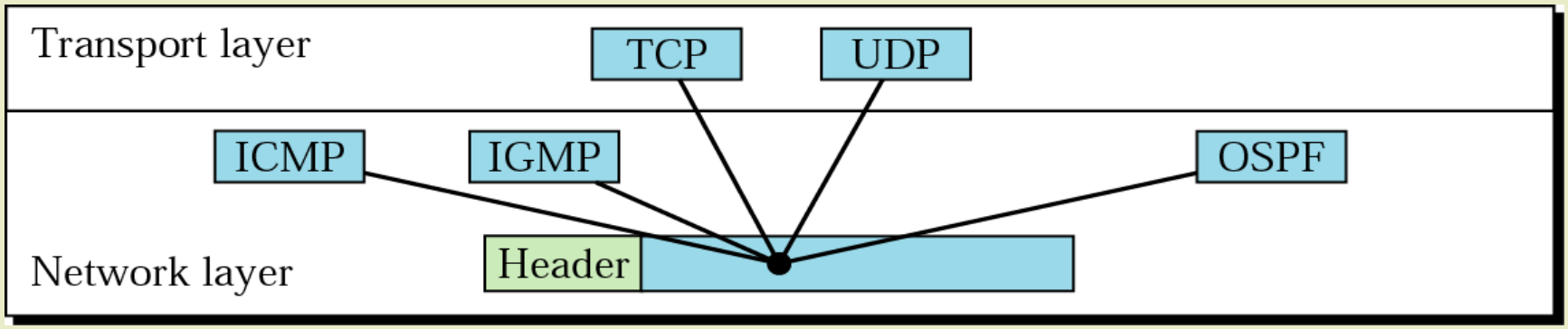
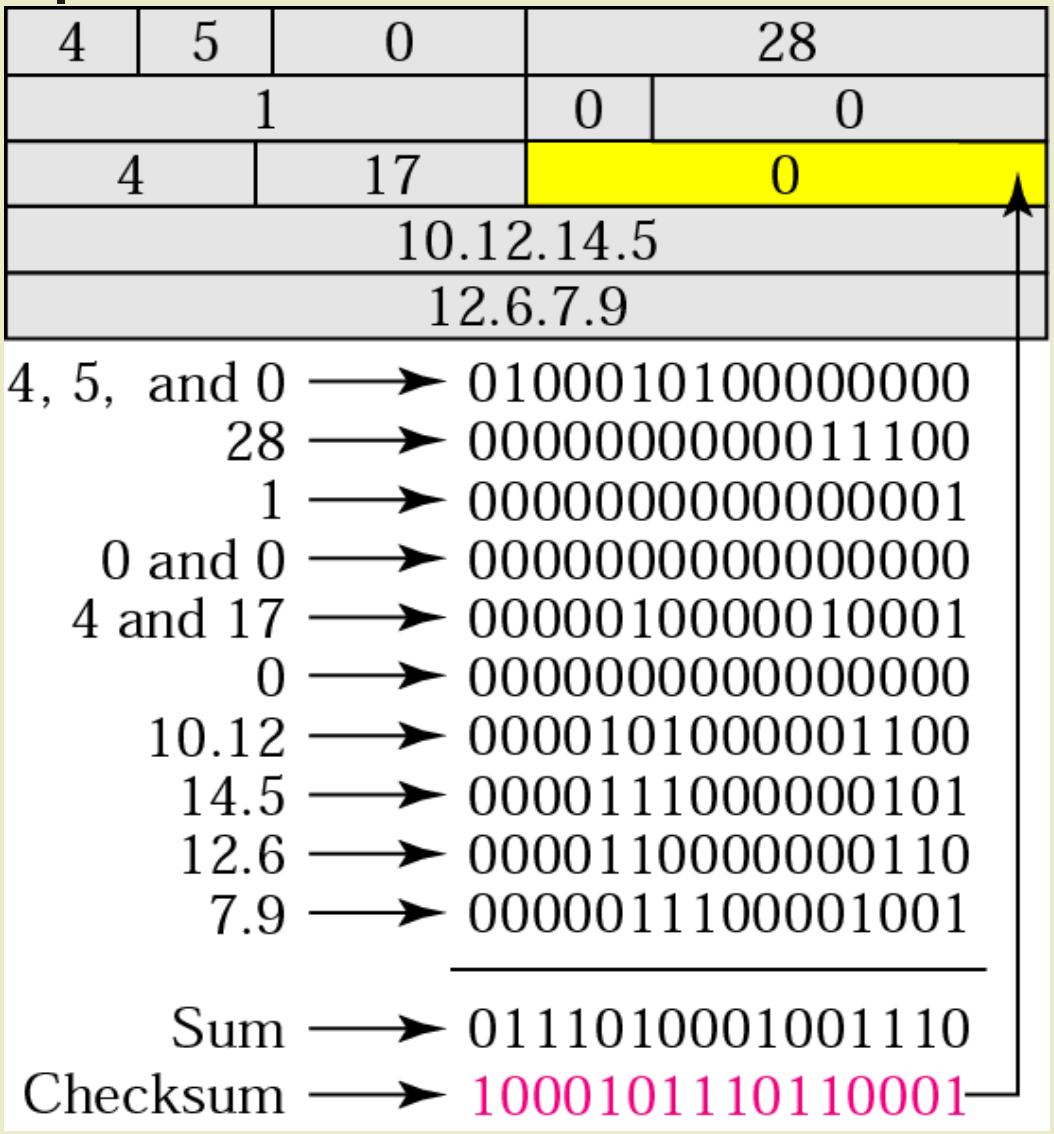


Figure 20.9 Example of checksum calculation



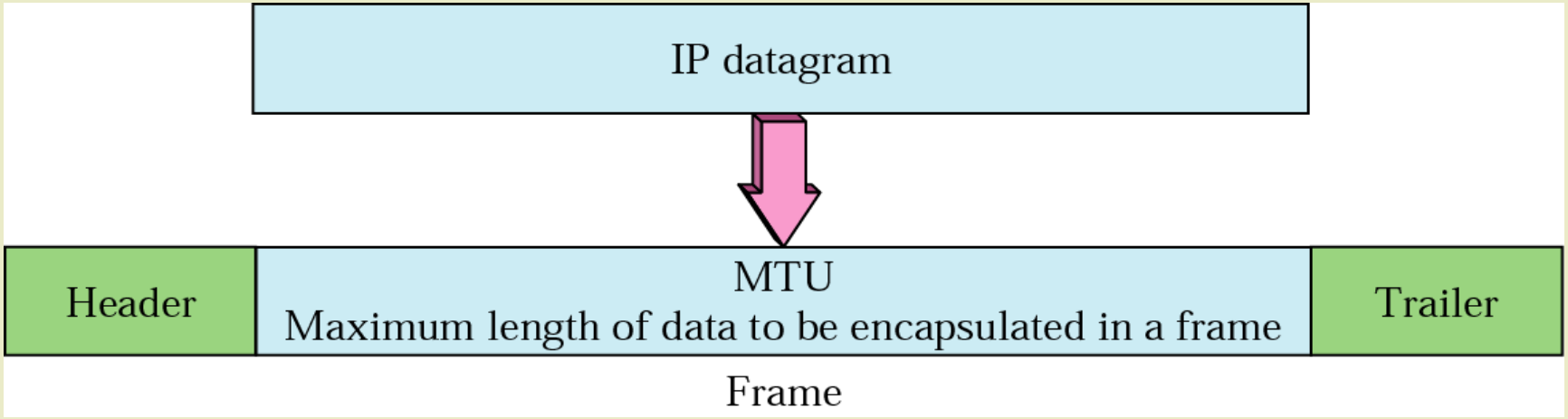
NOTE:

Only the HEADER has any kind of error detection – IP is NOT responsible for checking or correcting data

At the next 'hop', the Sum is recalculated and added to the received checksum. If the answer is not zero, the packet is discarded.

The checksum changes at each router, because the TTL changes each time.

Figure 20.10 MTU



The data link layer will have a maximum frame size (this varies depending on layer 2 protocol). This affects the maximum data payload, or MTU – maximum transfer unit.

The diagram above illustrates E N C A P S U L A T I O N - the layer 3 datagram es the layer 2 data

Figure 20.11 Fragmentation example

When a packet is fragmented at layer 3, each fragment's 'offset' needs to be stored within the fragmented packet so that it can be reconstructed later.
The fragmented packet also needs to record which original packet it belonged to – hence the need for the 'Identification' field

The 'flags' field contains bits which indicate that there are more fragments to come, or that fragmentation is not allowed