

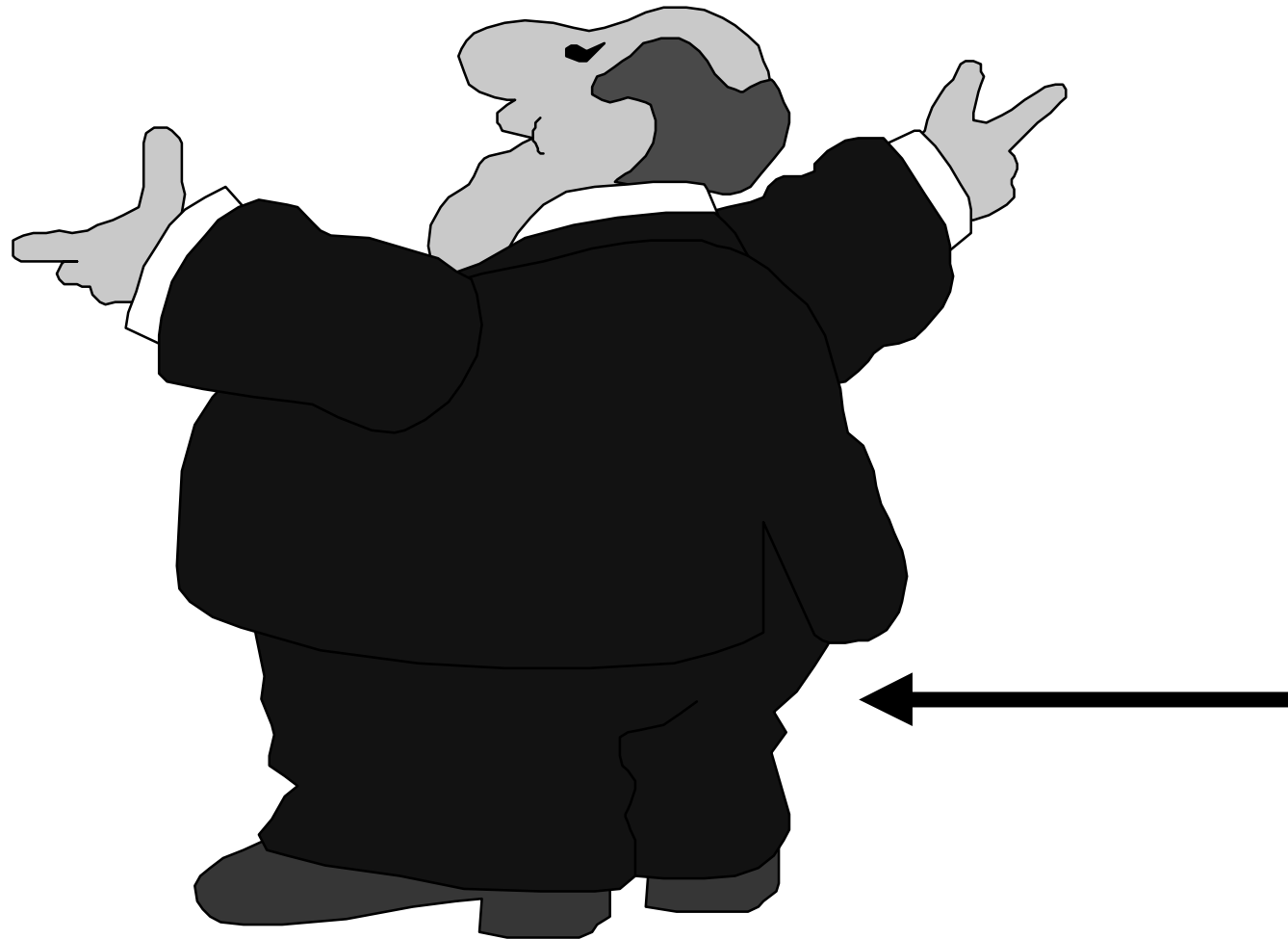
TCP/IP

*Transmission Control Protocol / Internet
Protocol*

TCP/IP & OSI

- ◆ In OSI reference model terminology -the TCP/IP protocol suite covers the network and transport layers.
- ◆ TCP/IP can be used on many data-link layers (can support many network hardware implementations).

But First ...



Ethernet - A Real Data-Link Layer

- ◆ It will be useful to discuss a real data-link layer.
- ◆ Ethernet (really IEEE 802.3) is widely used.
- ◆ Supported by a variety of physical layer implementations.

Ethernet

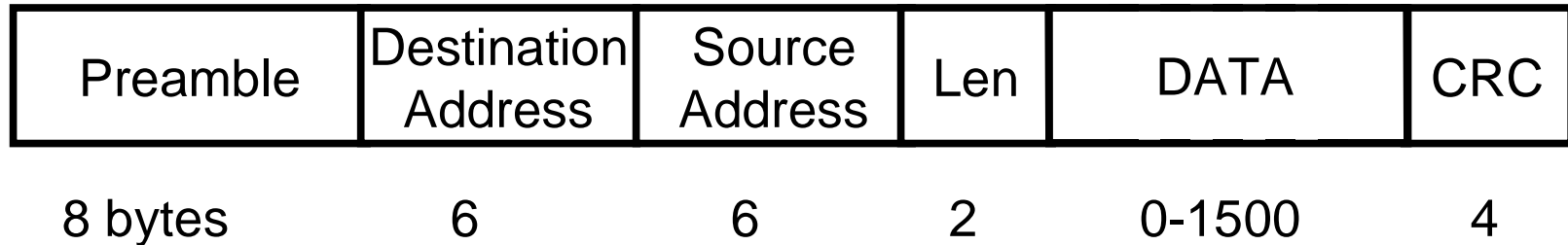
- ◆ Multi-access (shared medium).
- ◆ Every Ethernet interface has a unique 48 bit address (a.k.a. *hardware address*).
- ◆ Example: `C0:B3:44:17:21:17`
- ◆ The broadcast address is all 1's.
- ◆ Addresses are assigned to vendors by a central authority.

CSMA/CD

Carrier Sense Multiple Access with Collision Detection

- ◆ *Carrier Sense*: can tell when another host is transmitting
- ◆ *Multiple Access*: many hosts on 1 wire
- ◆ *Collision Detection*: can tell when another host transmits at the same time.

An Ethernet Frame

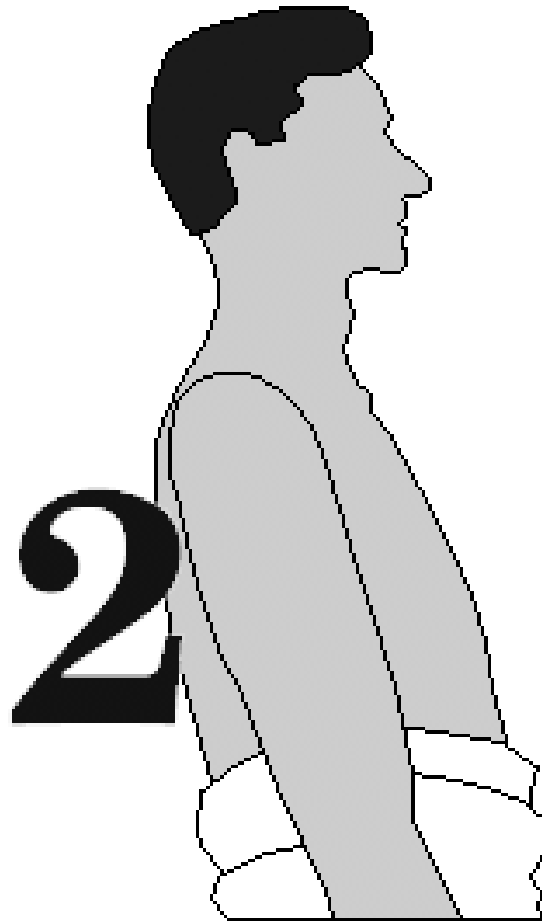


- ◆ The preamble is a sequence of alternating 1s and 0s used for synchronization.
- ◆ CRC is Cyclic Redundancy Check

Ethernet Addressing

- ◆ Each interface looks at every *frame* and inspects the destination address. If the address does not match the hardware address of the interface or the broadcast address, the frame is discarded.
- ◆ Some interfaces can also be programmed to recognize multicast addresses.

Back to TCP/IP



Internet Protocol

The IP in TCP/IP

- ◆ IP is the network layer
 - ◆ packet delivery service (host-to-host).
 - ◆ translation between different data-link protocols.

IP Datagrams

- ◆ IP provides connectionless, unreliable delivery of *IP datagrams*.
- ◆ *Connectionless*: each datagram is independent of all others.
- ◆ *Unreliable*: there is no guarantee that datagrams are delivered correctly or at all.

IP Addresses

- ◆ IP addresses are not the same as the underlying data-link (MAC) addresses.



Why ?

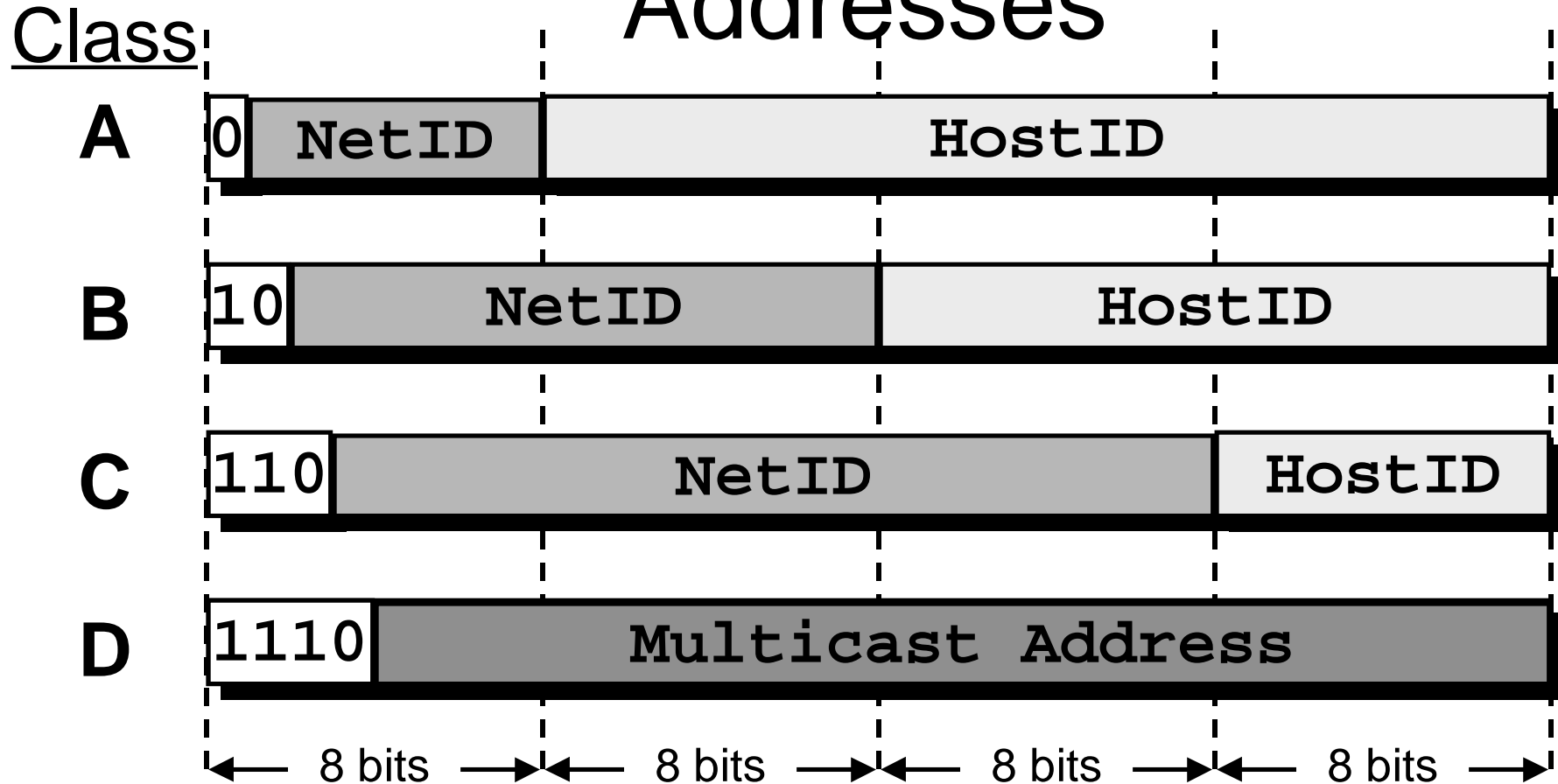
IP Addresses

- ◆ IP is a network layer - it must be capable of providing communication between hosts on different kinds of networks (different data-link implementations).
- ◆ The address must include information about what *network* the receiving host is on. This makes routing feasible.

IP Addresses

- ◆ IP addresses are *logical* addresses (not physical)
- ◆ 32 bits.
- ◆ Includes a network ID and a host ID.
- ◆ Every host must have a unique IP address.
- ◆ IP addresses are assigned by a central authority (the NIC at SRI International).

The *four* formats of IP Addresses



Class A

- 128 possible network IDs
- over 4 million host IDs per network ID

Class B

- 16K possible network IDs
- 64K host IDs per network ID

Class C

- over 2 million possible network IDs
- about 256 host IDs per network ID

Network and Host IDs

- ◆ A Network ID is assigned to an organization by a global authority.
- ◆ Host IDs are assigned locally by a system administrator.
- ◆ Both the Network ID and the Host ID are used for routing.

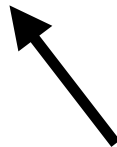
IP Addresses

- ◆ IP Addresses are usually shown in *dotted decimal* notation:

1.2.3.4 → 00000001 00000010 00000011 00000100

- ◆ cs.rpi.edu is 128.213.1.1

10000000 11010101 00000001 00000001



CS has a class B network

Host and Network Addresses

- ◆ A single network interface is assigned a single IP address called the *host* address.
- ◆ A host may have multiple interfaces, and therefore multiple *host* addresses.
- ◆ Hosts that share a network all have the same IP *network* address (the network ID).

IP Broadcast and Network Addresses

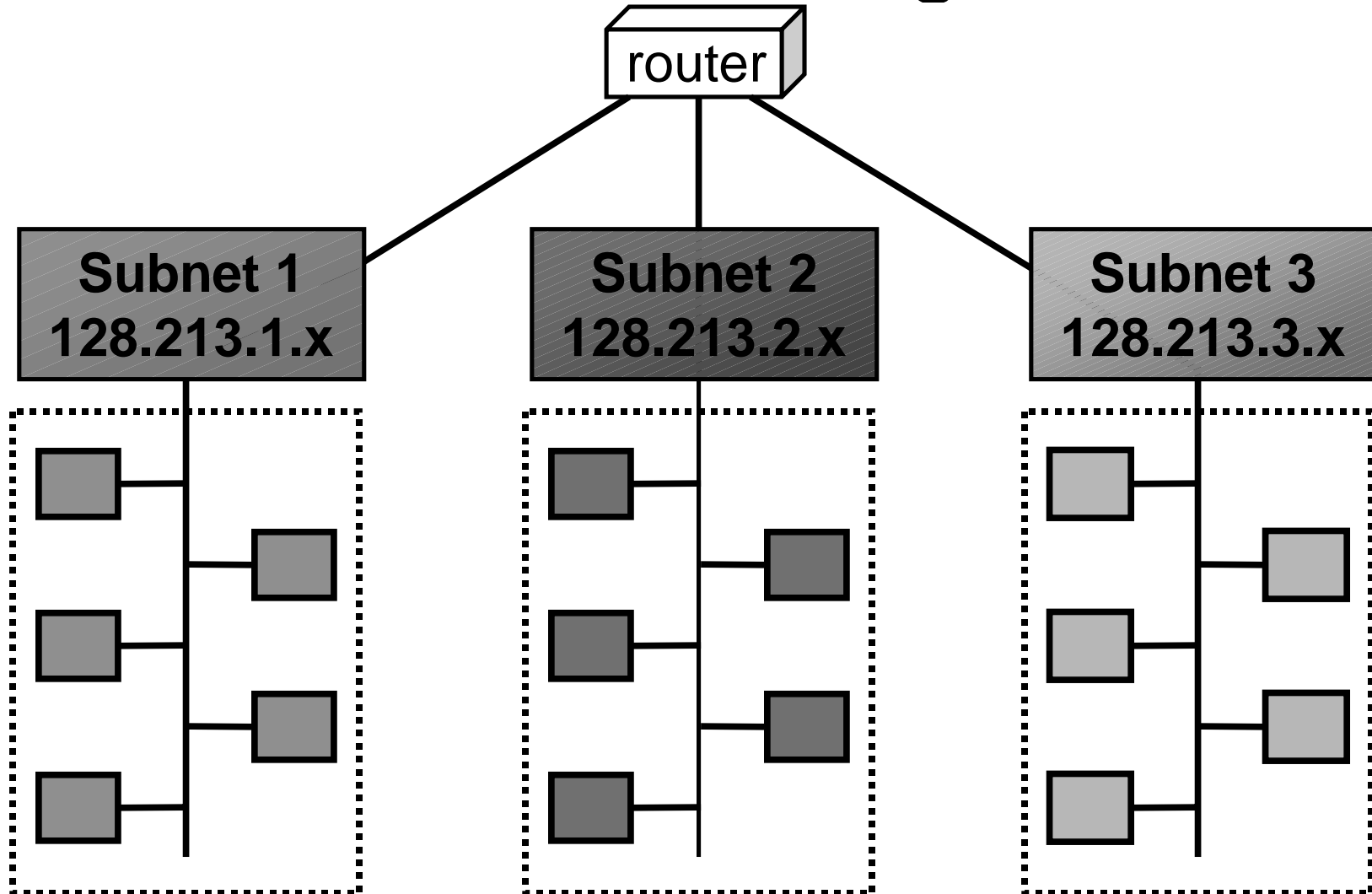
- ◆ An IP broadcast address has a host ID of all 1s.
- ◆ IP broadcasting is not necessarily a true broadcast, it relies on the underlying hardware technology.
- ◆ An IP address that has a host ID of all 0s is called a *network address* and refers to an entire network.

Subnet Addresses

- ◆ An organization can subdivide its host address space into groups called subnets.
- ◆ The subnet ID is generally used to group hosts based on the physical network topology.

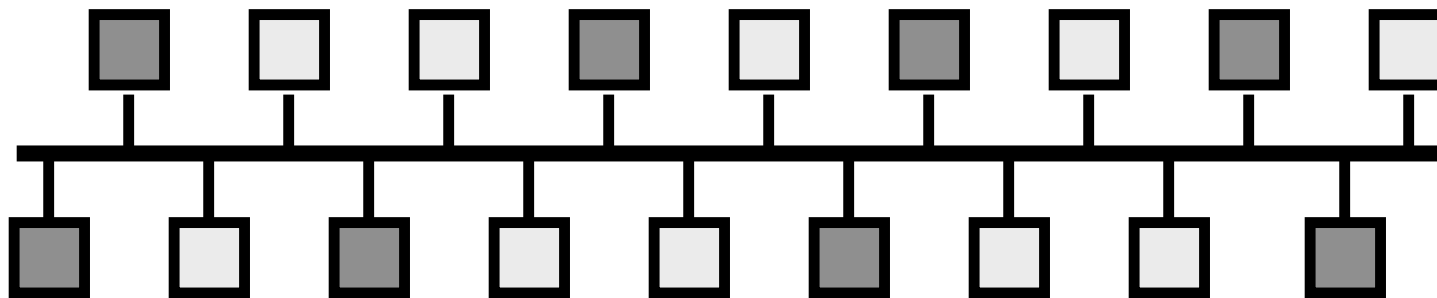


Subnetting



Subnetting

- ◆ Subnets can simplify routing.
- ◆ IP subnet broadcasts have a hostID of all 1s.
- ◆ It is possible to have a single wire network with multiple subnets.



Mapping IP Addresses to Hardware Addresses

- ◆ IP Addresses are not recognized by hardware.
- ◆ If we know the IP address of a host, how do we find out the hardware address ?
- ◆ The process of finding the hardware address of a host given the IP address is called

Address Resolution

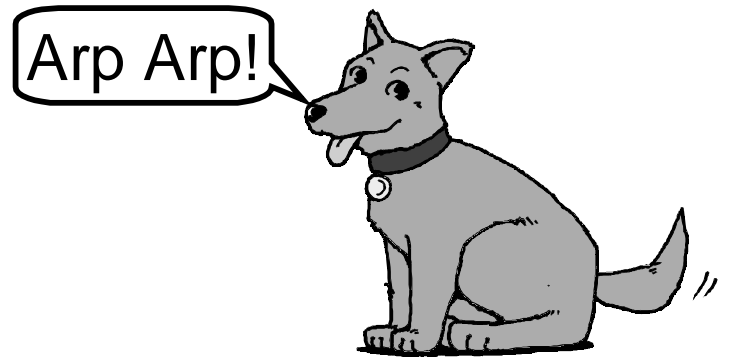
Reverse Address Resolution

- ◆ The process of finding out the IP address of a host given a hardware address is called

Reverse Address Resolution

- ◆ Reverse address resolution is needed by diskless workstations when booting.

ARP

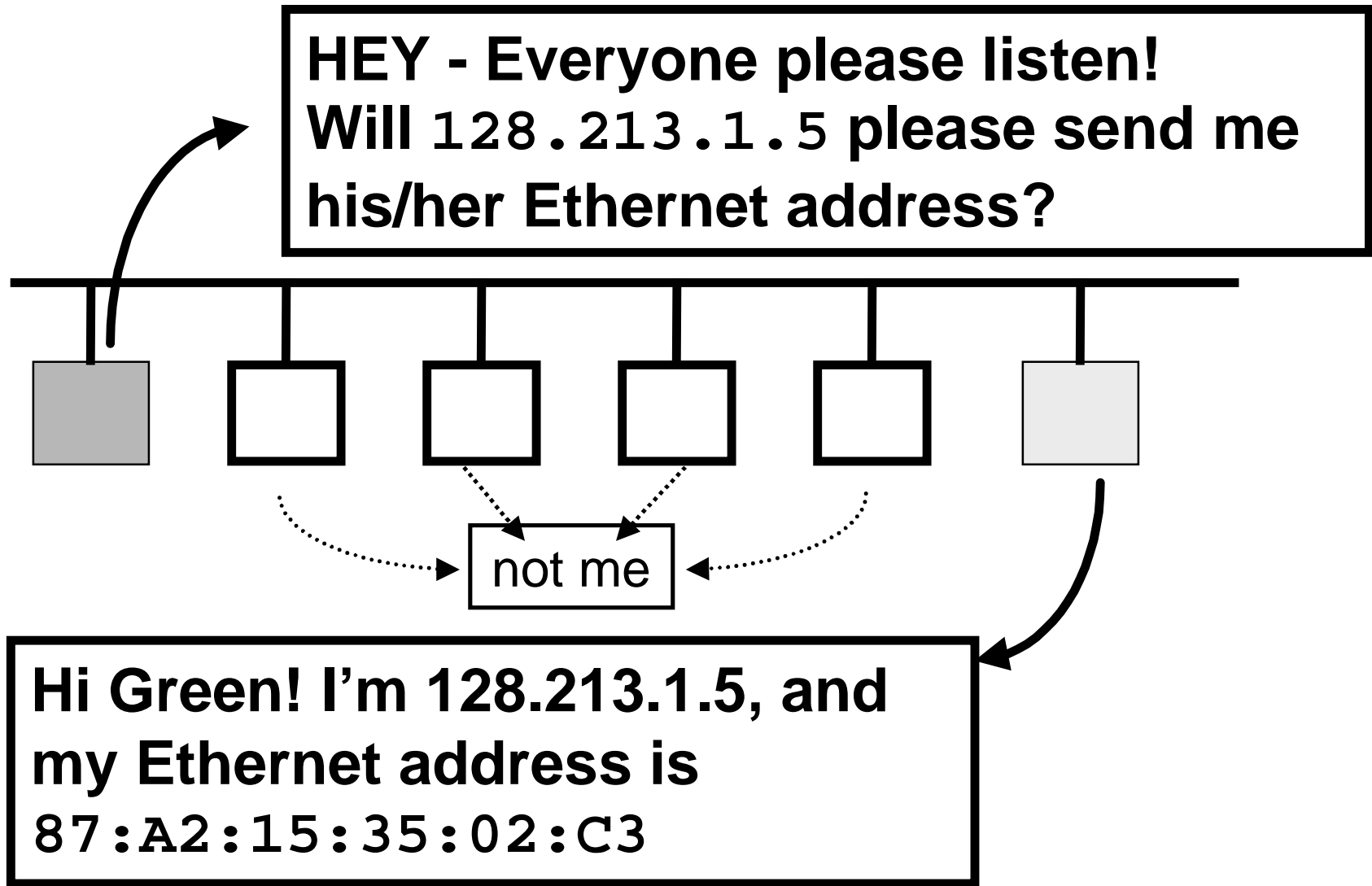


- ◆ The *Address Resolution Protocol* is used by a sending host when it knows the IP address of the destination but needs the Ethernet address.
- ◆ ARP is a broadcast protocol - every host on the network receives the request.
- ◆ Each host checks the request against its IP address - the right one responds.

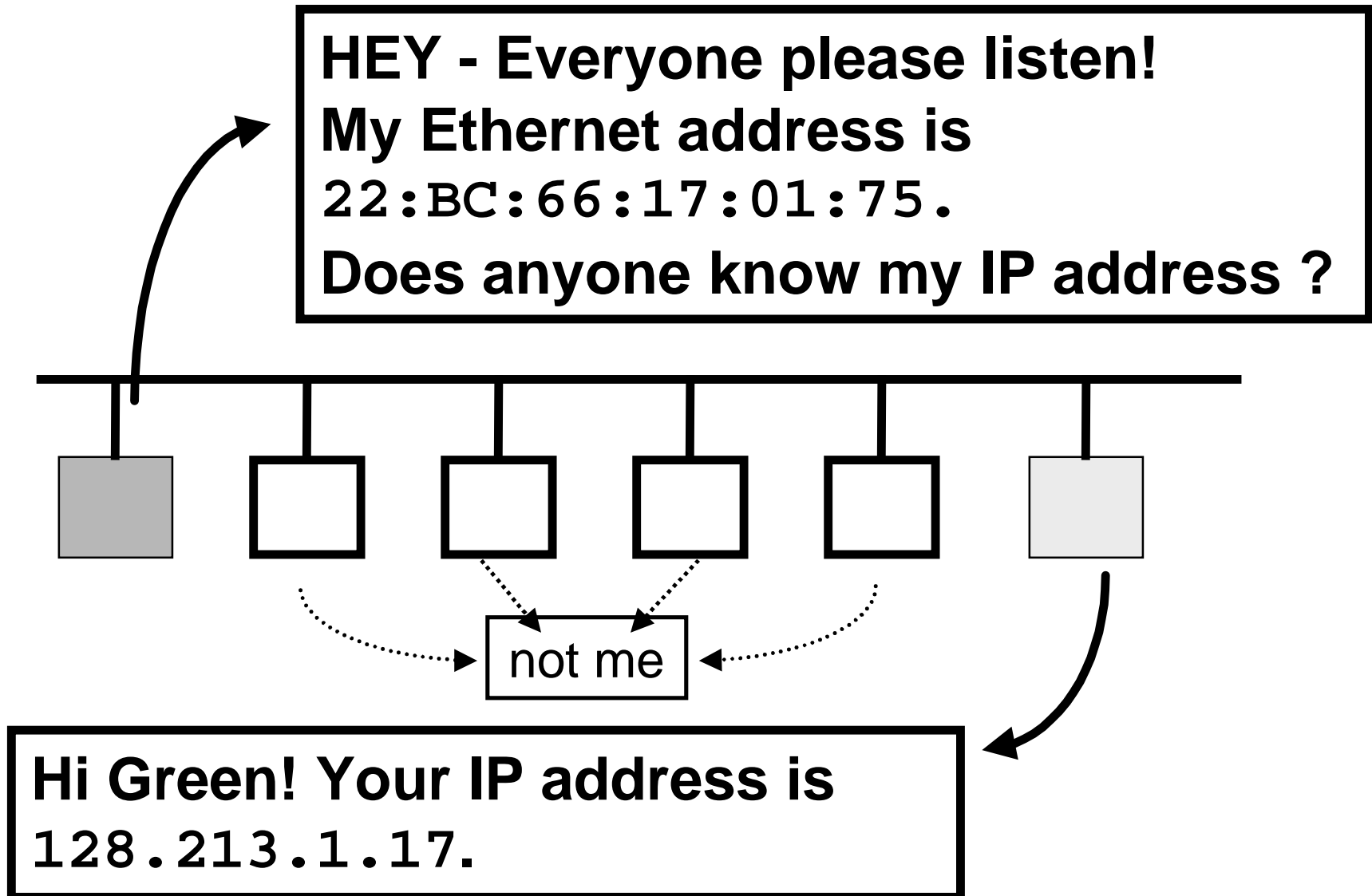
ARP (cont.)

- ◆ ARP does not need to be done every time an IP datagram is sent - hosts *remember* the hardware addresses of each other.
- ◆ Part of the ARP protocol specifies that the receiving host should also remember the IP and hardware addresses of the sending host.

ARP conversation



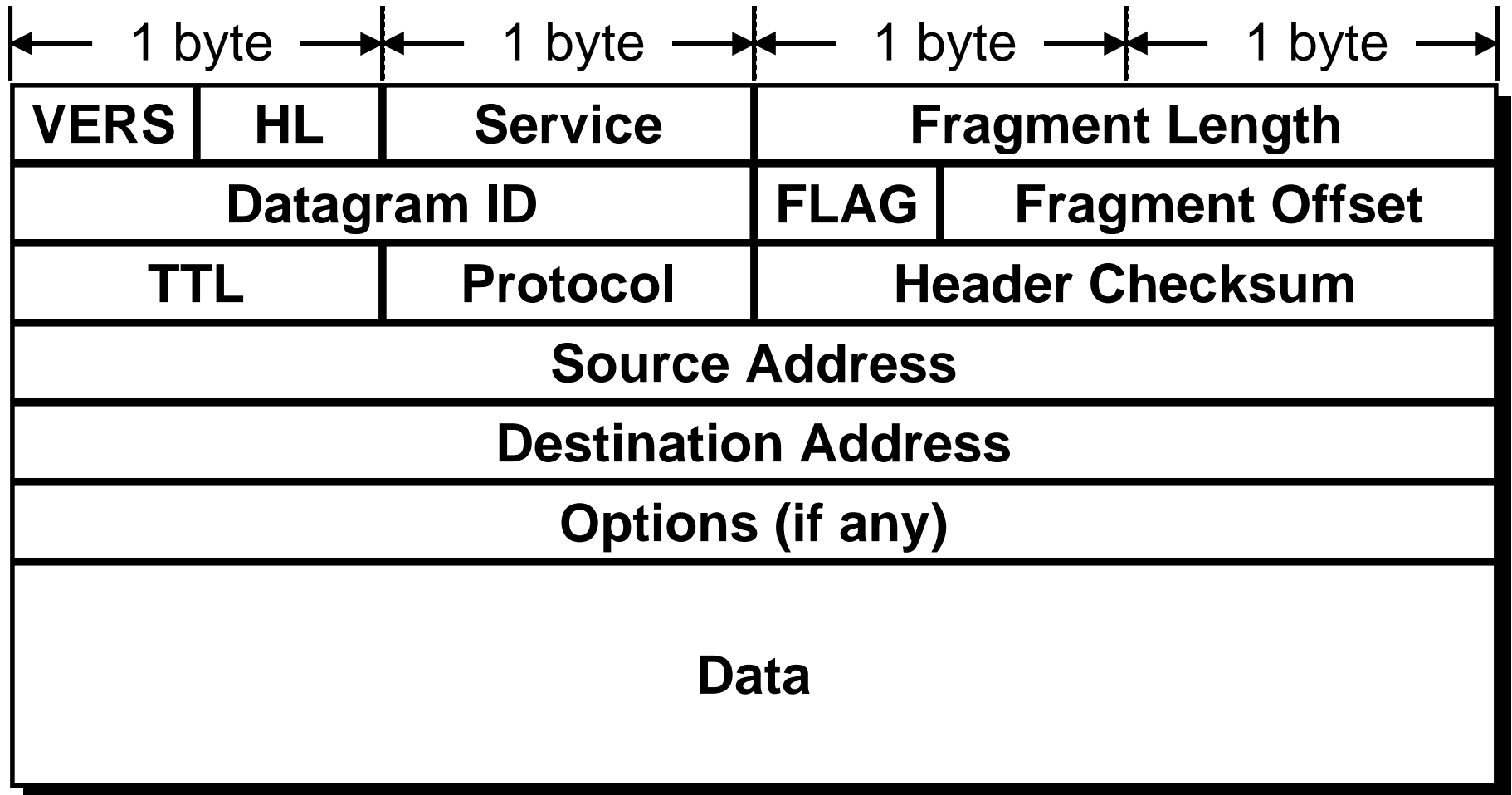
RARP conversation



Services provided by IP

- ◆ Connectionless Delivery (each datagram is treated individually).
- ◆ Unreliable (delivery is not guaranteed).
- ◆ Fragmentation / Reassembly (based on hardware MTU).
- ◆ Routing.
- ◆ Error detection.

IP Datagram



IP Datagram Fragmentation

- ◆ Each fragment (packet) has the same structure as the IP datagram.
- ◆ IP specifies that datagram reassembly is done only at the destination (not on a hop-by-hop basis).
- ◆ If any of the fragments are lost - the entire datagram is discarded (and an ICMP message is sent to the sender).

IP Datagram Fragmentation

- ◆ If packets arrive too fast - the receiver discards excessive packets and sends an ICMP message to the sender (SOURCE QUENCH).
- ◆ If an error is found (header checksum problem) the packet is discarded and an ICMP message is sent to the sender.

ICMP

Internet Control Message Protocol

- ◆ ICMP is a protocol used for exchanging control messages.
- ◆ ICMP uses IP to deliver messages.
- ◆ ICMP messages are usually generated and processed by the IP software, not the user process.

ICMP Message Types

- ◆ Echo Request
- ◆ Echo Response
- ◆ Destination Unreachable
- ◆ Redirect
- ◆ Time Exceeded
- ◆ Redirect (route change)
- ◆ there are more ...

IP/BYE-BYE

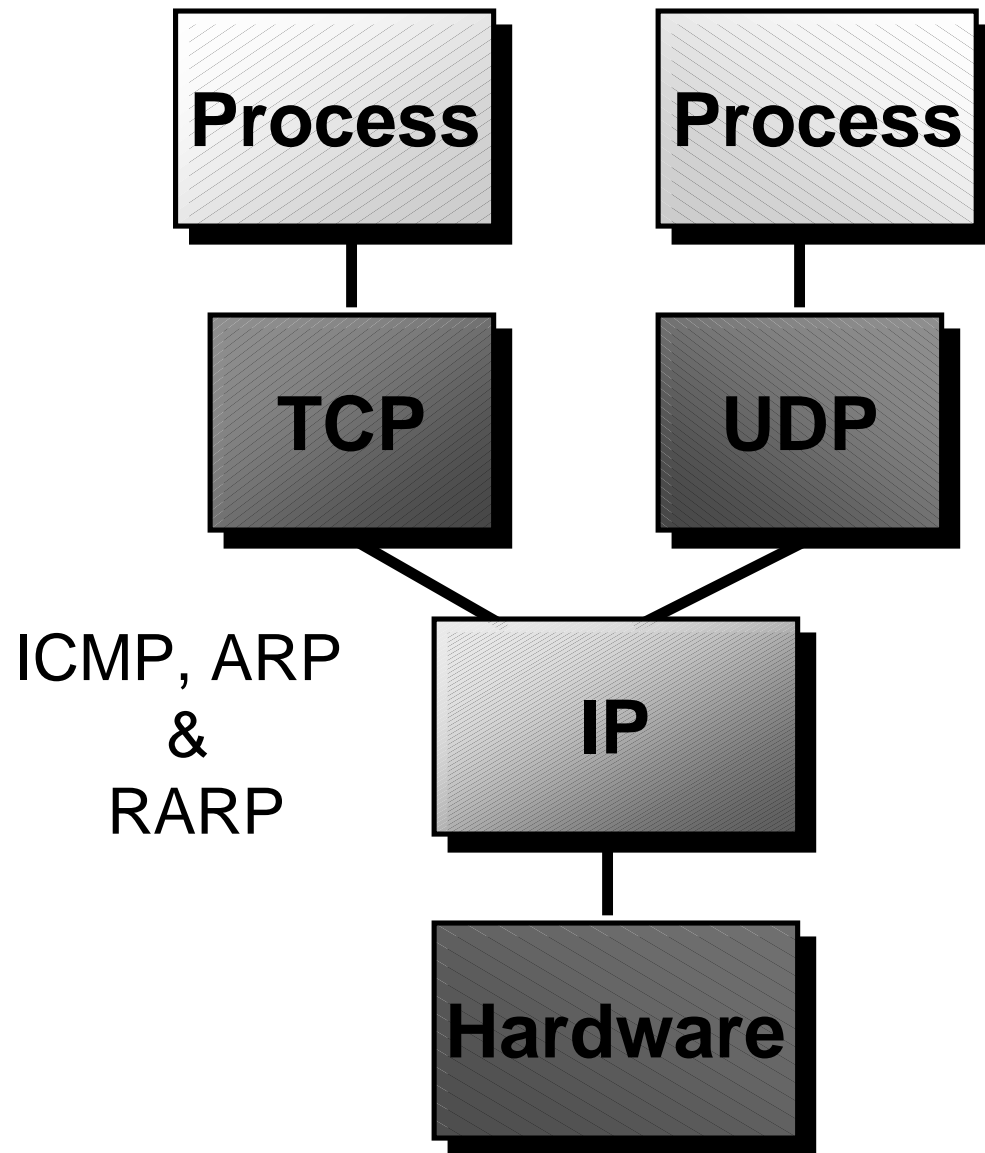
- ◆ IP/BYE-BYE is a lecture protocol used to signal the class that we have just finished our discussion of IP - the network layer of TCP/IP.
- ◆ The appropriate response to an IP/BYE-BYE request is immediate applause, although simply opening your eyes is enough (known as a WAKEUP response).

Transport Layer & TCP/IP

Q: We know that IP is the network layer - so TCP must be the transport layer, right ?

A: No.

TCP is only part of the TCP/IP transport layer - the other part is UDP (User Datagram Protocol).



Process Layer

Transport Layer

Network Layer

Data-Link Layer

UDP User Datagram Protocol

- ◆ UDP is a transport protocol
 - ◆ communication between processes
- ◆ UDP uses IP to deliver datagrams to the right host.
- ◆ UDP uses *ports* to provide communication services to individual processes.

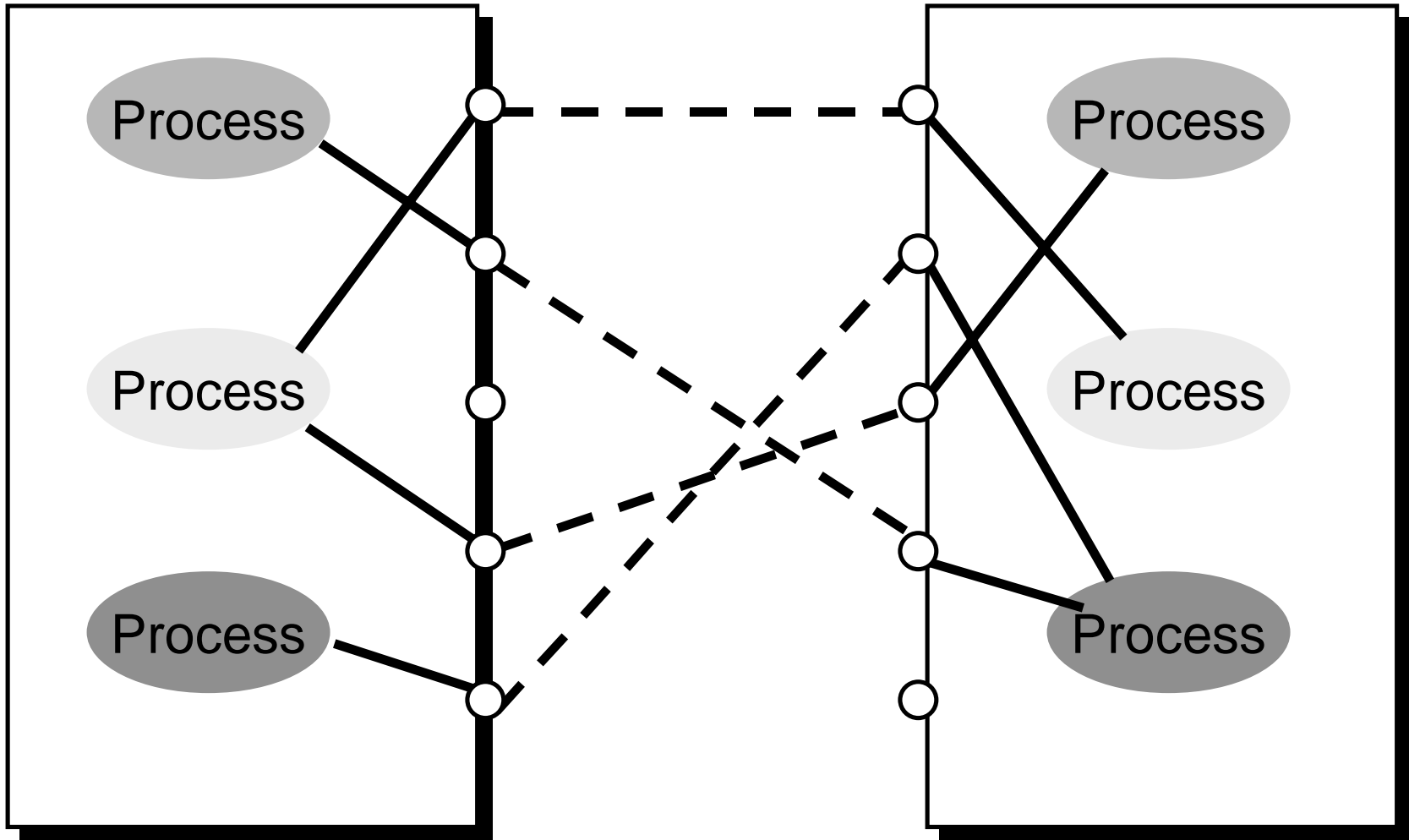
Ports

- ◆ TCP/IP uses an abstract destination point called a protocol port.
- ◆ Ports are identified by a positive integer.
- ◆ Operating systems provide some mechanism that processes use to specify a port.

Ports

Host A

Host B



UDP

- ◆ Datagram Delivery
- ◆ Connectionless
- ◆ Unreliable
- ◆ Minimal

UDP Datagram Format

Source Port	Destination Port
Length	Checksum
Data	

TCP

Transmission Control Protocol

- ◆ TCP is an alternative transport layer protocol supported by IP.
- ◆ TCP provides:
 - ◆ Connection-oriented
 - ◆ Reliable
 - ◆ Full-duplex
 - ◆ Byte-Stream

Connection-Oriented

- ◆ *Connection oriented* means that a virtual connection is established before any user data is transferred.
- ◆ If the connection cannot be established - the user program is notified.
- ◆ If the connection is ever interrupted - the user program(s) is notified.

Reliable

- ◆ *Reliable* means that every transmission of data is acknowledged by the receiver.
- ◆ If the sender does not receive acknowledgement within a specified amount of time, the sender retransmits the data.

Byte Stream

- ◆ *Stream* means that the connection is treated as a stream of bytes.
- ◆ The user application does not need to package data in individual datagrams (as with UDP).

Buffering

- ◆ TCP is responsible for buffering data and determining when it is time to send a datagram.
- ◆ It is possible for an application to tell TCP to send the data it has buffered without waiting for a buffer to fill up.

Full Duplex

- ◆ TCP provides transfer in both directions.
- ◆ To the application program these appear as 2 unrelated data streams, although TCP can piggyback control and data communication by providing control information (such as an ACK) along with user data.

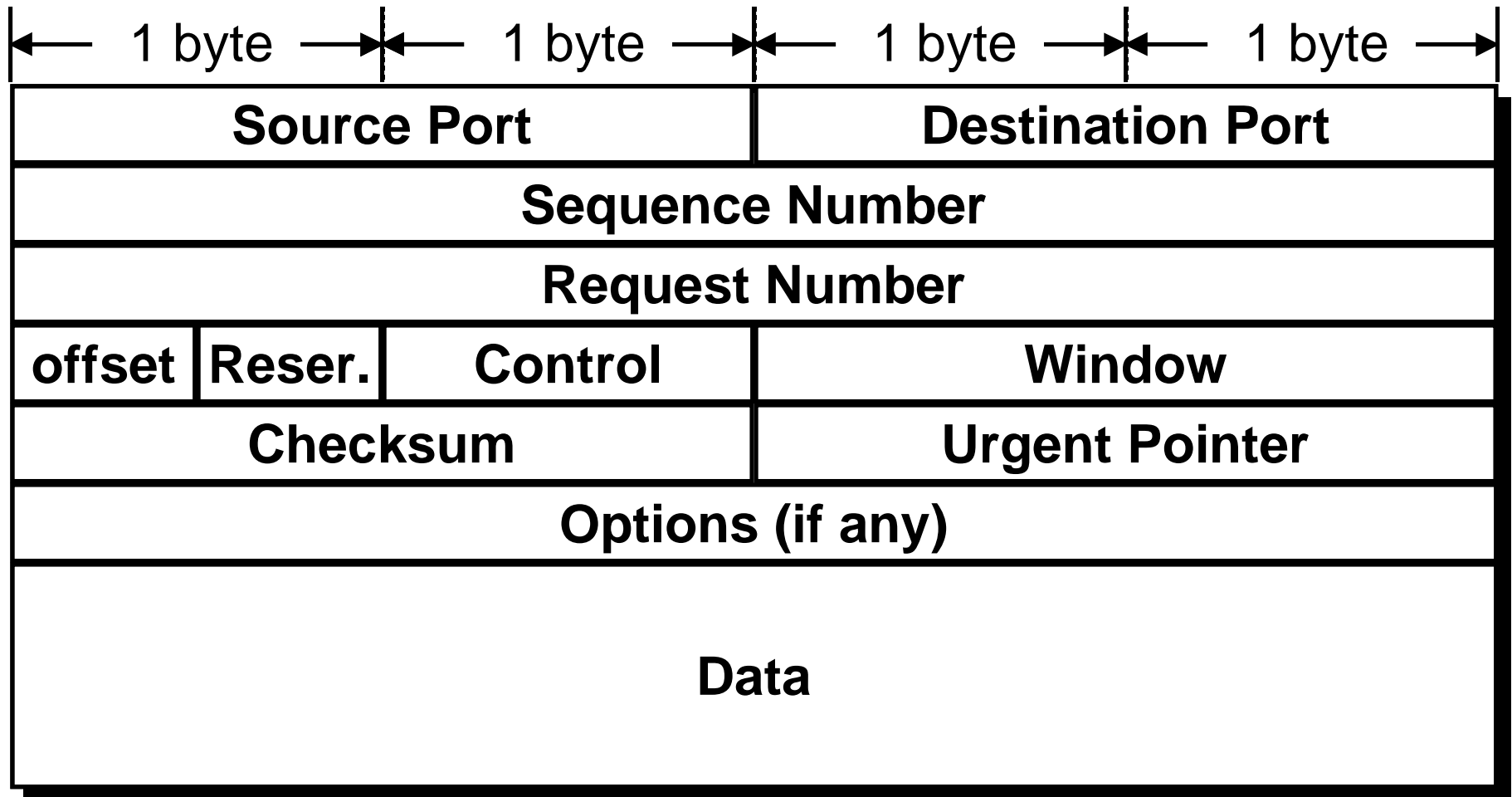
TCP Ports

- ◆ Interprocess communication via TCP is achieved with the use of ports (just like UDP).
- ◆ UDP ports have no relation to TCP ports (different name spaces).

TCP Segments

- ◆ The chunk of data that TCP asks IP to deliver is called a *TCP segment*.
- ◆ Each segment contains:
 - ◆ data bytes from the byte stream
 - ◆ control information that identifies the data bytes

TCP Segment Format



Addressing in TCP/IP

- ◆ Each TCP/IP address includes:
 - ◆ Internet Address
 - ◆ Protocol (UDP or TCP)
 - ◆ Port Number

TCP vs. UDP

Q: Which protocol is better ?

A: It depends on the application.

TCP provides a connection-oriented, reliable byte stream service (lots of overhead).

UDP offers minimal datagram delivery service (as little overhead as possible).

TCP/IP Summary

- ◆ IP: network layer protocol
 - ◆ unreliable datagram delivery between hosts.
- ◆ UDP: transport layer protocol
 - ◆ unreliable datagram delivery between processes.
- ◆ TCP: transport layer protocol
 - ◆ reliable, byte-stream delivery between processes.

Hmmmm. TCP or UDP ?

- ◆ Internet commerce ?
- ◆ Video server?
- ◆ File transfer?
- ◆ Email ?
- ◆ Chat groups?
- ◆ Robotic surgery controlled remotely over a network?