Networking Background

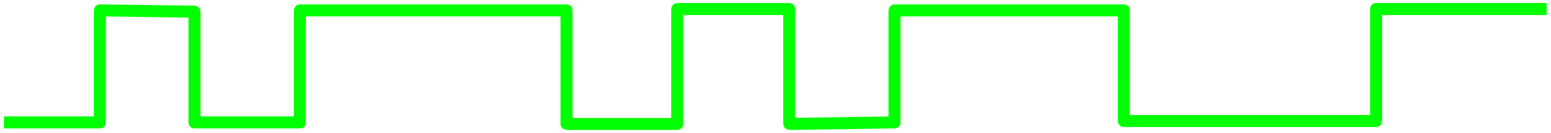
- OSI Reference Model
- Terminology
- TCP/IP
    - IP
    - UDP
    - TCP

OSI Reference Model

- Layered model:
  7. Application
  6. Presentation
  5. Session
  4. Transport
  3. Network
  2. Data Link
  1. Physical

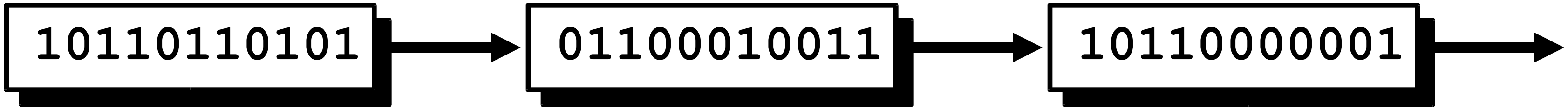
The Physical Layer

- Responsibility:
  - transmission of raw bits over a communication channel.
- Issues:
  - mechanical and electrical interfaces
  - time per bit
  - distances



The Data Link Layer - Data Link Control

- Responsibility:
  - provide an error-free communication link
- Issues:
  - *framing* (dividing data into chunks)
    - header & trailer bits
  - addressing



The Data Link Layer - The MAC sublayer

- Medium Access Control - needed by mutiaccess networks.

- MAC provides DLC with “virtual wires” on multiaccess networks.

The Network Layer

- Responsibilities:
  - path selection between end-systems (routing).
  - subnet flow control.
  - fragmentation & reassembly
  - translation between different network types.
- Issues:
  - *packet* headers
  - virtual circuits

The Transport Layer

- Responsibilities:
  - provides virtual end-to-end links between peer processes.
  - end-to-end flow control
- Issues:
  - headers
  - error detection
  - reliable communication

The Session Layer

- Responsibilities:
  - establishes, manages, and terminates sessions between applications.
  - service location lookup

- Many protocol suites do not include a session layer.

The Presentation Layer

- Responsibilities:
  - data encryption
  - data compression
  - data conversion
- Many protocol suites do not include a Presentation Layer.

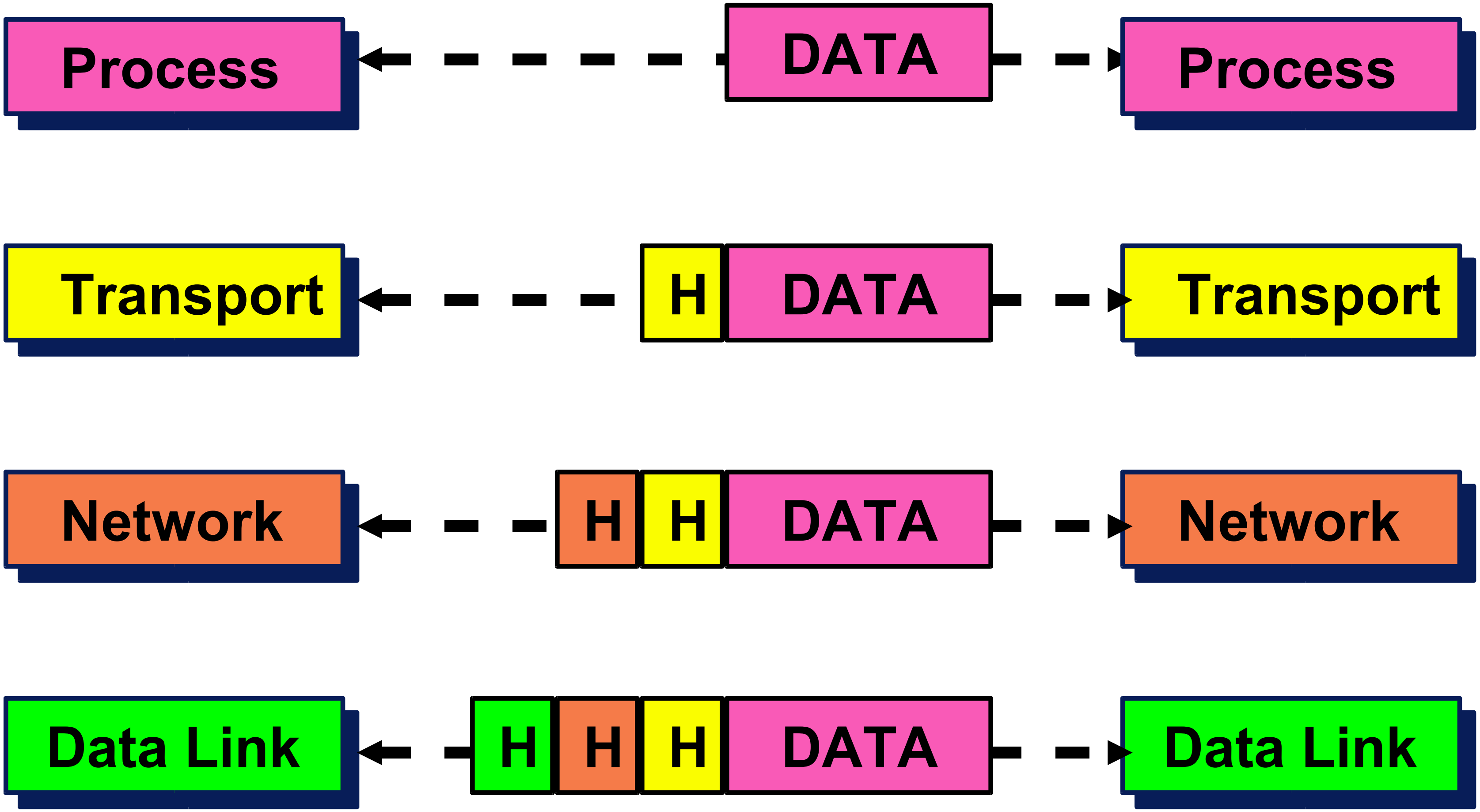
The Application Layer

- Responsibilities:
  - anything not provided by any of the other layers
- Issues:
  - application level protocols
  - appropriate selection of “type of service”

Layering & Headers

- Each layer needs to add some control information to the data in order to do it’s job.
- This information is typically prepended to the data before being given to the lower layer.
- Once the lower layers deliver the the data and control information - the peer layer uses the control information.

Headers


Process

DATA

Process

Transport

H

DATA

Transport

Network

H

H

DATA

Network

Data Link

H

H

H

DATA

Data Link

Important Summary

- Data-Link: communication between machines on the same network.
- Network: communication between machines on possibly different networks.
- Transport: communication between processes (running on machines on possibly different networks).

Modes of Service

- connection-oriented vs. connectionless
- sequencing
- error-control
- flow-control
- byte stream vs. message based
- full-duplex vs. half-duplex.

Network Byte Order

- Conversion of application-level data is left up to the presentation layer.
- But hold on !!! How do lower level layers communicate if they all represent values differently ? (data length fields in headers)
- A fixed byte order is used (called *network byte order*) for all control data.

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).

Internet Protocol
The IP in TCP/IP

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

An IP packet is called a *datagram*

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 even delivered at all.

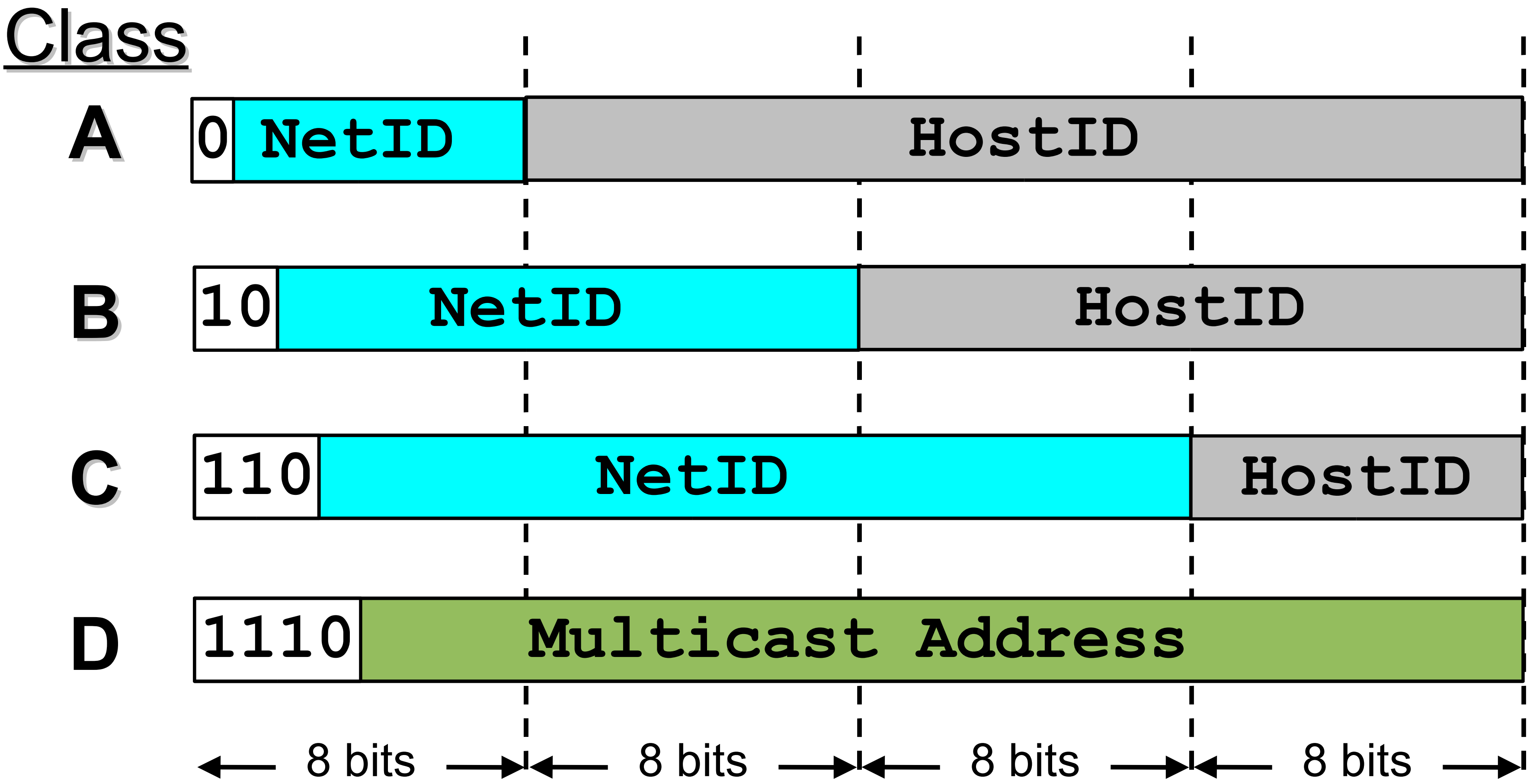
IP Addresses

- 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 is what makes routing feasible.

IP Addresses

- IP addresses are *logical* addresses (not physical)
- 32 bits. IPv4 (version 4)
- Includes a network ID and a host ID.
- Every host must have a unique IP address.
- IP addresses are assigned by a central authority (*American Registry for Internet Numbers* for North America).

The *four formats* of IP Addresses



Class

A 0 NetID HostID

B 10 NetID HostID

C 110 NetID HostID

D 1110 Multicast Address

8 bits 8 bits 8 bits 8 bits

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

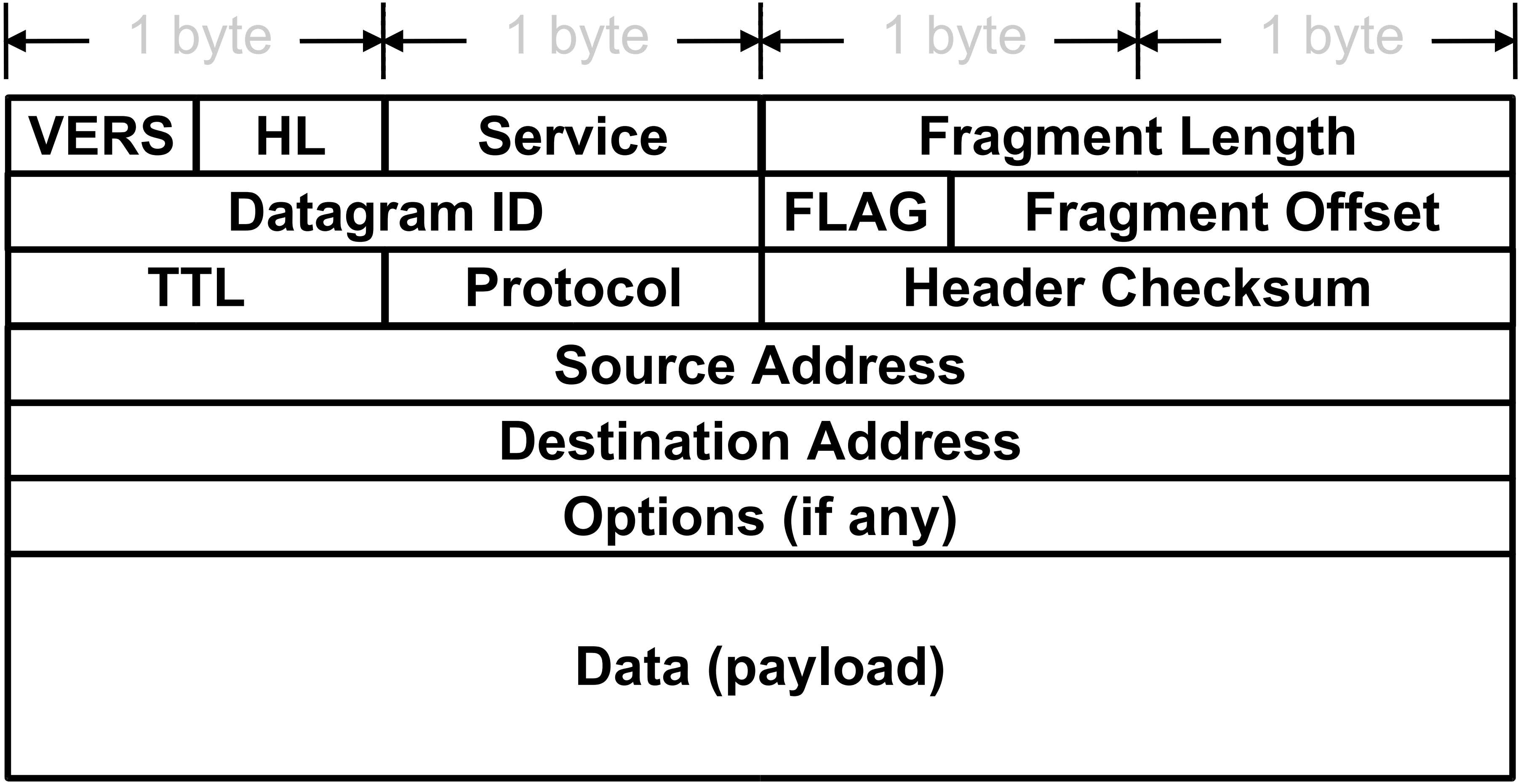
Host and Network Addresses

- 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).

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



1 byte | 1 byte | 1 byte | 1 byte

VERS | HL | Service | Fragment Length

Datagram ID | FLAG | Fragment Offset

TTL | Protocol | Header Checksum

Source Address

Destination Address

Options (if any)

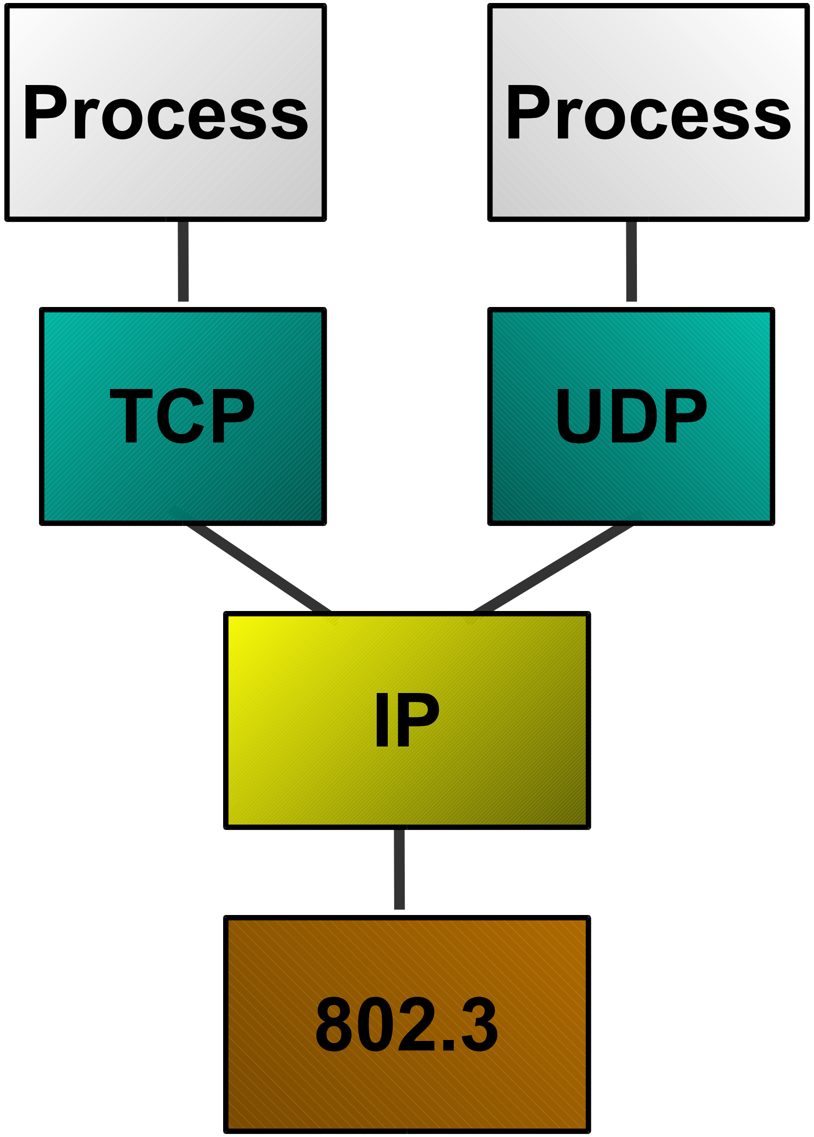
Data (payload)

Transport Layer & TCP/IP

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

A: No… well, almost.

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



Process

Process

TCP

UDP

IP

802.3

Process Layer

Transport Layer

Network Layer

Data-Link Layer

UDP User Datagram Protocol

- UDP is a transport protocol
  - communication between processes

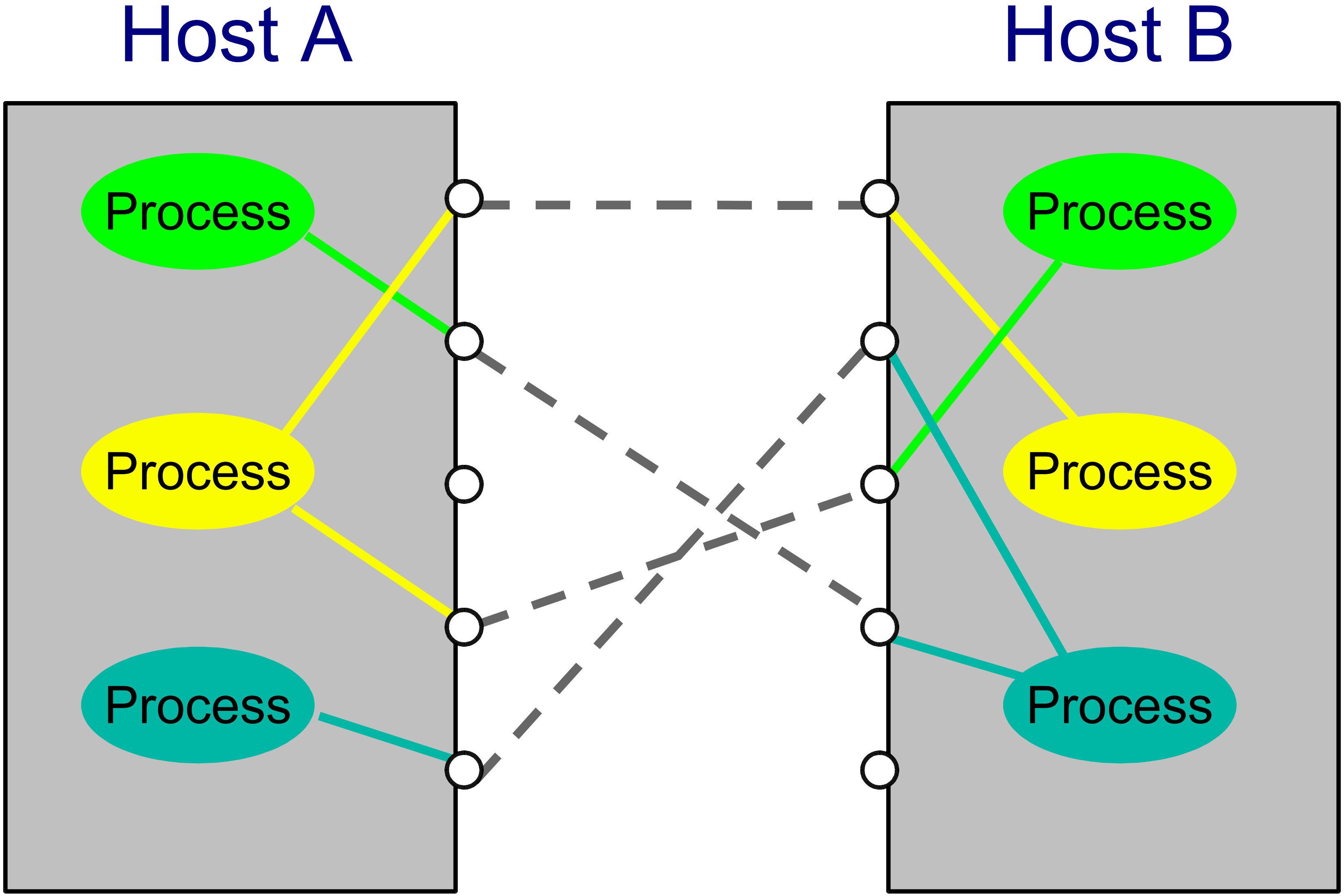
- UDP uses IP to deliver datagrams to the right host.
- UDP uses *ports* to provide communication services to individual processes.

Ports spelled backwards is *strop ...*
Coincidence? (I think not!)

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
Process
Process
Process
Process
Process
Process

UDP

The term *datagram* is also used to describe the unit of transfer of 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 TCP/IP.
- TCP provides:
  - Connection-oriented
  - Reliable
  - Full-duplex
  - Byte-Stream

Wow!

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 (finds out).

- If the connection is ever interrupted - the user program(s) is finds out there is a problem.

Reliable does not mean that things don't go wrong, it means that we find out when things go wrong.

Reliable

- *Reliable* means that every transmission of data is acknowledged by the receiver.
- If the sender does not receive acknowledgment 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).

  Somebody needs to do this since IP is delivering all the data, it's just that the application layer doesn't need to do this!

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 (over a single virtual connection).

- 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).

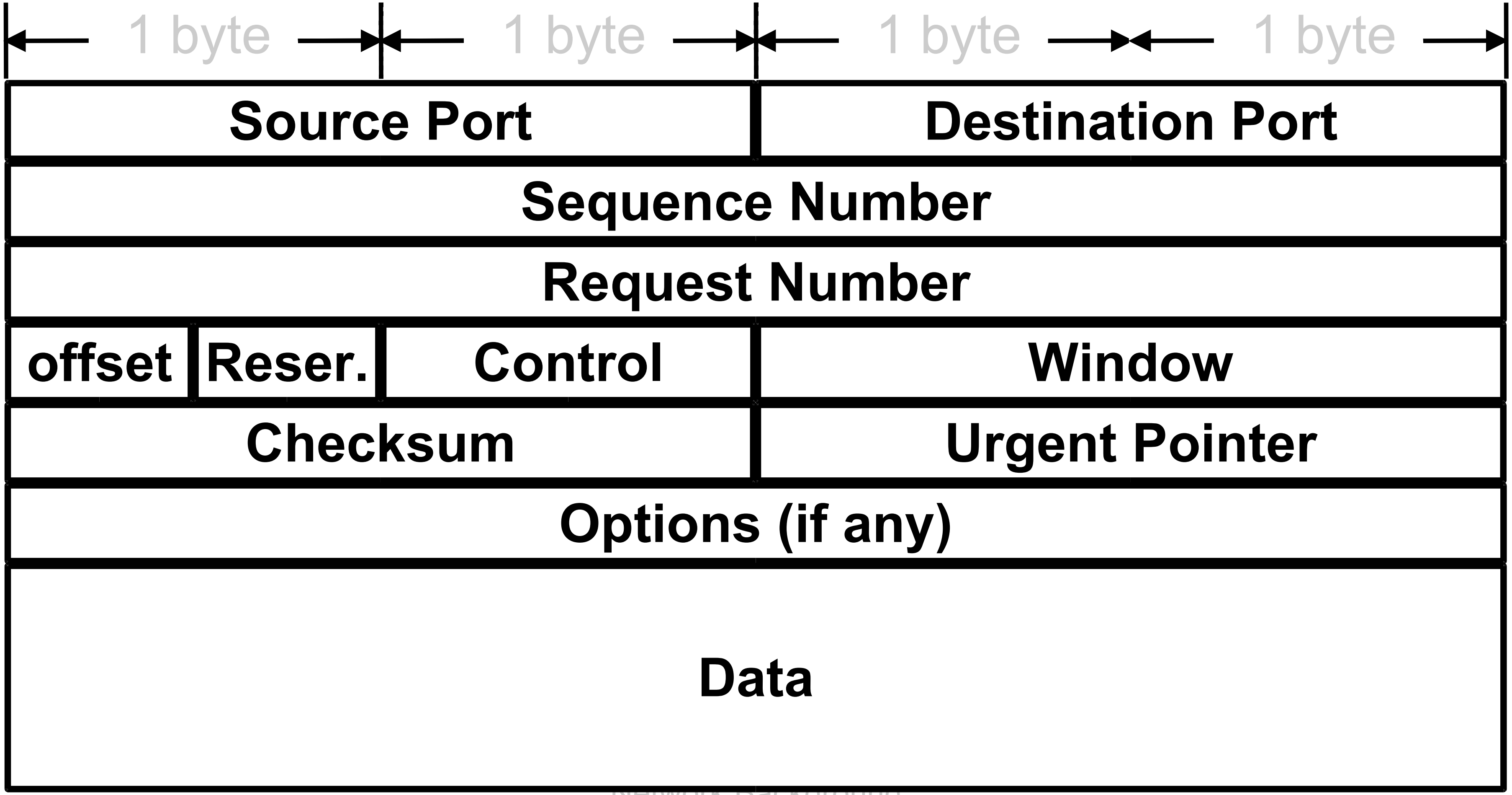
- UDP ports have no relation to TCP ports (different name spaces).

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



1 byte | 1 byte | 1 byte | 1 byte

Source Port | Destination Port

Sequence Number

Request Number

offset | Reser. | Control | Window

Checksum | Urgent Pointer

Options (if any)

Data

Addressing in TCP/IP

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

NOTE: TCP/IP is a *protocol suite* that includes IP,
TCP and UDP.

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.

Hmmmmm. TCP or UDP ?

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