SECTION – B

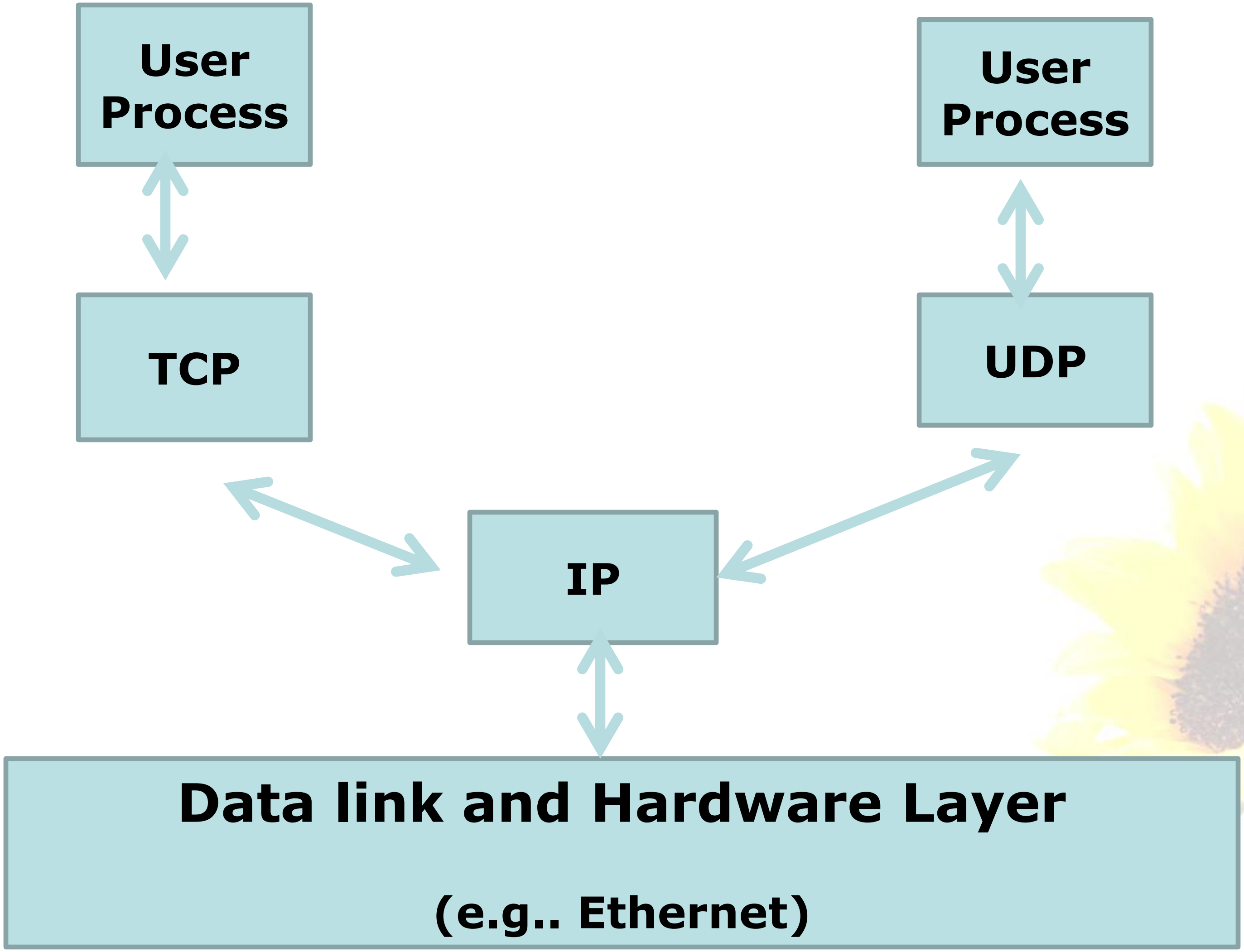
TCP/IP

Part – III

Introduction

- In TCP/IP, the transport layer consists of two different protocols.
  - Transmission control protocol (TCP)
  - User datagram protocol (UDP)
- Basic idea:
  - User processes (application) interact with the TCP/IP protocol suite by sending/receiving TCP or UDP data.
  - Both TCP and UDP in turn uses the IP layer for delivery of packets.

TCP and UDP


User Process
User Process
TCP
UDP
IP
Data link and Hardware Layer
(e.g.. Ethernet)

Role of TCP

- Provides a connection-oriented, reliable, full-duplex, byte-stream service.
  - Underlying IP layer is unreliable and provides connectionless delivery service.
  - TCP provides end-to-end reliability using
    - Checksum
    - Positive acknowledgements
    - Timeouts
    - End-to-end flow control

Role of TCP(contd.)

- TCP also handles
  - Establishment and termination of connections between processes.
  - Sequencing of data that might reach the destination in any arbitrary order.

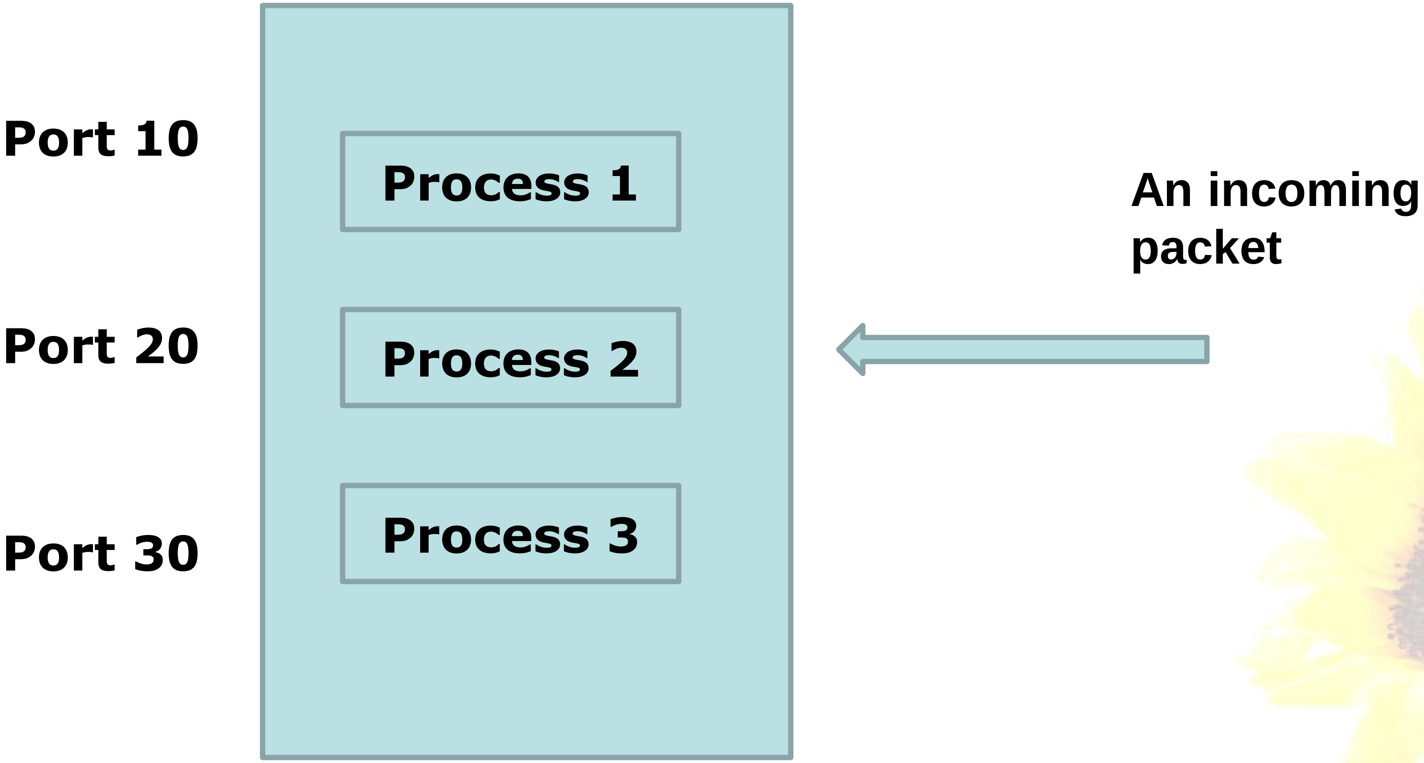
Role of UDP

- UDP provides a connectionless and unreliable datagram service.
  - Very similar to IP in this respect.
  - Provides two features that are not there in IP:
    - A Checksum to verify the integrity of the UDP packet.
    - Port numbers to identify the processes at the two ends.

Port numbers

- Multiple user processes on a machine may use TCP or UDP at the same time.
- There is need for a mechanism to uniquely identify the data packets associated with each process.

Port Numbers (contd.)



Port 10

Port 20

Port 30

Process 1

Process 2

Process 3

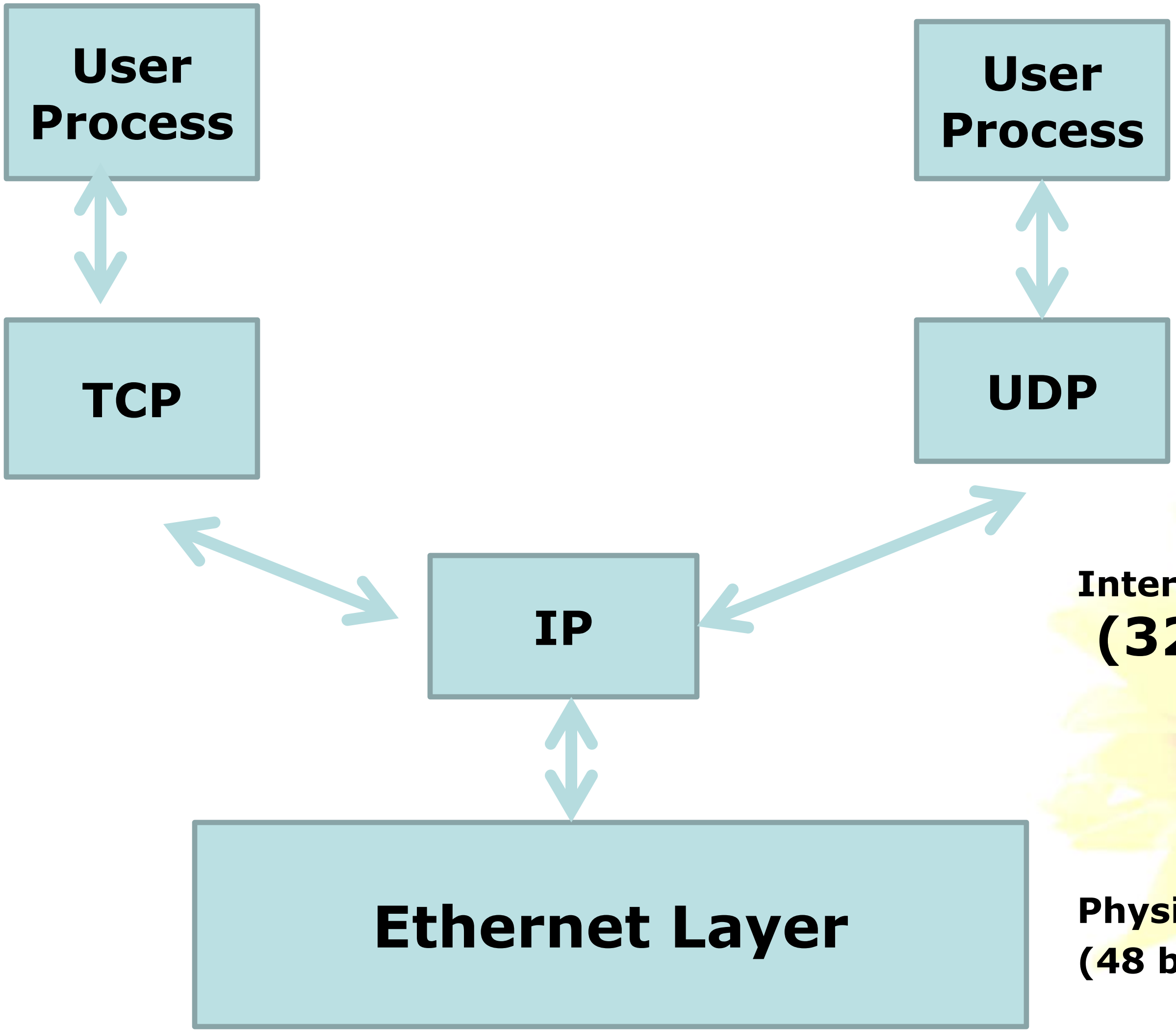
An incoming packet

A host on the Internet

Port Numbers (contd.)

- How this is done?
  - Both TCP and UDP uses 16 bit integer port numbers.
  - Different applications are identified by different port numbers.
  - Port numbers are stored in the headers of TCP or UDP packets.

Port Numbers (contd.)



User Process

User Process

TCP

UDP

IP

Ethernet Layer

Port Address

(16 bits)

Internet Address

(32bits)

Physical Address

(48 bits)

Port Numbers (contd.)

- **Client-server scenario**
  - By knowing the 32-bit IP address of the server host, a client host can connect to the server.
  - To identify a particular process running on the server host, the client must also know the corresponding port number.
- **Well known port numbers**
  - Predefined, and publically known.
  - FTP uses port 21, SMTP uses port 25.

Port Numbers (contd.)

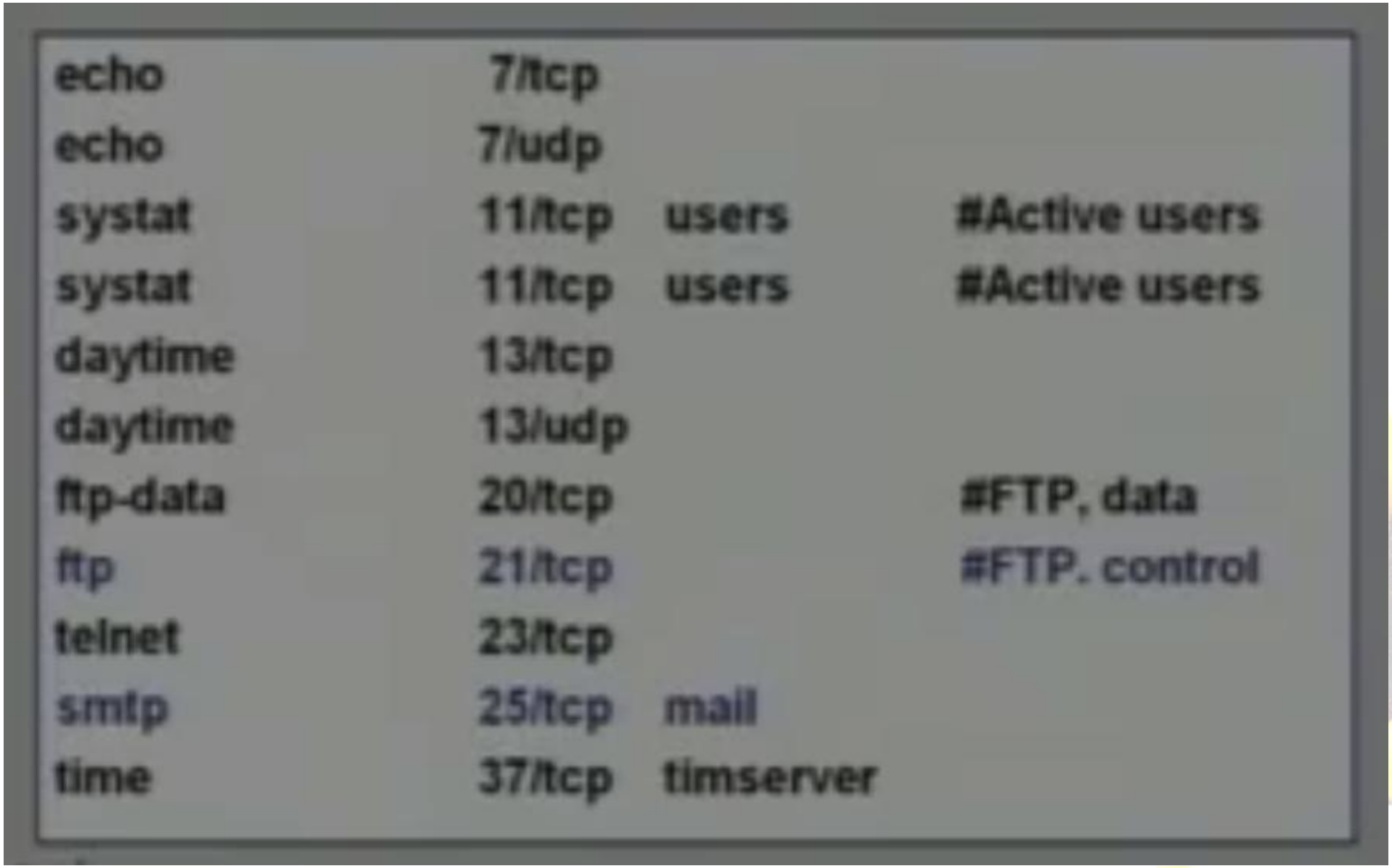
- Well-known port numbers are stored in a particular file on the host machine.
  - Unix:: **/etc/services**
  - XP::

  C:\WINDOWS\system32\drivers\etc
  - Each line has the format:

    <service name><port number>/<protocol>

    [aliases...]
  - **Few lines of the file are shown next.**

/etc/services



```
echo          7/tcp
echo          7/udp
systat        11/tcp   users       #Active users
systat        11/tcp   users       #Active users
daytime       13/tcp
daytime       13/udp
ftp-data      20/tcp               #FTP, data
ftp           21/tcp               #FTP. control
telnet        23/tcp
smtp          25/tcp   mail
time          37/tcp   timserver
```

Ephemeral Port Numbers

- A typical scenario:
  - A client process sends a message to a server process located on some host at port 1534.
  - How will the server know where to respond?
    - Client process requests unused port number from the TCP/UDP module on its local host.
    - These are temporary port numbers, called **ephemeral port numbers**
    - Send along with the TCP or UDP header.

Ephemeral Port Numbers

- How are the port numbers assigned?
  - Port numbers from 1 to 1023 are reserved for well-known ports.
  - Has been extended to 4095.
- Numbers beyond this range and up to 65535 are used as ephemeral port numbers.

Connection Establishment

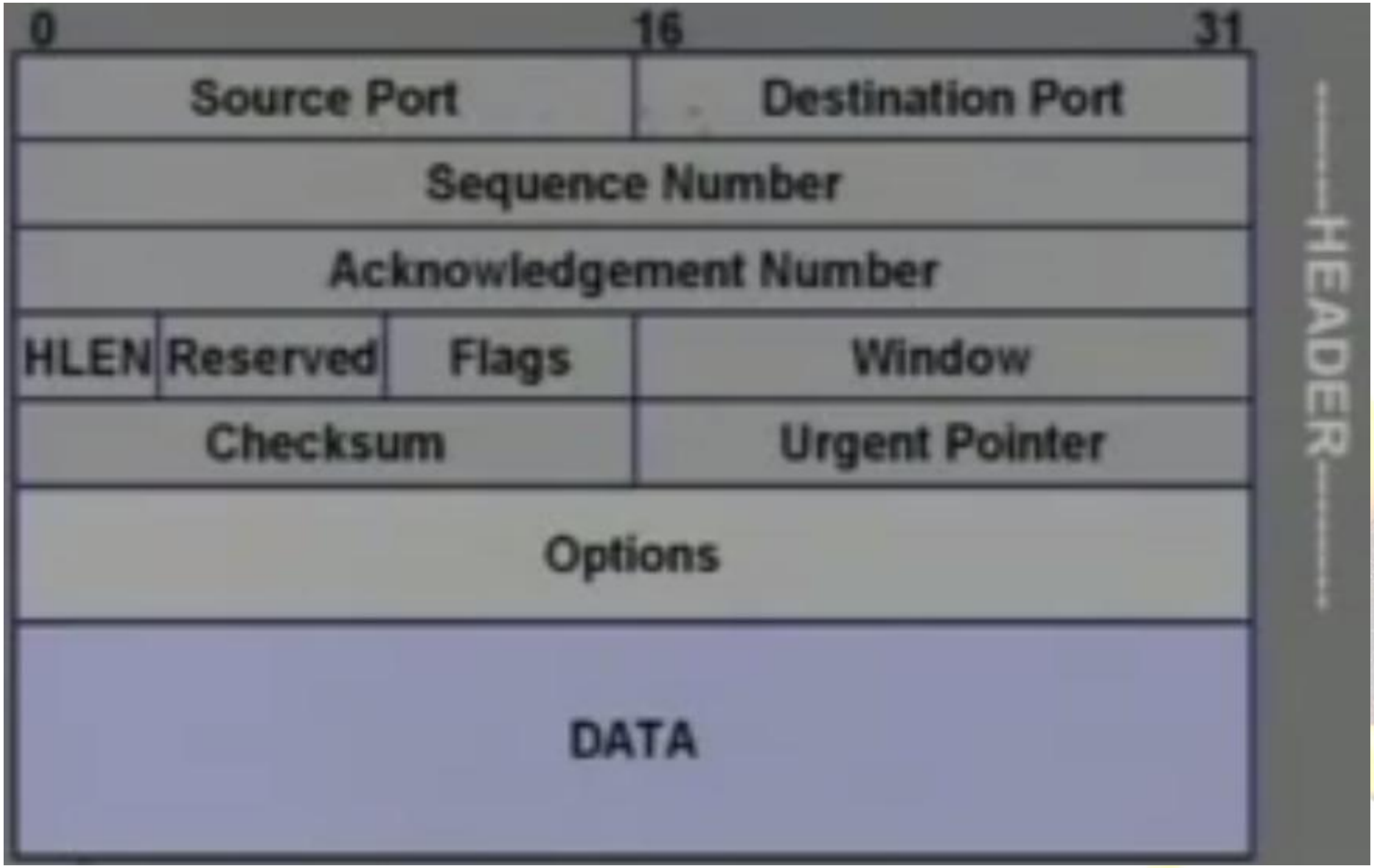
- A hierarchical addressing scheme is used to define a connection path between two hosts.
  - IP address
    - Identifies the communicating hosts.
  - Protocol identifier
    - Identifies the transport later protocol being used (TCP, UDP or anything else)
  - Port number
    - Identifies the communicating processes in the two hosts.

Association

- A set of five values that describe a unique process-to-process connection is called an **association.**
  - **The protocol (TCP or UDP)**
  - **Local host IP address (32-bit value)**
  - **Local port number (16-bit value)**
  - **Remote host IP address (32-bit value)**
  - **Remote port number (16-bit value)**
- **Example of an association:**
- **(TCP, 144.16.192.5,1785,144,16,202,57,21)**

TCP Encapsulation

Format of TCP Segment



0 16 31

Source Port | Destination Port

Sequence Number

Acknowledgement Number

HLEN | Reserved | Flags | Window

Checksum | Urgent Pointer

Options

DATA

HEADER

TCP Header Fields

- Source port (16 bits)
  - Identifies the process at the local end.
- Destination port (16 bits)
  - Identifies the process at the remote end.
- Sequence number (32 bits)
  - Used for reliable delivery of message.
  - Each byte of message is assigned a 32-bit number that is incremented sequentially.
  - The fields holds the number of the first byte in that TCP segment.

TCP Header Fields (contd.)

- Acknowledgement Number (32 bits)
  - Used by remote host to acknowledge receipt of data.
  - Contains the number of the next byte expected to be received.
- HLEN(4 bits)
  - Specifies the header length in number of 32-bit words.

TCP Header Fields

- **Flags (6 bits)**
  - There are six flags.
    - URG is set to 1 if the urgent pointer is in use.
    - A connection request is sent by making SYN=1 and ACK=0.
    - A connection is confirmed by sending SYN=1 and ACK=1.
    - When the sender has no more data, FIN=1 is sent to release the connection.

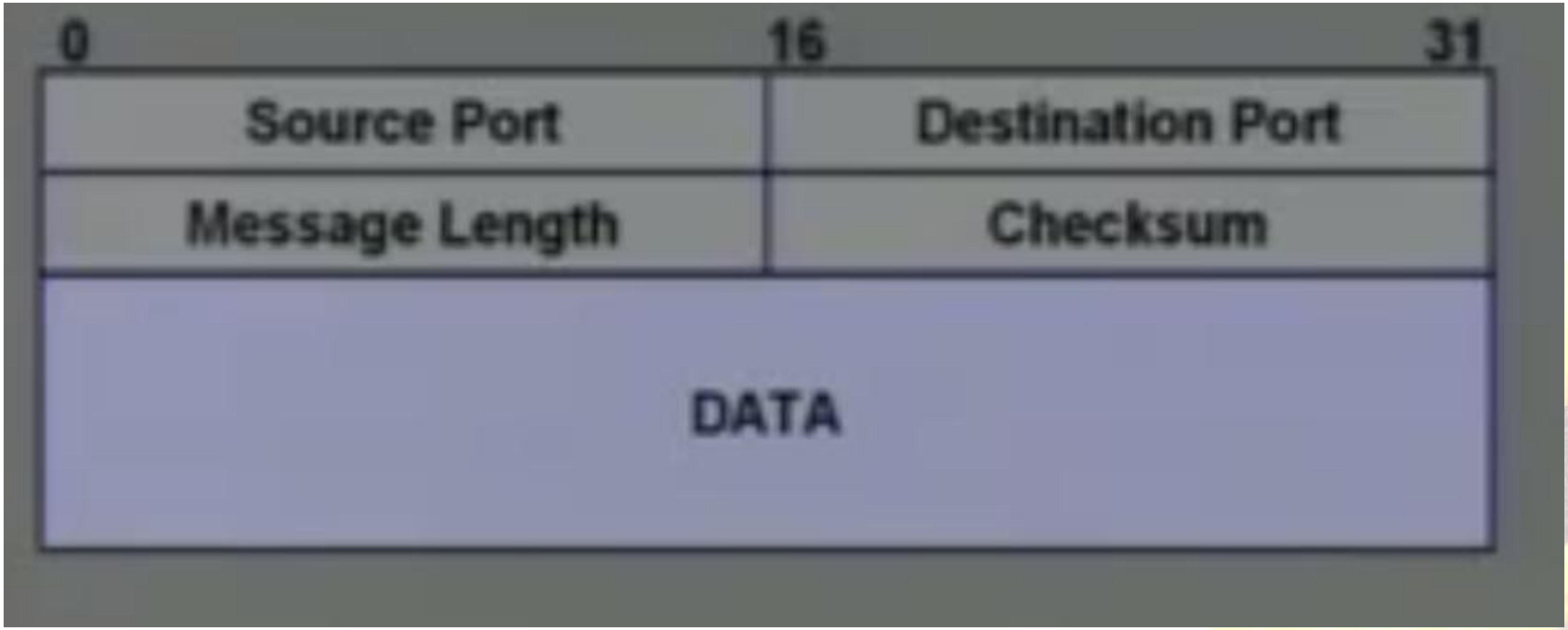
TCP Header Fields (contd.)

  - RST bit is used to reset a connection, it is also used to reject a connection attempt.
  - PSH bit indicates the push function. Used to indicate end of message.
- **Window (16 bits)**
  - Specifies how many bytes may be sent beyond the byte acknowledged.
  - This number, called **window advertisement,** can increase or decrease as needed.
  - A value of zero closes the window altogether.

TCP Header Fields (contd.)

- Checksum (16 bits)
  - Applies to the entire segment and a pseudo-header.
  - The pseudo-header contains the following IP header fields:
    - Source IP address, destination IP address, protocol, segment length.
    - TCP protects itself from misdelivery by IP (delivered to wrong host).
  - Same algorithm as used in IP.

Format of UDP Segment



0 16 31

Source Port	Destination Port
Message Length	Checksum
DATA	

UDP Header Fields

- Source Port (16 bits)
  - Identifies the process at the local end.
- Destination port (16 bits)
  - Identifies the process at the remote end.
- Message length (16 bits)
  - Specifies the size of the datagram in bytes (UDP header plus data)
- Checksum (16 bits)
  - Computed in the same way as TCP.
  - This is optional; set to zero if not used.

Berkeley Socket Interface

- How to develop a network application?
  - The best way is to use some standard and well-accepted protocol.
    - At the data link layer level, use Ethernet.
    - At the network layer level, use IP.
    - At the transport layer level, use TCP.
    - At the application layer level, use a standard API like the **Berkeley Socket Interface.**