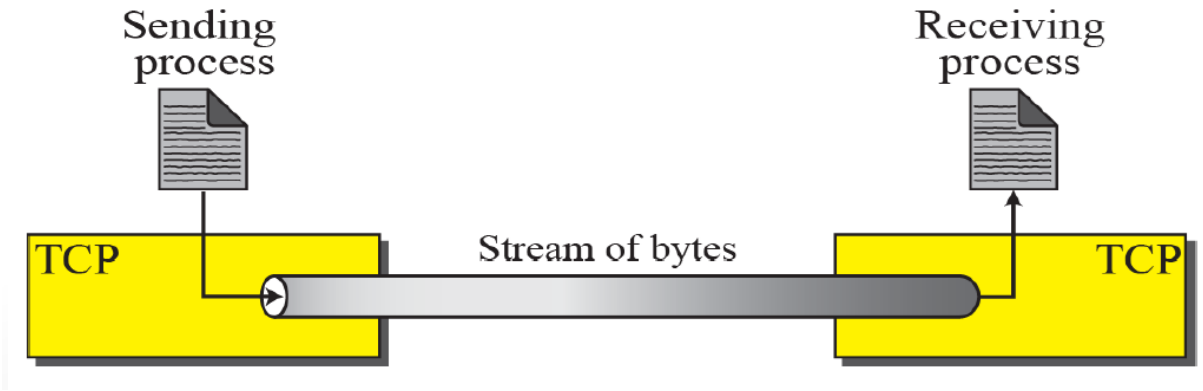
# TCP
# (Transmission Control Protocol)

- TCP is a process-to-process, reliable protocol which provides the following functionalities:
  - Stream Data transfer reliability
  - Efficient flow control
  - Full-duplex operations
  - Multiplexing




**Stream Delivery**

- TCP service is obtained by both the sender and receiver creating end points, called **sockets**.

- TCP delivers an unstructured stream of bytes identified by sequence nos.

- A key feature of TCP, and one which dominates the protocol design, is that every byte on a TCP connection has its own 32-bit sequence number.

- **Flow control** is an important feature of TCP, where in sender would not overwhelm the receiver. **Sliding window flow control** is used to achieve this.

- **Full-duplex operations** take place in TCP data transmission as data is transmitted using three-way handshaking technique.

- **Multiplexing** is used for numerous simultaneous upper-layer conversations which can be multiplexed over a single channel.

## Port addresses:

- **Range** → (0 to $2^{16}$ - 1) = (0 to 65, 535) port addresses.

- **0 – 1023** : Predefined ports/ Universal ports/ Fixed ports/ Well-known ports
- **1024 – 49,151** : Registered ports
- **49,152 – 65,535** : Dynamic ports/ Private ports

- **Predefined ports** are used for some predefined applications like http, SMTP, FTP, etc.

  http : 30

  SMTP : 25

  FTP : 21 & 20
- **Registered ports** are used for testing the software designed by different companies.
- **Dynamic ports** distinguish different process in the network environment.

*Well-known Ports used by TCP*

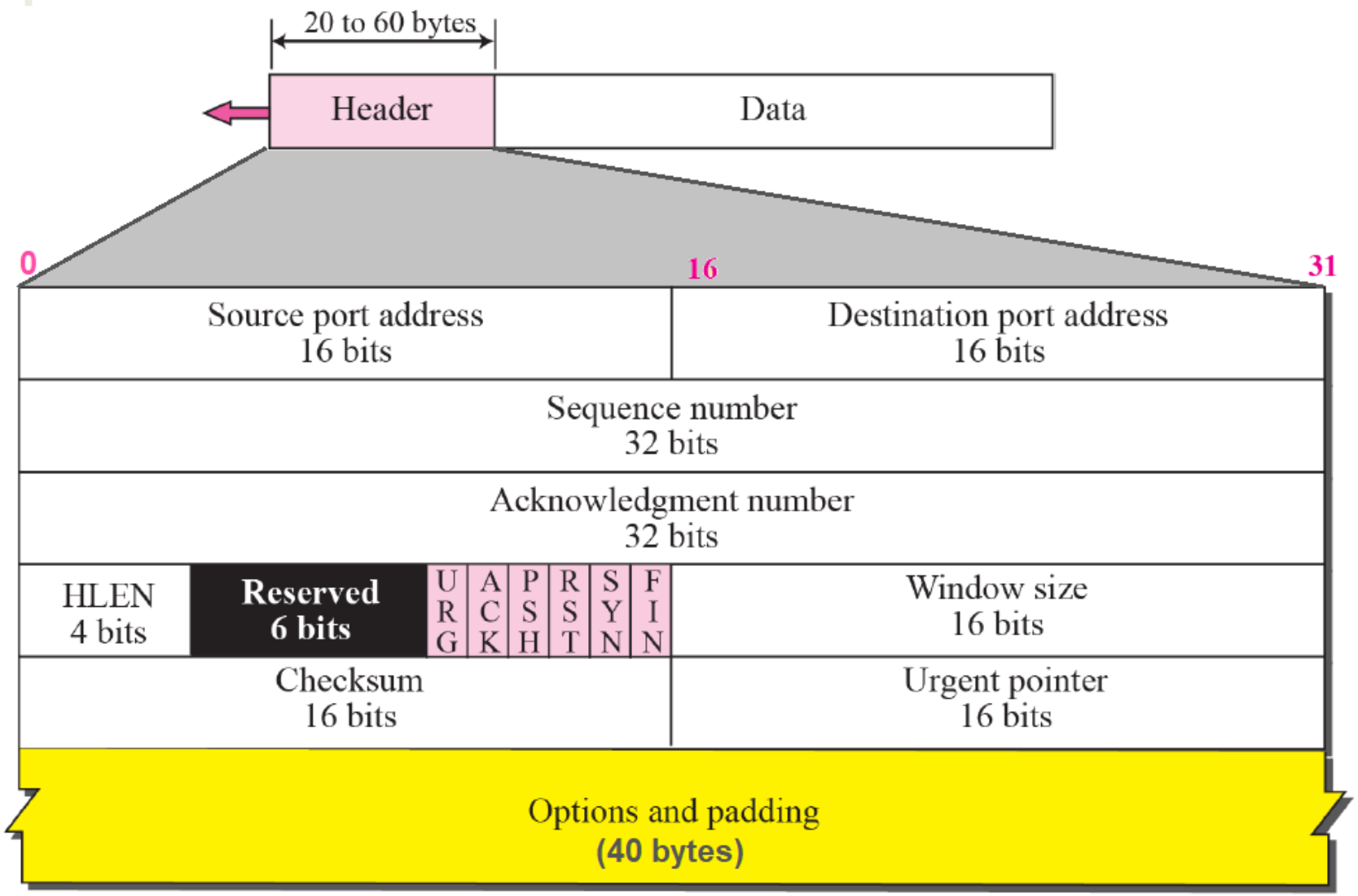
| *Port* | *Protocol* | *Description* |
|---|---|---|
| 7 | Echo | Echoes a received datagram back to the sender |
| 9 | Discard | Discards any datagram that is received |
| 11 | Users | Active users |
| 13 | Daytime | Returns the date and the time |
| 17 | Quote | Returns a quote of the day |
| 19 | Chargen | Returns a string of characters |
| 20 and 21 | FTP | File Transfer Protocol (Data and Control) |
| 23 | TELNET | Terminal Network |
| 25 | SMTP | Simple Mail Transfer Protocol |
| 53 | DNS | Domain Name Server |
| 67 | BOOTP | Bootstrap Protocol |
| 79 | Finger | Finger |
| 80 | HTTP | Hypertext Transfer Protocol |

Port 20 and 21 is used for FTP (Data and Control)

Port 20 : The server initiates a data channel to the client from its port 20, the FTP server data port.

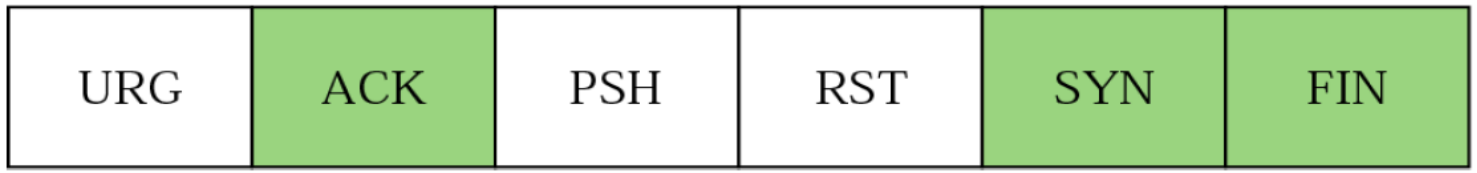
Port 21 : To establish a connection to a host to transfer a file using FTP

# TCP Segment


20 to 60 bytes
Header
Data
0
16
31
Source port address
16 bits
Destination port address
16 bits
Sequence number
32 bits
Acknowledgment number
32 bits
HLEN
4 bits
Reserved
6 bits
URG
ACK
PSH
RST
SYN
FIN
Window size
16 bits
Checksum
16 bits
Urgent pointer
16 bits
Options and padding
(40 bytes)

- **Source and Destination port fields** identify the local end points of the connection.
- **Sequence no.** identifies the current position of the 1st data byte in the segment within the entire byte stream.
- **Acknowledgement no.** identifies the next data byte the sender expects from the receiver. This is only used when ACK bit is set to 1.
- **HLEN:** Header length → 4 bits. Without options, TCP header is always 20 bytes. Max HLEN is 60 bytes.
- **Control Flags** : 6 bits are reserved for 6 TCP control flags.

| | |
|---|---|
| URG: Urgent pointer | RST: Reset the connection |
| ACK: Acknowledgment | SYN: Synchronize sequence numbers |
| PSH: Request for push | FIN: Terminate the connection |



- **URG Flag :** It is set to 1 if URGENT pointer is in use, which indicates beginning of urgent data.

- **ACK Flag :** When ACK bit is set to 1, it indicates that acknowledgement number contained in the TCP header is valid. For all TCP segments except request segment, ACK bit is set to 1.Request segment is sent for connection establishment during **Three-Way Handshake.**

- **PSH Flag :** This is the push function, which tells an application that the data should be transmitted immediately and that we don't want to wait to fill the entire TCP segment.

- **RST Flag :** This resets the connection, when you receive this you have to terminate the connection right away. This is only used when there are unrecoverable errors and it's not a normal way to finish the TCP connection.

- **SYN Flag :** SYN bit is used to synchronize the sequence numbers. When SYN bit is set to 1, it indicates the receiver that the sequence number contained in the TCP header is the initial sequence number. Request segment sent for connection establishment during Three way handshake contains SYN bit set to 1.

- **FIN Flag :** FIN bit is used to terminate the TCP connection. When FIN bit is set to 1, it indicates the receiver that the sender wants to terminate the connection. FIN segment sent for **TCP Connection Termination** contains FIN bit set to 1.

- **Window size** specifies how many bytes the receiver is willing to receive, that is, it indicates the receiving window size.

- **Checksum** field is used for error control. It verifies the integrity of data in the TCP payload. Sender adds CRC checksum to the checksum field before sending the data. Receiver rejects the data that fails the CRC check.

- **Urgent pointer :** if the URG flag is set, then this 16-bit field is an offset from the sequence number indicating the last urgent data byte. Thus, it is used to indicate where the urgent data ends.

- **Options :** Variable 0–320 bits. Options have up to three fields: Option-Kind (1 byte), Option-Length (1 byte), Option-Data (variable).

# UDP
# (User Datagram Protocol)

- The **User Datagram Protocol (UDP)** is called a connectionless, unreliable transport protocol.

- It does not add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication.

- Data transmitted using this protocol are UDP data units called **datagrams,** which has a fix-sized header of 8 bytes.

# UDP Datagram