Computer Network

Lecture-38

Dharmendra Kumar (Associate Professor)
Department of Computer Science and Engineering
United College of Engineering and Research,
Prayagraj

Process-to-Process Delivery

Transmission Control Protocol(TCP)

- Unlike UDP, TCP is a connection oriented protocol; it creates a virtual connection between two TCPs to send data. In addition, TCP uses flow and error control mechanisms at the transport level.

- TCP is called a connection-oriented, reliable transport protocol. It adds connection-oriented and reliability features to the services of IP.

Process-to-Process Delivery

TCP Services

Following are the services offered by TCP to the processes at the application layer.

Process-to-Process Communication

Like UDP, TCP provides process-to-process communication using port numbers.

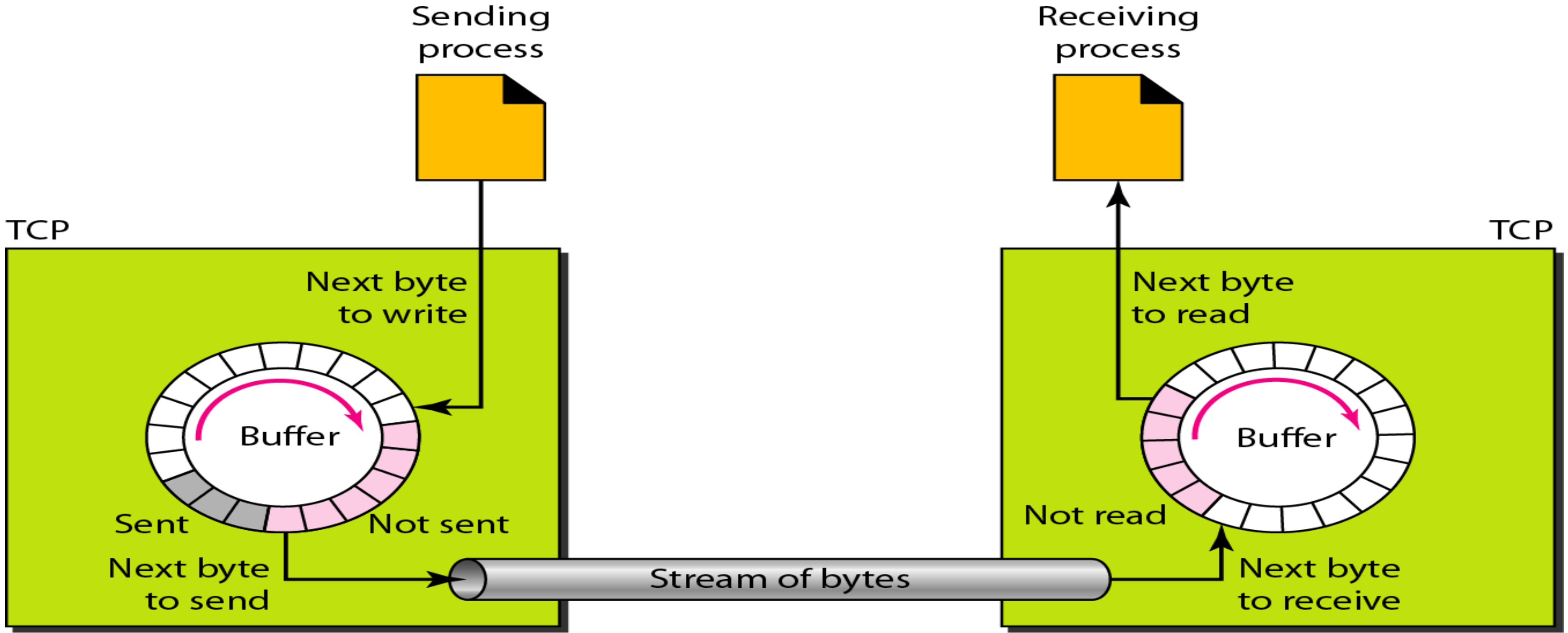
Stream Delivery Service

- ❖ TCP, unlike UDP, is a stream-oriented protocol.
- ❖ TCP allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes.
- ❖ TCP creates an environment in which the two processes seem to be connected by an imaginary "tube“ that carries their data across the Internet.
- ❖ The sending process produces (writes to) the stream of bytes, and the receiving process consumes (reads from) them.

Process-to-Process Delivery

Sending and Receiving Buffers

Because the sending and the receiving processes may not write or read data at the same speed, TCP needs buffers for storage. There are two buffers, the sending buffer and the receiving buffer, one for each direction.



Process-to-Process Delivery

- At the sending site, the buffer has three types of chambers. The white section contains empty chambers that can be filled by the sending process (producer). The gray area holds bytes that have been sent but not yet acknowledged. TCP keeps these bytes in the buffer until it receives an acknowledgment. The colored area contains bytes to be sent by the sending TCP. After the bytes in the gray chambers are acknowledged, the chambers are recycled and available for use by the sending process.
- The operation of the buffer at the receiver site is simpler. The circular buffer is divided into two areas (shown as white and colored). The white area contains empty chambers to be filled by bytes received from the network. The colored sections contain received bytes that can be read by the receiving process. When a byte is read by the receiving process, the chamber is recycled and added to the pool of empty chambers.

Process-to-Process Delivery

Segments

At the transport layer, TCP groups a number of bytes together into a packet called a segment. TCP adds a header to each segment (for control purposes) and delivers the segment to the IP layer for transmission. The segments are encapsulated in IP datagrams and transmitted.

Full-Duplex Communication

TCP offers full-duplex service, in which data can flow in both directions at the same time. Each TCP then has a sending and receiving buffer, and segments move in both directions.

Process-to-Process Delivery

Connection-Oriented Service

TCP, unlike UDP, is a connection-oriented protocol.

Reliable Service

TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data.

Process-to-Process Delivery

TCP Features

To provide the services mentioned in the previous section, TCP has several features.

Numbering System

To keeps track of the segments being transmitted or received, TCP uses two fields called the sequence number and the acknowledgment number. These two fields refer to the byte number and not the segment number.

Byte Number

TCP numbers all data bytes that are transmitted in a connection. Numbering is independent in each direction. When TCP receives bytes of data from a process, it stores them in the sending buffer and numbers them. The numbering does not necessarily start from 0. Instead, TCP generates a random number between 0 and 232 - 1 for the number of the first byte. For example, if the random number happens to be 1057 and the total data to be sent are 6000 bytes, the bytes are numbered from 1057 to 7056.

Process-to-Process Delivery

Sequence Number

After the bytes have been numbered, TCP assigns a sequence

number to each segment that is being sent. The sequence number for each segment is the number of the first byte carried in that segment.

Acknowledgment Number

The acknowledgment number defines the number of the next byte that the party expects to receive. The acknowledgment number is cumulative.

Process-to-Process Delivery

Flow Control

TCP, unlike UDP, provides flow control. The receiver of the data controls the amount of data that are to be sent by the sender. This is done to prevent the receiver from being overwhelmed with data. The numbering system allows TCP to use a byte-oriented flow control.

Error Control

To provide reliable service, TCP implements an error control mechanism.

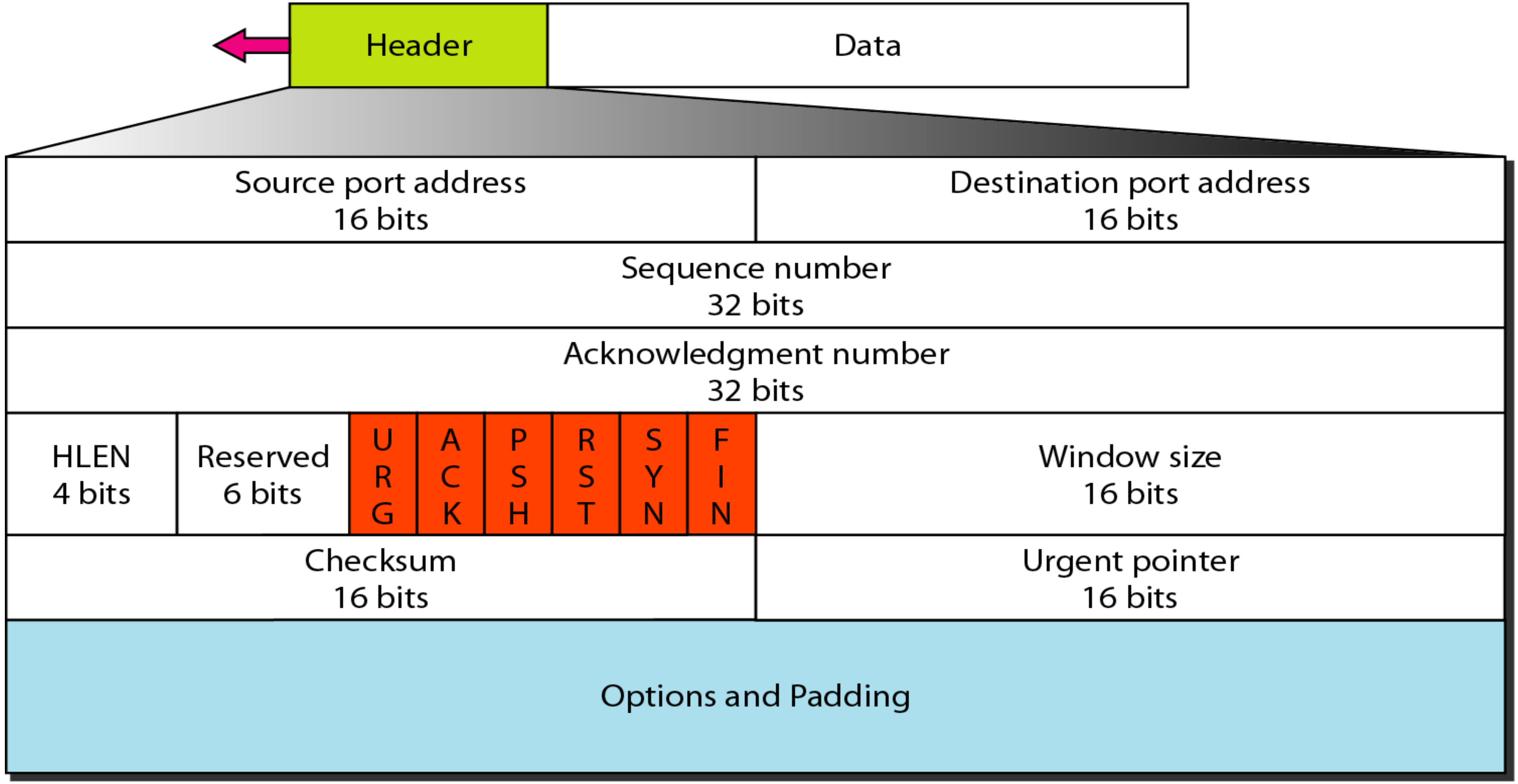
Congestion Control

TCP, unlike UDP, takes into account congestion in the network. The amount of data sent by a sender is not only controlled by the receiver (flow control), but is also determined by the level of congestion in the network.

Process-to-Process Delivery

TCP segment format

The format of a segment is shown in the following figure:-



Header

Data

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

Process-to-Process Delivery

The segment consists of a 20- to 60-byte header, followed by data from the application program. The header is 20 bytes if there are no options and up to 60 bytes if it contains options.

Source port address: This is a 16-bit field that defines the port number of the application program in the host that is sending the segment.

Destination port address: This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment.

Sequence number: This 32-bit field defines the number assigned to the first byte of data contained in this segment.

Acknowledgment number: This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number x from the other party, it defines x + 1 as the acknowledgment number. Acknowledgment and data can be piggybacked together.

Process-to-Process Delivery

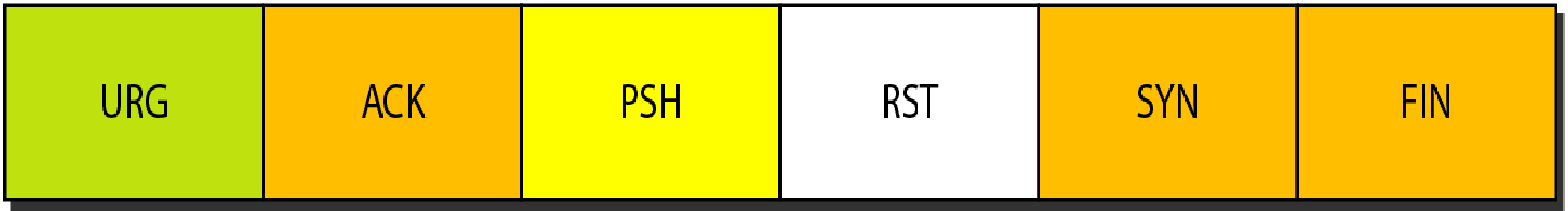
Header length: This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes.

Reserved: This is a 6-bit field reserved for future use.

Control: This field defines 6 different control bits or flags as shown in figure. One or more of these bits can be set at a time.

URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push

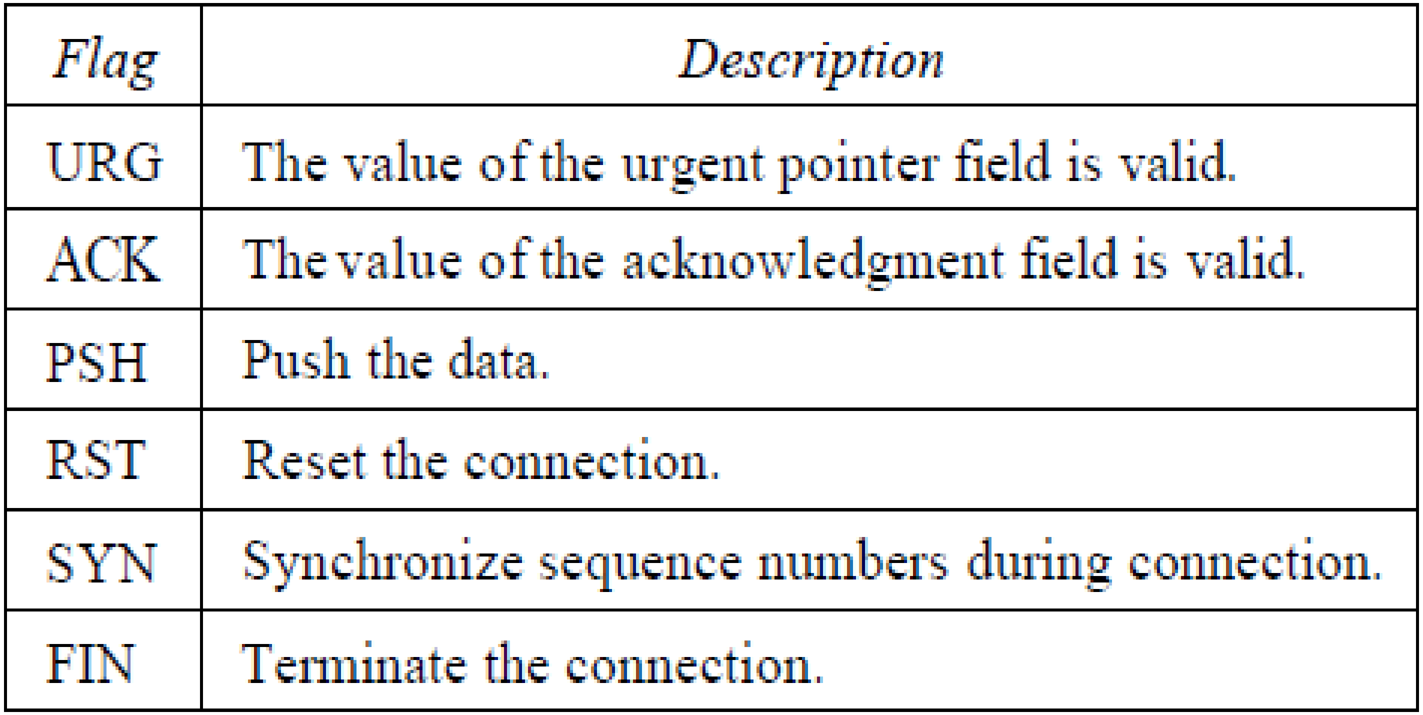
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection



URG	ACK	PSH	RST	SYN	FIN

Process-to-Process Delivery

These bits enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP.



Flag	*Description*
URG	The value of the urgent pointer field is valid.
ACK	The value of the acknowledgment field is valid.
PSH	Push the data.
RST	Reset the connection.
SYN	Synchronize sequence numbers during connection.
FIN	Terminate the connection.

Process-to-Process Delivery

Window size: This field defines the size of the window, in bytes, that the other party must maintain. Note that the length of this field is 16 bits, which means that the maximum size of the window is 65,535 bytes. This value is normally referred to as the receiving window (rwnd) and is determined by the receiver. The sender must obey the dictation of the receiver in this case.

Checksum: This 16-bit field contains the checksum.

Urgent pointer: This 16-bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data. It defines the number that must be added to the sequence number to obtain the number of the last urgent byte in the data section of the segment.

Options: There can be up to 40 bytes of optional information in the TCP header.