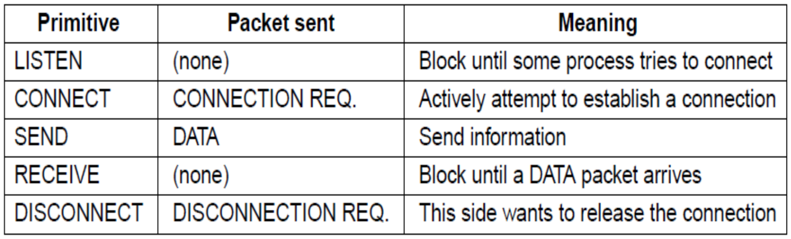
Chapter 6: The Transport Layer

Transport Service Primitives (1)

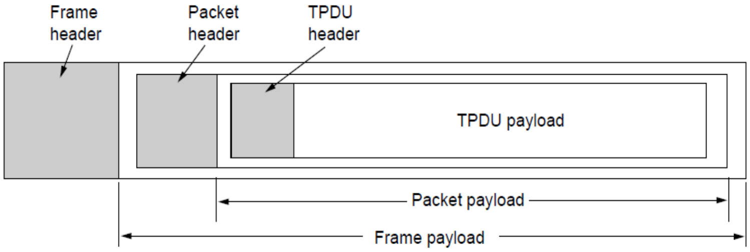
The primitives for a simple transport service



Primitive	Packet sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

Transport Service Primitives (2)

Nesting of TPDUs, packets, and frames



The Internet Transport Protocols

- The Internet has two main protocols in the transport layer, a connectionless protocol and a connection-oriented one.
- The connectionless protocol is UDP. The connection-oriented protocol is TCP.
- TCP (Transmission Control Protocol) to ensure destination received segments
- UDP (User Datagram Protocol) to send segments without assurance of delivery

Introduction to UDP

User Datagram protocol is connectionless and unreliable.

It has very limited error checking capability.

It is a very simple protocol and it can be used with minimum overhead.

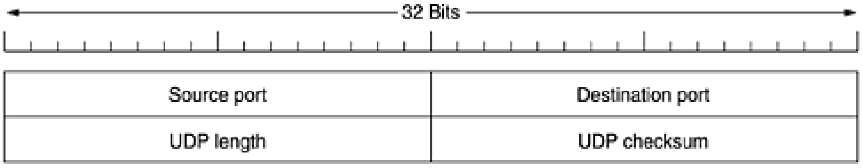
UDP is used when a process needs to send a small data without any issue of reliability.

UDP takes less time as compared to TCP.

It is a good protocol for data flowing in one direction.

It is simple and suitable for query based communication.

Figure 6-27. The UDP header.



UDP packets are called “User Datagrams” which contains a fixed size header of 8 bytes.

UDP Header contains four main parameters.

Source Port: This 16 bit information is used to identify the source port of the packet.

Destination Port: This 16 bit information is used to identify the destination port of the packet.

UDP length: Specifies entire length of UDP packet (including header).It is a 16 bit field. Minimum size of header is 8 byte

UDP checksum: Stores the checksum value guaranteed by the sender before sending.

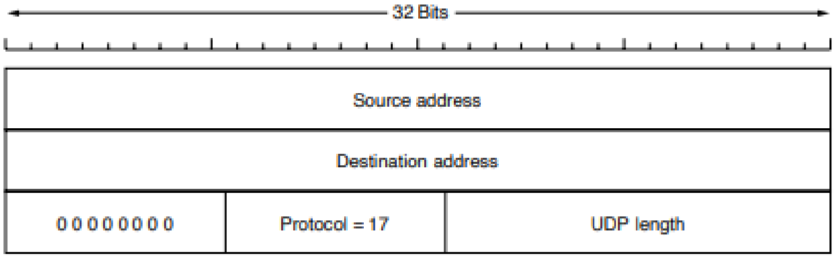


Figure 6-28. The IPv4 pseudoheader included in the UDP checksum.

The pseudoheader for the case of IPv4 is shown in Fig. 6-28.

It contains the 32-bit IPv4 addresses of the source and destination machines, the protocol number for UDP (17), and the byte count for the UDP segment (including the header).

Including the pseudoheader in the UDP check- sum computation helps detect misdelivered packets, but including it also violates the protocol hierarchy.

<u>The Internet Transport Protocols: TCP</u>

It is a connection oriented protocol

It is a reliable protocol as it uses flow and error control mechanism of transport layer.

TCP ensures the data reaches the destination in order.

In TCP, a segment carries data and control information**.**

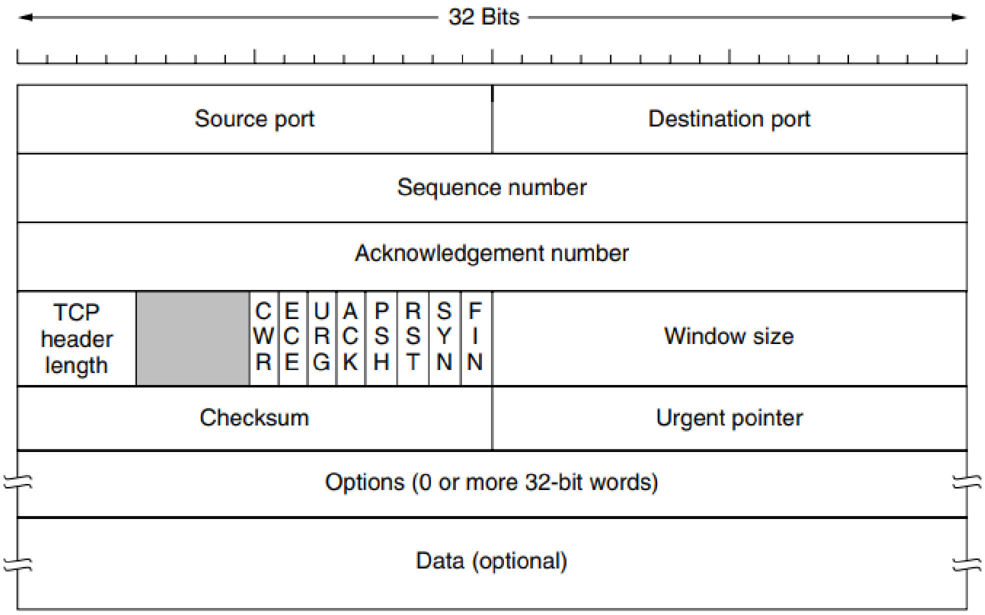


Figure 6-36. The TCP header.

- **Source port**: this is a 16 bit field that specifies the port number of the sender.
- **Destination port**: this is a 16 bit field that specifies the port number of the receiver.
- **Sequence number**: the sequence number is a 32 bit field that indicates how much data is sent during the TCP session. When you establish a new TCP connection (3 way handshake) then the initial sequence number is a random 32 bit value. The receiver will use this sequence number and sends back an acknowledgment.
- **Acknowledgment number**: this 32 bit field is used by the receiver to request the next TCP segment. This value will be the sequence number incremented by 1.
- **DO/TCP Header Length**: this is the 4 bit data offset field, also known as the header length. It indicates the length of the TCP header so that we know where the actual data begins.
- **RSV**: these are 3 bits for the reserved field. They are unused and are always set to 0.
- **Flags**: there are 8 bits for flags, we also call them control bits. We use them to establish connections, send data and terminate connections:
  - **CWR and ECE**: TCP supports ECN (Explicit congestion notification) using two flags. CWR (Congestion window reduced) ECE(ECN Echo)
  - **URG**: urgent pointer. When this bit is set, the data should be treated as priority over other data.
  - **ACK**: used for the acknowledgment.
  - **PSH**: this is the push function. This 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**: 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**: we use this for the initial three way handshake and it’s used to set the initial sequence number.
- **FIN**: this finish bit is used to end the TCP connection. TCP is full duplex so both parties will have to use the FIN bit to end the connection. This is the normal method how we end an connection.

- **Window**: the 16 bit window field specifies how many bytes the receiver is willing to receive. It is used so the receiver can tell the sender that it would like to receive more data than what it is currently receiving. It does so by specifying the number of bytes beyond the sequence number in the acknowledgment field.
- **Checksum**: 16 bits are used for a checksum to check if the TCP header is OK or not.
- **Urgent pointer**: these 16 bits are used when the URG bit has been set, the urgent pointer is used to indicate where the urgent data ends.
- **Options**: this field is optional and can be anywhere between 0 and 32 bits.

TCP vs UDP

- Both use **port numbers**
  - ❑ application-specific construct serving as a communication endpoint and consist of 16-bit unsigned integer, thus ranging from 0 to 65535 to provide **end-to-end** transport

Differences between TCP and UDP:

The main differences between TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are:

Basis	**Transmission Control Protocol (TCP)**	**User Datagram Protocol (UDP)**
Type of Service	TCP is a connection-oriented protocol. Connection orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data.	UDP is the Datagram-oriented protocol. This is because there is no overhead for opening a connection, maintaining a connection, or terminating a connection. UDP is efficient for broadcast and multicast types of network transmission.
Reliability	TCP is reliable as it guarantees the delivery of data to the destination router.	The delivery of data to the destination cannot be guaranteed in UDP.
Error checking mechanism	TCP provides extensive error-checking mechanisms. It is because it provides flow control and	UDP has only the basic error-checking mechanism using checksums.

	acknowledgment of data.	
Acknowledgment	An acknowledgment segment is present.	No acknowledgment segment.
Sequence	Sequencing of data is a feature of Transmission Control Protocol (TCP). this means that packets arrive in order at the receiver.	There is no sequencing of data in UDP. If the order is required, it has to be managed by the application layer.
Speed	TCP is comparatively slower than UDP.	UDP is faster, simpler, and more efficient than TCP.
Retransmission	Retransmission of lost packets is possible in TCP, but not in UDP.	There is no retransmission of lost packets in the User Datagram Protocol (UDP).
Header Length	TCP has a (20-60) bytes variable length header.	UDP has an 8 bytes fixed-length header.
Weight	TCP is heavy-weight.	UDP is lightweight.
Handshaking Techniques	Uses handshakes such as SYN, ACK, SYN-ACK	It’s a connectionless protocol i.e. No handshake
Broadcasting	TCP doesn’t support Broadcasting.	UDP supports Broadcasting.
Protocols	TCP is used by HTTP, HTTPs, FTP, SMTP and Telnet.	UDP is used by DNS, DHCP, TFTP, SNMP, RIP, and VoIP.
Stream Type	The TCP connection is a byte stream.	UDP connection is a message stream.
Overhead	Low but higher than UDP.	Very low.
Applications	This protocol is primarily utilized in situations when a safe and trustworthy communication procedure is necessary, such as in email, on the web surfing, and in military services.	This protocol is used in situations where quick communication is necessary but where dependability is not a concern, such as VoIP, game streaming, video, and music streaming, etc.