Following five functions are used in data delivery process: -

* Segmentation
* Connection multiplexing
* Connection oriented or connection less delivery
* Reliability through acknowledgement and sequencing
* Flow control through windowing

From these functions, to ensure the accuracy in delivery process, TCP supports all functions while in order to provide the highest possible speed, UDP supports only the second function.

Let’s understand each function in detail and compare the way in which both protocols provide it.

**Segmentation**

Segmentation is the process of dividing large data stream into smaller pieces. This functionality allows a host to send or receive a file of any size over the any size of network. For example, if network bandwidth is 1 Mbps and file size is 100 Mb, host can divide the file in 100 or more pieces. Once a piece becomes less or equal to the network bandwidth in size, it can be transferred easily. Destination host, upon receiving all pieces, joins them back to reproduce the original file.

TCP supports segmentation while UDP does not. It means if an application wants to use the TCP to send its data, it can give the data to TCP in actual size. Based on several conditions such as data size and available network bandwidth, if segmentation is required, TCP does it on its own before packing data for transmission.

But if an application wants to use UDP to send its data, it can’t give the data to UDP in actual size. It has to use its own mechanism to detect whether segmentation is required or not. And if segmentation is required, it has to do it on its own before giving data to UDP.

**Packing data for transmission**

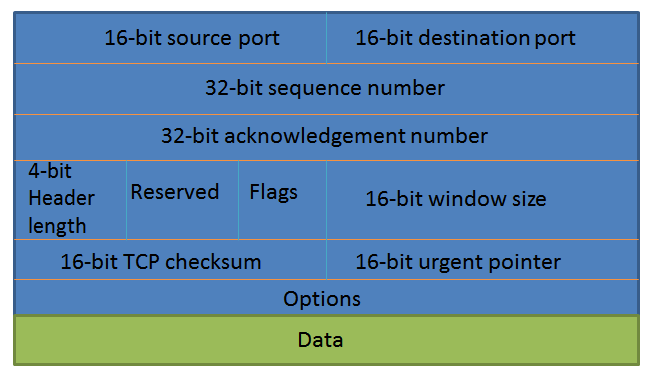
Both protocols pack data in similar fashion. Both add a header with each data piece. A header mainly contains two types of information;

1. The information that is required to send the segment at the correct destination.
2. The information that is required to support the protocol specific features.

Both TCP and UDP add first type of information in same manner. Both use two fields for this information; source port and destination port. Information about the application that is sending the data and the information about the application that will receive the data are added in source port field and in destination port field respectively.

Protocols add second type of information based on the services they offer. TCP offers several protocol specific services such as segmentation, windowing, flow control, etc. To provide these services, it adds the necessary information in the header.

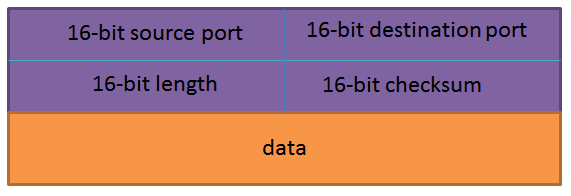
Following figure shows a data piece with the TCP header.



|  |  |
| --- | --- |
| Field | Description |
| Source port | Used to identify the application that is sending data from the source host |
| Destination port | Used to identify the application that will receive the data at destination host |
| Sequence Number | Used to identify the lost segments and maintain the sequencing in transmission. |
| Acknowledgment Number | Used to send a verification of received segments and to ask for the next segments |
| Header Length | A number that indicates where the data begin in segment |
| Reserved | Reserve for future use. Always set to zero. |
| Code Bits | Used to define the control functions such as setting up and terminating the session |
| Window size | Used to set the number of segments that can be sent before waiting for a confirmation from the destination. |
| Checksum | CRC (cyclic redundancy check) of the header and data piece. |
| Urgent | Used to point any urgent data in the segment. |
| Options | Used to define any additional options such as maximum segment size |
| Data | A data piece that is produced from the segmentation |

On other hand, UDP neither provides any protocol specific service, nor adds any additional information in the header.

Following figure shows data with UDP header.

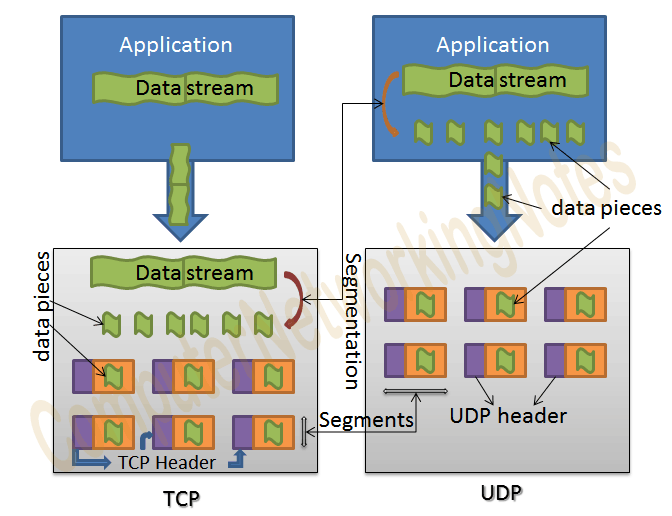


|  |  |
| --- | --- |
| Field | Description |
| Source port | Port number of the application that is transmitting data from the source computer |
| Destination port | Port number of the application that will receive the data at destination. |
| Length | Denotes the length of the UDP header and the UDP data |
| Checksum | CRC of the complete segment |
| Data | Data which it received from the application |

**Segment**

Once a header is attached with the data piece (generated from the segmentation in TCP or received from the application in UDP), it is referred as a segment.

Following figure shows how segmentation works in both protocols.



**Key points**

* TCP uses segmentation while UDP does not.
* Both protocols use different types of header to pack the data for transmission.
* UDP header contains information only about the compulsory functions and it is 8 bytes in the length.
* TCP header contains information for both compulsory and optional functions. TCP header is 20 bytes and 24 bytes in length without options and with options respectively.