



White Paper on Basics of Networking

Part 1 - TCP in Brief

by Tasawar Jalali

Friday, October 17, 2003



Table of Contents

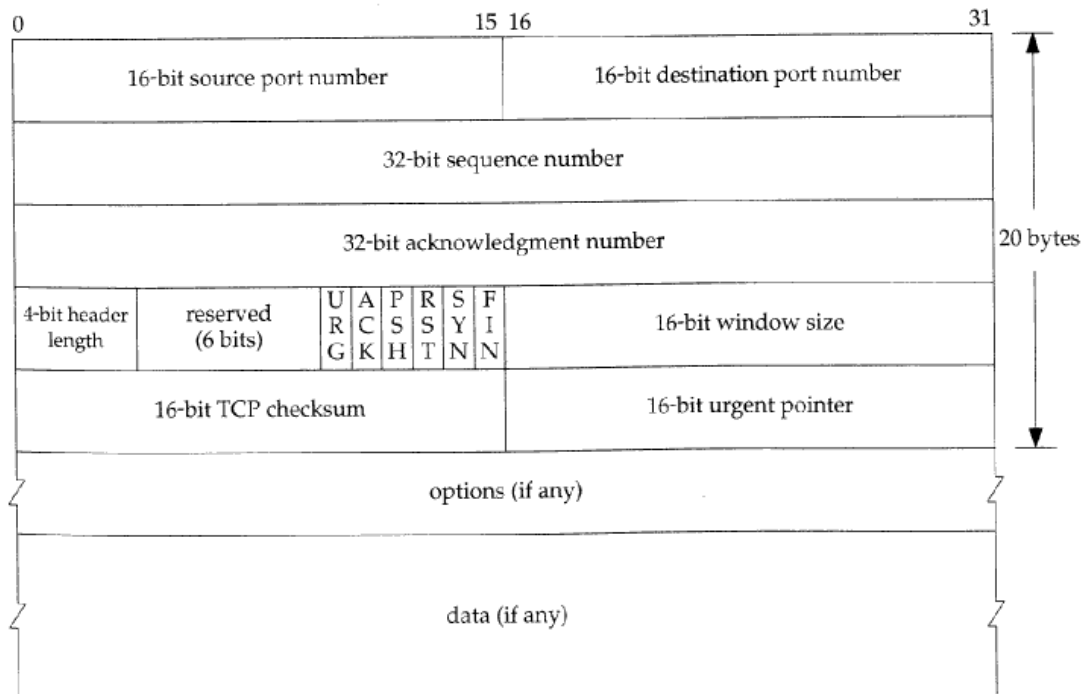
TCP Summary	3
TCP Header	3
TCP State Transition Diagram.....	5
Sliding Window Algorithm.....	6
TCP Packet Exchange Diagram.....	7
References	8

TCP Summary

TCP provides a connection oriented, reliable, byte stream service. The term connection-oriented means the two applications using TCP must establish a TCP connection with each other before they can exchange data. It is a full duplex protocol, meaning that each TCP connection supports a pair of byte streams, one flowing in each direction. TCP includes a flow-control mechanism for each of these byte streams that allows the receiver to limit how much data the sender can transmit. TCP also implements a congestion-control mechanism.

TCP Header

TCP header consists of following information:



The **SrcPort** and **DstPort** fields identify the source and destination ports, respectively. These two fields plus the source and destination IP addresses, combine to uniquely identify each TCP connection.

The **sequence number** identifies the byte in the stream of data from the sending TCP to the receiving TCP that the first byte of data in this segment represents.



The **Acknowledgement number** field contains the next sequence number that the sender of the acknowledgement expects to receive. This is therefore the sequence number plus 1 of the last successfully received byte of data. This field is valid only if the ACK flag is on. Once a connection is established the **ACK** flag is always on.

The **Acknowledgement**, **SequenceNum**, and **AdvertisedWindow** fields are all involved in TCP's sliding window algorithm. The **Acknowledgement** and **AdvertisedWindow** fields carry information about the flow of data going in the other direction. In TCP's sliding window algorithm the receiver advertises a window size to the sender. This is done using the **AdvertisedWindow** field. The sender is then limited to having no more than a value of **AdvertisedWindow** bytes of unacknowledged data at any given time. The receiver sets a suitable value for the **AdvertisedWindow** based on the amount of memory allocated to the connection for the purpose of buffering data.

The **header length** gives the length of the header in 32-bit words. This is required because the length of the options field is variable.

The 6-bit **Flags** field is used to relay control information between TCP peers. The possible flags include **SYN**, **FIN**, **RESET**, **PUSH**, **URG**, and **ACK**.

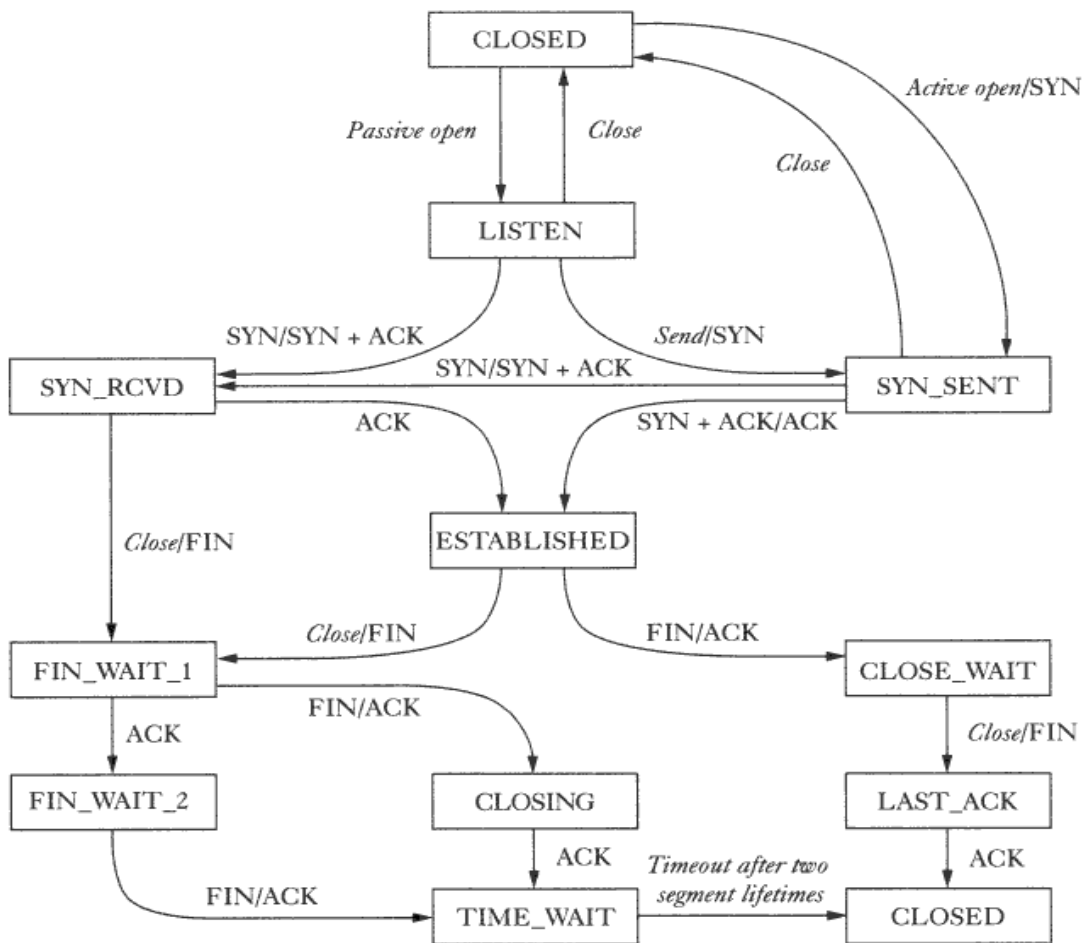
- The **SYN** and **Fin** flags are used when establishing and terminating a TCP connection, respectively.
- The **ACK** flag is set any time the **Acknowledgement** field is valid, implying that the receiver should pay attention to it.
- The **URG** flag signifies that this segment contains urgent data. When this flag is set, the **UrgPtr** field indicates where the non-urgent data contained in this segment begins.
- The **PUSH** flag signifies that the sender invoked the push operation, which indicates to the receiving side of TCP that it should notify the receiving process of this fact.
- Finally, the **RST** flag signifies that the receiver has become confused and so wants to abort the connection.

The **Checksum** covers the TCP segment: the TCP header and the TCP data. This is a mandatory field that must be calculated by the sender, and then verified by the receiver.

The **Option** field is the maximum segment size option, called the MSS. Each end of the connection normally specifies this option on the first segment exchanged. It specifies the maximum sized segment the sender wants to receive. The **data** portion of the TCP segment is optional.

TCP State Transition Diagram

The operation of TCP with regard to connection establishment and connection termination can be specified with a state transition diagram. There are 11 different states defined for a connection as shown below.





Sliding Window Algorithm

The "window" is the maximum amount of data server can send without having to wait for ACKs.

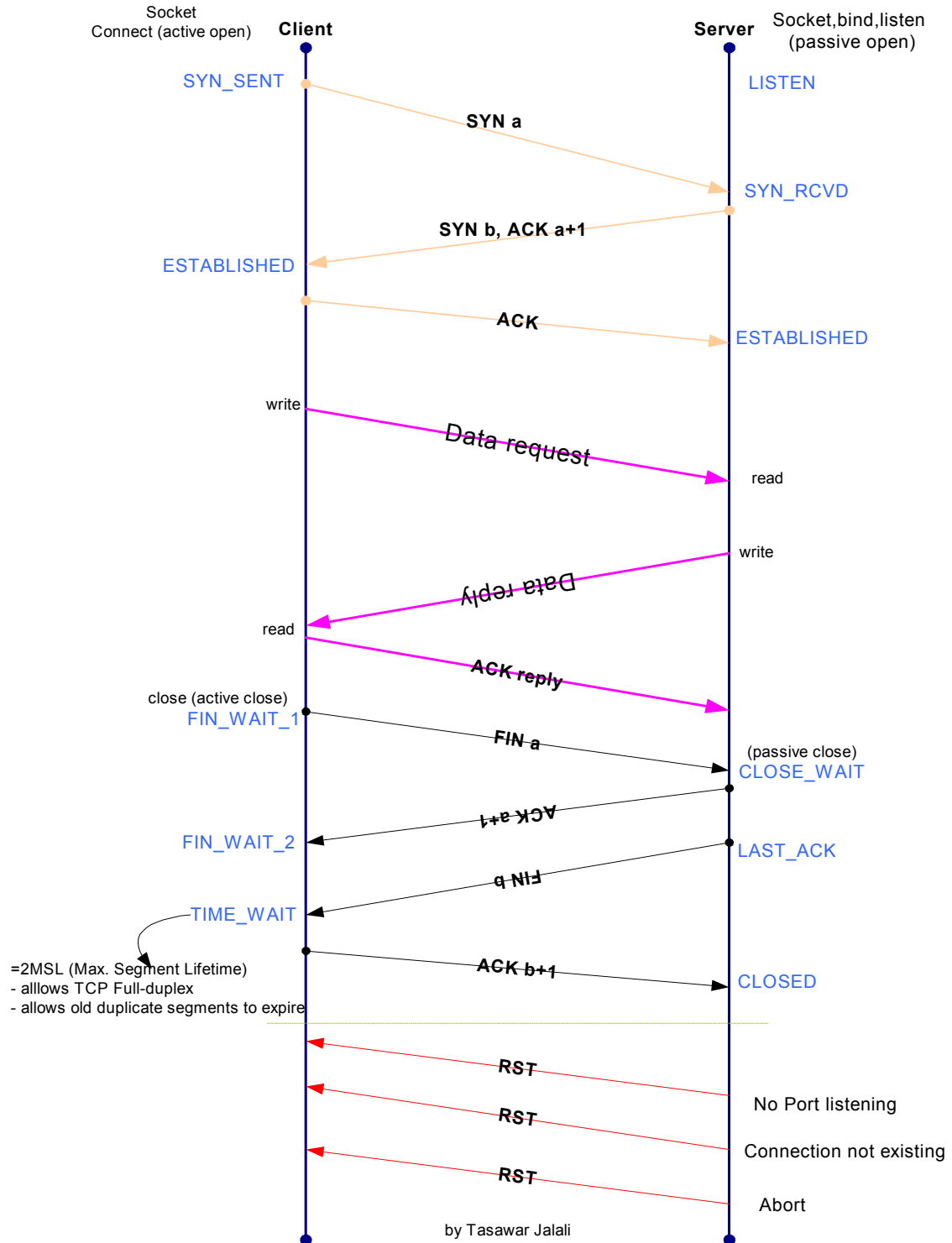
In summary, the operation of the algorithm is as follows:

1. Transmit all the new segments in the window.
2. Wait for acknowledgement/s to come (several packets can be acknowledged in the same ACK).
3. Slide the window to the indicated position and set the window size to the value advertised in the acknowledgement.

When we wait for an acknowledgement to a packet for some time and it has not arrived yet, the packet is retransmitted. When the acknowledgement arrives, it causes the window to be repositioned and the transmission continues from the packet following the one transmitted last.

TCP Packet Exchange Diagram

Following diagram shows how a three-way TCP handshake occurs and different TCP transition states





References

- Stevens, Richard W.: *UNIX Network Programming: Networking APIs: Sockets and XTI*. Volume 1. Prentice Hall; Englewood Cliffs 1997.
- <http://www4.ulpgc.es/tutoriales/tcpip/pru/3376fm.htm>