HOCHSCHULE FURTWANGEN UNIVERSITY

# TERM PAPER

Title: **TRANSMISSION CONTROL PROTOCOL**

Name: Ravi Kiran Sattineni
Imm Nr: 230458

Course: E-Business Technology
Professor: Dr. Eduard Heindl
Date: 21.05.2008

**Declaration**

**I here by declare that work done in this term paper is based on my own research and thoughts. The sources that are used as references are mentioned at the end of the paper.**

Ravi Kiran Sattineni

TABLE OF CONTENTS

# 1 INTRODUCTION

The **Transmission Control Protocol** (**TCP**) is one of the core protocols of the Internet protocol suite. TCP provides reliable, in-order delivery of a stream of bytes, making it suitable for applications like file transfer and e-mail. It is so important in the Internet protocol suite that sometimes the entire suite is referred to as "the TCP/IP protocol suite." TCP is the transport protocol that manages the individual conversations between web servers and web clients.

## 1.1 Reasons for using TCP

Tcp achieves a seemingly impossible task: it uses the unreliable datagram service Offered by IP when sending data to another computer,but provides a reliable data delivery service to application programs. Tcp must compensate for loss or delay in an internet to provide efficient data transfer, and it must do so with out overloading the underlying networks and routers. After reviewing the service that tcp provides to applications, we examine the techniques tcp uses to achieve reliability.

Tcp is designed to operate over a wide ariety of networks and to provide virtual circuit service with orderly, reliable transmission of user data. Tcp serves as the basis for a reliable Inter-process communication mechanism on top of unreliable subnetworking of packets where loss, damage , duplication , delay or misordering of packets can occur. It is a complex protocol – having to deal , for example with detection of lost packets, automatic retransmission and pathological problems like the handling of delayed duplicate packets.

The potential for providing robustness in the face of unreliable media makes tcp well-suited to a wide variety of multi-machine communications applications. Tcp supports interconnected networks. It was specifically designed to operate above IP, which is at ISO layer three (the network layer).

## 1.2 The services tcp provides to applications

From an application program's nt of view, the service offered by tcp has seven major features:

1.connection orientation: TCP provides connection-oriented service in which an application must first require a connection to a destination ,and then use the connection to transfer data.

2.point-to-point communication: Each tcp connection has exactly two endpoints

3.complex reliability: TCP guarantees that the data sent across a connection will be delivered exactly as sent, with no data missing or out of order.

4.Full duplex communication: A TCP connection allows data to flow in either direction, and Allows either application program to send data at any time. Tcp can buffer outgoing and incoming data in both directions, making it possible for an application to send data and then To continue computation while the data is being transferred.

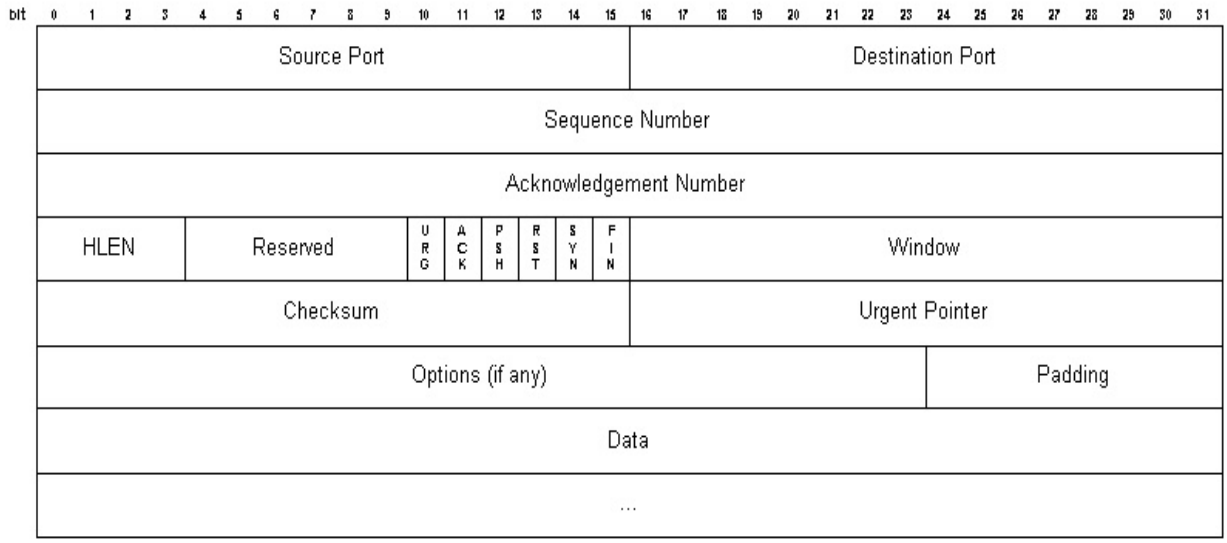
5.Stream interface: we say that TCP provides a stream interface in which an application sends a continuous sequence of octets across a connection. That is, TCP does not provide a notion of records, and does not guarantee that data will be delivered in the same pieces at the receiving application in the same pieces that it was transferred by the sending application.

6.Reliable connection startup: TCP requires that when two applications create a connection, both must agree to the new connection; duplicate packets used in previous connections will Not appear to be valid responses or otherwise interfere with the new connection.

7.Graceful connection shutdown: An application program can open a connection, send arbitary amounts of data, and then request that the connection be shut down. Tcp guarantees to deliver all data reliably before closing the connection.

## 1.3 TCP SEGMENT FORMAT

Remember that the combination of TCP header and TCP in one packet is called a TCP segment. Figure 1 depicts the format of all valid TCP segments. The size of the header without options is 20 bytes



Tcp Segment consists of a source port,destination port,sequence number,acknowledgement number,data offset,reserved,urgent pointer,checksum for errors.

# 2.OPERATION OF TCP

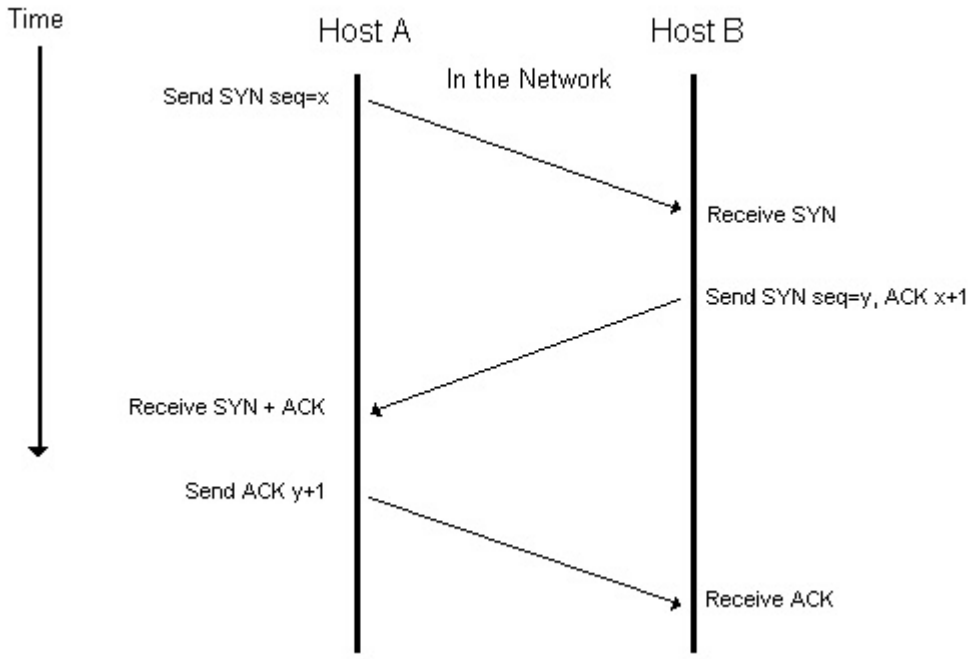
## 2.1 Connection Establishment and Termination

TCP provides a connection-oriented service over packet switched networks. Connection-oriented implies that there is a virtual connection between two endpoints. There are three phases in any virtual connection. These are the connection establishment, data transfer and connection termination phases.

## 2.2 Three-Way Handshake

In order for two hosts to communicate using TCP they must first establish a connection by exchanging messages in what is known as the three-way handshake. The diagram below depicts the process of the three-way handshake.

**Figure 2.1 - TCP Connection Establishment**



From figure 2.1, it can be seen that there are three TCP segments exchanged between two hosts, Host A and Host B. Reading down the diagram depicts events in time.

To start, Host A initiates the connection by sending a TCP segment with the SYN control bit set and an initial sequence number (ISN) we represent as the variable x in the sequence number field.

At some moment later in time, Host B receives this SYN segment, processes it and responds with a TCP segment of its own. The response from Host B contains the SYN control bit set and its own ISN represented as variable y. Host B also sets the ACK control bit to indicate the next expected byte from Host A should contain data starting with sequence number x+1.

When Host A receives Host B's ISN and ACK, it finishes the connection establishment phase by sending a final acknowledgement segment to Host B. In this case, Host A sets the ACK control bit and indicates the next expected byte from Host B by placing acknowledgement number y+1 in the acknowledgement field.

In addition to the information shown in the diagram above, an exchange of source and destination ports to use for this connection are also included in each senders' segments.

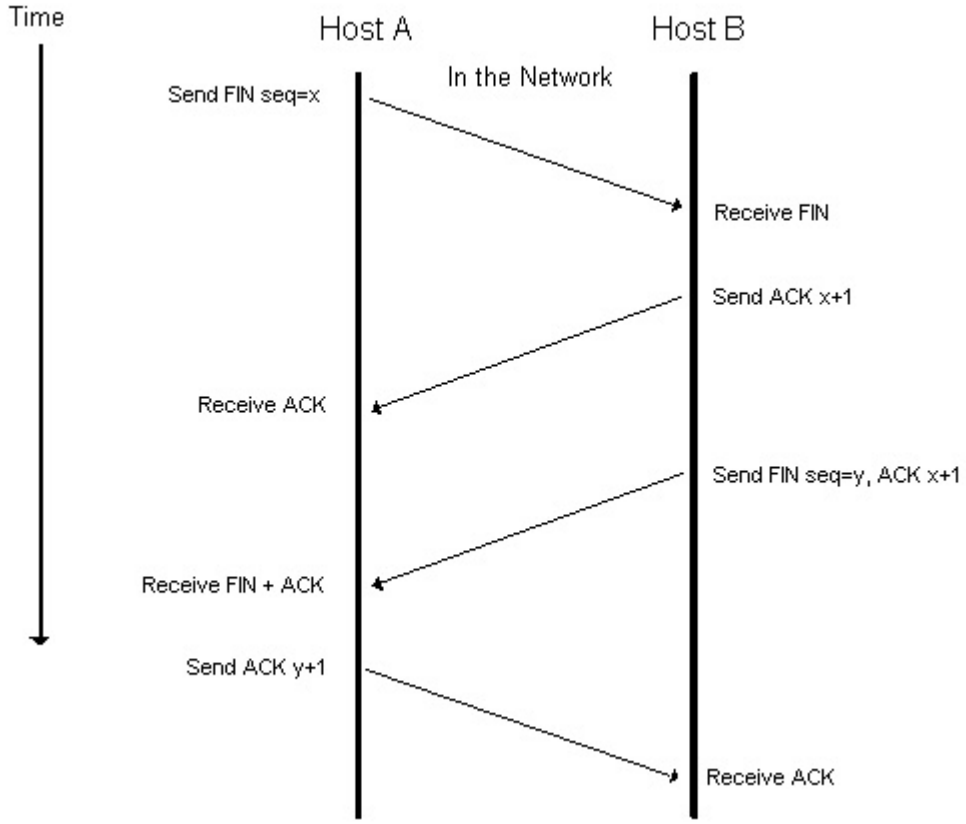
### 2.3 Connection Termination

In order for a connection to be released, four segments are required to completely close a connection. Four segments are necessary due to the fact that TCP is a full-duplex protocol, meaning that each end must shut down independently. The connection termination phase is shown in figure 3 below.

Notice that instead of SYN control bit fields, the connection termination phase uses the FIN control bit fields to signal the close of a connection.

To terminate the connection in our example, the application running on Host A signals TCP to close the connection. This generates the first FIN segment from Host A to Host B. When Host B receives the initial FIN segment, it immediately acknowledges the segment and notifies its destination application of the termination request. Once the application on Host B also decides to shut down the connection, it then sends its own FIN segment, which Host A will process and respond with an acknowledgement.

**Figure 2.2 TCP Connection Termination**




## 2.4 Achieving Reliability

One of the most important features of TCP is reliable end-to-end data delivery. In order to provide reliability, TCP must recover from data that is damaged, lost, duplicated, or delivered out of order by the Network layer. TCP uses the Positive Acknowledgment Retransmission (PAR) scheme for achieving reliability. Figure 2, below, shows the PAR scheme, where all data and acknowledgments are received as expected. Data is sent only after previously sent data has been acknowledged.

TCP implements PAR by assigning a sequence number to each octet that is transmitted and by requiring a positive acknowledgment (ACK) from the receiving TCP module. If the ACK is not received within a time-out interval, the data is retransmitted. At the receiver TCP module, the sequence numbers are used to correctly order segments that may have arrived out of order and to eliminate duplicates. Corruption of data is detected by using a checksum field filled in the TCP packet header. Data segments that are received with a bad checksum field are discarded.

## 2.5 Flow control (buffers and window)

Tcp's flow control mechanism permits a receiving tcp to govern the amount of data dispatched by a sending tcp. Tcp uses a window mechanism to control the flow f data. When a connection is established, each end of the connection allocates a buffer to the other end. As, data arrives the receiver sends acknowledgements, which also specify the remaining buffer size. The amount of buffer available at any time is called window and a notification that specifies the size is called a window advertisement. A receiver sends a window advertisement with each acknowledgement. As data are accepted, TCP slides theWindow upward in the sequence number space. The current window is specified in every segment and enables peer TCPs to maintain up-to date information.
If the receiving application can read data as quickly as it arrives, a receiver will Send a positive window advertisement along with each acknowledgement . However , if the sending side operates faster than the receiving side( the cpuIs faster),incoming data will eventually fill the receivers buffer, causing the receiver to advertise a zero window. A sender that receives a zero window advertisement must stop sending until the receiver again advertises a positive window.

## 2.6 Error-free data transfer

Sequence numbers and acknowledgments cover discarding duplicate packets, retransmission of lost packets, and ordered-data transfer. To assure correctness a checksum field is included (see TCP segment structure for details on checksumming).

The TCP checksum is a quite weak check by modern standards. Data Link Layers with high bit error rates may require additional link error correction/detection capabilities. If TCP were to be redesigned today, it would most probably have a 32-bit cyclic redundancy check specified as an error check instead of the current checksum. The weak checksum is partially compensated for by the common use of a CRC or better integrity check at layer 2, below both TCP and IP, such as is used in PPP or the Ethernet frame. However, this does not mean that the 16-bit TCP checksum is redundant: remarkably, introduction of errors in packets between CRC-protected hops is common, but the end-to-end 16-bit TCP checksum catches most of these simple errors . This is the end-to-end principle at work.Applicability of TCP

TCP is used extensively by many of the Internet's most popular application protocols and resulting applications, including the World Wide Web, E-mail, File Transfer Protocol, Secure Shell, and some streaming media applications.

However, because TCP is optimized for accurate delivery rather than timely delivery, TCP sometimes incurs relatively long delays (in the order of seconds) while waiting for out-of-order messages or retransmissions of lost messages, and it is not particularly suitable for real-time applications such as Voice over IP. For such applications, protocols like the Real-time Transport Protocol (RTP) running over the User Datagram Protocol (UDP) are usually recommended instead.

TCP is a reliable stream delivery service that guarantees to deliver a stream of data sent from one host to another without duplication or losing data. Since packet transfer is not reliable, a technique known as positive acknowledgment with retransmission is used to guarantee reliability of packet transfers. This fundamental technique requires the receiver to respond with an acknowledgment message as it receives the data. The sender keeps a record of each packet it sends, and waits for acknowledgment before sending the next packet. The sender also keeps a timer from when the packet was sent, and retransmits a packet if the timer expires. The timer is needed in case a packet becomes lost or corrupt.

TCP (Transmission Control Protocol) consists of a set of rules, the protocol, that are used with the Internet Protocol, the IP, to send data “in a form of message units” between computers over the Internet. At the same time that the IP takes care of handling the actual delivery of the data, the TCP takes care of keeping track of the individual units of data “packets” that a message is divided into for efficient routing through the net. For example, when an HTML file is sent to you from a Web server, the TCP program layer of that server takes the file as a stream of bytes and divides it into packets, numbers the packets, and then forwards them individually to the IP program layer. Even though every packet has the same destination IP address, they can get routed differently through the network. When the client program in your computer gets them, the TCP stack (implementation) reassembles the individual packets and ensures they are correctly ordered as it streams them to an application.

## 2.7 Adaptive Retransmission

Before tcp was invented, transport protocols used a fixed value for retransmission delay-the protocol designer or network manager chose a value that was large enough for the expected delay. Designers working on TCP realized that Fixed timeout would not operate well for an internet. Thus, they chose to make TCP’s retransmission adaptive. That is, TCP monitors current delay on each

Connection, and adapts the retransmission timer to accommodate changing conditions. How can TCP monitor internet delays? In fact, TCP cannot know the Exact delays for all parts of an internet at all times. Instead, TCP estimates round-trip delay for each active connection by measuring the time needed to receive a response. Whenever it sends a message to which it expects a response, TCP records the time at which the message was sent. When a response arrives, tcp subtracts the time at which the message was sent from the current time to Produce a new estimate of the round trip delay for that connection. As it sends Data packets and receives acknowledgements , TCP generates a sequence of round trip estimates and uses a statistical function to produce a weighted average. In addition to a weighted average, TCP keeps an estimate of the variance, and uses a linear combination of the estimated mean and variance as a Value for retransmission.

Experience has shown that TCP adaptive retransmission works well. Using the variance helps TCP react quickly when delay increases following a burst of Packets. Using a weighted average helps reset the retransmission timer if the delay returns to a lower value aftera temporary burst. When the delay remains constant , TCP adjusts the retransmission timeout to a value that is slightly longer than the mean round-trip delay. When delays start to vary, TCP adjusts the retransmission timeout to a value greater than the mean to accommodate peaks.

## 2.8 Congestion control

One of the most interesting aspects of TCP is a mechanism for congestion control. In most modern internets , packet loss(or extremely long delay) is more Likely to be caused by congestion than a hardware failure. Interestingly, transport protocols that retransmit can exacerbate the problem of congestion by injecting additional copies of a message. If congestion triggers excessive retransmission, the entire system can reach a state of congestion collapse, analogous to a traffic jam on a highway. To avoid the problem TCP always uses Packet loss as a measure of congestion, and responds by reducing the rate at Which it retransmits data.

Whenever a message is lost, TCP begins congestion control. Instead of retransmitting enough data to fill the receivers buffer(window size if receiver) Tcp begins by sending a single message containing data. If the acknowledgement arrives without additional loss, TCP doubles the amount of data being sent , and sends two additional messages. If acknowledgements arrive For those two, TCP sends four and more, and so on. The exponential increase continues until TCP is sending half of the recevier's advertised window, at which time TCP slows down the rate of increase.

TCP's congestion control scheme responds well to increased traffic in an internet. By backing off quickly, TCP is able to alleviate congestion. More important, because it avoids adding retransmissions to a congested internet, TCP's congestion control scheme helps prevent congestion collapse.

# 3 DEVELOPMENT OF TCP OVER YEARS

TCP is a complex and evolving protocol. However, while significant enhancements have been made and proposed over the years, its most basic operation has not changed significantly since its first specification RFC 675 in 1974, and the v4 specification RFC 793, published in September 1981 RFC 1122, Host Requirements for Internet Hosts, clarified a number of TCP protocol implementation requirements.RFC2581, TCP Congestion Control, one of the most important TCP related RFCs in recent years, describes updated algorithms to be used in order to avoid undue congestion. In 2001,RFC 3168 was written to describe explict congestion notification (ECN), a congestion avoidance signalling mechanism.

The original TCP congestion avoidance was known as "TCP Tahoe", but many alternative algorithms have since been proposed (including TCPReno, Vegas, FAST TCP;New Reno and Hybla .

Another scheme looked how to engineer various extensions into TCP.TCP Interactive(iTCP) allows applications to subscribe to TCP events and respond accordingly enabling various functional extensions to TCP from outside TCP layer including application assisted congestion control.

# 4 TCP over wireless

TCP has been optimized for wired networks. Any packet loss is considered to be the result of congestion and the congestion window size is reduced dramatically as a precaution. However, wireless links are known to experience sporadic and usually temporary losses due to fading, shadowing, hand off, and other radio effects, that cannot be considered congestion. After the (erroneous) back-off of

the congestion window size, due to wireless packet loss, there can be a congestion avoidance phase with a conservative decrease in window size. This causes the radio link to be underutilized. Extensive research has been done on the subject of how to combat these harmful effects. Suggested solutions can be categorized as end-to-end solutions (which require modifications at the client and/or server), link layer solutions (such as RLP in CDMA 2000), or proxy based solutions (which require some changes in the network without modifying end nodes

# 5 THE FUTURE OF TCP

Advances in the architecture of Transmission Control Protocol/Internet Protocol (TCP/IP) continue to benefit millions of end users, gigabit-per-second backbone links and up to gigabit-per-second communication between single pairs of end users. Different requirements represent each of these three aspects of growth, prompting the IBM and the Internet community to continuously upgrade TCP/IP to meet these requirements.

## References

Computer networks and internets with internet applications third edition
Douglas E.Comer

An introduction to TCP/IP by John Davidson

## Internet references

http://www.encyclopedia.com/doc/1G1-17568923.html
http://condor.depaul.edu/~jkristof/technotes/tcp.html
http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/tcp.html