International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 06, June-2015

A Comparative Study of Transmission Control Protocols

Nur-a Nusrat Nazmi, Md. Wafi Islam Omi
Department of Computer Science and Engineering
IUBAT— International University of Business Agriculture and Technology
Uttara, Dhaka-1230, Bangladesh

***Abstract*— Transmission Control Protocol (TCP) is important because it is the dominant protocol used in the Internet today. In this paper, three way- and four way- handshaking of TCP are described by following various mechanisms of TCP. Besides describing other mechanisms, here mainly the comparison of 3-way and 4-way handshaking mechanisms are considered and discussed. This paper attempts to list some of the TCP mechanisms and describe the three way and four way handshaking mechanisms.**

I. INTRODUCTION

TCP/IP first came on the scene in 1973. Later, in 1978, it was divided into two distinct protocols: TCP and IP. Then, in 1983, TCP/IP replaced the Network Control Protocol (NCP) and was authorized as the official means of data transport for anything connecting to ARPANET, the Internet's ancestor that was created by ARPA, the DOD's Advanced Research Projects Agency, and way back in 1957 in reaction to the Soviet's launching of Sputnik. ARPA was soon redubbed DARPA, and it was divided into ARPANET and MILNET (also in 1983); both were finally dissolved in 1990. But contrary, most of the development work on TCP/IP happened at UC Berkeley in Northern California, where a group of scientists were simultaneously working on the Berkeley version of UNIX, which soon became known as the BSD, or Berkeley Software Distribution, series of UNIX versions. Of course, because TCP/IP worked so well, it was packaged into subsequent releases of BSD UNIX and offered to other universities and institutions if they bought the distribution tape. So basically, BSD Unix bundled with TCP/IP began as shareware in the world of academia and, as a result, became the basis of the huge success and exponential growth of today's Internet as well as smaller, private and corporate intranets. As usual, what may have started as a small group of TCP/IP aficionados evolved, and as it did, the U.S. government created a program to test any new published standards and make sure they passed certain criteria. This was to protect TCP/IP's integrity and to ensure that no developer changed anything too dramatically or added any proprietary features. It's this very quality—this open-systems approach to the TCP/IP family of protocols—that pretty much sealed its popularity because it guarantees a solid connection between myriad hardware and software platforms with no strings attached.

The Transmission Control Protocol (TCP) standard is defined in the Request for Comment (RFC) standards document number 793 by the Internet Engineering Task Force (IETF). The original specification written in 1981 was based on earlier research and experimentation in the original ARPANET. The design of TCP was heavily influenced by what has come to be known as the "end-to-end argument".

As it applies to the Internet, the end-to-end argument says that by putting excessive intelligence in physical and link layers to handle error control, encryption or flow control you unnecessarily complicate the system. The end-to-end argument helped determine how two characteristics of TCP operate; performance and error handling. TCP performance is often dependent on a subset of algorithms and techniques such as flow control and congestion control. Flow control determines the rate at which data is transmitted between a sender and receiver. Congestion control defines the methods for implicitly interpreting signals from the network in order for a sender to adjust its rate of transmission.

The term congestion control is a bit of a misnomer. Congestion avoidance would be a better term since TCP cannot control congestion per sec. Ultimately intermediate devices, such as IP routers would only be able to control congestion. Congestion control is currently a large area of research and concern in the network community. A companion study on congestion control examines the current state of activity in that area. Timeouts and retransmissions handle error control in TCP. Although delay could be substantial, particularly if you were to implement real-time applications, the use of both techniques offers error detection and error correction thereby guaranteeing that data will eventually be sent successfully.

The nature of TCP and the underlying packet switched network provide formidable challenges for managers, designers and researchers of networks. Once regulated to low speed data communication applications, the Internet and in part TCP are being used to support very high speed communications of voice, video and data. It is unlikely that the Internet protocols will remain static as the applications change and expand. Understanding the current state of affairs will assist us in understanding protocol changes made to support future applications.

II. RELATED WORK

A simple handshaking protocol might only involve the receiver sending a message meaning "I received your last message and I am ready for you to send me another one." A more complex handshaking protocol might allow the sender to ask the receiver if he is ready to receive or for the receiver to reply with a negative acknowledgement meaning "I did not receive your last message correctly, please resend it" (e.g. if the data was corrupted en route).

There have been several proposals for reliable link-layer protocols. The two main classes of techniques employed by these protocols are: error correction (using techniques such as forward error correction (FEC)), and retransmission of lost packets in response to automatic repeat request (ARQ) messages.

The link-layer protocols for the digital cellular systems in the U.S. — both CDMA and TDMA— primarily use ARQ techniques. While the TDMA protocol guarantees reliable, in-order delivery of link-layer frames, the CDMA protocol only makes a limited attempt and leaves it to the (reliable) transport layer to recover from errors in the worst case. The AIRMAIL protocol employs a combination of FEC and ARQ techniques for loss recovery. The main advantage of employing a link-layer protocol for loss recovery is that it fits naturally into the layered structure of network protocols. The link-layer protocol operates independently of higher-layer protocols (which makes it applicable to a wide range of scenarios), and consequently, does not maintain any per-connection state. The main concern about link-layer protocols is the possibility of adverse effect on certain transport-layer protocols such as TCP. We investigate this in detail in our experiments. Indirect-TCP (I-TCP) protocol was one of the early protocols to use the split-connection approach.

It involves splitting each TCP connection between a sender and receiver into two separate connections at the base station — one TCP connection between the sender and the base station, and the other between the base station and the receiver. In our classification of protocols, ITCP is a split-connection solution that uses regular TCP for its connection over wireless link. I-TCP, like other split-connection proposals, attempts to separate loss recovery over the wireless link from that across the wire line network, thereby shielding the original TCP sender from the wireless link. However, as our experiments indicate the choice of TCP over the wireless link results in several performance problems. Since TCP is not well-tuned for the loss link, the TCP sender of the wireless connection often times out, causing the original sender to stall.

In addition, every packet incurs the overhead of going through TCP protocol processing twice at the base station (as compared to zero times for a non-split-connection approach), although extra copies are avoided by an efficient kernel implementation. Another disadvantage of this approach is that the end-to-end semantics of TCP acknowledgments is violated, since acknowledgments to packets can now reach the source even before the packets actually reach the mobile host. Also, since this protocol maintains a significant amount of state at the base station per TCP connection, handoff procedures tend to be complicated and slow.

The snoop protocol introduces a module, called the snoop agent, at the base station. The agent monitors every packet that passes through the TCP connection in both directions and maintains a cache of TCP segments sent across the link that have not yet been acknowledged by the receiver. A packet loss is detected by the arrival of a small number of duplicate acknowledgments from the receiver or by a local timeout. The snoop agent retransmits the lost packet if it has it cached and suppresses the duplicate acknowledgments. In our classification of the protocols, the snoop protocol is a link-layer protocol that takes advantage of the knowledge of the higher-layer transport protocol (TCP).Snoop protocol is designed to be TCP aware, and to mask unreliability of wireless layer. Snoop is implemented as a layer in TCP/IP architecture stack. It is located just below TCP layer. Snoop can be located at both the access point and the mobile nodes. It is not necessary to use it at mobile nodes, which makes it easier to implement, but transfer of data from mobile host to wired node will not benefit from snoop. Snoop at the access point is only able to improve TCP performance of connections from wired host to mobile hosts.

Explicit Feedback (EF) is a mechanism used by the access point to inform TCP sender (located in the wired network) that wireless channel is currently experiencing a lot of errors and that it should not invoke congestion avoidance procedure on lost segment timeouts. This requires modifications at both the access point and the TCP sender. The explicit feedback messages are sent to the sender after every failed transmission to a mobile node from the access point. In access point is assumed to send acknowledgments to senders on the wired network for every segment it receives. These acknowledgments indicate to the TCP sender the segment reached the access point and if it does not receive the acknowledgment for it, then the sender can assume that the loss occurred due to corruption over wireless medium, and congestion avoidance should not be initiated. The last hop acknowledgment scheme assumes that losses over wireless network happen only due to corruption and that wireless network is the last hop on the TCP segment path.

The acknowledgment from the access point is called last hop ACK (LHACK). In the case that TCP sender does not receive LHACK, then congestion in the wired network caused packet to be dropped and therefore TCP sender should start congestion avoidance procedure In [3], TCP segment inter-arrival times at TCP receiver are used to distinguish between congestion and wireless losses. It is assumed that TCP segments will queue at the access point in the case when TCP receiver is on a wireless node. Queuing occurs here because of small wireless bandwidth as compared to wired bandwidth. TCP receiver looks at inter-arrival time between every segment. If the inter-arrival time between two segments is a multiple of a segment transmission time over wireless network, but the two segments arrived out-of-order, then TCP receiver assumes that all segments between last in-order received segment and the segment just received are lost due to congestion in the wired network.

This scheme assumes that due to queuing at the access point, all segments will be sent back-to-back to the wireless node. It also assumes that there is no congestion in the wireless link and that only bulk transfers are used. In the case that segments are lost because of congestion, the queue at the access point will have gaps in sequence numbers, but inter-arrival times at the mobile node will be the same for all packets. From these gaps, TCP receiver can conclude that congestion is the cause of the losses. On the other hand, if losses occurred in wireless part then the inter-arrival times will not be a multiple of segment transmission times. From this, TCP receiver can conclude that the losses occurred because of wireless error and it does not initiate congestion avoidance.

Mobile-TCP is another solution that is designed mostly for problems of disconnections. Mobile-TCP informs TCP-sender (on wired network) that a disconnection occurred. If TCP sender detects a loss (duplicate acknowledgments or timeout) it will perform retransmissions but without reducing its send window. Once disconnection ends, TCP sender is informed to resume normal operation.

III. DISCRIPTION OF THREE WAY AND FOUR WAY HANDSHAKING

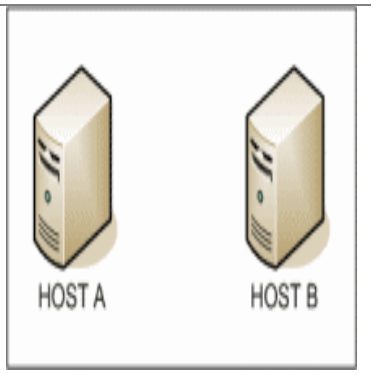
The Transmission Control Protocol (TCP) level of the TCP/IP transport protocol is connection-oriented. Connection-oriented means that, before any data can be transmitted, a reliable connection must be obtained and acknowledged. TCP level data transmissions, connection establishment, and connection termination maintain specific control parameters that the entire process.TCP connections begin with a "handshake". Like three way handshaking, four way handshaking etc. here is a description of three way handshaking and four way handshaking -

A. Three Way Handshaking (TWH):

The TCP three-way handshake is in Transmission Control Protocol. It also called the TCP-handshake, three message handshake and or SYN-SYN-ACK is the method used by TCP set up a TCP/IP connection over an Internet Protocol based network. TCP's three way handshaking technique is often referred to as "SYN-SYN-ACK" or more accurately SYN, SYN-ACK, ACK because there are three messages transmitted by TCP to negotiate and start a TCP session between two computers. The TCP handshaking mechanism is designed so that two computers attempting to communicate can negotiate the parameters of the network TCP socket connection before transmitting data such as SSH and HTTP web browser requests. This 3-way handshake process is also designed so that both ends can initiate and negotiate separate TCP socket connections at the same time. Being able to negotiate multiple TCP socket connections in both directions at the same time allows a single physical network interface, such as ethernet, to be multiplexed to transfer multiple streams of TCP data simultaneously.

TWH Diagram

Below there is a very simplified diagram of the TCP 3-way handshake process. There is a diagram of a three way handshaking-

EVENT	DIAGRAM
Host A sends a TCP Synchronize packet to Host B Host B receives A's SYN Host B sends a Synchronize Acknowledgement Host A receives B's SYN-ACK Host A sends Acknowledge Host B receives ACK. TCP socket connection is ESTABLISHED.	

TCP Three Way Handshake (SYN,SYN-ACK,ACK)

Synchronize and Acknowledge messages are indicated by a either the SYN bit, or the ACK bit inside the TCP header, and the SYN-ACK message has both the SYN and the ACK bits turned on (set to 1) in the TCP header. TCP knows whether the network TCP socket connection is opening, synchronizing, established by using the Synchronize and Acknowledge messages when establishing a network TCP socket connection. When the communication between two computers ends, another 3-way communication is performed to tear down the TCP socket connection. This setup and teardown of a TCP socket connection is part of what qualifies TCP a reliable protocol. TCP also

acknowledges that data is successfully received and guarantees the data is reassembled in the correct order.

FTP, Telnet, HTTP, HTTPS, SMTP, POP3, IMAP, SSH and any other protocol that rides over TCP also has a three way handshake performed as connection is opened. HTTP web requests, SMTP emails, FTP file transfers all manage the messages they each send. TCP handles the transmission of those messages.

TCP 'rides' on top of Internet Protocol (IP) in the protocol stack, which is why the combined pair of Internet protocols is called TCP/IP . TCP segments are passed inside the payload section of the IP packets.

IP handles IP addressing and routing and gets the packets from one place to another, but TCP manages the actual communication sockets between endpoints computers at either end of the network or internet connection.

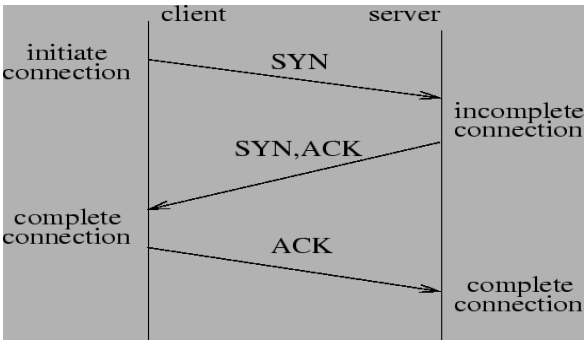


Fig.1

The purpose of the three-way handshake is to synchronize the sequence number and acknowledgment numbers of both sides of the connection and exchange TCP window sizes or the use of large window sizes or TCP timestamps. Here we summarize the process:

1. The initiator of the TCP connection, typically a client, sends a TCP segment to the server with an initial Sequence Number for the connection and a window size indicating the size of a buffer on the client to store incoming segments from the server.

2. The responder of the TCP connection, typically a server, sends back a TCP segment containing its chosen initial Sequence Number, an acknowledgment of the client's Sequence Number, and a window size indicating the size of a buffer on the server to store incoming segments from the client.

3. The initiator sends a TCP segment to the server containing an acknowledgment of the server's Sequence Number.

TCP uses a similar handshake process to end a connection. This guarantees that both hosts have finished transmitting and that all data was received.

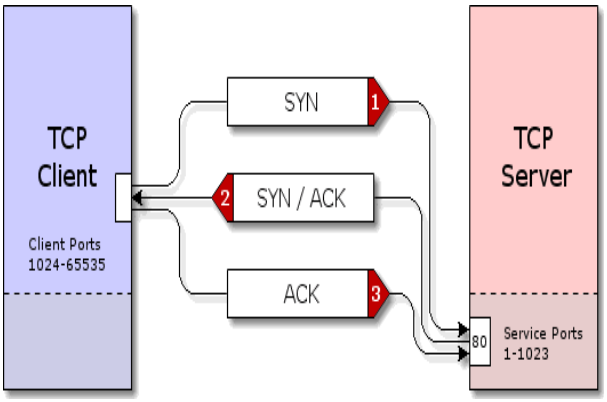


Fig. 2

B. Four Way Handshaking(FWH):

A four-way handshake is a type of network authentication protocol established by IEEE-802.11i that involves standards set up for the construction and use of wireless local area networks (WLANs). The four-way handshake provides a secure authentication strategy for data delivered through network architectures. The four-way handshake uses a pass key called Pair wise Master Key (PMK), and concatenation of various data items to set up the encryption of data. These include single-use items called ANonce and Snonce, as well as the Mac addresses of the two endpoints involved. The main processes of the four-way handshake are done to enable an access point to authenticate itself to the client, and to provide secure encryption. The PMK is generally not sent over the network, leaving this component unshared and thus strengthening the security of the process.

While there is some debate about the specific points of four-way handshake authentication, it is used to send messages between an access point and a client in a secure way. This complex setup allows for a more secure authentication process that matches the complexity and vulnerabilities of modern networks.

Here is a figure of Four Way Handshake:

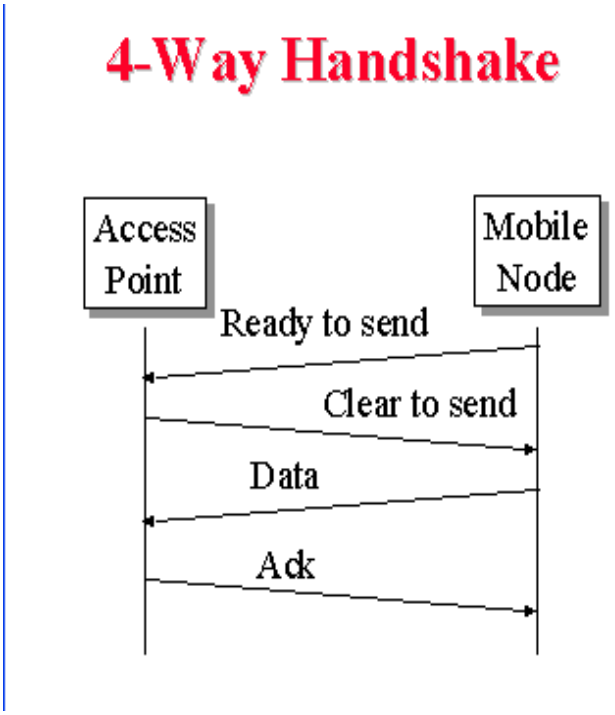


Fig. 3

The 802.11i key derivation procedure is based on a 4-way handshake. The primary activities in a 4-way handshake are to verify the existence of the same Pair wise Master Key (PMK) between the client and the AP and to derive the Pair wise Transient Key (PTK). The 4-way handshake consists of four messages, from Message-1 to Message-4. The 4-way handshake is not the only way to implement this process. For example Altunbasak and Owen suggested performance improvements by reducing the number of messages and the time delay.

Comparison of TWH and FWH:
We use Three Way Handshaking for not dropping call. But if we use Four Way Handshaking there, maybe there will be mismanage of network. So in the mobile area network we mostly use Three Way Handshaking.

Same case in many network we cannot use Three Way Handshaking mechanism; we use there Four Way Handshaking mechanism.

IV. COMPARATIVE STUDY

TCP implements an "urgent mechanism" that allows the sending user to stimulate the receiving user to accept some "urgent data" and that permits the receiving TCP to indicate to the receiving user when all the currently known "urgent data" have been read.

The TCP urgent mechanism permits a point in the data stream to be designated as the end of urgent information. Whenever this point is in advance of the receive sequence number at the receiving TCP, that TCP must tell the user to go into "urgent mode"; when the receive sequence number catches up to the urgent pointer, the TCP must tell user to go into "normal mode". This means, for example, that data was received as "normal data" might become "Urgent data" if an urgent indication is received in some successive TCP segment before that data is consumed by the TCP user. The URG control flag indicates that the "Urgent Pointer" field is meaningful and must be added to the segment sequence number to yield the urgent pointer. The absence of this flag indicates that there is no "urgent data" outstanding. The TCP urgent mechanism is NOT a mechanism for sending "out-of-band" data: the so-called "urgent data" should be delivered "in-line" to the TCP user. Here we are going to describe about three way handshake and four way handshake. The four-way handshake provides a secure authentication strategy for data delivered through network architectures.

We can tell that both 3-way and 4- way mechanisms are not perfect for all systems. Three Way Handshake is mainly using in mobile network system where Four Way Handshake is not.

The connection establishment in TCP is called Three Way Handshaking. In our example, an application program, called the client, wants to make a connection with another application program, called the server, using TCP as the transport layer protocol. The process starts with the server. The server program tells its TCP that it is ready to accept a connection. This is called a request for a passive open. Although the server TCP is ready to accept any connection from any machine in the world, it cannot make the connection itself. The client program issues a request for an active open. A client that wishes to connect to an open server tells its TCP that it needs to be connected to that particular server. TCP can now start the three-way handshaking process as shown in Figure 23.18. To show the process, we use two time lines: one at each site. Each segment has values for all its header fields and perhaps for some of its option fields, too.

Three Way Handshaking solves the duplicate SYN problem, in which an obsolete SYN arrives after the close of a connection. The particular TCP handshaking procedure was made in order that a pair of pcs wanting to talk could work out the guidelines from the system joining before the begin interaction. This method is usually designed making sure that both comes to an end can certainly enlightened along with negotiate individual associations at the same time.

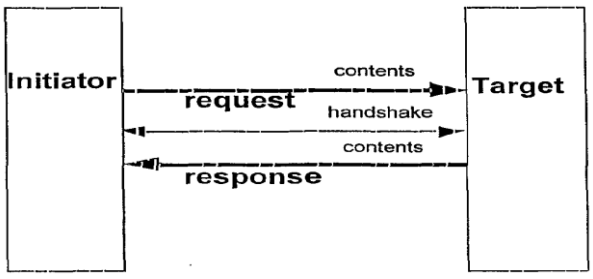


Fig. 4

A four-way handshake is a network type which is network authentication protocol that involves standards set up for the construction and use of wireless local area networks. There are two new discussion points for handling a four-way handshake. First, when a connection in LISTEN state receives a SYN packet, it has to decide based on the contents of that packet whether or not the remote side understands the four-way handshake. This is accomplished through the allocation of one of the unused bits in the TCP header, the Four Way bit. The client sets the Four Way bit in the initial SYN. If the server receives a Four Way bit in the initial SYN, then it will set the Four Way bit in the SYN/ACK. If the client receives a SYN/ACK without the Four Way bit set, it proceeds with the normal three-way handshake. If it receives a SYN/ACK with the Four Way bit set, then based on the options in the SYN/ACK it can chose to either proceed with the normal three-way handshake, or to continue with the four-way handshake. If a packet is received with the Four Way bit set, but not the SYN bit, the Four Way bit is ignored. When sending a packet without the SYN bit set, the Four Way bit must not be set. TCP interoperability issues with the CWR and ECE bits, but the Four Way bit does not have the same issues.

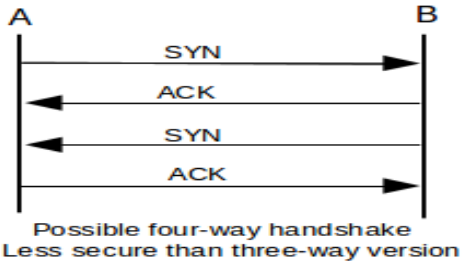


Fig. 5

Here is a table comparing Three Way Handshaking(TWH), Four Way handshaking(FWH) and Snoop Protocol-

Table 1

Keywords	TWH	FWH	Snoop protocol
1)Mobile network	It works	It doesn't	I doesn't
2)Wireless Network	It doesn't	It works	It does
3)bits	3bits	4bits	No bits
4)mobile nodes	no	no	Yes, it works

V. CONCLUSION

The Transmission Control Protocol (TCP) is one of the core protocols of the Internet protocol suite (IP), and is so common that the entire suite is often called TCP/IP. TCP provides reliable, ordered and error-checked delivery of a stream of octets between programs running on computers connected to a local area network, intranet or the public Internet. It resides at the layer. The transport layer behavior regarding TCP protocols is mainly considered here. The Transmission Control Protocol is one of the core protocols of the Internet protocol suite (IP), and is so common that the entire suite is often called TCP/IP. Here we discussed about different mechanism of Transport Control Protocol like snoop protocol, urgent data, Three Way handshake, Four Way Handshake etc. Handshake is usually a process that takes place when a computer is about to communicate with a foreign device to establish rules for communication. When a computer communicates with another device like a modem, printer, or network server, it needs to handshake with it to establish a connection. The 3-way handshaking and 4-way handshaking mechanism of TCP is mostly popular and briefly discussed here. Both characteristics are compared here.

VI. REFERENCES

[1] Douglas E. Comer: "Internetworking with TCP/IP", Vol. 1, Prentice Hall Inc., 1991
[2] T. Socolofsky, C. Kale: "A TCP/IP Tutorial", RFC 1180, Spider Systems Limited, January 1991
[3] A. Milanović, S. Srbljić, and V. Sruk thesis paper Performance of TCP Communication on Personal Computer
[4] TCP Mechanisms- http://search.ieice.org/bin/summary.php?id=e85-b_4_796
[5] 3- Way Handshaking. URL- http://www.inetdaemon.com/tutorials/internet/tcp/3-way_handshake.shtml
[6] 4- Way Handshake. URL- http://www.techopedia.com/definition/27188/four-way-handshake