Renovating TCP Segment by Reducing the Length of Sequence number & Acknowledgement Number

Joya Das[1] **Md. Rezaul Karim**[2] **Prosun Mozumder**[3]

***Abstract*—*This paper deals with throughput by reducing the sequence number bits and acknowledgement bits of Transmission Control Protocol ((TCP). TCP segment uses 32 bit for sequence numbering. But it is actually unnecessary to allocate such a large number of bits for sequencing which has a direct effect on data transfer speed. In this paper it is observed that the same activity is done by allocating only 16 bits for sequence numbering. For sequence numbering, a random number is generated between 100 and 1000 that can increase the number up to 65534 for a queue. After this sequencing technique the packet will be transferred and the receiver will be informed to lock the packet slot. If there is a need for extra sequence number, a new random number is selected and a new queue will be designed. When all the packet transfer has been finished, those will be sorted in First-In-First-Out (FIFO) basis. As a consequence 16 bits from the acknowledgement slot is also reduced.*

Keywords: *Segment, Initial Sequence Number (ISN), TCP, FIFO*

1. Introduction

TCP uses a sequence number to identify each byte of data. The sequence number identifies the order of bytes sent from each computer so that the data can be reconstructed in order, regardless of any fragmentation, disordering, or packet loss that may occur during transmission. For every payload byte transmitted, the sequence number must be incremented. In the first two steps of the 3-way handshake, both computers exchange an initial sequence number (ISN). During a connection via TCP/IP to a host, the host produces an Initial TCP Sequence Number, known as ISN. This sequence number is then used in the conversation occurring between itself and the host to assist in keeping track of each data packet. This sequence number is also helpful in ensuring the conversation continues in an adequate and appropriate fashion. Both the host and the client produce and use these sequence numbers in TCP connections [2].

Even as early as 1985, security experts said that by being able to come up with the next ISN, crackers could fake a one-way connection to a server by spoofing the source IP address of a trusted system. Therefore, to assist in the integrity of TCP/IP connections, security experts affirm that every stream, or communication using TCP/IP, should be given a unique, random sequence number [3].

1. Dept. of Computer Science & Engineering University of Information Technology & Sciences, email: joya.cse.sust@gmail.com
2. Dept. of Computer Science & Engineering University of Information Technology & Sciences, email: masum.luf@gmail.com
3. Dept of CD-Radio Access Network-Site Solution LM Ericsson Bangladesh Limited, email: jituprasun@yahoo.com

TCP primarily uses a cumulative acknowledgment scheme, where the receiver sends an acknowledgment signifying that the receiver has received all data preceding the acknowledged sequence number. The sender sets the sequence number field to the sequence number of the first payload byte in the segment's data field, and the receiver sends an acknowledgment specifying the sequence number of the next byte they expect to receive. For example, if a sending computer sends a packet containing four payload bytes with a sequence number field of 100, then the sequence numbers of the four payload bytes are 100, 101, 102 and 103. When this packet arrives at the receiving computer, it would send back an acknowledgment number of 104 since that is the sequence number of the next byte it expects to receive in the next packet. In addition to cumulative acknowledgments, TCP receivers can also send selective acknowledgments to provide further information. If the sender infers that data has been lost in the network, it retransmits the data [1].

2. Architecture of TCP Segment

The unit of data transfer between two devices using TCP is a segment [5][6].The format of a segment is projected below:

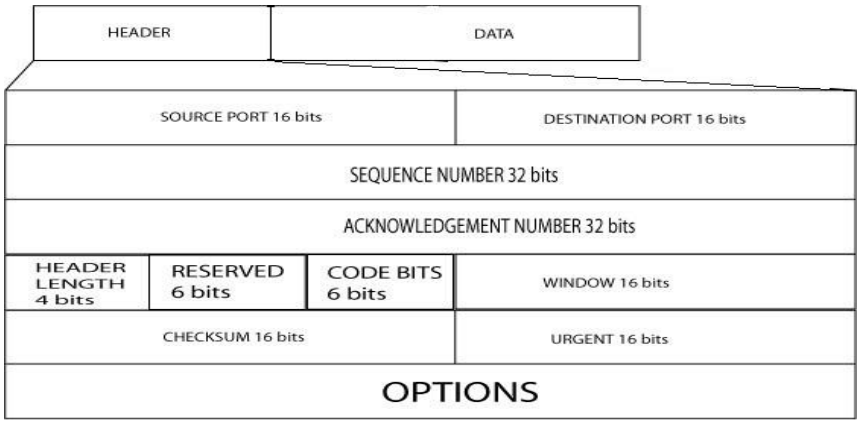


Fig 01: Existing TCP segment

Source port (16 bits) – identifies the sending port
Destination port (16 bits) – identifies the receiving port
Sequence number (32 bits) – has a dual role:
If the SYN flag is set (1), then this is the initial sequence number. The sequence number of the actual first data byte and the acknowledged number in the corresponding ACK are then this sequence number plus 1.

If the SYN flag is clear (0), then this is the accumulated sequence number of the first data byte of this segment for the current session.

Acknowledgment number (32 bits) – if the ACK flag is set then the value of this field is the next sequence number that the receiver is expecting. This acknowledges receipt of all prior bytes (if any). The first ACK sent by each end

acknowledges the other end's initial sequence number itself, but no data.

Data offset (4 bits) – specifies the size of the TCP header in 32-bit words. The minimum size header is 5 words and the maximum is 15 words thus giving the minimum size of 20 bytes and maximum of 60 bytes, allowing for up to 40 bytes of options in the header. This field gets its name from the fact that it is also the offset from the start of the TCP segment to the actual data.

Reserved (6 bits) – for future use and should be set to zero.

Control Bits (6 bits):

Urgent Pointer (URG). If this bit field is set, the receiving TCP should interpret the urgent pointer field (see below).

Acknowledgement (ACK). If this bit field is set, the acknowledgement field described earlier is valid.

Push Function (PSH). If this bit field is set, the receiver should deliver this segment to the receiving application as soon as possible. An example of its use may be to send a Control-BREAK request to an application, which can jump ahead of queued data.

Reset the Connection (RST). If this bit is present, it signals the receiver that the sender is aborting the connection and all queued data and allocated buffers for the connection can be freely relinquished.

Synchronize (SYN). When present, this bit field signifies that sender is attempting to "synchronize" sequence numbers. This bit is used during the initial stages of connection establishment between a sender and receiver.

No More Data from Sender (FIN). If set, this bit field tells the receiver that the sender has reached the end of its byte stream for the current TCP connection.

Window size (16 bits) – the size of the *receive window*, which specifies the number of window size units (by default, bytes) (beyond the sequence number in the acknowledgment field) that the sender of this segment is currently willing to receive .

Checksum (16 bits) – The 16-bit checksum field is used for error-checking of the header and data

Urgent pointer (16 bits) – if the URG flag is set, then this 16-bit field is an offset from the sequence number indicating the last urgent data byte

Options (Variable 0–320 bits, divisible by 32) – In order to provide additional functionality, several optional parameters may be used between a TCP sender and receiver. Depending on the option(s) used, the length of this field will vary in size, but it cannot be larger than 40 bytes due to the size of the header length field (4 bits). The most common option is the maximum segment size (MSS) option. A TCP receiver tells the TCP sender the maximum segment size it is willing to accept through the use of this option. Other options are often used for various flow control and congestion control techniques.

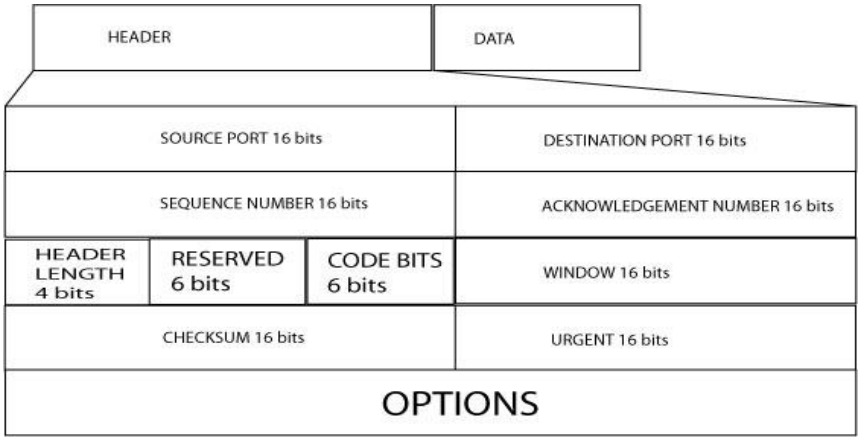


Fig 02: Proposed TCP Segment format

Here it can be stated that , traditionally the segment cosists of 20 to 60 byte header followed by data from the application program. In our proposed architecture we assign 16 bits for Sequence number instead of 32 bits.Acknowledgement field is also 16 bits in lieu of 32 bits.As a result the segment consists of 16 to 56 byte header followed by data from application program.The proposed TCP segment format is given here[7].

3. The Proposed Methodology for enhancing TCP/IP Throughput

This proposed technique will not only reduce the packet size by 32 bits but also reduce the cost of TCP/IP packet segments. To digest this theorem we have to keep a keen eye on the following procedures.

At first a random number will be selected as a starting sequence number which will be looped between 100 to 1000 ranges. After selecting the initial sequence number we will increase each packet's sequence number by 1. A queue will be created after numbering which can allocate packet for delivery .The queue delivery will be done sequentially that is after first queue second queue will be delivered then the third & forth queue(as shown in figure 03). The acknowledgement process will remain as previous process. If a queue is successfully delivered to its destination without any corruption or data loss, receiver will send a successful message and will request for next queue. When all queues will be sent to the receiver end, the queue counter will be set to 0.

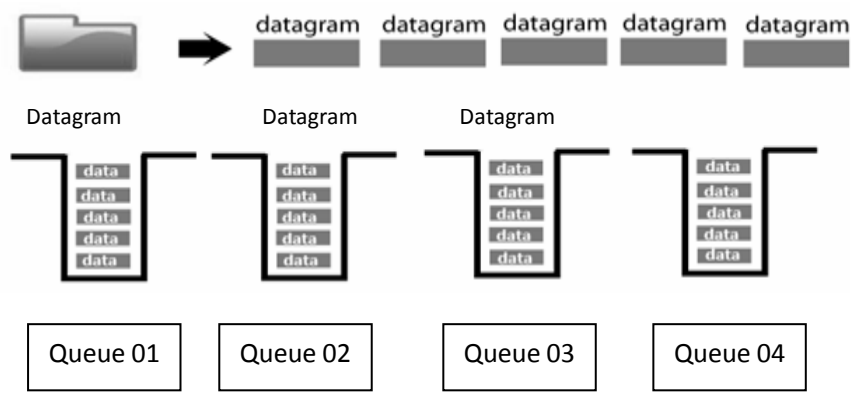
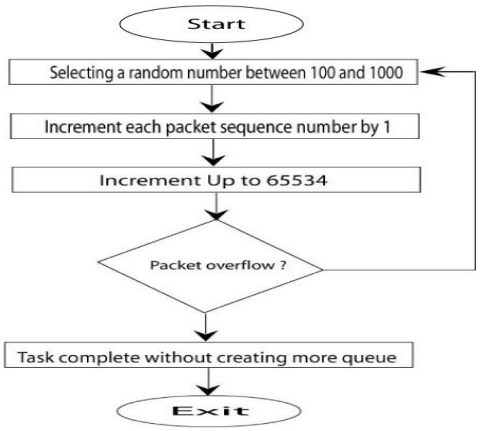


Fig 03: Extracting Packet Process

Flow chart for working procedure of proposed tcp segmen



4. Mathematically proof

From equation $2^{*}n-2$

n=16

[1]

***Earlier method*:**

Suppose, we have 1020 packets after extracting the original file .If we send this file (comprised with 1020 packets) using the earlier method, each packet needs 32bits for sequence number and 32bits for acknowledgment number. As a result (1020x32)+ (1020x32) bits i.e.65280 bits for sending.

1020 packets x 32bits (sequence number) =32640 bits (for sequence field)
1020 packets x 32bits (acknowledgement) =32640 bits (for acknowledgement field)

65280 bits for sending

In proposed method:

However in proposed method,16bits (instead of 32bits)are used for sequence number and 16bits(instead of 32bits)are used for acknowledgment number are used . For this reason (1020 x 16)+ (1020 x 16) bits i.e.32640 bits are needed for sending.

1020 packets x 16bits (sequence number) =16320 bits (for sequence field)
1020 packets x 16bits (acknowledgement) =16320 bits (for acknowledgement field)

32640 bits for sending

5. Mechanism For Handling Overflow In Queue At Reciver Section

Though the queue has a limitation which only allows 65534 number ranges not more, so if queue is overflow, we will select a new random number for new queue initializing as earlier process [8][9].

Suppose we required two queues for transmitting packet to receiver, After delivering successfully to receiver side we will join second queue with 1st queue from the 1st packet of 2nd queue[10] (As shown in Fig 04)

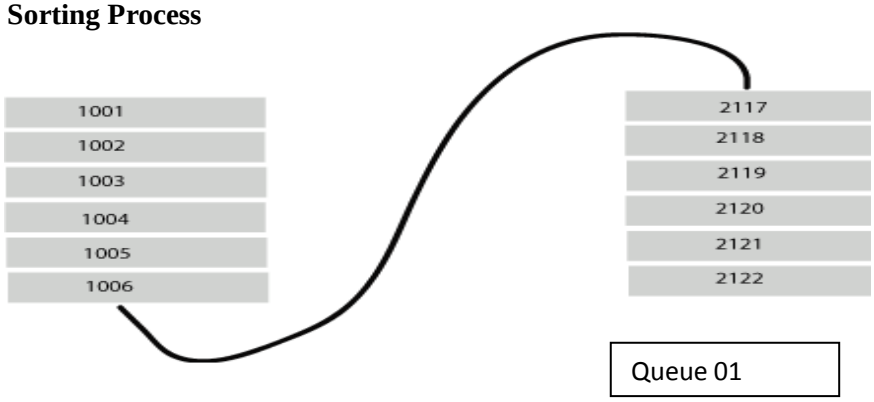


Fig 04: Packet sorting at the receiver end.

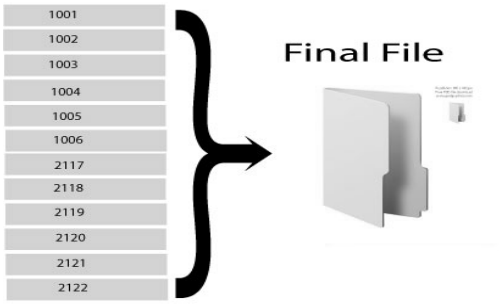


Fig 05: Final View of TCP/IP Packet Format

6. **Throughput achieved by applying proposed method**

If new TCP sending process is applied it will be acquired :

File size	Transfer speed	throughput
1GBytes	256kbps	65576 bytes

Whereas the earlier TCP method includes extra 65576 more byte for transmitting data.

So the throughput time is 34 Minutes 58.43 Seconds regard 65576 kilobytes over 256kbps transfer speed. Then a file is downloaded which size is 1GBytes over 256kbps speed, 34 Minutes 58.43 seconds earlier than previous TCP downloading process.

7. Summary

In conclusion, it can be stated that by reducing the length of sequence bit and acknowledgement bit, the TCP performance over network is enhanced. By applying this technique, the message can be transmitted to the receiver section with the lower cost at the same time maintaining the higher speed

which is the burning issue of the technological world.

Reference

1] Tanenbaum, Andrew S. (2003-03-17), "*Computer Networks"* (Fourth ed.)" Prentice Hall ISBN 0066102-3, http://www.amazon.com/Computer-Networks-Edition-Andrew-Tanenbaum/dp/0130661023)

[2] The TCP and IP packet format , http://linuxgazette.net/issue86/vinayak.html

[3] RFC 6528, Defending against Sequence Number Attacks, February 2012 Standard Track Steven M. Bellovin(http://tools.ietf.org/html/rfc6528)

[4] Donald Knuth. "*The Art of Programming*, *Computer* Volume 2: *Seminumerical Algorithms"*, Third Edition. Addison-Wesley, 1997. ISBN 0-201-89684-2,Chapter 3, pp. 1–193. Extensive coverage of statistical tests for non-randomness.(http://en.wikipedia.org/wiki/Pseudo-random_number_generator)

[5] Peterson, Larry (2003), "*Computer Networks"*,Morgan Kaufmann. pp. 401", ISBN 1-55860-832-X, http://www.amazon.com/Computer-Networks-Third-Edition-Networking/dp/155860832X

[6] W. Richard Stevens and Gary R Wright,*TCP/IP Illustrated, "Volume 2: The Implementation*" ISBN 0-201-63354-X http://www.amazon.com/TCP-IP-Illustrated-Vol-Implementation/dp/020163354X

[7] Kurose, James F. & Ross, Keith W. (2007), "Computer Networking" A Top-Down Approach" ISBN 0-321-49770-8 http://www.amazon.com/Computer-Networking-Top-Down-Approach-Edition/dp/0136079679

[8] The Transport Layer Security (TLS) Protocol, version 1.2", IETF.(http://tools.ietf.org/html/rfc5246)

[9] A. L. Barab´asi, R, Albert; Barabási, Albert-László (2002), "Statistical mechanics of complex networks"

http:/ / rmp. aps. org/ abstract/RMP/ v74/ i1/ p47_1. *Rev. Mod. Phys* **74**: 47–94. doi:10.1103/RevModPhys.74.47.

[10] The Internet Society (http:/ /www. isoc.org/)
[11] http://www.tamos.net/~rhay/overhead/ip-packet-overhead.htm
[12] Lessons from the History of the Internet, http:/ / www. oup. com/ us/ catalog/ general/ subject/ Sociology/Environment Technology/ ?view=usa& ci=9780199255771,

Manuel Castells, in *The Internet Galaxy*, Ch. 1, pp 9–35, Oxford University Press, 2001, ISBN 978-0-19-925577-1 ISBN10: 0-19-925577-6

[13] http://condor.depaul.edu/jkristof/technotes/tcp.html