ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 2, August 2012

Implementation of TCP_Reno Algorithm in the ns2

Dr. Neeraj Bhargava, Dr. Ritu Bhargava, Bharat Kumar,Shilpi Gupta, Naresh Kumar Senwaliya, Kamal Kumar Jyotiyana

Abstract – NS2 is the software simulation network tool which is used to describe the flow of data and algorithm in practical way. Many of the simulation tools are available and ns2 is one of them. In ns2 we have implemented the TCP-Reno algorithm which is efficient and easier algorithm than other algorithms. TCP_Reno is the congestion avoidance and fast transmission algorithm. In this paper we have described the SACK (selective acknowledgement) algorithm & how it work with congestion avoidance algorithm. This is the practical approach to determine the message transmission with TCP_Reno algorithm and SACK acknowledgement. The next section deals with the TCP_Reno & SACK algorithm.

Index Terms **-- SACK, TCP_Reno, NS2, XGraph, Full-Duplex.**

I. INTRODUCTION

TCP/IP is the communication protocol for Internet. TCP handles packet flow between many systems and IP handles the routing of packets. TCP will set up a "full-duplex" communication between the two applications [5]. The "full-duplex" communication will occupy the communication line between the two computers until it is closed by one of the two applications. TCP/IP handles the responsibilities of layers such as Session, Presentation, and Application in the OSI model. With IP, messages are broken up into small independent "packets" and sent between computers via Internet. IP is responsible for "routing" each packet to the correct destination [6]. The TCP standard is defined in the Request for Comment (RFC) standards document number 793 by the Internet Engineering Task Force (IETF) [4]. TCP is a transport layer protocol used by applications that require guaranteed delivery. It is a sliding window (determines the number of bytes of data that can be sent before an acknowledgement) protocol that provides handling for both timeouts and retransmissions. TCP is used for many internet applications such as www, remote administration and for file transfer. TCP divide the message into segments, small chunks and packets. Each packet carries the separate part of message. TCP/IP provides end to end connectivity between nodes and specifies how the data should be transmitted, formatted, routed and received at the receiving end. TCP IP has four layers and each layer work individually and hides the method from another layer [12].

I. Link Layer (contains Ethernet) used in the local communication link for local network.
II. Internet Layer (contains IP) performs in a local area network or in a wide area network.
III. Transport Layer (contains TCP) used in communication between two or more nodes.
IV. Application Layer (contains HTTP) used to transmits applications, messages and more. HTTP also used in communication services such as client to server for providing web services and also much more.

TCP IP has many versions which are designed for different hardware platform [12]. Four versions were developed by DARPA for TCP/IP. They are TCPv1, TCPv2, TCPv3, IP v3, TCP/IP v4 and now a day's the protocol that is still running is TCP/IPv6 [17]. TCP/IP assigns a unique number to every workstation. This "IP NO." is a four byte value 192.168.5.26 [16]. Using TCP/IP protocol we can connect different types of network. TCP/IP has a network support capabilities and provide communication across diverse interconnected networks. TCP/IP makes easier transmission between networks. It can carry large amount of data from one node to another and also makes easier communication between them. In the network area each node has a unique id. To reduce the network traffic many more restrictions and techniques are used. Through TCP/IP protocol networks divide into sub networks, sub network reduce the traffic of networks [16]. TCP/IP built the communication path (pipe), in which data is travelled. TCP gives reliability, trustworthiness, ensures about the data, message. TCP/IP established the link between two nodes for conversion. The establishment of TCP/IP is same as telephone conversion. If a node request to another node for any service, then the protocol check that requested node is ready for conversion. If the node is ready then the response will be given to the requestor (Client), but if the node is not in working state then the requested client gives a message to indicate that the node is not in running mode [12]. TCP/IP uses logical and physical address to transmit any message [13]. A Physical address is a 48-bit flat address burned into ROM of the NIC card which is a Layer1 device of the OSI model (MAC Address). This IIS is divided into 24-bit vendor code and 24-bit serial address. This is unique for each system and cannot be changed. A Logical address is a 32- bit address assigned to each system in a network. This works in Layer-3 of OSI Model. This would be generally the IP address. TCP/IP called just a "protocol" instead of a "protocol suite" because it behaves like single protocol. Without IP we cannot use TCP or Without TCP we cannot use IP. It is combined set of protocol which is used for networking. TCP/IP protocol can be used with any protocol suite such as HTTP, FTP, PPP, Telnet etc. It will create an I-way path. In this any protocol can be use to transmit data.

A. SELECTIVE ACKNOWLEDGEMENT OF TCP IP:

Multiple packet loss from a window of data can have extremely harmful effect on TCP throughput.TCP uses a

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 2, August 2012

cumulative acknowledgment scheme in which received segments that are not at the left edge of the receive window are not acknowledged. This forces the sender to either wait for a roundtrip time to find out about each lost packet, or to unnecessarily retransmit segments which have been correctly received.

Left Edge of Block: This is the first sequence number of the block.

Right Edge of Block: This is the sequence number immediately following the last sequence.

Block: Number of this block.

Selective Acknowledgement (SACK) [1] is a strategy which corrects this behavior in the face of multiple dropped segments. With selective acknowledgments, the data receiver can inform the sender about all segments that have arrived successfully, so the sender need to retransmit only those segments that have actually been lost. The selective acknowledgment extension uses two TCP options [1].

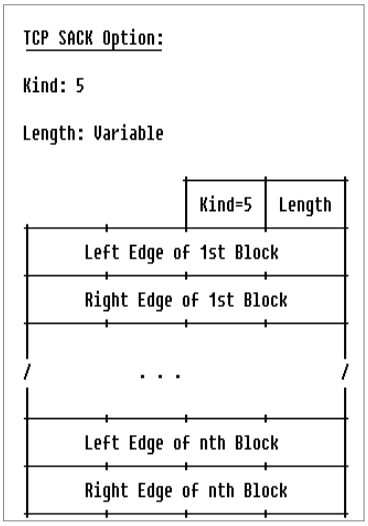

Fig 1. The TCP SACK Option.

Figure 1 the first is an enabling option, "SACK-permitted" (Kind-4) [1], which may be sent in a SYN segment to indicate that the SACK option can be used once the connection is established. The other is the SACK option (kind-5) itself, which may be sent over an established connection once permission has been given by SACK-permitted. The 2-byte TCP *Sack-Permitted* option may be sent in a SYN by a TCP that has been extended to receive the SACK option once the connection has opened. TCP Option indicates kind of options 3(window scale), 4(SACK permitted), 5(SACK), 6(request), 7(response). Window scale indicates the receiving capacity of the end user. SACK-permitted allows user for any permission which is related to SACK. SACK-permitted gives many more options to the user, user can choose this option for reliable message delivery. SACK gives the permission of resending acknowledgement of the segment. Request and response is the common used term in messaging for relevant communication.

II. TCP RENO

TCP uses various network congestion avoidance algorithms. In network, **network congestion** occurs when a link or node is carrying so much data that its quality of service neglected. TCP Reno is the fast recovery and fast retransmission algorithm. It is very successive and easy data transmission algorithm. The new algorithm prevents the communication path ("pipe") from going empty after Fast Retransmit, thereby avoiding the need to Slow-Start to re-fill it after a single packet loss. If duplicate ACK received at the receiver hand it represents a single packet having left the pipe. TCP Reno induces packet loss to estimate the available bandwidth in the network. When there are no packet loss, TCP Reno continues to increase its window size by one, during each round trip time. When it experiences a packet loss, it reduces its window size to one half of the current window size. This is called additive increase and multiplicative decrease. TCP Reno reduces the packet loss to estimate the network traffic [15]. TCP Reno continues to increase the window size during each round trip. If the packet is lost then it reduces the window size to one half of the current window size. This process is called additive increase and multiplicative decrease.TCP Reno Periodically update the size of window. In case of, if a Packet is lost then the lost packet is retransmitted in the available bandwidth. The receiving window size is dependent on the round trip delay of the connection. If the connection speed is fast then the window size increases shortly [15]. It is dependent on the connection speed. TCP Reno fails to achieve the fairness because TCP is not a synchronized based scheme. It can only avoid the congestion in their exist limit. When a message is send it will start with slow speed and after that it will automatically cover the transmission speed. If the message is slowly started then the flow of line is in the form of curly shape. It will denote that the speed of transmission is slow. But if the line is in diagonal form then it is showing the congestion avoidance technique.

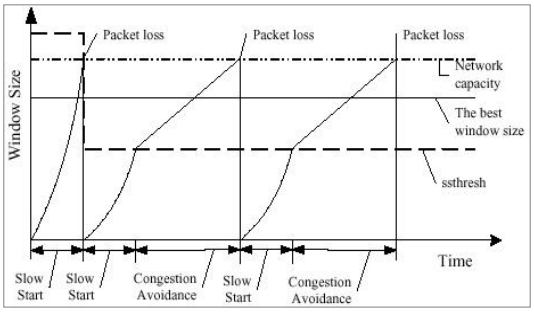

Fig 2. Window Size during the Transmission of Packets

Figure 2 shows the window size and transmission of package. Vertical line shows the size of window and horizontal line shows the time of transmission. There many sections in the diagram, each section controlling different parts of data transmission. Network Capacity shows the capacity of network, how much data can be carried by the network at a time. SSthresh (Slow-Start Threshold) to

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 2, August 2012

specify the no. of sent packets [15][18]. This algorithm is used to solve the lost packet problem, when the speed of packet transmission is larger than the router speed. This algorithm is used to avoid the slow start of packet or to avoid the packet lost problem they are: SSthresh (Slow-Start Threshold) and CWND (Congestion Window) [18]. The result of this algorithm is dependent on these two methods. The best window size shows the capacity of window to handle the amount of data. In this diagram diagonal line crossing the limit of window size means that only those data packets which is in limit of the window will remain in window and overflow packets will be misplaced by this window.

III. Problem Phase

This phase consider the problems that are occurred in simulation network during the data sending and receiving time. This phase includes all the problems related to a packet or to complete message, it considers some problems like PER (packet error rate), BER (bit error rate), data rate, delay time. TCP IP checks the network errors and find out all possible errors for removing them. If PER and BER are neglect-able then, it will not be considered, but if high error rate is found then it will order to retransmit the packets. Data rate consider the rate of data transmission and delay time consider the gap between packet transmission times. It is also responsible for buffer size of memory. When sender transmit large amount of data to the receiver and receiver has no capacity to handle this data then the efficiency of transmission reduced automatically. This kind of problem is called "silly window syndrome". Another example is that, if sender transmits 0 amount of data to the receiver it may create some exceptions. To reduce this exception problem a blank window will be shown to the receiver.

IV. Methodology

A. *TCP Dump Module* :

TCP Dump Module are used for capturing the message that are sends and received by the nodes. Captured logs can be viewed with packet analyzing tools like WIRESHARK (open source packet analyzer tool). TCP Dump Module traces the message transmission and delivery of the message. This technique is very helpful in many cases, when any fault occurs in the network system.

B. *Data Lost And Recovery*:

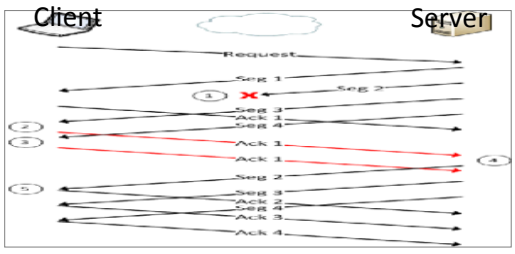

Fig 3. Transmission between Two Nodes (Client and Server).

First of all client request to the server, then server sends a segment one to the client, and immediately he sends another segment to the client but it cannot reaches its destination. This fault can be handled by the SACK [2]. Other Packet of the message has been transmitted by the server to client and client gives the response to the server as an acknowledgement; each packet is transmitted to the client and each packet acknowledgement are given by the client to server to ensure that the packet is successfully transmitted. This process deals with another system until the message is successfully delivered. Data lost and recovery is a very useful mechanism to recover the undelivered data report. It is a two way process to hold and transmit data. This process gives the guarantee that the message is delivered. When data transmission is started between two nodes then the data

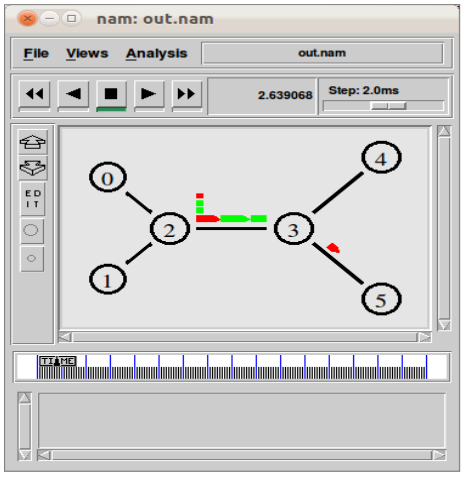

FIG 4. NODE OF NETWORK

is delivered in segment form. The id of the segment is generated by the system and this is used for re-arranging the message on the receiving system. Sender receives the acknowledgement of each packet. This acknowledgement ensures that the receiver has received all the packets.

V. Results

Here, we consider the six component of network they are client (0)(1), router(2)(3), server(4)(5).

```
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1Mb 20ms DropTail
$ns duplex-link $n3 $n4 2Mb 20ms DropTail
$ns duplex-link $n3 $n5 2Mb 20ms DropTail
```

These are the common parameter which is used to handle the flow of data in the network. Duplex link between n0 to n2 create a path between them and also define the data rate (2Mb), gap between packets (10ms). The last term defined here is the queue management technique (Drop Tail) of the node through which message is handled. A single line is used to create the connection between only two nodes and this sequence is followed by us to connect more than two nodes. Here five lines code connect the topologies node. First line connects n0 to n1, Second line connects n1 to n2, Third line connects n2 to n3, Fourth line connects n3 to n4

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 2, August 2012

and last Fifth line connects n3 to n5. Here we can set the gap between packets 10ms, 20ms and DropTail queue management technique handling the connection between two nodes. Here we can display a time line which indicates the time period of the model according to that we can easily find the data flow time and traffic between nodes.

A. *Throughput Analysis*:

Throughput is the average rate of successful message delivery over a communication channel. This data may be delivered over a physical or logical link, or pass through a certain network node. The throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second or data packets per time slot.

There are various ways to check the throughput result. To check the throughput result throughput meter module are used on the server side to measure the average connection throughput. To solve the BER problem TCP_Reno uses SACK protocol [19]. It will enable and disable the SACK to perform the error checking operation. IF BER = 0 then no error in the block, but if BER = 1 then error may be here. 1 indicates the error in the message.

B. *Transport Flow / Monitoring The Task:*

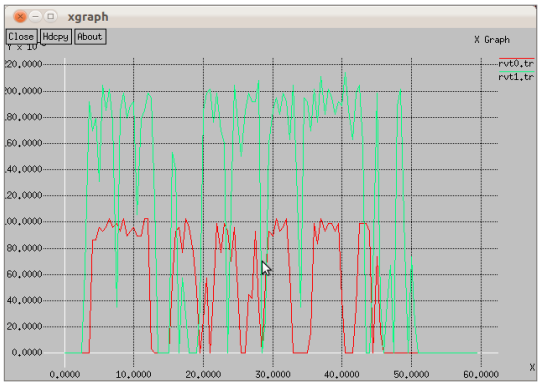

Fig 5. Xgraph (Starting/Ending Time of the Data Flow between Two Nodes).

This type of graph is called the X graph which is used for monitoring the task of nodes. This graph show, at what time data flow will be started and what is the ending time of data flow. Green line shows TCP Reno node data flow and Red line shows the sink data flow. The height of line shows the CBR (Constant Bit Rate), rate of the message transmission. TCP Reno checks the capacity of network to carry the data and after that it will transmit the data packets. The signals in the diagram show the data flow. Figure 5 display the time line of message. We can easily identify the flow and time line of data transportation. Horizontal Line represents maximum time of data flow and minimum time of data flow. Generally XGraph is used for measuring the data rate and bandwidth of transmission. These signals are normally encoded in the digital form. These cannot be seen directly and for this we need XGraph tool. XGraph generate the graph according to the digital value. This value is recorded into trace file.

```
#Setup a TCP connection
    set tcp [new Agent/TCP]
    $tcp set packet_size_ 1000
    $ns attach-agent $n0 $tcp
    set sink [new Agent/TCPSink]
    $ns attach-agent $n4 $sink
    $ns connect $tcp $sink
    $tcp set fid_ 1
#Attach-expoo-traffic
proc attach-expoo-traffic { node sink size burst idle rate } {
    #Get an instance of the simulator
    set ns [Simulator instance]
    #Create a UDP agent and attach it to the node
    set source [new Agent/UDP]
    $ns attach-agent $node $source
    #Create an Expoo traffic agent and set its configuration
    parameters
    set traffic [new Application/Traffic/Exponential]
    $traffic set packetSize_ $size
    $traffic set burst_time_ $burst
    $traffic set idle_time_ $idle
    $traffic set rate_ $rate
    # Attach traffic source to the traffic generator
    $traffic attach-agent $source
    #Connect the source and the sink
    $ns connect $source $sink
    return $traffic
}
#Record the flow of data
proc record {} {
    global sink0 sink1 sink2 f0 f1
    #Get an instance of the simulator
    set ns [Simulator instance]
    #Set the time after which the procedure should be called
    again
    set time 0.5
    #How many bytes have been received by the traffic
    sinks?
    set bw0 [$sink0 set bytes_]
    set bw1 [$sink1 set bytes_]
    #Get the current time
    set now [$ns now]
    #Calculate the bandwidth (in MBit/s) and write it to the
    files
    puts $f0 "$now [expr $bw0/$time*8/1000000]"
    puts $f1 "$now [expr $bw1/$time*8/1000000]"
    #Reset the bytes_ values on the traffic sinks
    $sink0 set bytes_ 0
    $sink1 set bytes_ 0
    #Re-schedule the procedure
    $ns at [expr $now+$time] "record"
}
#Execute NAM on the trace file
    exec nam out.nam &
```

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 2, August 2012

exec xgraph rvt0.tr rvt1.tr -geometry 800x400 &
exit 0

ALGORITHM:-

Step1. Create a Network Simulator Object.

Step2. Create the link between nodes. With duplex-link, data rate, interval and queue management technique.
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1Mb 20ms DropTail
$ns duplex-link $n3 $n4 2Mb 20ms DropTail
$ns duplex-link $n3 $n5 2Mb 20ms DropTail

Step3. Setup a TCP connection
set TCP Agent =[new Agent/TCP]
set packet size
Attach-agent with node $n0 $tcp and connect all the agents.

Step4. Attach expoo traffic with the node.

Step5. Attach the trace code for the XGraph file write the code to calculate the bandwidth of network.
puts $f0 "$now [expr $bw0/$time*8/1000000]"
puts $f1 "$now [expr $bw1/$time*8/1000000]"

Step6. Execute nam on the trace file and finish
exec nam out.nam &
exec xgraph rvt0.tr rvt1.tr -geometry 800x400 &
exit 00

VI. CONCLUSION

In this paper TCP module has been implemented to prove the message transmission rate and its performance. This paper describes the implementation of TCP_Reno & SACK. In this paper we have superimposed TCP_Reno over TCP_Tahoe algorithm. The main purpose of the paper is to explain the TCP_Reno algorithm implementation. We have also described the flow control, BER, PER and Silly Window Syndrome Problem in this paper. We have described the TCP IP Module integration in simulation network that specify how can we implement simulation module in NS2. Overall working of TCP_Reno & SACK with other helping tools can be seen in this paper.

REFERENCES

[1] M.Mathis, S.Floyd. TCP Selective Acknowledgement Options (SACK) http://tools.ietf.org/html/rfc2018.

[2] Jeremy Stretch. TCP Selective Acknowledgement (SACK) http://packetlife.net/blog/2010/jun/17/tcp-selective acknowledgments-sack/.

[3] M. Tuxen, I. Rungeler, and E. P. Rathgeb. Interface Connecting the INET Simulation Framework with the Real World. In Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools), pages 1-6, Marseille/France, Mar. 2008.

[4] A Profile for TCP/IP (ROHC-TCP), http://tools.ietf.org/id/draft-sandlund-rfc4996bis-01.txt.

[5] Marina del Rey, Information Sciences Institute University of Southern California, http://www.ietf.org/rfc/rfc793.txt.

[6] Computer Networks, Fourth Edition By Andrew S. Tanenbaum,http://downloads.ziddu.com/downloadfiles/17254500/ComputerNetworks4thEd-AndrewS.Tanenbaum.rar.

[7] TCP Utilization for TCP Reno, http://idosi.org/wasj /wasj7(c&it)/11.pdf.

[8] TCP IP model, InetDaemon, http://www.inetdaemon. com/tutorials/basic concepts/network models/TCP IP_model/index.shtml

[9] TCP IP Architecture ,by DARPA(United States Department of Defense), http://www.tcpipguide.com/ free/t_TCPIPArchitectureandtheTCPIPModel.htm

[10] TCP Reno, http://www2.ensc.sfu.ca/~ljilja/cnl/ presentations/grace/modeling_TCP/sld001.htm

[11] Computer networks - Andrew S. Tanenbaum TCP Reno, Congestion Avoidance Algorithm

[12] TCP IP, by DARPA (United States Department of Defense), http://en.wikipedia.org/wiki/Internet_protocol_ suite

[13] TCP IP Addresses, http://www.coolinterview.com/ interview/9285/

[14] TCP/IP Protocol Suite 3rd Edition by Behrouz A. Forouzan, OSI Model and TCP/IP Protocol Suite, IP Addressing, IP, RFCs.

[15] TCP Reno, Department of Electrical Engineering and Computer Sciences University of California at Berkeley, Jeonghoon Mo, Richard J. La, Venkat Anantharam, and Jean Walrand http://netlab.caltech.edu/FAST/references /Mo_comparisonwithTCPReno.pdf

[16] Introduction to TCP/IP http://www.yale.edu/pclt/ COMM/TCPIP.HTM

[17] Transmission Control Protocol http://en.wikipedia.org/ wiki/Transmission_Control_Protocol#cite_note-12

[18] An Analysis on TCP Reno, http://inst.eecs.berkeley.edu /~ee122/fa05/projects/Project2/SACKRENEVEGAS.pdf

[19] Enhancement of TCP Module, http://www.tdr.wiwi.uni - due.de/fileadmin/fileupload/I-TDR/SCTP/Paper/OMNeT_ _Workshop2010-TCP.pdf.

Authors Profile

Dr. Neeraj Bhargava : Presently working as Associate Professor & Head, Department of Computer Science, School of Engineering & System Sciences, MDS University , Ajmer. He has more than 23 yr of experience for teaching in the University. His areas of interest are Spatial Database, Image Processing and Ad hoc Network.

Dr. Ritu Bhargava: Presently working as lecturer, Department of MCA, Govt. Women's Engineering College, Ajmer. She has more than 9 yr of experience for teaching MCA. Her areas of research are Web GIS, Wireless Network.

Bharat Kumar: Pursuing Ph.D Computer Science from MJRP University Jaipur. Presently he is involved in Ad hoc network research.