Performance Evaluation and Optimization of TCP Parameters over Frame Relay Network

Gursimran Singh
Department Of Computer Science & Engineering
Continental Institute Of Engineering & Technology, Fatehgarh Sahib, Punjab, India
gkhalsa@theciis.ac.in

Jatinder Singh
Department Of Computer Science & Engineering
Baba Banda Singh Bhadur Engineering College, Fatehgarh Sahib, Punjab, India
jatinder.saini@bbsbec.ac.in

***Abstract* --- The TCP window size value, which is contained in the window size field of the TCP segment is a very important TCP Parameter. The window size value determines the number of bytes of data that can be sent before an acknowledgement from the reciever is necessary. If the window size field value is made too small, the performance deteriorates. This will slow throughput considerably. On the other hand, if the window size field is made too large, the sender may transmit so many segments that reciever will be overloaded. The window size field provides flow control, regulating the rate at which the transport processes on the two hosts may transmit. In this paper we have proposed modifications in TCP Window Size, wherein the performance in the network is improved by changing the TCP parameters. Simulations done on Opnet Simulator shows significant improvement in throughput, reduction in upload response time and delay.**

***Keywords* --- TCP Protocol, Buffer, Average end-to-end Delay, TCP Window Size, Throughput, Frame Relay Network, Opnet, Residual Error Rate**

I. INTRODUCTION

A) Frame Relay Network

Recently, the high performance WAN encapsulation method known as Frame Relay has become one of the most popular technologies in use. It operates at the Physical and Data Link layers of the OSI reference model and was originally designed for use across Integrated Services Digital Network (ISDN) interfaces. Frame Relay assumes that the facilities used are less error prone than when X.25 was used and that they transmit data with less overhead. Frame Relay is more cost-effective than point-to-point links and can typically run at speeds of 64Kbps to 1.544Mbps. Frame Relay provides features for dynamic-bandwidth allocation and congestion control. But today, Frame Relay is used over a variety of other network interfaces.

Frame Relay provides connection-oriented, Data Link layer communication via virtual circuits just as X.25 does. These virtual circuits are logical connections created between two DTEs across a packet-switched network, which is identified by a DLCI, or Data Link Connection Identifier. This virtual circuit provides the complete path to the destination network prior to the sending of the first frame. Popular opinion maintains that Frame Relay is more efficient and faster than X.25 because it assumes error checking will be done through higher layer protocols and application services. Frame Relay provides a communications interface between DTE (DataTerminal Equipment) and DCE (Data Circuit-Terminating Equipment, such as packet switches) devices. DTE consists of terminals, PCs, routers, and bridges—customer-owned end-node and internetworking devices. DCE consists of carrier-owned internetworking devices.

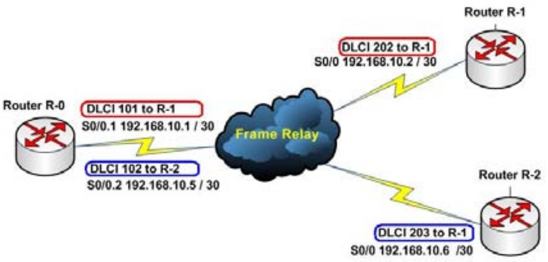

Fig 1. Frame Relay Network

There are two types of services which are used for routing of packets from source to destination. These are

- Connection Oriented Service: Here, first a connection is established and then all packets of the application follow the same route. To understand this concept, we can also draw an analogy from the real life. Connection oriented service is modeled after the telephone system. All voice packets go on the same path after the connection is established till the connection is hung up. It acts like a tube ; the sender pushes the objects in at one end and the receiver takes them out in the same order at the other end.

- Connection Less Service: Each packet of an application is treated as an independent entity. On each packet of the application the destination address is provided and the packet is routed. Connection less service is modeled after the postal system. Each letter carries the destination address and is routed independent of all the others. Here, it is possible that the letter sent first is delayed so that the second letter reaches the destination before the first letter.

B) TRANSMISSION CONTROL PROTOCOL

TCP is a transport layer protocol used by applications that require guaranteed delivery. It is a sliding window protocol that provides handling for both timeouts and retransmissions.

TCP establishes a full duplex virtual connection between two endpoints. Each endpoint is defined by an IP address and a TCP port number. The operation of TCP is implemented as a finite state machine.

The byte stream is transfered in segments. The window size determines the number of bytes of data that can be sent before an acknowledgement from the receiver is necessary.

The Transmission Control Protocol (TCP), makes up for IP's deficiencies by providing reliable, stream-oriented connections that hide most of IP's shortcomings. The protocol suite gets its name because most TCP/IP protocols are based on TCP, which is in turn based on IP. TCP and IP are the twin pillars of TCP/IP.

TCP adds a great deal of functionality to the IP service it is layered over:

- **Streams.** TCP data is organized as a stream of bytes, much like a file. The datagram nature of the network is concealed. A mechanism (the Urgent Pointer) exists to let out-of-band data be specially flagged.
- **Reliable delivery.** Sequence numbers are used to coordinate which data has been transmitted and received. TCP will arrange for retransmission if it determines that data has been lost.
- **Network adaptation.** TCP will dynamically learn the delay characteristics of a network and adjust its operation to maximize throughput without overloading the network.
- **Flow control.** TCP manages data buffers, and coordinates traffic so its buffers will never overflow. Fast senders will be stopped periodically to keep up with slower receivers.
- **IP** - is responsible for moving packet of data from node to node. IP forwards each packet based on a four byte destination address (the IP number). The Internet authorities assign ranges of numbers to different organizations. The organizations assign groups of their numbers to departments. IP operates on gateway machines that move data from department to organization to region and then around the world.
- **TCP** - is responsible for verifying the correct delivery of data from client to server. Data can be lost in the intermediate network. TCP adds support to detect errors or lost data and to trigger retransmission until the data is correctly and completely received.
- **Sockets -** is a name given to the package of subroutines that provide access to TCP/IP on most systems.

Window Size and Buffering

Each endpoint of a TCP connection will have a buffer for storing data that is transmitted over the network before the application is ready to read the data. This lets network transfers take place while applications are busy with other processing, improving overall performance.

To avoid overflowing the buffer, TCP sets a Window Size field in each packet it transmits. This field contains the amount of data that may be transmitted into the buffer. If this number falls to zero, the remote TCP can send no more data. It must wait until buffer space becomes available and it receives a packet announcing a non-zero window size.

Sometimes, the buffer space is too small. This happens when the network's bandwidth-delay product exceeds the buffer size. The simplest solution is to increase the buffer, but for extreme cases the protocol itself becomes the bottleneck because it doesn't support a large enough Window Size. Under these conditions, the network is termed an LFN (Long Fat Network).

Round-Trip Time Estimation

When a host transmits a TCP packet to its peer, it must wait a period of time for an acknowledgment. If the reply does not come within the expected period, the packet is assumed to have been lost and the data is retransmitted. The obvious question - How long do we wait? - lacks a simple answer. Over an Ethernet, no more than a few microseconds should be needed for a reply. If the traffic must flow over the wide-area Internet, a second or two might be reasonable during peak utilization times. If we're talking to an instrument package on a satellite hurtling toward Mars, minutes might be required before a reply. There is no one answer to the question - How long?

All modern TCP implementations seek to answer this question by monitoring the normal exchange of data packets and developing an estimate of how long is "too long". This process is called Round-Trip Time (RTT) estimation. RTT estimates are one of the most important performance parameters in a TCP exchange, especially when you consider that on an indefinitely large transfer, all TCP implementations eventually drop packets and retransmit them, no matter how good the quality of the link. If the RTT estimate is too low, packets are retransmitted unnecessarily; if too high, the connection can sit idle while the host waits to timeout.

II. PROBLEM FORMULATION

Axis Bank has one of its branches located in Chandigarh from where it transfers daily account and transaction information of 30 MB to its backup center (data center) in Mumbai.

The branch and the backup station are connected through a frame relay network with a latency of 5 ms. The time to transfer a 30 MB file over a T1 link is estimated to be approximately 674 sec.

IT team decides to increase the transfer rate by upgrade the link to the frame relay cloud to a T3 link, assuming that the delay is caused due to low bandwidth. The time to transfer a 30 MB file over a T3 link is estimated to be approximately 570 sec. Even the upgrade in bandwidth does not give the desired results.

Another problem arises when error may occur in the networks. Most of the networks are reliable today but still packets may be dropped due to congestion in the network. If there is a congestion then frame relay has to drop packets to clear the congestion. The packets which are dropped due to congestion will have to be retransmitted and this will lead to increase in upload response time, which reduces the efficiency of the network.

This paper relates to finding the solutions to the above mentioned problems by optimizing the Configuration Parameter which is free while upgrading the Wan link is Expensive.

The TCP window size value, which is contained in the window size field of the TCP segment is a very important TCP Parameter. The window size value determines the number of bytes of data that can be sent before an acknowledgement from the reciever is necessary. If the window size field value is made too small, the throughput decreases considerably. On the other hand, if the window size field is made too large, the sender may transmit so many segments that reciever will be overloaded. The window size field provides flow control, regulating the rate at which the transport processes on the two hosts may transmit. We will investigate the performance of an account transfer application with different TCP window sizes.

III. RELATED WORK

Brief survey of literature in area related to the Frame Relay Network has been conducted and summarized as below.

According to [1], Frame relay is a technology for access to wide area networks. The frame relay protocol operates at a sublayer of the data link layer, which provides the minimum functionality needed to take advantage of the statistical properties of the communication. Unlike preexisting protocol suites in use in general topology subnetworks, it also performs relaying, and multiplexing at this sublayer. Other data link layer functions are provided only in systems connected to the edges of the frame relay subnetwork.

In [2], Frame relay is a standardized interface that provides multiplexed access to bandwidth-on-demand backbone networks and delivers LAN-like performance over a wide area. It is shown that use of frame relay can benefit private networks. Frame relay's data link connection identifier (DLCI) addressing allows a single frame-relay access device to communicate as if directly attached to nearly 1000 other access devices. The bandwidth-on-demand characteristics give end systems and intermediate systems the appearance of having far more bandwidth available than they physically have dedicated, and the optional local management interface (LMI) extensions simplify the configuration and management of frame-relay-based networks. Most important is that the application of frame relay interfaces reduces the latency of private network.

In [3], Various aspects of the implementation of access traffic control mechanisms for frame relay networks are considered. Emphasis is placed on the distinction between two related rate control mechanisms, i.e., access policing by the network and traffic shaping by the user. Frame discarding performance policer strictness is compared for two alternative policing techniques, i.e., leaky bucket and quantized moving window.

In [4], frame relay is starting to be deployed within relatively small networks to support applications such as LAN interconnect. It is also being used to increase bandwidth utilisation on trunk kinks within corporate networks. ISDN frame relay capability is expected within a few years, but in this role it will demand more from the equipment and a wider range of issues need to be addressed. Nevertheless such early networks can form the foundation for an evolution path towards full integration of frame relay services within the ISDN.

In [5], the study is to quantify performance of network file transfer for research testbed. This study was carried out because Internet users nowadays has surfing tendency to stream video, download and upload video which need more Internet bandwidth. In addition, there is also a requirement for clarification between theoretical and actual performance of Internet Protocol version 6 (IPv6). This study is important for researchers and network administrators because output from this study is useful for them as an input for simulation and decision making process. Testbed used for this study consists of computers, network switch, server and routers. Multiple sizes of files were transferred by using multiple sizes of Transmission Control Protocol (TCP) Window and IP address version. Result from the test shows that there is a performance gap for file transfer between different IP address versions. More study in this research area will be done because there is a need for enhancement in IP based network performance.

According to [6], the three main approaches to congestion control are compared, and an implicit congestion avoidance technique based on Kalman filtering is summarised. A four-stage cycle of throughput observation, congestion level prediction, traffic load adjustment decision and window modification is simulated for comparison, and performances are plotted against normalised background traffic intensity.

In [7], Flow-control issues in frame-relay networks are investigated. The intermediate nodes perform the routing and store-and-forward functions. In the case of transmission error or buffer overflow, these nodes simply discard frames. Error recovery and flow control are performed by the end nodes. Simulation results show that a properly tuned dynamic window flow control mechanism provides both good throughput/delay performance and fairness. In the cases considered, a linear window increase strategy performs slightly better than a parabolic one. However, the parabolic strategy is less susceptible to transmission errors.

In [8], A design is proposed for an n×n switch to be used in frame relay networks. The design is based on a single storage unit for packets and hardware-based mechanism for handling simultaneously arriving packets over different input channels which may be intended for the same output channel. The switch is flexible in that it can handle variable length packets, a large number of input/output channels, and a wide range of channel speeds. It can perform cut-through switching and thus decrease packet delay through the network.

I V. PROPOSED SOLUTION

The problem of high Upload Response Time and less Throughput which was still not solved by upgrading the T1 link to the Frame Relay Cloud to T3 link can be solved without spending the expensive money by optimizing configuration parameters such as window sizes on the sender and receiver node which is free. This will help in reducing the delay, Round Trip Time and Jitter in the network and factors like throughput and end-to-end delay, will have positive effects on them. We intend to make modifications in TCP Window Size algorithm, present in Opnet Network Simulator to achieve this task.

The second problem of dropped packets which is also leading to increase in upload response time can be solved by implementing an algorithm which will change the rate of packets that it can safely have in transit. A variable is used for each connection. This variable will control how much data is allowed to have in transit at a given time. This variable is called congestion window. Opnet simulator provides with the option of implementing and modifying TCP RENO Algorithm.

V. EXPERIMENTAL SETUP

This section deals with the detailed description of experimental scenario that has been setup to evaluate various traffic parameters of Frame Relay Network. We have built our Frame Relay Network test bed using Opnet Network Simulator. OPNET or OPtimized Network Engineering Tool, provides a Virtual Network Environment that models the behaviour of your entire network, including its routers, switches, protocols, servers, and individual applications. By working in the Virtual Network Environment, IT managers, network and system planners, and operations staff are empowered to diagnose difficult problems more effectively, validate changes before they are implemented, and plan for future scenarios including growth and failure. OPNET's Application Characterization Environment (ACE) module for IT Guru enables enterprises to identify the root cause of end-to-end application performance problems and to solve them cost-effectively by understanding the impact of changes. To implement the Frame Relay Network functionality in the simulation model, we have used 2 Cisco C4700 series Routers, 2 Frame Relay Switches, 4 Ethernet Workstation, 1 Ethernet Server, 1 Frame Relay Cloud, 2 Frame Relay T1 Links, 2 T3 Links, and 1 FR PVC are used. The FR T1 and T3 links are used to connect Chandigarh Branch Router to Frame Relay Cloud and Frame Relay Cloud to Mumbai Branch Router. The Ethernet 10 BaseT is used to connect the frame relay Switch to Router and frame Relay Switch to Ethernet Workstation. Similarly Ethernet 10 BaseT is used to connect the frame relay Switch to Router and frame Relay Switch to Ethernet Server on the backup station side. The designed scenario is shown in fig 2 below.

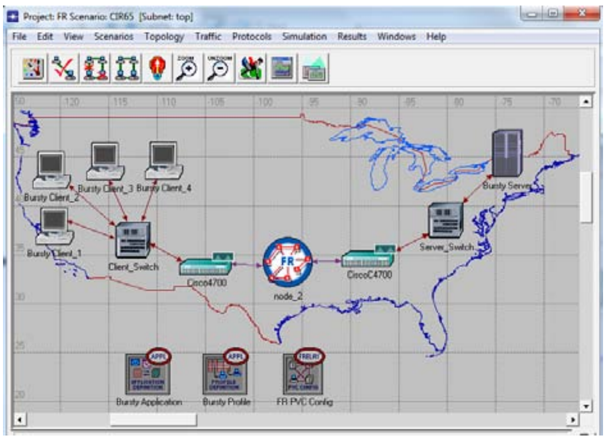

Fig 2. Frame Relay Scenario in Opnet

Thirty MB of file (FTP Data) is transferred from the Chandigarh branch to Mumbai Branch. Values for different parameters are given in the table below.

Parameter	Value
Data	FTP
Routing Protocol	TCP
Simulation Time	1 Hour
Total No of Bytes Sent	30000000
Packet Size	1500 Bytes
TCP Receive Buffer	8760
TCP Receive Buffer	65535 (Modified)

Table 1. Parameter Values

All the Workstations in the Chandigarh Branch are connected to frame relay Switch and the Switch is connected with Router. TCP Routing protocol is used to analyze the different traffic parameters.

VI. RESULTS

The purpose of this experiment is to check the functioning of our scenario and to evaluate the performance of frame Relay network after the proposed modifications. The above mentioned scenario was run for 1 Hour. Total of 30 MB of data were to be transmitted at interval of 1ms, starting from 0 seconds. The branch and the backup station are connected through a frame relay network. The actual Ftp Upload response Time to transfer a 30 MB file over a T1 link is 674 sec. After observing very high transfer times for the file transfer, we decide to upgrade the links between the routers and the frame relay cloud from T1 to T3, assuming that the delay is caused due to low bandwidth. The actual Ftp Upload response Time to transfer a 30 MB file over a T3 link is 570 sec. Even the upgrade in bandwidth does not give the desired results. Then we decide to go back to a T1 link and increase the TCP window size from the initial 8K to 65K. While upgrading a WAN link is expensive, optimizing configuration parameters is free. In the modified scenario, in which TCP Window size is increased from 8K to 64K is implemented, and the time to transfer a 30 MB file over a T1 link was reduced to 192 sec. Fig 3 show the original and modified results after increasing TCP Window Size.

The second parameter which is checked is throughput. In the original scenario, the throughput is 50 packets per second. In the modified scenario when the data is transferred through T3 link throughput is improved to 58 packets per second. Even the upgrade in bandwidth does not give the desired results. In the modified scenario, in which TCP Window size is increased from 8K to 64K mean throughput is improved to 186 packets per second. This was done by reducing the Delay and thus improving the efficiency of the network. Fig. 4 shows the comparison between original and modified throughput values.

The third parameter which is checked is Residual Error Rate. In the original scenario, the residual Error Rate is 0.01 percent. In the modified scenario when the data is transferred through T3 link the residual Error Rate is again 0.01 percent. In the modified scenario, in which TCP Window size is increased from 8K to 64K , the residual Error Rate is 0.01 percent. This shows that by changing Tcp window size we can increase the throughput and Residual Error rate still has no effect. It is same and thus improving the efficiency of the network. Fig. 5 shows the comparison between original and modified Residual Error Rate Values.

The fourth parameter which is checked is TCP Delay. In the original scenario, the TCP Delay is 307 seconds. In the modified scenario when the data is transferred through T3 link the TCP Delay is reduced to 255 seconds. In the modified scenario, in which TCP Window size is increased from 8K to 64K , the TCP Delay is reduced to 75 seconds . This shows that by changing Tcp window size we can reduce the delay and increase the throughput and thus improving the efficiency of the network. Fig. 6 shows the comparison between original and modified TCP Delay.

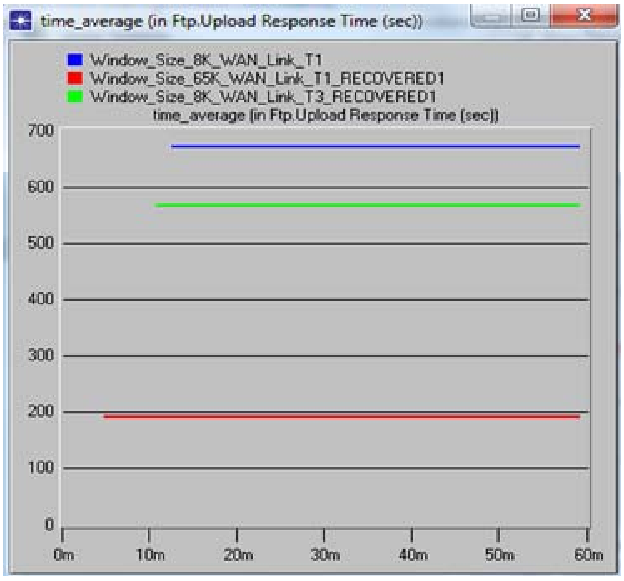

Figure 3. FTP Upload Response Time

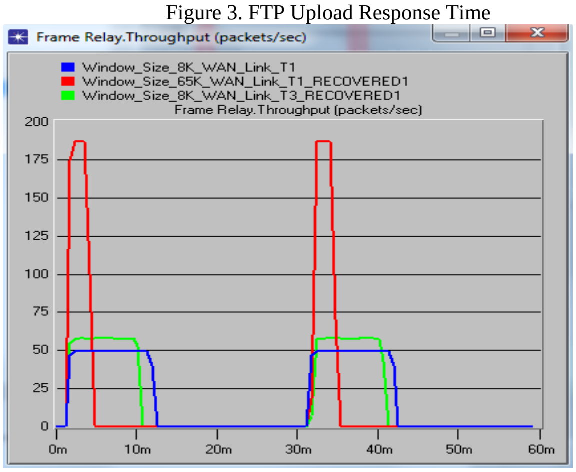

Figure 4.Throughput

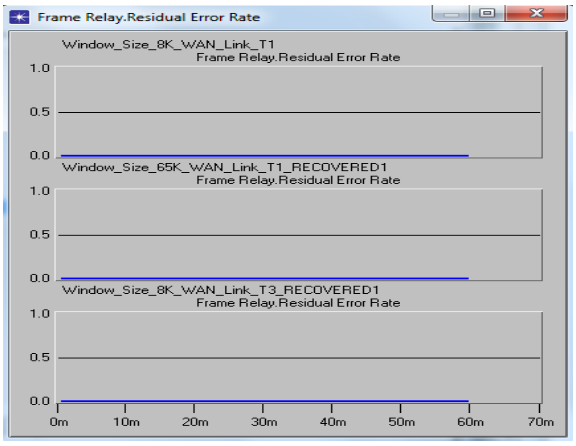

Figure 5. Residual Error Rate

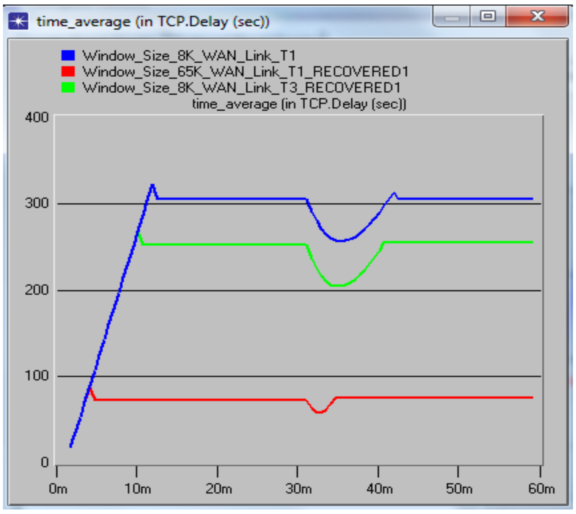

Figure 6. TCP Delay

All the graphs which are shown above are taken under network in which we are having no drops that is no error has occurred during the transmission. Error may occur in any networks. Most of the networks are reliable today but still packets may be dropped due to congestion in the network. Acknowledgement of dropped packets is not received by source so the source will retransmit packets which will increase the upload response time. Now this Upload Response Time which is increased due to packets dropped in the network can be reduced by implementing RENO Algorithm.

The idea of TCP congestion control is for each source to determine how much capacity is available in the network, so that it knows how many packets it can safely have in transit. It maintains a state variable for each connection, called the congestion window, which is used by the source to limit how much data it is allowed to have in transit at a given time. TCP uses a mechanism, called additive increase/ multiplicative decrease, that decreases the congestion window when the level of congestion goes up and increases the congestion window when the level of congestion goes down. TCP interprets timeouts as a sign of congestion. Each time a timeout occurs, the source sets the congestion window to half of its previous value. This halving corresponds to the multiplicative decrease part of the mechanism. The congestion window is not allowed to fall below the size of a single packet. Every time the source successfully sends a congestion window's worth of packets, it adds the equivalent of one packet to the congestion window; this is the additive increase part of the mechanism.

TCP uses a mechanism called slow start to increase the congestion window rapidly from a cold start in TCP connections. It increases the congestion window exponentially, rather than linearly. Finally, TCP utilizes a mechanism called fast retransmit and fast recovery. Fast retransmit is a heuristic that sometimes triggers the retransmission of a dropped packet sooner than the regular timeout mechanism.

As we are having drops in our network due to congestion in the same above mentioned scenario. Total of 30 MB of data is again transferred over T1,T3 Link and T1 Link with window size 65K and now the upload response time increases to 833, 702, 538 seconds respectively. Now this Upload Response Time which is increased due to packets dropped in the network can be reduced by implementing RENO Algorithm. After implementing the TCP RENO Algorithm the upload response time reduces to 802, 695,506 seconds over T1, T3 and T1 with window size 65K Link respectively. Fig 7 show the original and modified results after implementing the TCP RENO Algorithm

As shown below in the graphs we can see that when packets are dropped the graph of the congestion window size goes down and packets have to retransmitted but here we have not implemented algorithm so it takes more time as it can be seen in the graph But when we have implemented TCP RENO algorithm then it takes less time as dropped packets are transmitted earlier so results are improved. Fig 8 show the original and modified results after implementing the TCP RENO Algorithm

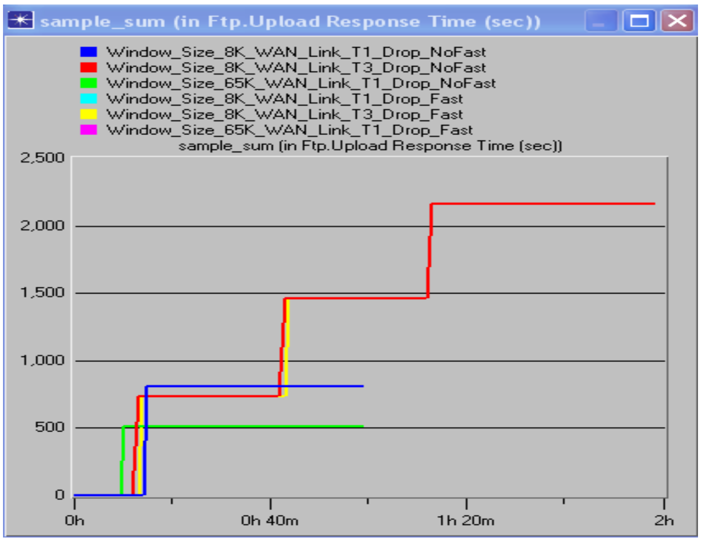

Figure 7. Upload Response Time

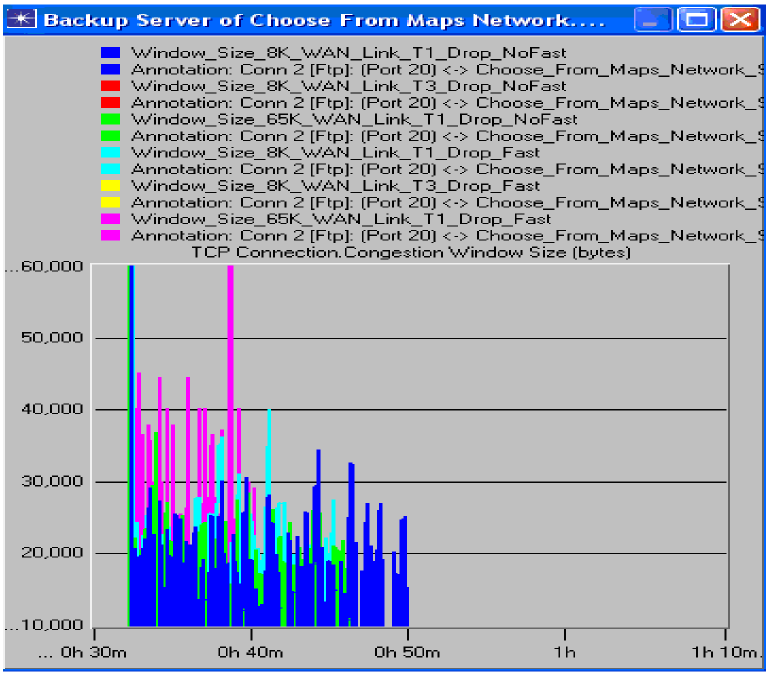

Figure 8. Congestion Window Size

VII. CONCLUSION

A Frame Relay Network scenario is developed in order to analyze and enhance the performance of various traffic parameters using TCP protocol and CBR traffic was used. A scenario is developed in Opnet having 2 Cisco C4700 series Routers, 2 Frame Relay Switches, 1 Frame Relay Workstation, 1 frame Relay Server, 1 Frame Relay Cloud, 2 Frame Relay T1 Links, 2 T3 Links, and 1 FR PVC . It is observed that by increasing the TCP window size from 8K to 65K and keeping the bandwidth to T1, we were able to achieve almost the estimated results. We were able to achieve maximum throughput by reducing delay. In other ways, intelligence in optimizing TCP parameters costs little or nothing but does a better job than an expensive WAN upgrade. These results show that bandwidth was not the cause for the high response times. We can certainly improve upon the throughput by modifying the TCP Window Size.

VIII. REFERENCES

[1] Grossman, D.B, “An overview of frame relay technology” Tenth Annual International Phoenix Conference on Computers and Communication, IEEE Volume: 43, Publication Year: 2002, Page(s): 539-545.
[2] Cavanagh, J.P. “Applying the frame relay interface to private networks” Communications Magazine, IEEE Volume: 30 , Issue: 3. Publication Year: 1992, Page(s): 48 – 54.
[3] Petr, D.W.; Evans, J.B.; Neir, L.; Singh Jasvindarjit; Frost, V.S.; " Access traffic control implementations for frame relay," Communications, 1993. ICC 93. Geneva. Technical Program, Conference Record, IEEE International Conference on Volume: 3 Publication Year: 1993 , Page(s): 1348 - 1352 vol.3.
[4] Cooper, N.J.P, “Frame relay: the implementer's view” Telecommunications, 2001. Third IEEE Conference on Publication Year: 2001 , Page(s): 78 – 83.
[5] Mohd. Khairil Sailan and Rosilah Hassan “Network File Transfer Performance: Effect of IP Address, TCP Window Size and File Size”. Network Applications Protocols and Services (NETAPPS), 2010 Second International Conference on Network Applications, Protocols and Services, Publication Year: 2010 , Page(s): 114 – 118.
[6] Waters, A.G.; Ab-Hamid, K.; “Congestion control for frame relay networks” Frame Relay, IEE Colloquium on Publication Year: 2003 , Page(s): 3/1 - 3/4.
[7] Nassehi, M.; “Window flow control in frame-relay networks”, Global Telecommunications Conference, 1988, and Exhibition. 'Communications for the Information Age.' Conference Record, GLOBECOM '88., IEEE, Publication Year: 2004 , Page(s): 1784 - 1790 vol.3.
[8] Birman, A.; “A high-performance switch with applications to frame relay networks”, INFOCOM '90. Ninth Annual Joint Conference of the IEEE Computer and Communication Societies. 'The Multiple Facets of Integration'. Proceedings., IEEE, Publication Year: 2005 , Page(s): 782 - 789 vol.2
[9] Cherukuri, R.J; Derby, J.H; “Frame relay: protocols and private network applications” INFOCOM '05. Proceedings of the Eighth Annual Joint Conference of the IEEE Computer and Communications Societies. Technology: Emerging or Converging, IEEE Publication Year: 2005 , Page(s): 676 - 685 vol.2
[10] Jain, R.; “Congestion control in Computer Networks: issues and trends”, IEEE ,Publication Year : 2000 , Page(s): 24 – 30.
[11] Larry L. Peterson and Bruce S. Davie, “Computer Networks A system Approach” Morgan Kaufmann Publishers by ElsevierScience(USA)in2007.