Simulation & Comparison of CSFQ, RED & FRED Queuing Techniques

Sandeep, Rambir Joon, Manveen Singh Chadha

Abstract: Today's Internet only provides Best Effort Service. Traffic is processed as quickly as possible, but there is no guarantee of timelines or actual delivery. With the rapid transformation of the Internet into a commercial infrastructure, demands for service quality have rapidly developed. People of the modern world are very much dependent on various network services like VOIP, Videoconferencing and File Transfer. Different types of Traffic Management systems are used in those services. Queuing is one of the very vital mechanisms in traffic management system. Each router in the network must implement some queuing discipline that governs how packets are buffered while waiting to be transmitted. This paper gives a comparative analysis of three queuing systems CSFQ, RED and FRED. The study has been carried out on some issues like: Throughput, packet end to end delay and packet delay fraction rate the simulation results shows that CSFQ technique has a superior quality than the oth techniques.

Keywords: RED (Random Early Drop), FRED (Flow random Early Drop) and CSFQ (Core Stateless fair Queuing)

I. INTRODUCTION

Today's Internet only provides best effort service and traffic is processed as quickly as possible, but there is no guarantee for the timely delivery of data. So, the ability to provide flow based quality of service (QoS) support has become very important for the design of modern switches and routers. With the development of the Internet network in recent years, a variety of novel Internet multimedia applications, such as voice over IP and videoconferencing, have been developed, which usually have different quality of service requirements. In order to complete various processes successfully, the network should maintain a good QoS (Quality of Service) to provide satisfactory

Results to the user. QoS must be efficient to differentiate the traffic and satisfy their specific requirements. As the traffic on network is increasing due to congestion, it decades the performance of network, the different priorities can be assigned to different applications to enhance the performance of network. This paper demonstrates the performance of a number of packet handling mechanisms and produces a comparative picture of them using the simulation software NS-2(Network Simulator-2).

Manuscript received on July, 2012

Manveen Singh Chadha, M.Tech, Department of ECE, Jagannath University, Jaipur (Rajasthan), India

Rambir Joon, M.Tech, Department of ECE, Jagannath University, Jaipur (Rajasthan), India

Sandeep, M.Tech, Department of ECE, Jagannath University, Jaipur (Rajasthan), India

II. PACKET HANDLING TECHNIQUES

Various queuing disciplines can be used to control which packets get transmitted and which packets which packets get dropped. The queuing disciplines are:

1. **RED** (Random Early Drop)
2. **FRED** (Flow random Early Drop)
3. **CSFQ** (Core Stateless fair Queuing)

A. Red (random early drop)

The basic idea behind RED queue management is to detect incipient congestion early and to convey congestion notification to the end-hosts, allowing them to reduce their transmission rates before queues in the network overflow and packets are dropped.

To do this, RED maintains an exponentially-weighted moving average (EWMA) of the queue length which it uses to detect congestion. When the average queue length exceeds a minimum threshold (min_{th}), packets are randomly dropped or marked with an explicit congestion notification (ECN) bit. When the average queue length exceeds a maximum threshold (max_{th}), all packets are dropped or marked. Random Early Detection (RED) keeps no per flow state information. Packets are dropped probabilistically based on the long-term average queue size and fixed indicators of congestion (thresholds). RED uses randomization to drop arriving packets to avoid biases against bursty traffic and roughly drops packets in proportion to the flows data rate at the router. However, flows with high RTTs and small window sizes are bursty, and this burstiness causes high variability in the perceived data rate of these flows as seen by RED routers.

B. flow random early drop (fred)

Flow Random Early Drop (FRED) is a modified version of RED, which uses per-active-flow accounting to make different dropping decisions for connections with different bandwidth usages. FRED only keeps track of flows that have packets in the buffer, thus the cost of FRED is proportional to the buffer size and independent of the total flow numbers (including the short-lived and idle flows). FRED can achieve the benefits of per-flow queuing and round-robin scheduling with substantially less complexity. Flow Random Early Drop (FRED) uses per-flow preferential dropping to achieve fairer allocation of bandwidth among flows. FRED builds per-flow state at the router by examining those packets that are currently in the queue. The packet drop rate for a flow is determined by the number of packets the flow has in the queue, and is not directly influenced by the flow's data rate or round trip time. We evaluate the

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

effectiveness of FRED as a less expensive means of attempting per-flow fairness.

Some other interesting features of FRED include: (1) penalizing non-adaptive flows by imposing a maximum number of buffered packets, and surpassing their share to average per-flow buffer usage; (2) protecting fragile flows by deterministically accepting flows from low bandwidth connections; (3) providing fair sharing for large numbers of flows by using "two-packet-buffer" when buffer is used up; (4) fixing several imperfections of RED by calculate average queue length at both packet arrival and departure (which also causes more overhead).

Two parameters are introduced into FRED: min_q and max_q, which are minimum and maximum numbers of packets that each flow is allow to buffer. In order to track the average per-active-flow buffer usage, FRED uses a global variable *avgcq* to estimate it. It maintains the number of active flows, and for each of them, FRED maintains a count of buffer packets, *qlen*, and a count of times when the flow is not responsive ($qlen > max_q$). FRED will penalize flows with high strike values

C. core stateless fair queuing (csfq)

The Core-Stateless FQ scheme, CSFQ, distinguishes core routers, the higher-speed and busier routers at the "core" of an Internet AS backbone from edge routers. In typical deployment, edge routers might handle thousands of flows, while core routers might handle 50k-100k flows. CSFQ exploits this gap by delegating the management of per-flow statistics to the edge routers. Edge routers then share this information with core routers by labeling each packet that they forward. Core routers, in turn, can use the labels to allocate bandwidth fairly among all incoming flows. It is important to realize that in the case of CSFQ, edge routers run essentially the same algorithm as core routers (including probabilistically dropping incoming packets); however, edge routers have the added responsibility of maintaining per-flow state. In general, of course, edge and core routers in such an approach could run very different algorithms. Here are the key points of CSFQ:

1. Dynamic Packet State: Edge routers label each packet with an estimate of the arrival rate for each flow. Per-flow statistics are maintained here.
2. Core routers use estimated arrival rates provided on packet labels, and An Internal measure of fair-share, to compute the probability of dropping each incoming Packet. Every packet that is accepted is processed and relabeled with new arrival rate information.
3. The estimation procedure for the "fair-share" value convergences rapidly to the optimal value. Cheaters cannot win too much extra bandwidth.

There are two goals for a CSFQ router:

1. Maintain max-min fairness for bandwidth allocation.
2. Avoid having to keep per-flow statistics in high-speed core routers.

Note that goal (2) prevents a core router from maintaining per-flow queues. Therefore, once a packet has been accepted by a core router, it sits in one of a small number of queues until it is eventually processed. Hence, the only action the core router can take in order to achieve (1) is to drop packets from greedy flows. Notably absent is the ability to schedule when a packet is to be sent. In order to avoid patterns of synchronization, packet dropping is done probabilistically using both information accumulated and added to the packet by an edge router, and a global parameter estimated by the core router.

CSFQ (Core Stateless fair Queuing) is a Queuing Technique used to achieve fair bandwidth allocation using differential packet dropping. The CSFQ architecture differentiates between 'edge' and 'core' nodes. The edge nodes performs per flow management, core nodes do not perform per flow management and therefore can be efficiently implemented at high speeds. The main objective of CSFQ is to achieve fair bandwidth allocation with a simpler and more scalable approach. Edge and core routers use first in first out (FIFO) queuing. Edge routers compute per-flow rate estimates and label the packets passing through them by inserting these estimates into each packet header. Edge and core routers estimate the fair share rate, and employ a probabilistic dropping algorithm that utilizes the information in the packet header and the fair share rate. There are several basic mechanisms for realizing CSFQ. They are flow arrival rate estimation algorithms, fair rate estimation algorithms, and packet dropping algorithms.

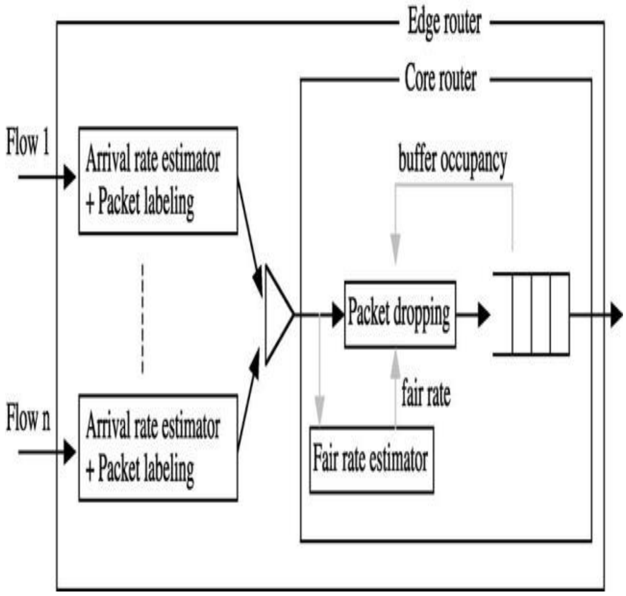

Figure 1: Architecture of edge and core routers (Nabeshima, 2002)

III. SIMULATION RESULT AND ANALYSIS

The main parameters that are taken into consideration are packet delay fraction, Throughput and end to end delay. The comparative investigation on different queuing disciplines based on heavy congestion is evaluated. The significant result has been investigated for CSFQ for all the factors. It has been evaluated from the simulation results that the CSFQ receives more packets, shows minimum delay and maximum Throughput. Therefore CSFQ may be referred or recommended for heavy traffic.

International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-2, Issue-3, July 2012

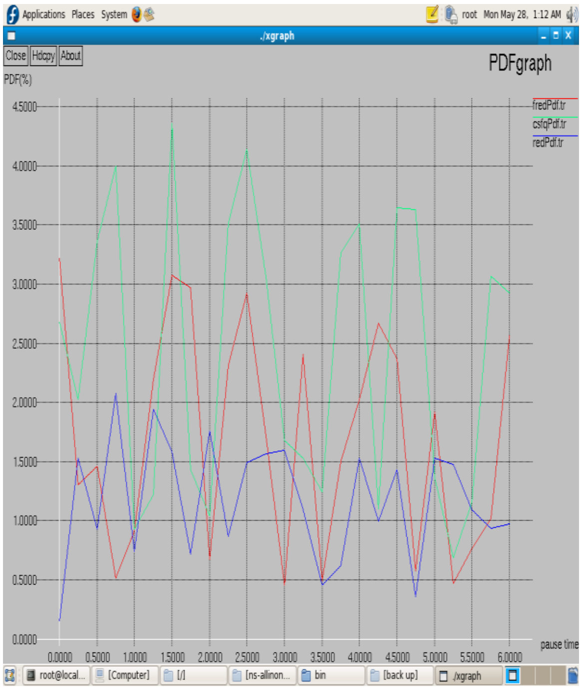

Figure 2: Comparison analysis for RED, FRED and CSFQ with packet delivery Fraction.

From Fig.2 it is clear that PDF% of CSFQ is better than FRED and RED. Average Packet delivery fraction of CSFQ is 2.42, FRED is 1.70 and RED is 1.18 is calculated for Fig. 2 at 15 nodes.

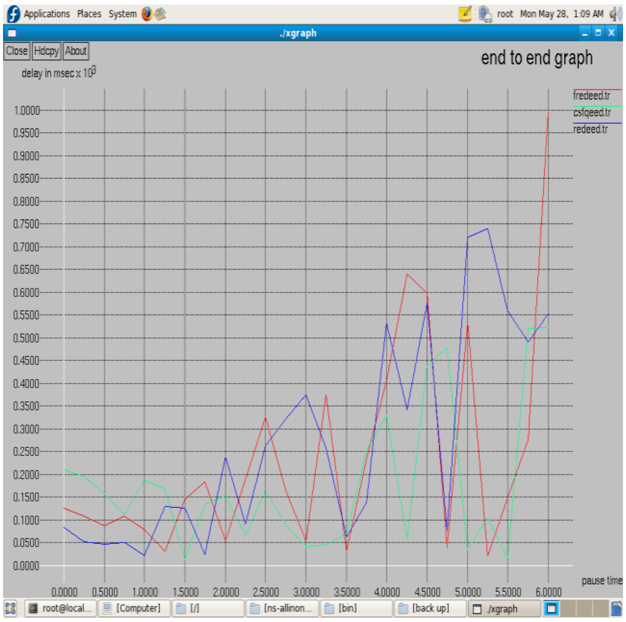

Figure 3: Comparison analysis for CSFQ, FRED and RED with End to End Delay

From Fig. 3 it is clear that End to End delay of CSFQ is maximum at time 0ms and minimum at time 6ms. Average End to End delay for CSFQ is 182.30, FRED is 238.21 and RED is 274.98 is calculated for Fig.3 at 15 nodes.

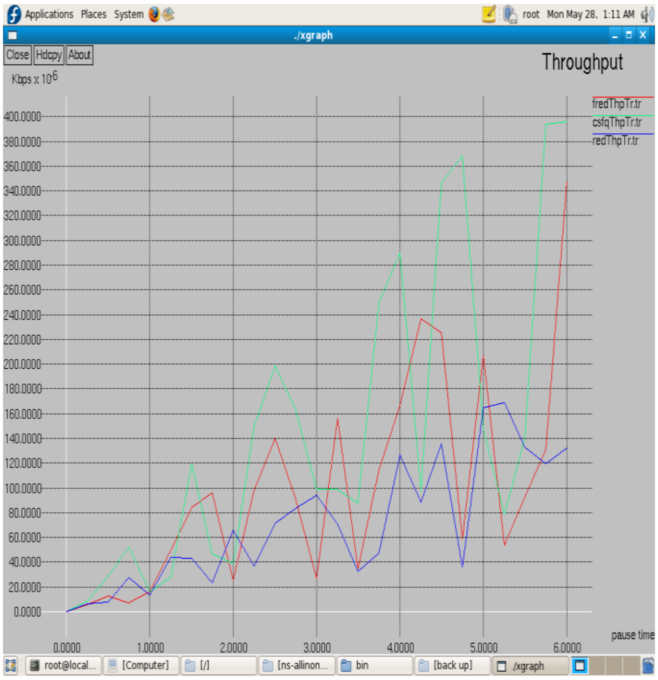

Figure 4: Comparison analysis for CSFQ, FRED and RED with Throughput.

From Fig. 4 it is clear that throughput is high for CSFQ as compared to FRED and RED.

IV. NAM (NETWORK ANIMATOR) FILES OF QUEUING TECHNIQUES

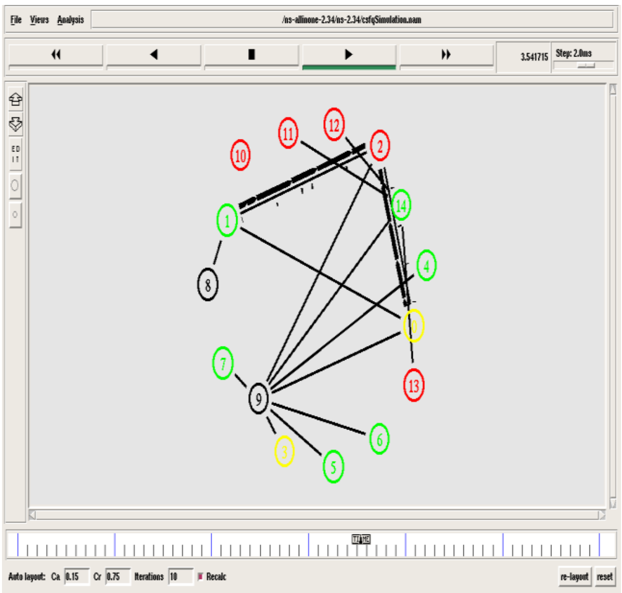

Figure 5: Screenshot of CSFQ NAM (Network Animator)

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

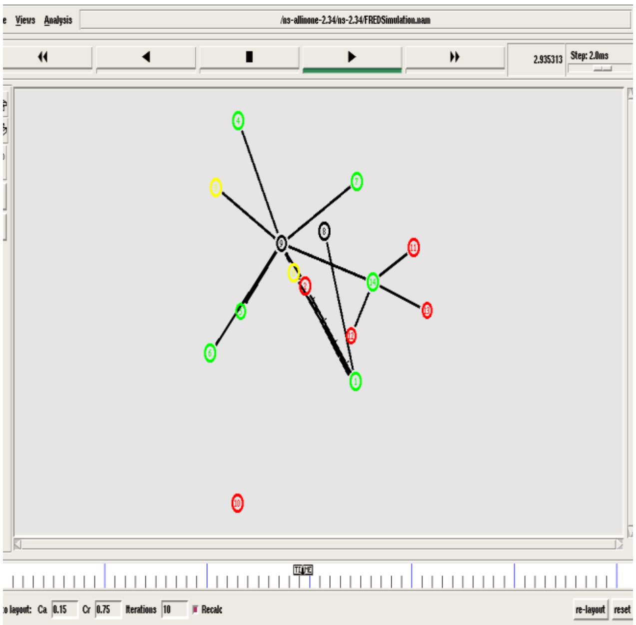

Figure 6: Screenshot of FRED NAM (Network Animator)

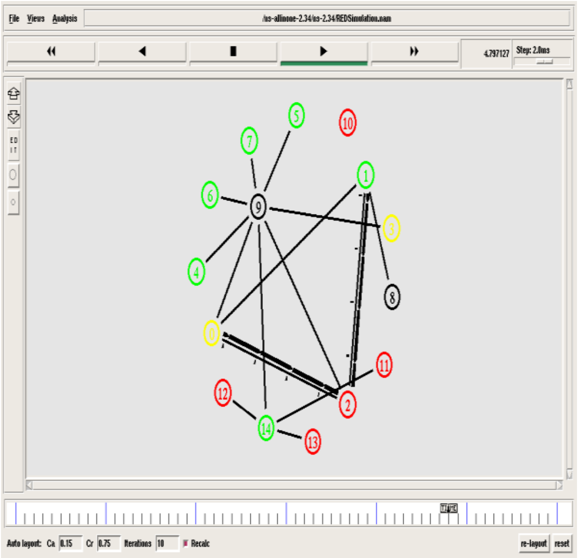

Figure 7: Screenshot of RED NAM (Network Animator)

V. CONCLUSION

Active queue management is the widely accepted mechanism for managing queues in routers servicing best effort traffic. Implementations such as RED and FRED have demonstrated better performance than traditional drop tail mechanisms. In this thesis a new queue management scheme is also introduced.

This work implements a simulation based performance evaluation and comparison of three queuing scheduling disciplines for different performance parameters. The study of these queuing disciplines also examined the impact of using RED (Random Early Drop) as compared to drop tail policy. The simulation results shows that CSFQ performs better than RED and FRED and other queuing disciplines in terms of packet delay fraction rate, throughput and end to end delay although FRED is very close to it for the considered traffic scenarios. CSFQ works on three algorithms: flow arrival rate estimation algorithm, fair rate estimation algorithm, packet dropping algorithm. FRED records per active flow management but shows longer delay as compared with CSFQ. CSFQ and several other algorithms are analyzed on wide variety of conditions. CSFQ achieves a significant degree of fairness in all of these circumstances. It is comparable or superior to FRED, and vastly better than the baseline cases of RED and FIFO.

REFERENCES

1. Masayoshi Nabeshima (2002) "Adaptive CSFQ: determining the packet dropping probability adaptively for achieving fair bandwidth allocations in SCORE networks" conference on computer communications, Yokosuka-shi, Kanagawa, Japan , pp. 1-2
2. Ion Stoica , Scott Shenker , Hui Zhang (2003) "CSFQ : Achieving Approximately Fair Bandwidth Allocations in High Speed Networks" IEEE/ACM transactions on Networking, Vancouver, pp 33-46.
3. H.Gundersen and F.T. Grimstad (2001), "QoS for Real Time Traffic" IEEE computer and Communication proceedings, New York, pp. 50-55
4. K.Ahlin (2003) "Quality of Service in IP Networks", Linkoping University, pp. 10-18
5. C.Semeria (2001) "Supporting differentiated service classes, queue scheduling disciplines" white paper, pp 5-25. www.juniper.net.
6. N.Alborz (2002) "Implementation and Performance Simulation of Virtual Clock Scheduling Algorithm in IP Networks" , Washington, pp.
7. Fourstics CTO Alan Noble (2003) "Network Performance Technology" Whitepaper, pp. 8
8. T. P. Lee and J. Siliquini (2007) "Deficit Round Robin with hop-by-hop credit based flow control" IEEE published, pp 1-4.
9. E.C. Popovici and T. Radulescu (2007) "Coding scheme impact on the IP QoS network utilization and voice quality" pp 2-9
10. T.Subash and S.Indira Gandhi (2006) "Performance Analysis of Scheduling Disciplines in Optical Networks" , IEEE Published, pp 1-5.
11. N. Sulaiman, R.Carrasco and G. Chester (2006) "Analyzing the Performance of Voice over Internet Protocol in a 3G Network", IEEE Published, pp 1-5
12. L.Le, K. Jeffay and F.D Smith (2006) "A Loss and Queuing-Delay Controller for Router Buffer Management", IEEE, pp 1-10
13. O.Salami and H.A Chan (2006) "Multistage Queuing and Scheduling of IP traffic for QoS Provisioning", IEEE, pp 1-6.
14. M. Gospodinov (2004) "The affects of different queuing disciplines over FTP, Video and VoIP Performance", International conference, pp 1-5.
15. I. Miloucheva, A. Nassri and A. Anzaloni (2004) "Automated analysis of network QoS parameters for Voice over IP applications", pp 1-10.
16. I. Bouazizi (2003) "Size-Distortion Optimization for Application-Specific Packet Dropping, the Case of Video Traffic", IEEE Published, pp 1-6.
17. H.M. Chaskar and U.Madhow (2003) "Fair Scheduling With Tunable Latency:A Round-Robin Approach" , IEEE , pp 1-5 ,9.
18. M. F. de Castro, A. M'hamed, Dominique gaiti, Mauro Oliveira (2003). "Simulated Internet Traffic Behavior under Different QoS Management Scenarios", IEEE, pp 1-7.
19. A. Bitorika, M.Robin and M. Huggard (2003) "An Evaluation Framework for Active Queue Management Schemes" , IEEE, pp 1-7.
20. F.Palmieri (2003) "Improving the performance in multimedia streaming networks: A differentiated service approach" , IEEE, pp 1-9
21. Salil S. Kanhere and H.sethu (2001) "Low latency gurantee rate scheduling using extended round robin" pp 1, 3,8.
22. G. Zhang, H.T. Mouftah (2001) "End-to-End QoS Guarantees over Diffserv Networks", IEEE, pp 1-9.
23. Salil S. Kanhere, Alpa B. Parekh and H.Sethu (2000) "Fair and Efficient Packet Scheduling in Wormhole Networks", IEEE, pp 1-9.
24. D.A. H, M.Rumsewicz and L.L.H. Andrew (1999) "Impact of Flow Control on Quality of Service Driven Packet Scheduling Disciplines", IEEE, pp 1-6.
25. P.Goyal, H.M.Vin, and H.Cheng (1997) "Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks", IEEE, pp 1, 2, 11-14.
26. M.Shreedar and G.Varghese (1995) "Efficient Fair Queuing using Deficit Round Robin", IEEE, pp 1-5, 7-12.
27. R.D. Fellman, R. Grygiel, and I.Chu (1995) "The Effect of Preemptive Queuing in a Priority-Based Real-Time Network", pp 1-4.
28. A. Parekh and G.Gallager (1994) "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case", IEEE, pp 1, 12-13.

29. A. Parekh and R.G. Gallager (1993) "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case", IEEE, pp 1-2.
30. A.Dhemers, S.Keshav and S. Shenkar (1990) "Analysis and Simulation of a Fair Queuing algorithm", pp 1-12
31. S. Jamaloddin Golestani (1990) "Congestion-Free Transmission of Real-Time Traffic in Packet Networks" pp 1-8.
32. Ningning Hu, Liu Ren, Jichuan Chang (2001) " Evaluation of Queue Management Algorithms" Course Project report for Computer Networks, pp. 1-14 .

AUTHORS PROFILE

Mr.Sandeep received his B.Tech from Gurgaon Institute of Technology and Management affiliated from MDU, Rohtak & M.Tech from Jagannath university Jaipur in 2009 and 2012 in Electronics and Communication. His field of interest include Wireless and Mobile communication.

Mr. Manveen Singh Chadha received his B.Tech from Dehradun Institute of Technology, Dehradun affiliated from UPTU & M.Tech from Jagannath University, Jaipur in 2008 & 2012 respectively. His field of interest includes Data Communication and Networking, Wireless and Mobile Communication and Digital Communication.

Mr. Rambir Joon received his B.Tech from S I T M affiliated from M.D.U. Rohtak & M.Tech from Jagannath University, Jaipur, in 2008 & 2012 in Electronics & Communication. His field of Research interest include Wireless Communication.