Middle-East Journal of Scientific Research 22 (8): 1150-1156, 2014
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2014.22.08.22055

VOIP Congestion Control with Adaptable Token Generation Rate

Sheikh Aqeel Abbas and Malik Sikandar Hayat Khiyal

Faculty of Computer Sciences, Preston University, Islamabad, Pakistan

Abstract: VoIP users are concerned about quality of voice communication. Voice quality depends upon many factors like delay, packet loss. Such factors consequently bring in congestion. Congestion occurs when sender do not decrease sending rate or there is delay or packet loss. A smooth flow of traffic is needed for VoIP to maintain its quality. It is also important for VoIP throughput. To maintain a smooth transfer rate of voice packets a method is presented to control traffic congestion. It reduces the chances of congestion occurrence almost to zero. It also adopts according to changing traffic rate, handles burst and regulates the traffic for a continuous flow. This paper includes a token bucket traffic shaper in which token generation rate is predicted according to data burst and hence tackles the problem of congestion control. Token bucket traffic shaper uses fuzzy logic to predict adoptable token generation rate. Token rate prediction helps dynamic traffic shaping. Consequently token bucket traffic shaper scheme achieves less cell loss, lower delay and higher throughput. A simulation study was performed for token bucket traffic shaper using OMNeT Simulator. Simulation results show that token bucket traffic shaper gives better results as compared to normal token bucket with fixed token generation rate.

Key words: Congestion · Fuzzy Logic · Token Bucket · Token Generation · Packet loss · Traffic Shaping · OMNeT · VoIP

INTRODUCTION

Voice communication is an emerging technology and has great importance in our routine life. VoIP is a popular technology for transmitting voice over internet protocol. Although it is a very popular and less expensive method of transferring voice packets over a single network but it has some major challenges too as is in evolution period. In other words, VoIP is still in process of development. Many challenges are faced by this new technology of transferring voice packets. It requires resolving many challenges before it outclass the existing way of voice transmission.

One of the prominent issue faced is maintaining quality of service for voice transmission. There are a number of factors which affect voice transfer, like codec, bandwidth utilization, delay, jitter, jitter buffer, packet loss, latency and traffic congestion. All of these factors lead to poor communication and low voice quality. Poor voice quality is experienced by delay, jitter and packet loss which is due to network congestion. Network congestion comprises of, one packet loss, delay which is further classified into fixed delay (codec processing, serialization, propagation) and variable delay (processing and queuing) and secondly of jitter (delay variation).

Voice quality depends on delay and these delays ultimately lead to congestion. Congestion is a critical challenge for VoIP and is likely to occur in VoIP. When offered load increases and resources are over utilized Traffic congestion likely to occur. Congestion occurs when the sending rate of packets increases than the available bandwidth. This cause excessive delay in transmitting or retransmission of packets and subsequently packet loss occurs. This results in low throughput of destination. The main objective to control or avoid congestion is to reduce or eliminate congestion. Any shared point within a network can be a prospective point of congestion.

Congestion control is a check on congestion through which smooth flow of traffic on an available bandwidth is assured in order to improve performance. Congestion control is classified in two major categories on the basis of place. First is source end control and second one is network control (routers). The other classification is

Corresponding Author: Sheikh Aqeel Abbas, Faculty of Computer Sciences , Preston University, Islamabad, Pakistan.
E-mail: msikandar59@gmail.com.

based on procedure and it also has two major categories. First is an open loop algorithm and second is a closed loop algorithm. Open loop is a preventive method. It controls the traffic and manages to avoid the congestion before it occurs. The closed loop control is reactive in nature and it starts working when congestion occurs.

Open loop algorithms which mostly used to control congestion are leaky bucket (LB) and token bucket (TB). LB works like a water bucket which has a hole in the bottom that allows a continuous flow of water through that hole. Same idea is used in the algorithm that traffic enters in the LB and transmits through it at a preset rate to the network. It uses a queue normally FIFO (first in first out) [1] to place the data and a specific buffer size to hold the data. It transmits the data at a preset rate so when a connection is established or when the data is sent to the network flow, no feedback is required as the data is sent at fixed rate. TB is a modified form of LB, the concept is same as LB except a token is generated for each data item (packet) depending on the size of the packet. The packet is allowed to be transmitted only when it receives a token.

Token traffic shaper is presented to control congestion. The idea is to use token bucket algorithms for peak and off peak rate. Token Bucket works for the transfer of packets by sending them to network. TB traffic shaper reduces the chances of occurrence of congestion. It handles the burst in data and regulates the data stream so that there is almost no chance for congestion occurrence. Hence increases the fairness, quality of voice packets, less packet loss and throughput.

TB traffic shaper works better than the normal token bucket by reducing chances of congestion and by increasing throughput.

Related Work: Traffic shaping is discussed by various authors using different parameters. Many of them works on Leaky Bucket [2, 3] and Token Bucket [4, 5] algorithms for shaping traffic and modeling [6]. Unstable and excessively long handoffs and unpredictable occurrence of bursts in VoWiFi (Voice over WiFi) are discussed [7] as limitations of current WiFi. Packet losses are not rare events in VoIP traffic stream [8]. These occurs often especially on international paths.

Author mostly views the aspect of Mobile VoIP Technology in business [9]. It identifies the relevant technologies in implementing mobile VoIP services. Main obstacles faced in VoIP are bandwidth utilization and traffic congestions [10]. The method MFSP (multiple frames into a single packet) reduces the traffic through the network and packet overhead.

Voice packets are transferred using RTP (real time protocol). RTP is suitable to maintain the QoS (quality of service) [11]. Delay, jitter and packet loss are some of the major factors which affects the speech quality in VoIP [12].

Applications those are sensitive and affected by delay and jitter needs small queues in the routers [13]. If low rate VoIP is transmitted over congested links, both coding and packet rate needs to be adapted in order to avoid congestion [14]. Internet's congestion control algorithms regulate the flow and prevent a congestion collapse [15]. TCP-friendly rate control (TFRC) algorithms are not suitable for voice flow that needs to transmit small packets [16]. Small packets can be treated by the Random Early Dropping [17]. One of the major limitations of TFRC is that it is an AIMD (Additive Increase Multiplicative Decrease) mechanism that leads to short term congestion which decreases quality of voice application. Implementation of TFRC with Token Bucket (TFRC-TB) focuses on providing a QoS mechanism for VoIP applications. The use of discrete sending rate and the token bucket strategy resulted in a more stable sending rate that result in a smoother traffic pattern [18-20].

Congestion in a network degrades performance and ultimately affects the quality of service. Two categories are described for congestion control [21]. Core Stateless Fair Queuing (CSFQ) is unable to estimate fairness during large traffic flows like VoIP. Enhanced Core Stateless Fair Queuing improves fairness and efficiency by reducing packet loss and delay [22].

Open Loop: Open loop congestion control regulates the traffic and prevents (avoid) the occurrence of congestion. Such a system does not use feedback and determine session route and resource requirements before session starts.

Closed Loop: Closed loop controls the traffic when congestion occurs and reacts to smooth traffic. Closed loop uses a feedback and it also monitors congestion.

Author worked on fuzzy predictor of bandwidth for differentiated services network (DiffServ network) [23]. It described improving bandwidth allocation to different traffic streams in DiffServ-Aware network. Two traffic classes Expedited Forwarding (EF) and Best Effort (BE) were used. EF class is related to high priority real time internet traffic and BE class is related to low priority non-real time internet traffic. The work is focused on bandwidth sharing between high and low priority traffic using fuzzy predictor.

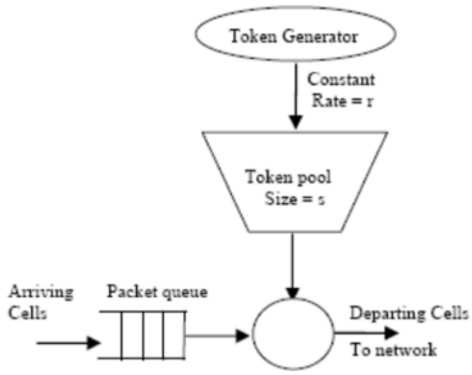

Fig. 1: Token Bucket

Token Bucket Traffic Shaper: Traffic shaping for VoIP is important to maintain its quality it is done to avoid congestion occurrence. Several methods are used to avoid congestion like Leaky Bucket algorithm and Token Bucket algorithm. The only difference between Leaky and Token bucket is the generation of token. Token bucket algorithm works on the mechanism of token generation. Each packet must get a token before going to output stream. If a packet did not receive a token then it will wait for the token.

Token bucket has token bucket to store tokens in it generated by a token generator at a specific interval of time. Arriving packets are placed in queue mostly FIFO. Only those packets are sent to network stream who receives the token. Other packets are discarded or resend [4, 5]. Figure 1 shows a token bucket with token pool size s and token generation rate r.

Token bucket receives the packets and put them in a bucket. It checks for the token available for the incoming packets and sends packets if enough token are there in the bucket, otherwise sends them into queue/buffer for later transmission.

Token Bucket Traffic Shaper (TB Traffic Shaper) uses a dynamic token generation rate module. Token generation rate is adapted according to the incoming traffic and available bandwidth.

Token Bucket architecture is shown in Figure 2. Incoming traffic is placed in a buffer (queue) inside the traffic shaping unit. Voice packets are served as first come first served basis. Token generation rate is set at a specific rate say x. when incoming voice packet rate increases and buffer reaches a specific level say m. Token generation rate changes accordingly to serve the incoming packets. Token generation rate is now some multiple of x.

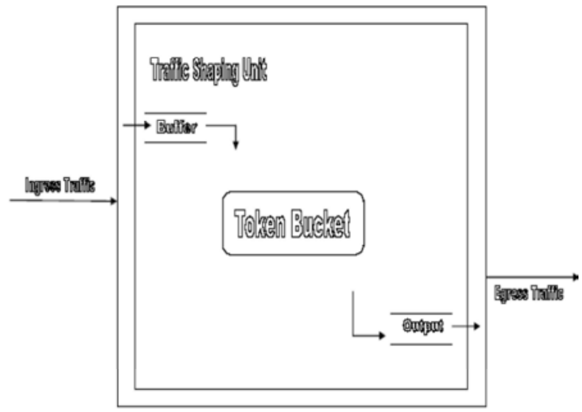

Fig. 2: Token Bucket Architecture

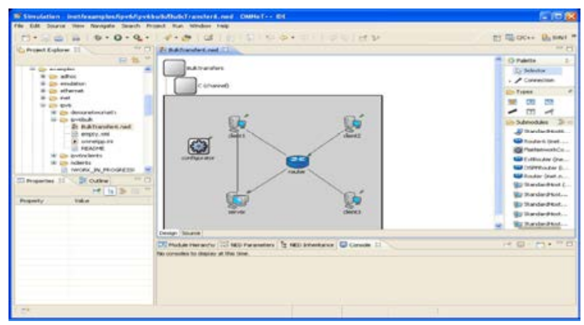

Fig. 3: OMNeT GUI

Simulation Work: Token bucket traffic shaper is implemented in OMNeT Simulator. The GUI of OMNeT Simulator is shown in Figure 3.

OMNeT [24] simulator is a powerful tool for network simulations and research applications. The tool gives the flexibility to create a topology with different nodes and routers. In other words one can design a model according to some topology and link the different modules of the model by explaining the logical instructions in its source files. Design view shows the model graphically, which is helpful in understanding the exact working of logic described in source code.

Token bucket traffic shaper simulation is performed according to the design mentioned in the design unit. Simulation is performed separately for simple token bucket and for fuzzy based token bucket.

In simple token bucket simulation is performed by keeping the token generation rate fixed. It is assumed that available bandwidth is fixed. It is predicted by SIP (session initiation protocol) at the time of connection. Token bucket has a buffer queue. When threshold reaches, incoming packets are discarded. Figure 4 shows simple token bucket start for simulation run.

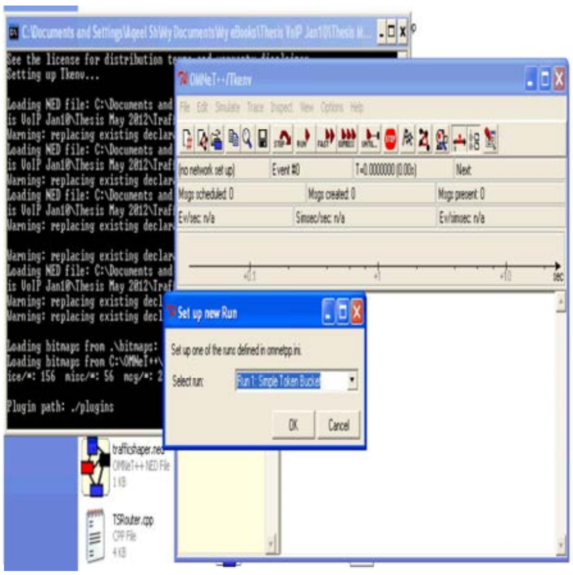

Fig. 4: Simple Token Bucket Start

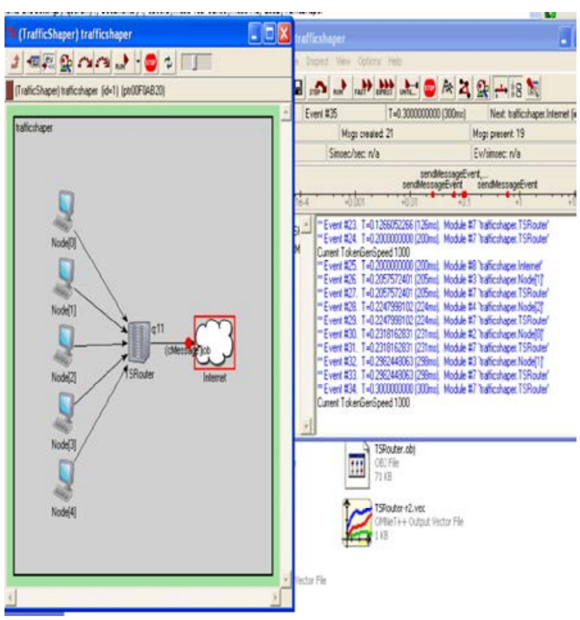

Fig. 5: Simple Token Bucket Run

Figure 5 shows the simulation Run for simple token bucket. Token bucket receives packets by different nodes and stores them in a buffer. Packets are sends to the network as an output as soon as they receive the token. No packet can be send without any token. Packets are discarded when the buffer reaches maximum level.

Simulation shows the event occurrence with time line and it also shows working of different modules with respect to timeline.

RESULTS

Simulation results show that TB Traffic Shaper performs much better as compared to simple token bucket.

In the initial stage of simulation run packet loss is low, but as the time goes on packet loss is higher and it remains higher after certain time as shown in Figure 6.

Simulation Run 2 simulates the token bucket traffic shaper with adoptable token generation rate prediction. Results for token bucket traffic shaper are shown in Figure 7.

There is no traffic loss for traffic shaper as long as the simulation runs. It means that there is almost zero packet loss in traffic shaper. This is because token generation rate is adjusted according to available bandwidth. Every time a packet comes in buffer a token is available to it and it immediately moves to output stream. Packet loss results are shown in Figure 7.

Figure 8 shows the situation of the buffer inside the simple token bucket during different periods of time. This shows that in fix token generation rate packets are always in hunt of token in order to move to output stream. Simple token bucket most of the time has almost full or full buffer space with the packets in it waiting for tokens.

Simple token bucket gives a downward trend for throughput. Throughput decreases reasonably with the passage of time. Throughput trend in simple token bucket with fixed token generation rate is shown in Figure 9.

Traffic shaper buffer condition during simulation run is shown in Figure 10. Token generation rate is adjusted at optimal level which generates more tokens. Consequently packets in the buffer find the token and instantly packets are moved towards the output stream.

Figure 11 shows the token generation rate adjustments with the passage of time. This shows that token generation rate become high when an optimal level reaches.

Output from traffic shaper is shown in Figure 12. Output remains higher for most of the time of simulation. It varies according to the varying situations.

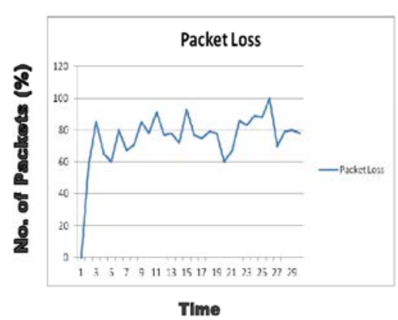

Fig. 6: Packet Loss in Simple Token Bucket

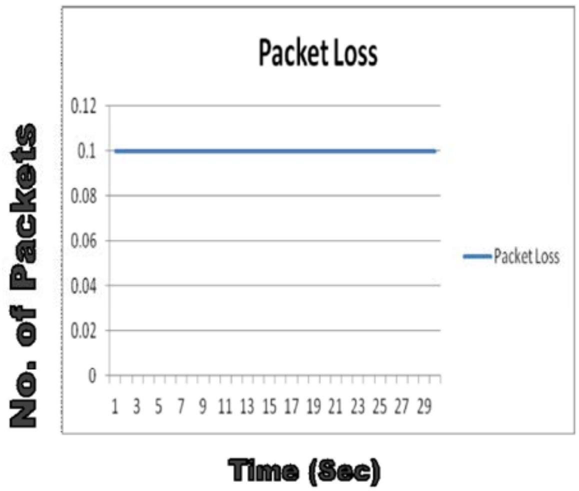

Fig. 7: Packet Loss in Token Bucket Traffic Shaper

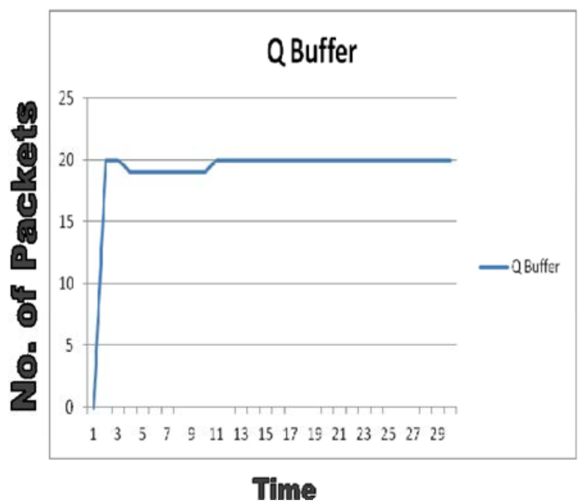

Fig. 8: Simple Token Bucket Buffer

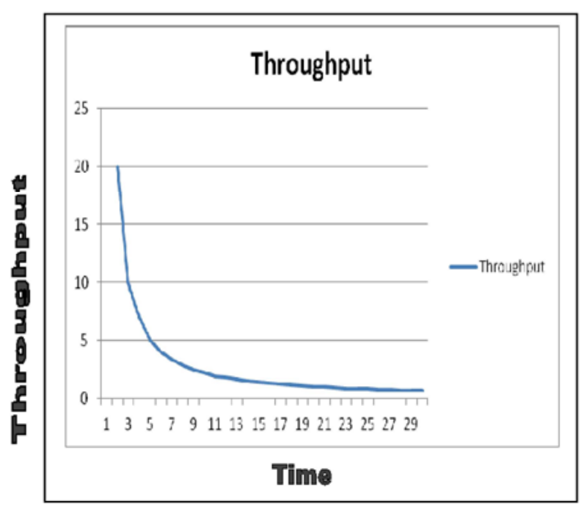

Fig. 9: Simple Token Bucket Throughput

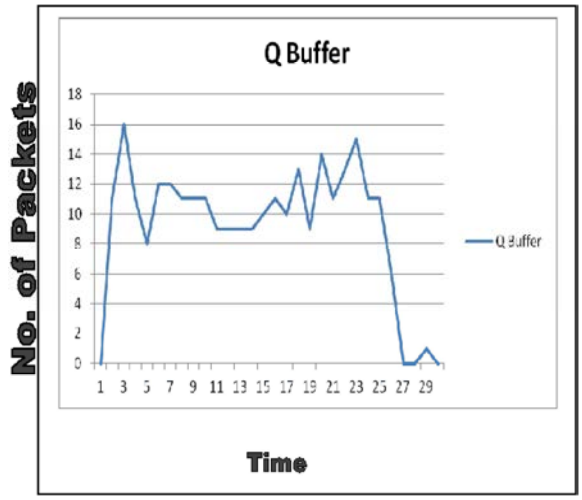

Fig. 10: Token Bucket Traffic Shaper Buffer

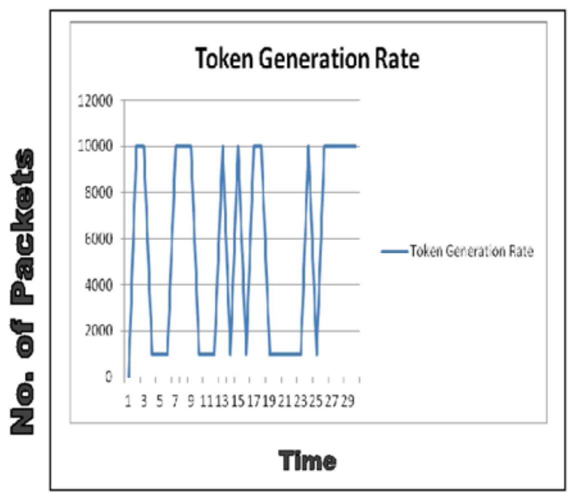

Fig. 11: Token Generation Rate in Token Bucket Traffic Shaper

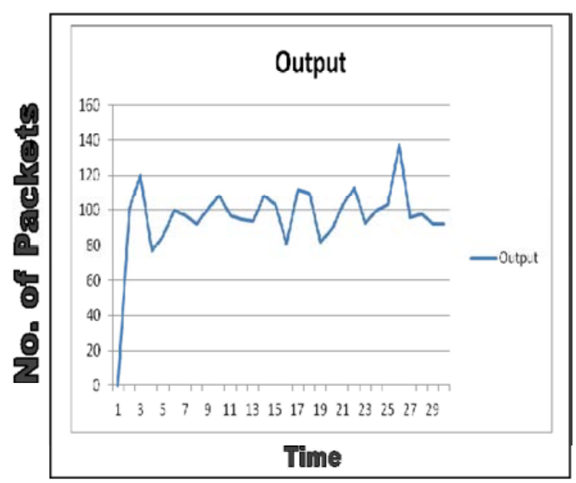

Fig. 12: Token Bucket Traffic Shaper Output

CONCLUSION

The study presented token bucket traffic shaper based on adoptable token generation rate. It is compared with the simple token bucket. Token generation rate is fixed in simple token bucket. Two main goals were set. One is to provide less packet loss and second one is to increase throughput. This ultimately shapes the traffic for voice packets and hence controls the traffic congestion.

Simulation results clearly show that token bucket traffic shaper performs better than the simple token bucket. Simple token bucket works well until threshold limit reaches. When maximum threshold reaches it starts discarding packets. This ultimately leads to higher packet loss and decreases the performance or quality of service. Token bucket traffic shaper performs very well by predicting token generation rate. It reduces the packet loss, reduces delay and increases the throughput. Consequently it shapes the incoming traffic in such a manner that there are very low chances of traffic congestion.

REFERENCES

1. Gospodinov, M., 2004. The affects of different queuing disciplines over FTP, Video and VoIP Performance, International Conference on Computer Systems and Technologies - CompSysTech.
2. Chen, Y., J. Huang, J.F. Hayes and M.M. Ali, 1995. Departure Process Characterization of the Leaky Bucket with Modified Geometric Model, 0-78G33-2766- Sep-1995 IEEE, 971-974.
3. Tinnakornsrisuphap, P., S. Vanichpun and R.J. La, 2003. Dynamic Resource Allocation of GPS Queues with Leaky Buckets. Department of Electrical and Computer Engineering University of Maryland.
4. Ahmed, N.U., Q. Wang and L. Orozco-Barbosa, 2002. Systems approach to modeling the Token Bucket Algorithm in Computer Networks, Mathematical Problems in Engineering, 8(3): 265-279.
5. Tanenbaum, A.S., 1996. Computer Networks, third ed. Prentice Hall PTR Upper Saddle River, New Jersey.
6. Ahmed, N.U., B. Li and L. Orozco-Barbosa, 2005. Modelling and optimization of computer network traffic controllers, Mathematical Problems in Engineering, 6: 617-640.
7. Da Conceic, A.F., J. Liu, D.A. Florencio and F. Kon, 2006. Is IEEE 802.11 ready for VoIP?, 8th International Workshop on Multimedia Signal Processing (IEEE MMSP), Victoria, Canada.
8. Jiang, W., 2003. Assessment of VoIP Service Availability in the Current Internet, Department of Computer Science Columbia University. Available: http://www1.cs.columbia.edu/~wenyu/papers/pam_2003.pdf.
9. Verkasalo, H., 2006. Emerging Trends in the Mobile VoIP Business, Helsinki University of Technology Networking Laboratory, Finland. Available: http://keskus.hut.fi/opetus/s383042/2006/papers_pdf.
10. Abdule1, S.M., W.T. Chee, A. Mustafa and A. Hassan, 2006. Efficient Scheme to Increase the Throughput of VoIP Traffic Over Satellite while Reducing Bandwidth Utilization, Proceedings International Conference on ICT for the Muslim World, ICT4M 2006, Kuala Lumpur, Malaysia, pp: 21-23.
11. Williams, R., Tackling push-to-talk challenges in handsets, E-mail: ray.williams@ttpcom.com.
12. Linden, J., 2003. Finding Harmony between VoIP and WLANs, Global IP Sound, Inc. http://www.commsdesign.com/design_corner/OEG20031105S0010.
13. Semeria, C., 2001. Supporting differentiated service classes: queue-scheduling disciplines, Juniper networks.
14. Mahlo, C., C. Hoene, A. Rostami and A. Wolisz, 2005. Adaptive Coding and Packet Rates for TCP-Friendly VoIP Flows. Technical University of Berlin, Germany.
15. Floyd, S. and K. Fall, 1999. Promoting the use of end-to-end congestion control in the internet, IEEE/ACM Trans. Netw. 7(4): 458-472.
16. Floyd, S., M. Handley, J. Padhye and J. Widmer, 2000. Equation-based congestion control for unicast applications, SIGCOMM, Stockholm, Sweden, pp: 43-56.
17. Floyd, S. and V. Jacobson, 1993. Random Early Detection Gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, 1(4): 397-413
18. Maldonado, M., S.A. Baset and H. Schulzrinne, 2005. TCP-Friendly Rate Control with Token Bucket for VoIP Congestion Control. Department of Computer Science Columbia University.

19. Floyd, S., E. Kohler and M.. Handley, 2005. Datagram congestion control protocol (dccp). Internet draft [Online]. Mar Available: http://www.ietf.org/internet-drafts/draft-ietf-dccp-spec-11.txt.
20. Sjoberg, J., M. Westerlund, A. Lakaniemi and Q. Xie, 2003. Real-time transport protocol (rtp) payload format and file storage format for the adaptive multi-rate (amr) and adaptive multi-rate wideband (amr-wb) audio codecs, RFC 3267.
21. Fulp, E.W., 2008. Congestion Control, Department of Computer Science Wake Forest University, North Carolina. Available: www. cse. wfu. edu/ ~ fulp/csc343/congest.pdf
22. Sivasubramaniam, N. and P. Senniappan, 2014. Enhanced Core Stateless Fair Queuing with Multiple Queue Priority Scheduler, The International Arab Journal of Information Technology, 11(2).
23. Din, N.M. and N. Fisal, 2008. Fuzzy Logic Bandwidth Predition and Policing in a DiffServ-Aware Network, Journal of Computers, Academy Publisher, 3(5): 18-23.
24. http://www.omnetpp.org/component/docman/cat_view/17-downloads/1-omnet-releases.