Virtual Clock Scheduling Algorithm for Video Streaming

Sankar.P and Chellamuthu .C

***Abstract*—Algorithms used for scheduling play a major role in providing quality of service in video streaming. This paper describes a simulation environment that has been designed combining network simulator ns2 with video codec to predict the quality of service. Virtual clock scheduling algorithm has been applied to evaluate the performance metrics and compared with the weighted round robin (WRR) and Deficit round robin (DRR) scheduling policies.**

***Index terms*—DRR, ffmpeg, MPEG4, ns2, Virtual Clock, WRR.**

I. INTRODUCTION

The development of video and multimedia applications enabled the users to handle large amount of visual information contained in video images and also audio data. Efficient delivery of video data over computer networks has been studied for a long time. However, multi-receiver video delivery presents a challenge. Video streaming refers to real-time transmission of video. There are two modes for transmission of video over the Internet. One is the download mode and the other is streaming mode. The former will download the entire video file and then plays back. Hence there is no real-time criticality. In contrast, in the streaming mode, it is mandatory that the video content need to be delivered as packets and played back in real-time as and when they arrive at the destination. The QoS of video streaming demands high bandwidth, low loss rate and stringent timing constraints. The best-effort scheduling algorithm used currently in the Internet does not offer guaranteed quality of service for video streaming.

Chih-Heng Ke et al [1] and Kulan Kao et al [2] have described about the advanced simulation tool to evaluate performance in video transmission. Suman et al [3] has explained a low overhead media streaming system that can be used to deliver streaming data to a large group of receivers. Jin-Cherng Lin et al [4] has applied the scheduling algorithms on WIMAX system on Ns2 to analyze the uplink and downlink directions to get better performances. The problem of secured efficient and scalable video streaming in IEEE 802.11e is discussed by Azfar Moid et al in [5]. Frank et al in [6] has explained how MPEG4 and H.263 encoded video has been used for a large portion of traffic in wired and wireless networks.

In IP networks, traffic management is important to achieve better QoS and high level of network utilization. Several scheduling algorithms have been introduced to have stream-like and loss less traffic to prevent quality degradation. Lofti et al [7] has described a buffer less scheduling algorithm to solve the complex and scalability issues The WRR scheduling algorithm is widely used for high speed switching networks. However it does not meet all QoS requirements. Lixia Zhang [8] has described a different concept namely virtual clock algorithm, which is also used effectively to control the data traffic in high-speed networks. Shreedhar et al has described the DRR algorithm that provides fair queuing of packets at gateways and routers.[9]. In this paper, virtual clock has been considered for scheduling to evaluate its suitability in wired networks for video streaming.

I. SIMULATION ENVIRONMENT FOR VIDEO STREAMING

A simulation environment for studying the network performance is shown in Fig 1. It has basic setup, video codec, and service utilities. The basic set up consists of ns2 simulator and its various components. It consists of Otcl linkage, event scheduler and video agents. The video codec converts the given raw video into a trace file, which is attached to the simulation program written in tool command language (tcl). The service utilities are used for evaluating performance and quality metrics.

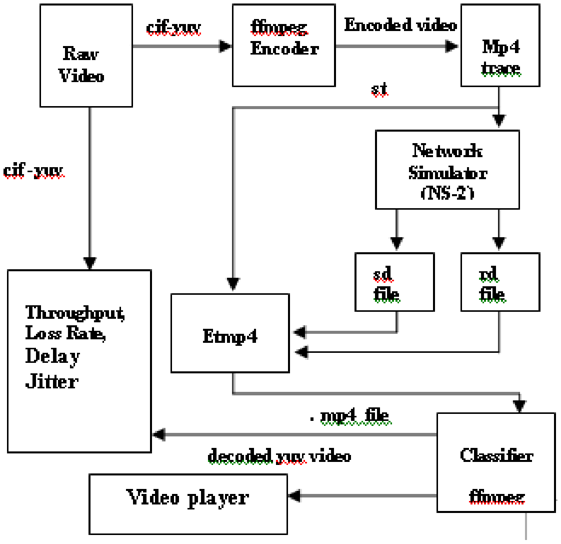


Fig. 1. Simulation environment for Video Streaming

Manuscript received June 29, 2009. The authors acknowledge the support Mr. Venkat Subramanian of Kumaran systems for this research. Also we thank the support of ISRO Sriharikota for useful discussions about the topic

Sankar.P is with the Jerusalem college of engineering, Narayanapuram, Pallikaranai, Chennai-600100, INDIA (phone: +91 044 22461404 ext62; fax:+91 044 22461904; e-mail: sanp_1@yahoo.com).

Chellamuthu.C. was with Anna University, Chennai-600025 INDIA. He is now with the Department of EEE, RMK Engineering College Chennai-INDIA (e-mail: malgudi60@hotmail.com).

A. Basic Setup

The NS2 is an object oriented network simulator. It is written in C++ with front end interpreter called object tool command language (Otcl).Events such as creation, transmission, reception of packets can be simulated. An event comprises a firing time and a function handler. The firing time of the event provides the starting time of the service and function handler selects the appropriate service for the event. There are four schedulers available in the network simulator: linked list, heap, calendar queue and real time scheduler. The scheduler initiates bunch of events one after other and runs it into completion It supports simulation of TCP, UDP, media access control layer protocols and various routing and multicast protocols over both wired and wireless networks.

To setup and run a simulation, a tcl program is written.It is a simulation program to initiate an event handler, set up the network topology and select the functions in the library. The program also tells the traffic sources when to start and stop transmitting the packets through the event scheduler. The Otcl interpreter of NS2 receives this tcl program sets up parameters of the environment specified. Calendar event scheduler is selected for the simulation because it is suitable for video flows. A framework for quality evaluation has been developed by researchers for MPEG4 video traffic over the network [10]. It includes two transport agents and an application/traffic. They are MvsUDP, MvsSink and MvsTrace.

MvsTrace: This agent extracts the type and size of the frame from the video trace file. It splits the video frames into smaller segments and transmits them to the lower UDP layer according to the user specified in the tcl simulation program. This can be invoked by the following command

Set video ($i) [new Application/Traffic/mvs]

MvsUDP: This is an extension of the UDP agent allows the users to specify the output sender trace file and records timestamp, packet id, and payload size of each transmitted packet. The UDP agent is called with the following command:

Set udp ($i) [new Agent/mvsUDP]

MvsSink: This is the receiving agent for the video frame packets sent by mvsUDP. It also records timestamp, packet id, and payload size of each received packet in the user specified file. The null agent is called with the following command:

Set null ($i) [new Agent/mvsSink]

B. Video Codec

The free software tool ‘ffmpeg [11] in the MPEG4 format is chosen as video codec for encoding the test video because it is widely used in the real world and provides better documented methods for comparing video qualities. The ‘ffmpeg’ is a collection of free software that can record, convert and stream digital audio and video frames. It is a command line tool used to convert video files from one formats to another such as DivX, H.264, VCD, DVD and several others. It also supports grabbing and encoding real time video.

C. Service Utilities

The service utilities of Video codec consists of error generator (eg), etmp4, psnr, hist and mos. The tool ‘etmp4’ evaluates the mp4 trace. It generates files for plotting frame loss, delay jitter, send rate, and receive rate. NS2 simulator has several tools such as graphical animator (nam), xgraph and gnuplot.These utilities help in generating and visualizing the simulation results.

II. PACKET SCHEDULING

Packet Scheduling is a mechanism in a router that allocates the video flow for a given output link. A router consists of flowing components; input buffer switching matrix and output buffer. The input buffer receives packet from the incoming link associated to the buffer. The switching matrix connects the output link to one of the input links according to switching strategy. Output buffer stores the packets received from the incoming links and connects to the output link. In this section, Virtual clock algorithm has been described for scheduling the video traffic.

A. Weighted Round Robin (WRR)

WRR is a best-effort connection scheduling discipline.There is multiple first in first out (FIFO) queues. A traffic classification unit assigns an incoming packet to one of the FIFO queues. Each queue in the input buffer has its own weight. Based on the weight the transmission time and selection of the packet is decided by the scheduler.

The WRR scheduler operates in rounds. In each round the scheduler visits each queue in a around robin fashion starting with queue one. During each visit of a queue one or more packets may be serviced. Each queue is given a weight which provides implied bandwidth for the flow in order to give priority to high priority traffic without starving low priority flow.

B. Deficit Round Robin (DRR)

DRR is a modified weighted round robin scheduling with fair queuing.. If the packet size is less than the quantum assumed it is accepted. If the packet size exceeds the quantum then, only quantum size is accepted from connection and a deficit counter is set with a value equal to difference between quantum and packet size. At the end of each Round, if there are any bytes remaining in deficit counter, the amount of bytes is carried over to the next round.

C. Virtual Clock

Virtual Clock (VC) is a traffic control algorithm for high-speed packet switching networks. VC controls the average rate of data transmission and enforces ‘each user’s resource usage according to its reservation, builds firewalls among flows, and supports multi-priority transmissions. The model of virtual clock algorithm is shown in Fig. 2.

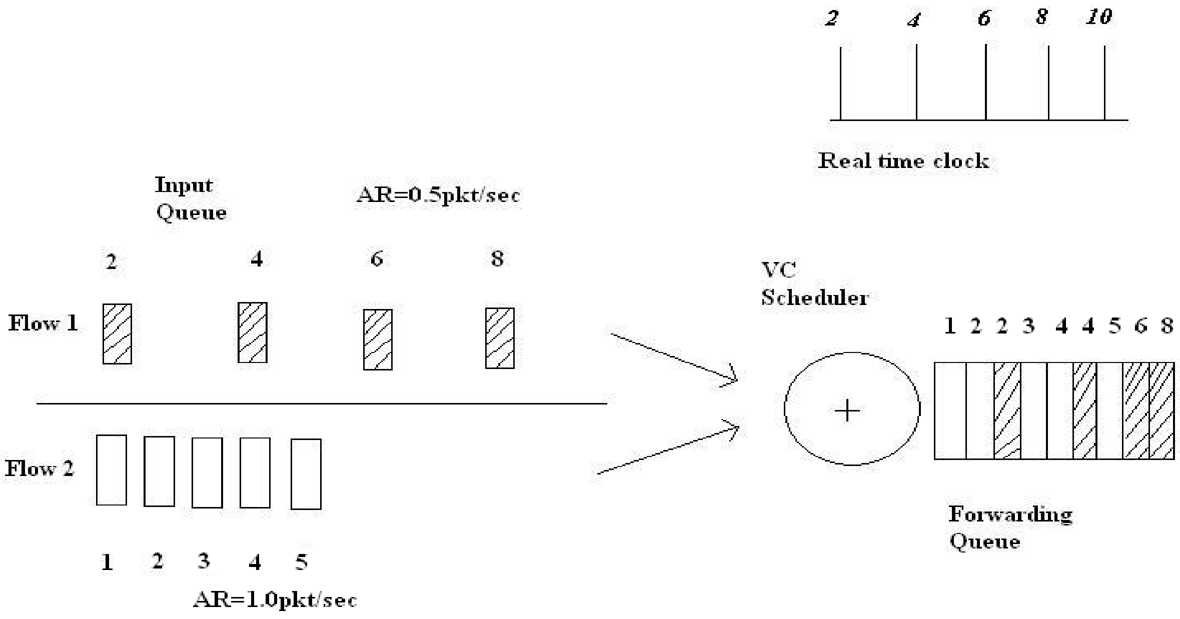


Fig. 2. Model of Virtual clock Algorithm

The algorithm assigns each data flow a virtual clock. Each virtual clock advances one tick at every packet arrival which is the mean packet inter arrival time. If a flow sends packets according to its specified average rate, virtual clock follows real time otherwise, the algorithm stamps the packets with virtual clock values and transmits the packets in the ascending order of their stamps. According to the flows specified transmission rates the virtual clock algorithm determines which packets should be forwarded next.

Choosing flow parameters

The following parameters are considered for each flowi entering a switch in a network:

i=1,2…. n for n flows
ARi, average transmission rate (packets/sec)
IRi, packet inter-arrival time (sec)
AIi, average observation interval (sec).

Choosing the AI value for each flow is very important.
AI = total transmitted data (over AI/AR)
The range of values for AI is: 1/AR <=AI <= total flow duration.

The value for AI should be small enough to give the network sufficient control and also it should be large enough to tolerate variations in packet arrival pattern. For example, if the lower bound value is chosen for AI, the source has to send packet at a constant rate. On the other hand, upper bound value (total transmission time) is assigned to AI, the source can send packet in any arbitrary manner.

Algorithm

A.Data forwarding:-

Step1: When the first packet is received from flowi, VCi and auxVCi are both set to the real time

Step2: Upon receiving each packet from flowi,
Vticki =1/ARi
auxVCi = max (real time, auxVCi)
auxVCi = auxVCi + Vticki
VCi = VCi + Vticki

Step3: Stamp the packet with auxVCi value.
Step4: Insert the packet in its outgoing queue.
Step5: Serve the packets according to their increasing stamp values.

B. Flow monitoring:-

The flow monitoring algorithm calculates a control variable AIRi.

AIRi = ARi × AIi for each flowi (in packets).

Upon receiving AIRi packets from flowi, the following conditions are checked

Condition1: If (VCi - real time) > T (a control threshold), then the flowi is warned.

Condition2: If (VCi < real time), let VCi = real time.

Thus, the VC variable plays the role of a flow meter and it is increased according to the negotiated packet arrival rate of flow. Hence, the difference between a VC of the flow and the real time shows how closely a flow is following its specified rate

By introducing a second parameter called the auxiliary VC, the algorithm prevents flows from accumulating credits. Consider the case when a source sends a burst of packets after remaining idle for a while. In this situation, although the VC value might fall behind the real time, the use of the auxiliary Virtual Clock will cause the packet’s auxVC stamp to be updated with the real time. Thus, the traffic burst will be interleaved with packets from other flows. Therefore, the auxVC is used to order the packets from distinct flows. By serving packets in the order of their auxVC values, the algorithm assures that flows use the bandwidth according to their specified packet arrival rates. Thus, nonconforming flows can use free bandwidth and they cannot affect conforming flows.

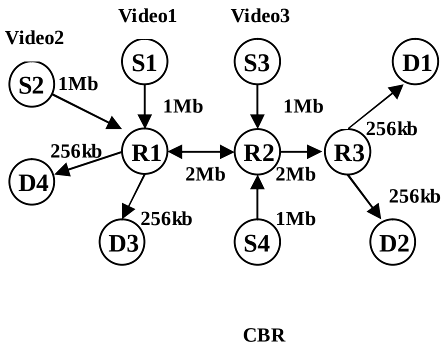
A pseudo code for virtual clock algorithm is as follows

```
Global Initialization:
 NoActiveFlows =0;

Flow Initialization:
(When flowi begins)
Set flow-specific variables VTi; PrevNoActiveFlowsi=0;
auxVCi = real time;
VirtualClocki = real time;
Enqueue: (For every packet from flowi)
if (Queue Length(flowi)=0)
then
PrevNoActiveFlowsi=NoActiveFlows;
NoActiveFlows ++;
if (PrevNoActiveFlowsi>NoActiveFlows)
then
VTi =Vticki;
while (PrevNoActiveFlowsi<NoActiveFlows)
 do
PrevNoActiveFlowsi++;
PrevNoActiveFlowsi = NoActiveFlows;
auxVCi = max (realtime, auxVCi);
VirtualClocki = (VirtualClocki+ VTi);
auxVCi = (auxVCi+VTi);
Stamp packet with auxVCi;

Dequeue: Transmit packets in increasing order of stamps;
```

III. SIMULATION NETWORK

A network interconnecting two networks is simulated for given scenario as shown in Fig. 3. A video traffic in S3 and CBR traffic in S4 are added through router 2 to vary the load in the network.

A raw cif-yuv video file (Video1) with resolution 352 x 288, 30 frames per second, 8 seconds play time and a bit rate of 47.3 kbps is encoded using MPEG4 encoder to generate video trace file of 250 frames. A trace file ‘st’ is created using the mp4 trace utility for the video file and it is attached to a ‘tcl’ program which simulate the given network topology. When the tcl file is made to run, it converts the ‘st’ file to a video data file and writes it with. ‘dat’ extension. It also generates a sender dump (sd) and a receiver dump (rd) files.



S1-S2-R1-D4-D3 → Campus 1
D1-R3-D2→ Campus 2
S3-S4-R2 → intermediate sources and router
CBR → Constant bit rate
Fig. 3. Network topology

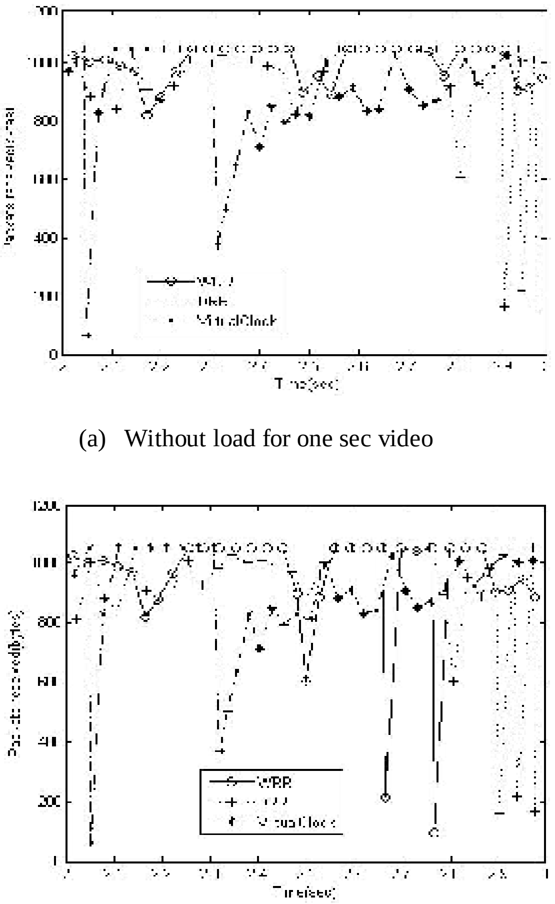
Using ‘etmp4’ tool along with ‘sd’,’rd’ and ‘st’ files the video file is generated and saved with .mp4 extension. This .mp4 is converted to ‘yuv’ using the ffmpeg that can be played on the players like VLC for visualization. The performance of video streaming are evaluated using the utilities of NS2, ffmpeg, matlab and excel

IV. SIMLATION RESULTS

The simulation has been carried out for video 1 with and without network load. The performance parameters throughput, loss rate and delay-jitter are evaluated at the destination D2 of campus2 where video 1 is received.

A. Throughput

A sample plot of packets received in bytes for a simulation period of 1 sec is shown in Fig. 4. Since packets are of varying size, the information content received is measured in bytes per sec. A comparative statement of the actual throughput in bits per sec for the three algorithms is given in Table. I. It is observed that DRR and VC gives a better throughput when compared to WRR.



(a) Without load for one sec video

(b) With full load for one sec video

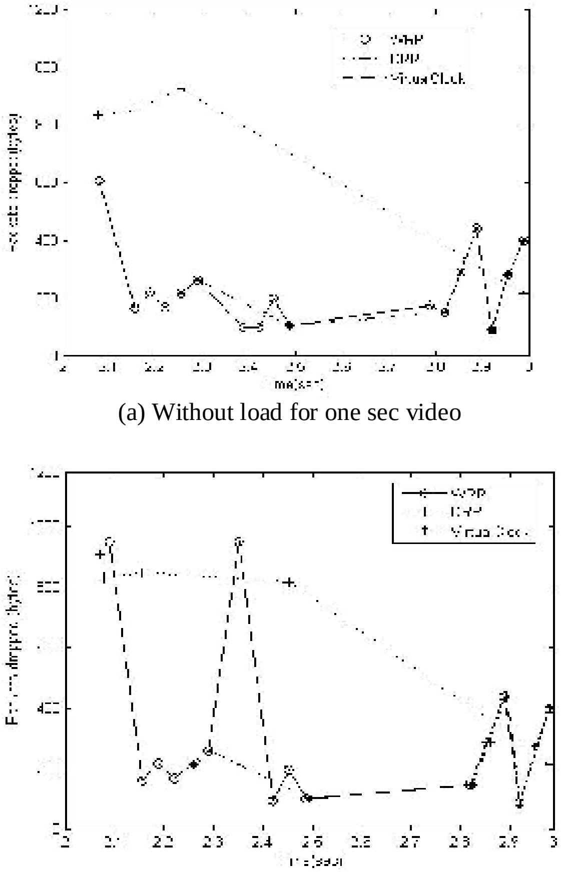
Fig. 4. Throughput

Table. I Full load condition

Algorithm	Throughput (Bits/sec)
Virtual Clock	**2,73368**
DRR	**2,58016**
WRR	**2,46723**

B. Loss Rate

Loss rate is defined as the video information lost per sec in the network due to dropping of packets. It is observed from the result that for the VC algorithm drop rate is lowest in comparison with WRR and DRR It is given in Table. II. The Fig 5 illustrates the drop priority of the all three algorithms in the destination node with and without congested traffic. It is concluded from the result that the dropping starts only when the routers are having insufficient buffer capacity.



(a) Without load for one sec video

(b) With Full load for one sec video

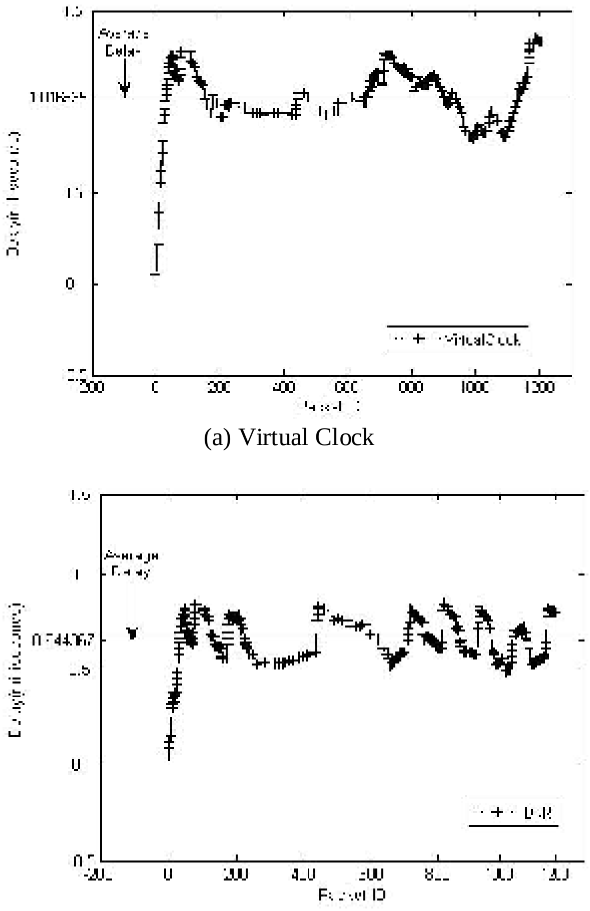
Fig. 5. Loss rate

Table II. Full load condition

Algorithm	Loss rate (Bits/sec)
Virtual Clock	**37853**
DRR	**50399**
WRR	**51185**

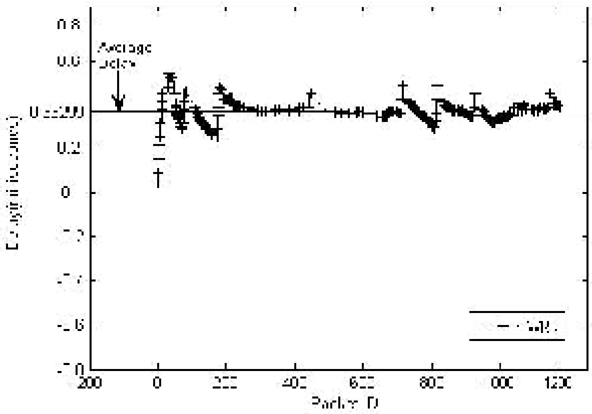
C. Delay Jitter

Delay jitter is a performance measure that shows the delay variation between arrival times of consecutive packets. From the simulated data, it is observed that the transmitted packets have experienced different time delays. The variations in the time delay (jitter) from the average value is plotted and shown in Fig 6 for all three types of algorithms. It is inferred from the graph that VC gives less jitter characteristics due to time stamping as compared to other algorithms. A comparative statement is given in Table III.



(a) Virtual Clock

(b) DRR



(c) WRR
Fig 6. Delay jitter

Table. III Delay Jitter comparison

DRR	WRR	VirtualClock
High	Low	Lowest

V.CONCLUSION

In this paper a simulation model for video streaming has been presented. A virtual clock scheduling algorithm has been applied to predict the performance measures like delay jitter, throughput and loss rate for the video flow. A performance comparison has been made with WRR and DRR. It is observed from the simulation that the virtual clock algorithm is a good choice for video streaming

REFERENCES

[1] Chih-Heng Ke, Cheng-Han Lin, Ce-Kuen Shieh, Wen-Shyang Hwang, "A Novel Realistic simulation tool for video transmission over wireless network", The IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC2006), June, Taichung, Taiwan

[2] Kulan Kao, Chih Heng Ke, Cekven Shieh, "An advanced simulation tool set for video transmission performance evaluation" Proc ACM, WNS2, Pisa, Italy, 2006, Oct10.

[3] Suman Banerjee, Seungjoon Lee, Ryan Braud, Bobby Bhattacharjee & Aravind Srinivasan, "Scalable Resilient Media Streaming", In NOSSDAV'04, 2004 ,June 16-18.

[4] Jin-Cherng Lin, Chun-Lun Chou and Cheng "Performance evaluation for scheduling algorithms in Wimax Network "Proceedings of IEEE Computer Society,2008, pp 68-74.

[5] Azfar Moid, Abraham O.Fapojuwo,Robert J Davies "Secure and Scalable video streaming over IEEE 802.11 e based home networks" Proceedings of IEEE, 2007, pp 445-448.

[6] Frank H.P. Fitzek, and Martin Reisslein,"MPEG-4 and H.263 video traces for network performance evaluation", IEEE Network Vol. 15, No. 6, 2004, pp 40-54.

[7] Lotfi Mhamdi and Mounir Hamdi," Practical Scheduling Algorithms for High- Performance Packet Switches", IEEE transaction on Comp Science, 2003, pp1659-1663.

[8] Lixia Zhang, "Virtual Clock: A new traffic control algorithm for packet switched networks,", ACM transactions on engineering, computing and technology ,1997, pp 19-29.

[9] M.Shreedhar and George Varghese,"Efficient Fair queuing using Deficit Round Robin" IEEE journal on selected areas of communication, November, 1994.

[10] Jirka Klaue, Berthold Rathke, and Adam Wolisz: "EvalVid - A Framework for video transmission and quality evaluation", proc. of 13th International conference on modeling techniques and tools for computer performance evaluation, September, 2003.

[11] http://ffmpeg.sourceforge.net/index.php

[12] http://www.isi.edu/nsnam/ns/

[13] Kevin Fall and Kannan Varadhan, 2007 the NS Manual,

[14] http://www.tkn.tu-berlin.de/research/evalvid/

Sankar.P received the master degree in engineering in communication systems from Madurai Kamaraj University, India in 1997, bachelor of engineering in electronics and telecommunication from University of Poona in 1994, bachelor of science in mathematics from University of Kerala, India in 1986.Currently he is working as asst professor in Department of electronics and communication in Jerusalem college of engineering and also a research scholar in Anna University, India. His areas of interests are Multimedia information systems, Image and video processing and Computer networks.

C.Chellamuthu received his PhD degree in Power Electronics from IIT Chennai, India. After working as associate lecturer in GCT, Coimbatore India; he joined the Anna University Chennai, India as lecturer, asst professor and then professor. He is former head of department of electrical and electronics engineering, Anna University, India. He is currently Professor at Jerusalem College of engineering, Chennai, India. He has published his works in several International and national journals, pioneered several new research directions, made a number of landmark contributions in his field. His areas of interests are Power electronics, PMBDLC designs, Wind energy; Microprocessor based system design, Image and video processing and Controller area networks.