International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 32 • 2017

Application Aware Scheduling over WLAN

Vimlesh Kumar[a] and Anurag Singh Baghel[b]

[a-b]School of Information and Communication Technology, Gautam Buddha University, Yamuna Expressway, Greater Noida – 201310, Gautam Budh Nagar, India. Email: [a]vimlesh@gbu.ac.in; [b]asb@gbu.ac.in

Abstract: IEEE 802.11 WLAN has become a necessity in wireless communication and is widely being used in intra-campus scenarios. An access point is used to handle the traffic in WLAN network. To manage the traffic efficiently, access point implements one of the scheduling mechanisms such as FIFO, PQ, FQ etc. that handle how packets are buffered while waiting to be transmitted. Due to the complex scheduling mechanisms applied at different network devices in the whole network, packets may suffer long delay, large packet delay variation and high packet loss. This becomes a problem when time sensitive applications VoIP, video streaming run on the network. In the present work, a non-complex scheduling process based on application is proposed to improve delay and delay variation of a time sensitive packet in the network and so to improve network performance. The proposed application aware scheduling process is applied at access point and network performance is evaluated by simulation using ns2.35 network simulator. Experimental results and post execution analysis show significant improvement in time related parameters delay and delay variation.

Keywords: WLAN, delay, PDV, scheduling, VoIP, video, ns2.35, simulator.

1. INTRODUCTION

WLAN based on IEEE 802.11 standard is a necessity now days as broadband wireless internet access - at home, at offices and in any limited boundary areas. With the increasing number of internet users, demands for more and more applications are rising. According to Cisco, there is a growth in volume of all applications as well as considerable variation in traffic mix in the network [1]. A network is a series of nodes interconnected by communication paths, wired or wireless where the applications are running on some of these nodes using these paths. So, basic requirement of any network is to run these applications without any congestion and delay. The performance of any network is analyzed by total number of packets transmitted per unit time (throughput), propagation delay (latency), delay variation and packet loss [2]. Among these, for multimedia applications such as VoIP, video conferencing and online gaming, which have high percentage share in the traffic as compare to P2P file sharing. Now a days, delay and delay variation for real time application are the main concern [3].

A wireless network is a large set of network packets, to carry information from one host to logically connect to another user. At any given moment, transmission of a packet is being processed by a network device, like

AP (Access Point), switch or router. Each device executes one of the scheduling algorithms (FIFO, PQ etc.) that manages packets in queues, waiting for transmission [4, 5] (Fig. 1). The main cause of delay in received packets is due to one of the scheduling mechanism running over each network device other than propagation time of packet in the network.

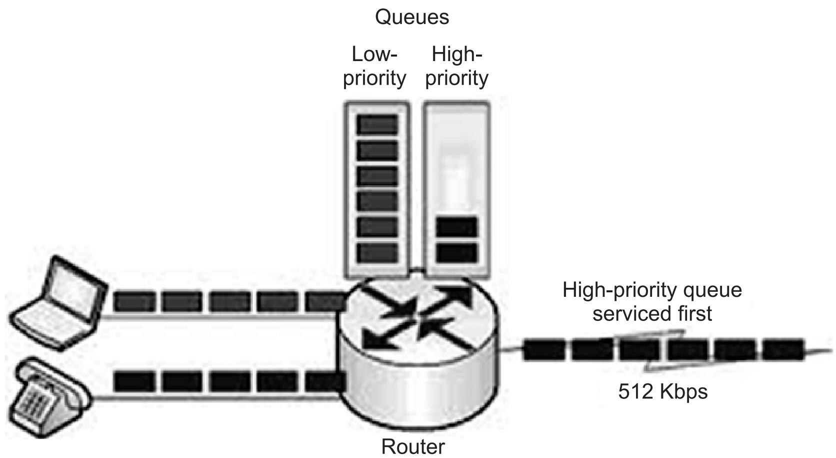

Figure 1: Scheduling of Packets

To improve delay and delay variations for any particular application from the mixed traffic, we propose a simple scheduling process. The proposed objective is to minimize time sensitive parameters of real time application to prioritize the application itself, while do not complex the scheduling process and so, to improve overall network performance.

In this paper, performance of application aware scheduling (AAS) on different traffic are presented using ns2.35 simulator. The complete scheduling process is executed on access point in WLAN.

The paper is organized as - section II has different scheduling mechanisms to improve time sensitive parameters and their pros and cons and shows related work. Section III introduces the process flow chart of proposed scheduling process whereas section IV and V show network design, configurations and results respectively. Finally, conclusion is drawn using post execution analysis in section VI.

2. PACKET SCHEDULING TECHNIQUES AND RELATED WORK

Packet scheduling is required when congestion is experienced. If the link is not congested, then there is no need to queue packets. In the absence of congestion, all packets are delivered directly to the interface [6]. Network devices handle an overflow of arriving traffic by different scheduling techniques, and then determine some method of prioritizing it onto an output link if required. Some of the scheduling mechanisms are DropTail (FIFO), Priority Queue (PQ), Fair Queue (FQ),Weighted Fair Queue (WFQ), Class Based Queues (CBQ), Deficit Round Robin (DRR), Deficit Transmission Time (DTT), Low Latency Queue (LLQ), Low Latency and Efficient Packet Scheduling (LLEPS), Credit Based-SCFQ (CB-SCFQ), Controlled Access Phase Scheduling (CAPS), Queue size Prediction - Computation of Additional Transmission (QP-CAT), Temporally-Weight Fair Queue (T-WFQ), Contention-Aware Temporally Fair Scheduling (CATS), and Decentralized-CATS (D-CATS). According to Nisar et. al. most of the scheduling mechanisms are for wired links and applied without any decision on managing traffic flow on real time traffic such as VoIP [7].

For time sensitive traffic, schedulers need efficient bandwidth and importantly minimized delay for packets during transmission. Most of the schedulers first classify packets and then manage the multiple flows with bandwidth and QoS guarantee but they do not fill the requirement of minimized delay and constant and controlled delay variation. To reduce delay variations in time sensitive applications in IP networks, two queuing mechanisms are used – PQ-CBWFQ and PQ-WFQ. Both belong to Low Latency Queuing (LLQ) [8].

Weighted Fair Queuing differentiates packets in different queue and assigns weight to provide priority. CBQ provides different classes of mixed traffic and fair link sharing but on high congestion it becomes the cause of delay on the network due to the used buffers. These algorithms and their hybrids LLQs do not assume the delay of packets in their execution and thus do not fulfill the need of constant and low delay on WLANs [9]. Further, use of de-jitter buffers in hybrid queues – LLQs require some extra output memory and even in case of large delay variations possibility of packet loss occurs.

2.1. Application Aware Scheduling (AAS)

AAS is proposed for WLAN 802.11 scenarios, in which data is transported through AP between servers and hosts. Sometimes, due to heavy traffic flow or congestive blocks it may not be possible to run an important application on the hosts. For example, in a university campus a lecture is organized on WSN and it is scheduled over video-conference for the students having wi-fi connection in their rooms. Network engineer do not want to block all other traffic. Application Aware Scheduler will keep prior that application without majorly blocking other applications without any complex process.

(a) **Architecture:** Architecture of app aware scheduler is shown in Figure 2. In simple FIFO queues appX and genX are generated with the help of classifier. Enqueued traffic is divided into two parts. The application specific packets are forwarded to appX and rest in genX through scheduler.

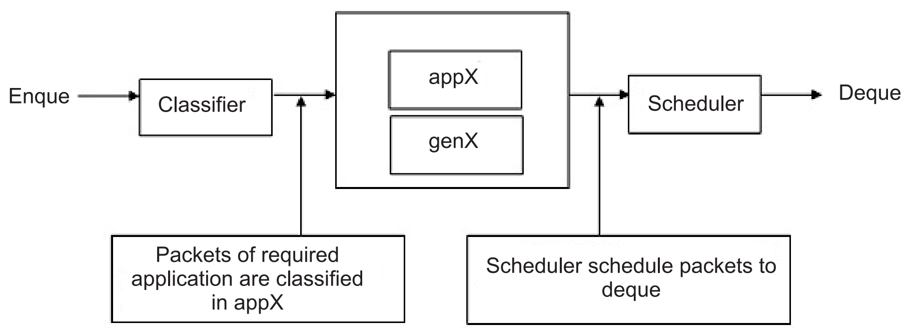

Figure 2: Architecture for Application Aware Scheduling

Now, scheduler fetches the packets from queues to transmit further. It fetches packets in a controlled manner according to the algorithm applied.

(b) **Process Flow Diagram:** The process flow diagram of the application aware scheduling is shown in Figure 3. During congestion when traffic need to be buffered in queue, scheduler first classifies packets of different traffic into two FIFO queue appX and genX. Number of packets in appX is monitored. A threshold range, *mnthrshld* to *mxthrshld* is set for number of packets in appX. The default mechanism for scheduler is set to round robin for both appX and genX.

If number of packets is within the range, scheduler gives preference to appX packets. It transmits those packets from appX first to deque and it keeps monitoring number of packets in appX. If number of packets exceeds the maximum threshold, scheduler schedules *mxthrshld* packets from appX to transmit further. If the number of packets does not lie within the range scheduler runs on round robin fashion for both the queues.

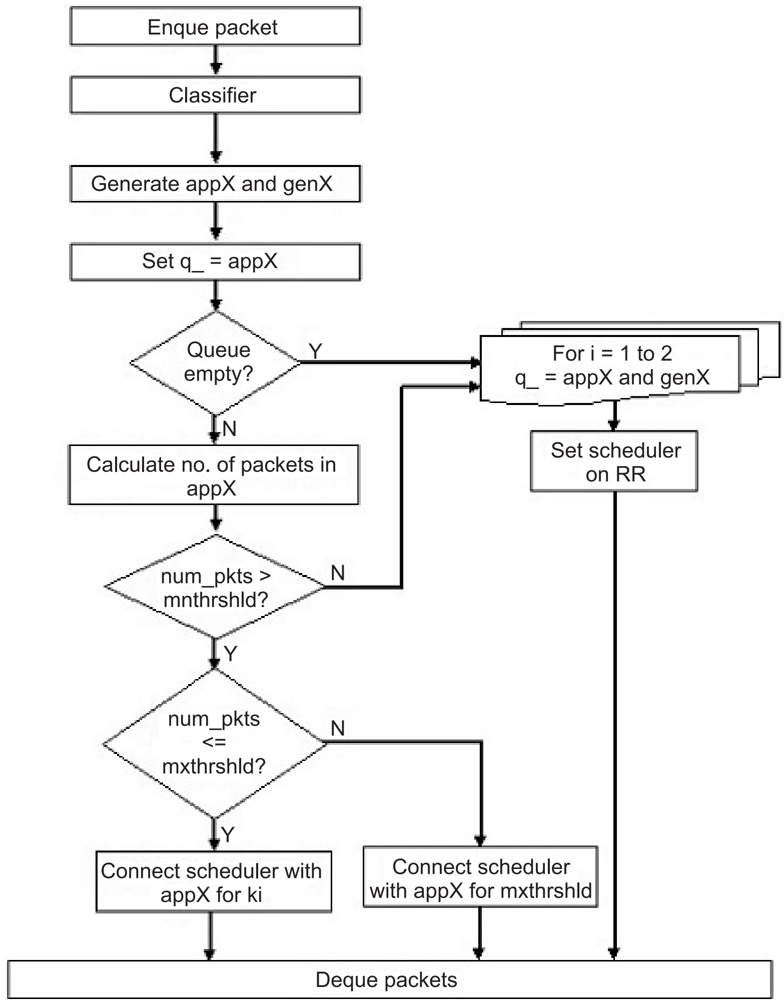

Figure 3: Process Flow Chart for App Aware Scheduling

3. NETWORK DESIGN AND EXPERIMENT SCENARIO

The present work presents simulation studies of WLAN in Network Simulator (ns2.35). In a WLAN, infrastructure mode, scenario is considered with severs, AP and hosts as illustrated in Figure 4.

AP is connected with servers in structured mode whereas nodes are wirelessly connected and movable within a building, though mobility is not considered in the analysis. Data traffic travels from servers to nodes through AP at the time of download and data from nodes to servers at the time of upload. The scenario of figure 5 is implemented using NS2 simulator.

A new uni-directional link as application aware scheduler is created between node-0 and node-4. To make this link bi-directional, simple link is created from node-4 to node-0. In ns2.35, buffer management, propagation delay calculation and callback mechanism are modeled at links not at nodes [10]. Figure 6 shows the composition of link, its components and architecture implemented in ns2.35.

The different simulation parameters are as per the Table 1. Traffic load – FTP, VoIP and Video are considered in the simulation experiments. Congestion is created at AP. Traffic is generated through agents and applications in ns2.35; VoIP is generated through UDP-CBR and video traffic with UDP-pareto distribution. Performances of scheduling technique have been applied and evaluated at the link between AP and network device in download state.

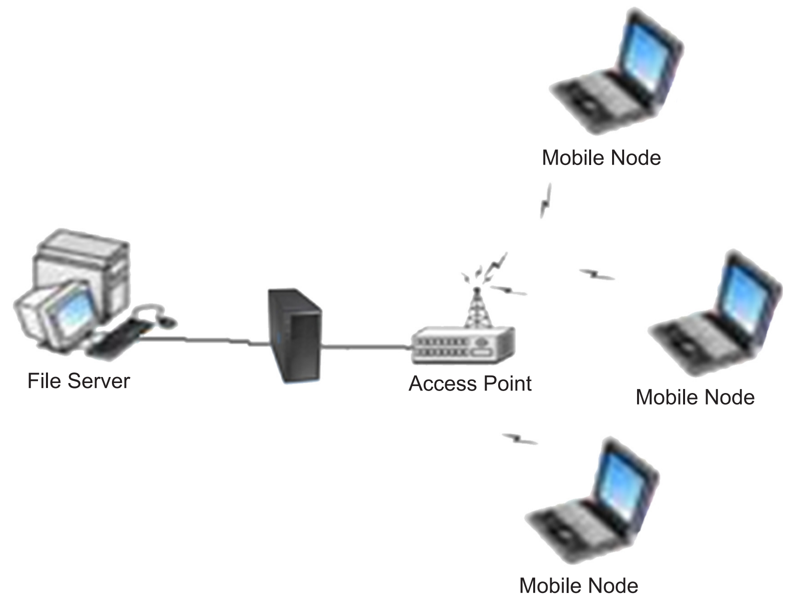

Figure 4: Network Scenario

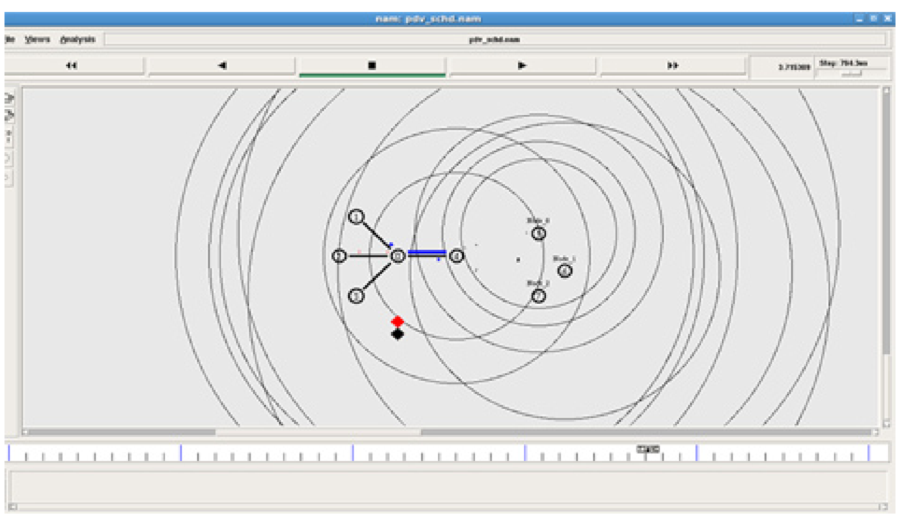

Figure 5: Scenario implemented in ns2.35

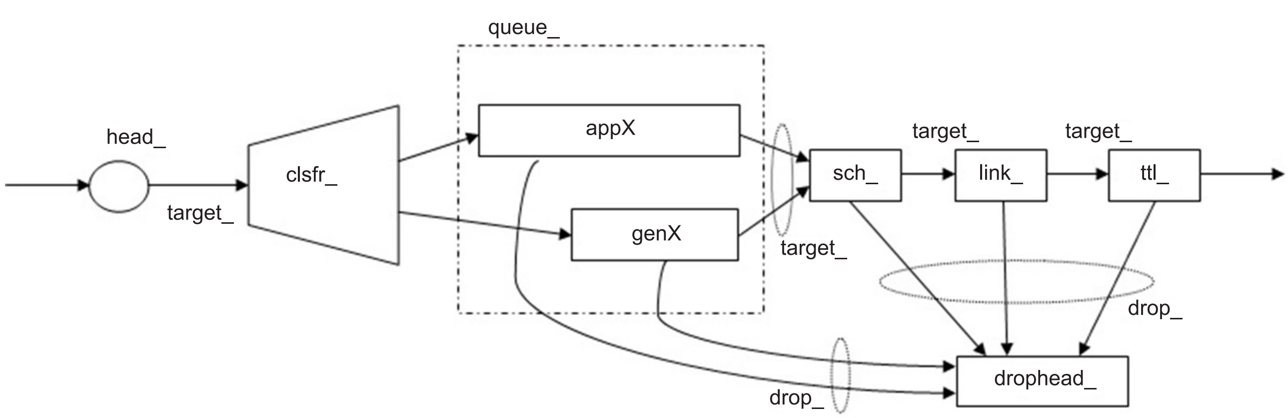

Figure 6: Architecture of app aware link created in ns2.35

Table 1
Simulation Parameters

Parameters	*Values*
WLAN Mode	Infrastructure
Channel	Wireless-phy
Radio Propagation Model	Two Ray Ground
Frequency	2.472 GHz
Band	ISM
PHY Modulation	DSSS
Simulation Time	5 Sec
Traffic	UDP, TCP
Application	FTP, VoIP, Video
RTS/CTS	On
Distributions Involved	CBR, Pareto
Routing	DSDV
Mobility of Nodes	Off

4. RESULTS AND ANALYSIS

The results of application aware scheduling in Figure 7 which is obtained from post-execution analysis of trace files. These files are obtained after executing Tcl scripts in ns2.35 for proposed appication aware scheduler. Data is communicated between servers and hosts and result is carried out in download state. Due to congestion at link with AP, some packets have dropped due to factors like scheduling technique, bandwidth, queue length etc.

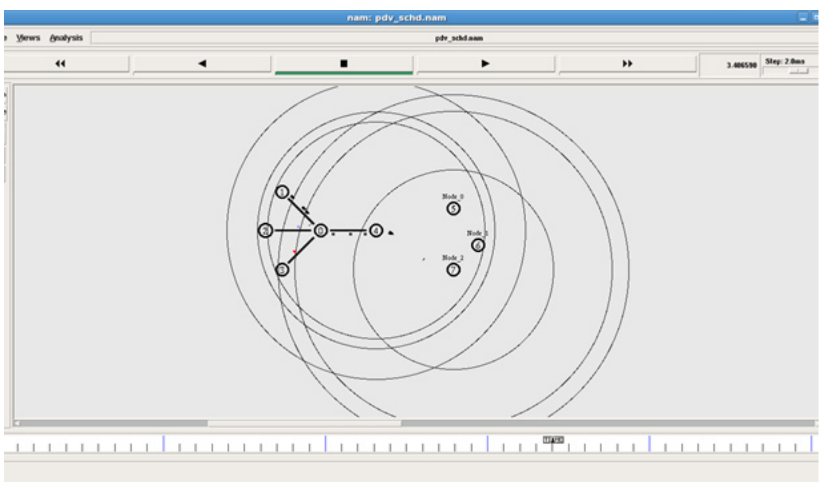

Figure 7: Nam output results for App Aware Scheduler

Performance parameters for traffic packets delay and delay variation is calculated.

Delay

Xgraph utility in ns2.35 shows the simulation result in Figure 8 for packet end to end delay. Packet delay is calculated as the time experienced by the packet when it leaves the source node and reaches at the destination node. The variations in delays like transmission delay, queuing delay etc. experienced by the packets will be covered in end to end delay itself. Post execution analysis is done with the help of trace file output and awk scripting.

Average delay is calculated as follow:

$$E\,[\text{delay}] = \Sigma \text{ all delay samples } / \text{Number of samples} \tag{1}$$

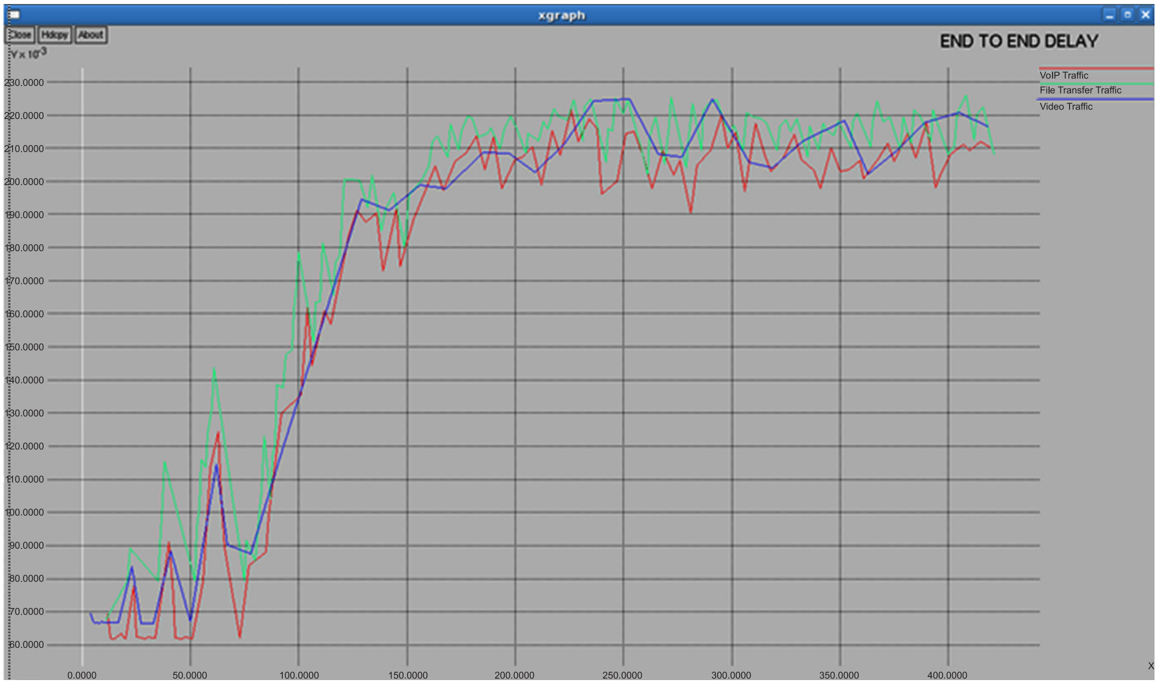

Figure 8: Packet Delay for FTP, VoIP and Video Traffic in App Aware Scheduler

Delay Variation

Packet delay variation is the time difference between the packets arrival time at destination. It can be constant or variable depend on behavior of network. If end to end delay of *i*th packet is δ_i and *j*th packet is $\delta_j (i < j)$, then P_{dv} can be calculated as –

$$P_{dv} = \delta_j - \delta_i / j - i \qquad (2)$$

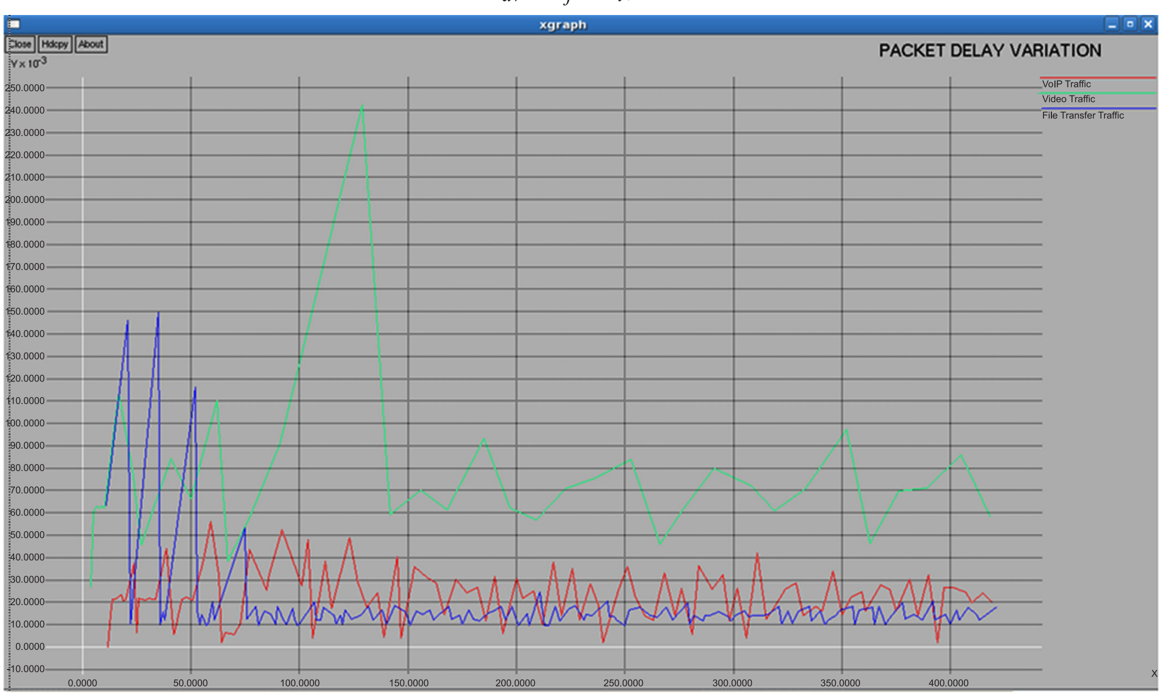

Figure 9: Packet Delay Variation for VoIP, Video and FTP in App Aware Scheduling

The performance parameters in different existing scheduling mechanisms (FIFO, SFQ) and proposed scheduling mechanism (AAS) are analyzed. The post-execution analysis results show satisfactorily improvement in time parameters as shown in Table 2, when AAS is applied on the link, end-to-end delay for 100 data packets of VoIP and FTP is clearly improved as compared to other scheduling techniques. It is important to note that delay is more for video traffic individually but AAS reduces overall delay of the network. It does not blocking the other traffic while application specific packets are prioritized.

Table 2
Average End to End delay (sec.)

Scheduling Technique	*VoIP*	*Video*	*FTP*
FIFO	0.5891	1.2464	1.0765
SFQ	0.3994	0.2995	0.9129
AAS	0.1541	0.3894	0.1315

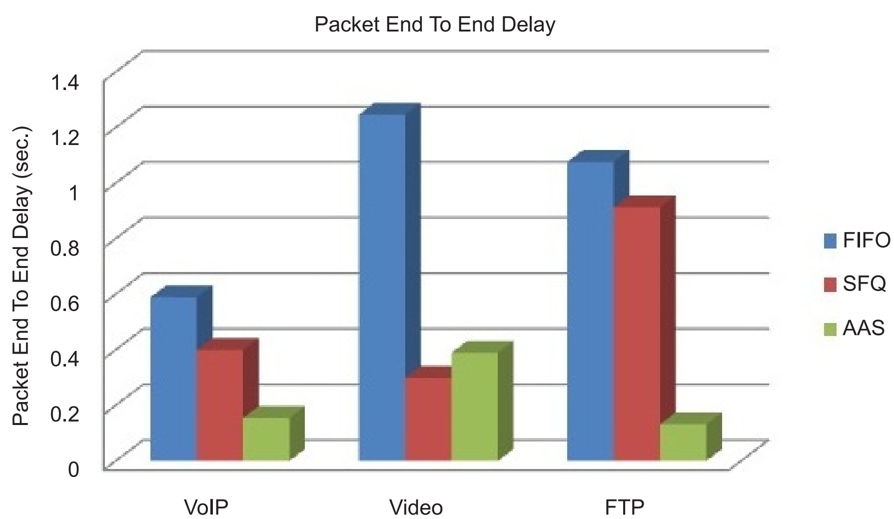

Figure 10: Comparison of Packet Delay in scheduling mechanisms

Now average PDV (packet delay variation) for 100 packets after implementation of AAS for VoIP and FTP traffic is minimized successfully as shown in Table 3. Minimized delay for VoIP and FTP clearly validate that scheduling of packets is balanced. In case of video traffic, SFQ mechanism shows low PDV due to the guaranteed and prioritized scheduling for video traffic but also SFQ increases PDV in FTP traffic significantly. This indicates that packet scheduling is not balanced in SFQ.

Table 3
Average PDV (sec.)

Scheduling Technique	*VoIP*	*Video*	*FTP*
FIFO	0.0433	0.3500	0.1123
SFQ	0.0328	0.1765	0.1239
AAS	0.0215	0.1869	0.0120

5. CONCLUSION

A new scheduling process is proposed as Application Aware Scheduling (AAS) in which packets of required application is scheduled on controlled round robin scheduling such that packets of that particular application is

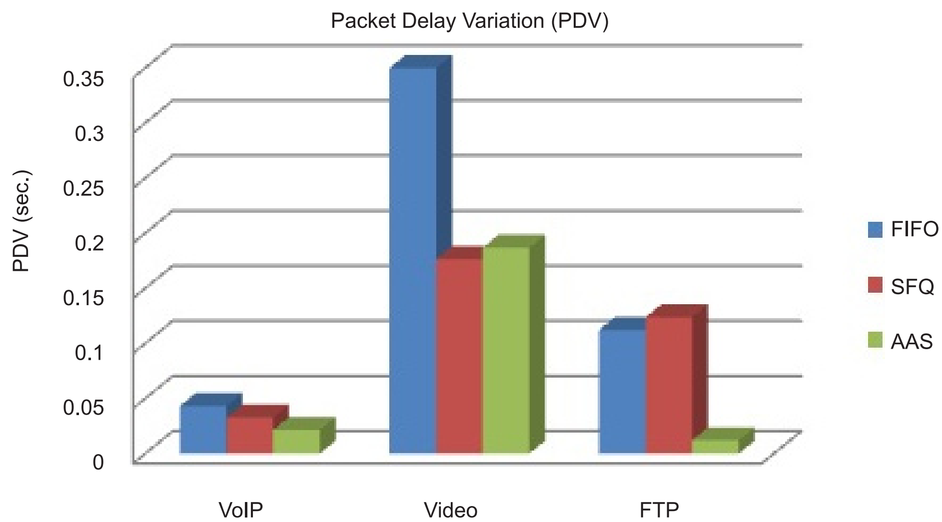

Figure 11: Comparison of PDV in scheduling mechanisms

prioritized but no other traffic is blocked. A new module (.cc, .h and OTcl files) for AAS process is developed and added in ns2.35 locally. AAS link is created and using this new link topology for WLAN is designed. Finally the performance of AAS has been evaluated and compared with existing scheduling techniques. It has been concluded that AAS performance is better than FIFO and SFQ scheduling techniques for various performance metrics.

REFERENCES

[1] Cisco Systems, "Cisco Visual Networking Index – Forecast and Methodology, 2007-2012," White Paper, June 2008

[2] Kashif Nisar, Angela Amphawana, Suhaidi Hassan, Nurul I., Sarkar Sintok Kedah, "A comprehensive survey on scheduler for VoIP over WLAN", Journal of Network and Computer Applications Vol. 36 pp. 933–948, 2013.

[3] Rastin Pries, Florian Wamser, Dirk Staehle, Klaus Heck, Phuoc Tran-Gia, "Traffic Measurement and Analysis of a Broadband Wireless Internet Access", IEEE 2009.

[4] RFC2475, Architecture for Differentiated Services, 1998.

[5] Article, link: http://www.biztechmagazine.com/article/2007/09/.

[6] S. Shah, W. Ali, I. Ahmad, S. Jabbar, A. Minhas, "Performance analysis of queuing mechanism in voice applications", Proceedings of World Congress on Engineering, July 3-5, 2013.

[7] Kashif Nisar, Angela Amphawana, Suhaidi Hassan, Nurul I., Sarkar, "A comprehensive survey on scheduler for VoIP over WLAN", Journal of Network and Computer Applications Vol. 36 pp. 933–948, 2013.

[8] Shahid Mumtaz, Jonathan Rodriquez, "Green Communication in 4G Wireless Systems", River Publishers, 2013.

[9] Striegel A, Manimaran G, "Dynamic class based queue management for scalable media servers", Journal of Systems and Software (USA) Vol. 3 pp. 1-17, 2003.

[10] The *ns* Manual, November 4, 2011.

[11] NS-2, link: http://www.isi.edu/nsnam/ns/tutorial/, November 2012.