Priority-based Packet Scheduling in Internet Protocol Television

Mehmet Deniz Demirci
Computer Science Department
Istanbul University
İstanbul, Turkey
e-mail:demircid@istanbul.edu.tr

Abdul Halim Zaim
Institute of Science and Engineering
Istanbul Commerce University
Istanbul, Turkey
e-mail:azaim@iticu.edu.tr

***Abstract*—Techniques to provide the quality of service for policing and scheduling IPTV are discussed (Weighted Fair Queuing and Alpha-Beta Virtual Clock). Also, a new scheduling algorithm is proposed for prioritized services over IPTV. Simulation results for the proposed algorithm is presented. In the proposed algorithm, the amount of packets that belongs each of the five priority classes is broadcast to every switcher. A series of estimated values is obtained by calculating the status of the amount of received packets. Then, these estimation values are used to calculate the credit of each packet received. Switcher selects the appropriate packet to forward to the next switcher by its credit value. The results are discussed as epilogue with objective comments.**

Keywords-IPTV; Priority; Scheduling; QoS; Class of Service; Priority Class

I. INTRODUCTION

Internet Protocol Television known as IPTV consists of several services that provide triple play entertainment. IPTV has 3 different services like television broadcasting, voice over IP [1] (Internet Protocol; RFC791 [2]) and data services bundled together to the subscribers. Providing all of these services at the same time to the subscribers is still a big challenge for researchers all around the world.

In IPTV, there have to be more than one types of packets. If one tries to transmit a real-time live soccer match with normal best-effort traffic then the packets belonging to the match's flow will suffer severe packet loss, delay and jitter.

Packet loss, delay and jitter are the most important aspects of Quality of Service (QoS). That influence the performance and Quality of Experience (QoE) received by the subscriber.

In multimedia streams, packet loss between 1% to 20%, and end-to-end delays until 100 ms are optimum but delays between 100 ms to even 1000 ms (with additional set top box buffering) are acceptable [3]. So, that traffic scheduling for IPTV must respect the difference of class for all kinds of triple-play packets.

The main challenge in IPTV implementation is distributing fairly the resources to each class of service. The Internet, is a best effort service which is neutral with respect to different services. The goal of this work is adjusting the flow of multi-serviced packets in terms of QoS.

Even though IP is a best effort service, the end-to-end delay and jitter are reduced using the proposed algorithm.

In Section 2, how the services are classified is presented. In Section 3, the proposed algorithm is compared to other work. In Section 4, the details of the proposed algorithm are given. In Section 5, the results, and in Section 6, the conclusion, are shown, respectively.

II. CLASSIFICATION OF SERVICES

IPTV packets can be grouped into five priority classes shown as in Figure 1. Priority class 1 is the real time video broadcasting, i.e. a live Champions League Match. This is the most important class of service because a live stream never stops and every seconds count. Priority class 2 is video on demand (VoD) an IPTV class of service which means, a multi million dollar budget new movie is rented by the user and sometimes user stops or pauses the movie even rewind or forward. Voice over IP applications such as internet telephony, belong to Priority Class 3. Best effort services such as web surfing, e-mail or ftp are members of the priority class 4. The last one is the priority class 5 that possess the signalization data. The importance of the class of services is given in descending order.

 ISBN: 978-1-61208-174-8

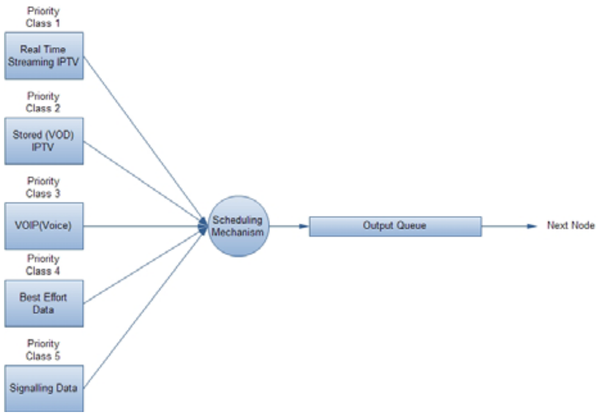

Figure 1. Priority Based Classified Scheduling Node Structure

III. RELATED WORK

The proposed algorithm is compared with two well-known algorithms which are Weighted Fair Queueing (WFQ) [4] and Alpha-Beta Virtual Clock (ABVC) [5].

In WFQ, incoming packets are classified by their type of service and placed into the appropriate queue. The packets are serviced by the node's scheduler in circular form. First of all, the first queue is serviced, then second, and so on. After serving the queue with the least priority, cycle restarts with the highest priority queue. During the cycle, an empty queue is skipped. WFQ is a little bit different than Round Robin [3] because every queue has a weight symbolized as w_i. Thanks to the weights assigned to each queue it guarantees (1)

$$w_i / \sum w_j \quad (1)$$

of the total bandwidth. Thus every time, for the transmission rate R, the class i has (2)

$$Rw_i / \sum w_j \quad (2)$$

of guaranteed rate. WFQ is depicted in Figure 2 [3].

The other algorithm, Alpha-Beta Virtual Clock (ABVC), is an enhanced version of Zhang's Virtual Clock [6]. ABVC forward packets to output queue of the node by inspecting which flow is sending packets and which not.

Initially, the number of active flows is n. The Virtual-Tick value of active flows is:

$$VT_i = \tfrac{1}{r_i}; \sum_i r_i = 1 \quad (3)$$

In any interval of time (n-j), flows start to send packets then the total bandwidth of all passive flows, δ becomes (4)

$$\delta = \sum_{i=j}^{n} r_i \quad (4)$$

δ is divided equally to j active flows. Then the virtual-tick value of active flows is calculated as (5)

$$VT_i = \frac{1}{(r_i + \frac{\delta}{j})} \quad (5)$$

Whenever a new flow starts to send packets, it takes equal amount of bandwidth from each flow without affecting the other flows. Virtual-tick value is updated as (6)

$$VT_i = \frac{1}{(r_i - \frac{r_{n+1}}{n})} \quad (6)$$

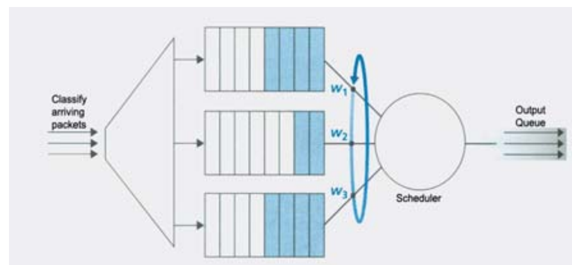

Figure 2. Schema of Weighted Fair Queuing

Each flow has an auxVC value which is equal with real time, after receiving new packets from a passive flow, the auxVC value is updating as (7)

$$auxVC_i = auxVC_i + VT_i \quad (7)$$

IV. PROPOSED WORK

Our scheduling algorithm is implemented on the topology depicted in Figure 3. Broadcasting is made by n units of IPTV service provider as 5 classes of service. Each node is an onboard switcher situated in IRIDIUM satellites. The proposed algorithm can be used in several different wireless and wired network topologies.

In simulations, for a scheduling task, packets generated by a single service provider are routed to a subscriber over a predefined route. In order to simulate a real time IPTV traffic other service providers are also included. The service provider sends an estimated value of packets that belongs to each of the priority classes to all switchers of how many packets from each priority class will be sent. The credit of each packet is calculated while the packet placed in the corresponding queue. An iterative estimation value is obtained by (8) whenever a packet is received from any priority class. (8) is a slightly specialized form of exponential smoothing or exponential averaging [7].

 ISBN: 978-1-61208-174-8

$$Est_i^{n+1} = \alpha(Est_i^n) + (1-\alpha)RT_i \text{ ,n=0,1,2,..} \quad (8)$$

Est_i^n : estimated packet count of i.th service class' n.th packet

α : alpha smoothing constant. $0 < \alpha < 1$. In the proposed algorithm the value 0.125 is selected.

RT_i : the number of received packets of i.th class

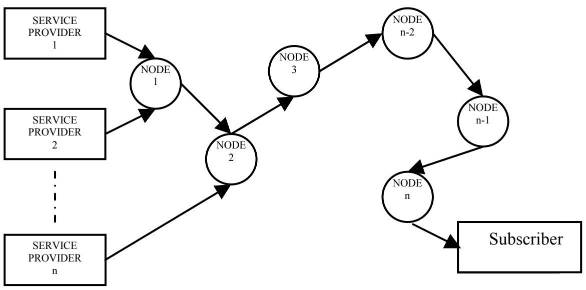

Figure 3. Exemplary Network Topology

After the estimated value is obtained, the credit of nth packet is computed as following:

$$Credit_i^n = \frac{Est_i^{n-1}}{PC} + SubCredit_i x Threshold_i \quad (9)$$

n=1,2,….

Est_i^{n-1} : estimation value of (n-1)th packet belonging ith service class.

$SubCredit_i$: momentary SubCredit value of ith service class

$Threshold_i$: momentary threshold value of ith service class

Table 1 shows the initial values of the SubCredit for each service class.

Threshold values starts all from zero and are updated when a packet is served by the scheduler from any of the service classes (SC). The value of and the other SC values are augmented as shown in Table 2, which indicates the values added to the previous value of the Threshold in each class. n is a natural number. Whenever the threshold of a SC reaches 35n, the first packet in the class queue is served automatically without checking the maximum credit.

If any of the threshold values reach 35n then the credit of each leading packet in the class queues are checked by the scheduler and the maximum one is routed to the output queue of the active switch.

All packets driven to the output queue of the current switcher are released to the link in the FIFO (First in First Out) order. At the next switcher all of the process is executed again.

The other packets coming from the other service providers through other switchers are also competing with our subscriber's packets.

TABLE I. SUBCREDIT INITIAL VALUES

SubCredit Initial Values	
Service Class 1 (PC1)	10
Service Class 2 (PC2)	8
Service Class 3 (PC3)	6
Service Class 4 (PC4)	4
Service Class 5 (PC5)	2

TABLE II. AUGMENTING THRESHOLD VALUES

Augmenting Threshold Values	
Service Class 1 (PC1)	+11n
Service Class 2 (PC2)	+7n
Service Class 3 (PC3)	+4n
Service Class 4 (PC4)	+2n
Service Class 5 (PC5)	+n

Table 3 shows how SubCredit of each SC is changed whenever a packet is selected and sent to the output queue. Figure 4 summarizes the algorithm by showing the flow diagram of the process.

TABLE III. SUBCREDIT UPDATE TABLE

SubCredit Update Table					
	PC1	***PC2***	***PC3***	***PC4***	***PC5***
A packet of PC1 is selected	$\frac{(\frac{PC2+PC3}{2}+10)xPS1}{50.Max(SubCredit)}$	PC2+20	PC3+30	PC4+40	PC5+50
A packet of PC2 is selected	PC1+10	$\frac{(\frac{PC3+PC4}{2}+20)xPS2}{50.Max(SubCredit)}$	PC3+30	PC4+40	PC5+50
A packet of PC3 is selected	PC1+10	PC2+20	$\frac{(\frac{PC4+PC5}{2}+30)xPS3}{50.Max(SubCredit)}$	PC4+40	PC5+50
A packet of PC4 is selected	PC1+10	PC2+20	PC3+30	$\frac{(\frac{PC3+PC5}{2}+40)xPS4}{50.Max(SubCredit)}$	PC5+50
A packet of PC5 is selected	PC1+10	PC2+20	PC3+30	PC4+40	$\frac{(\frac{PC3+PC4}{2}+50)xPS5}{50.Max(SubCredit)}$

PCi : the initial estimation value of ith service class

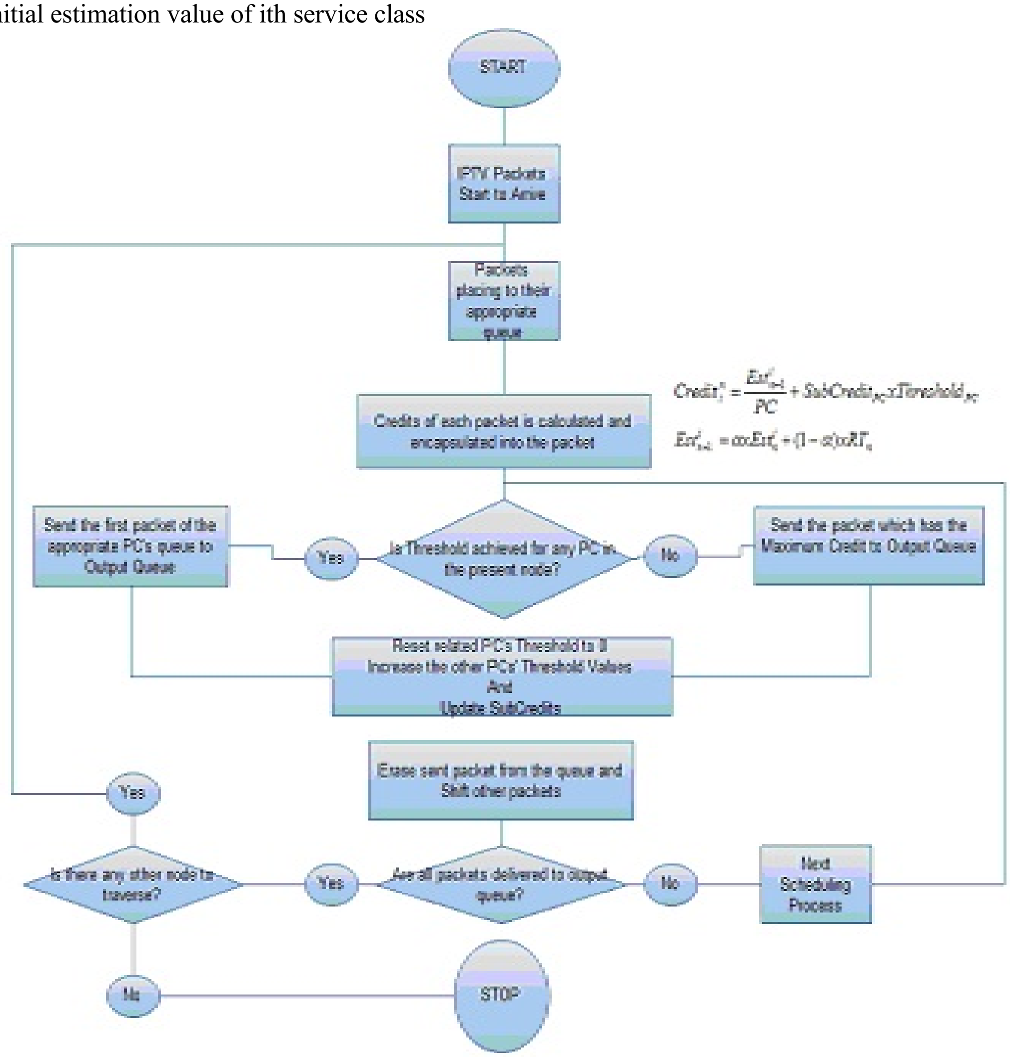

Figure 4. Flowchart of the Proposed Algorithm

V. RESULTS

The proposed algorithm is simulated versus WFQ and Alpha-Beta Virtual Clock algorithms in a framework written in MATLAB, at 100% load. In a flow started by the first service provider, packet distribution for each priority class is selected as:

PC1:2421
PC2: 2902
PC3: 3622
PC4: 8027
PC5: 815
Total duration of the real-time simulation is: 30955 ms

Queue length is chosen as 20 packets (all packets are at same size. Each node has a queue to buffer the arriving packets before forwarding occurs)

Mean packet delay variation and end-to-end mean delay for PC1 packets using one to nine switchers are shown in Figure 5 and 6 respectively.

VI. CONCLUSION

The proposed algorithm successfully staying below 30 ms through even 9 nodes offers a better solution against these two well-known algorithms. The advantage of the proposed algorithm is lied on its relative fairness by letting high priority class's class number in the denominator of the left side of the addition in the credit calculation equation number 9, giving advantage to itself. The other way around, if the right side of the equation is inspected, it is evident that SubCredit value favors the low priority classes. Threshold value itself favors all classes to reach good performance of packet delay variation.

In the future, the delay and the delay variation will be investigated thoroughly. It may be possible to be improved by changing the parameters and the algorithm formula.

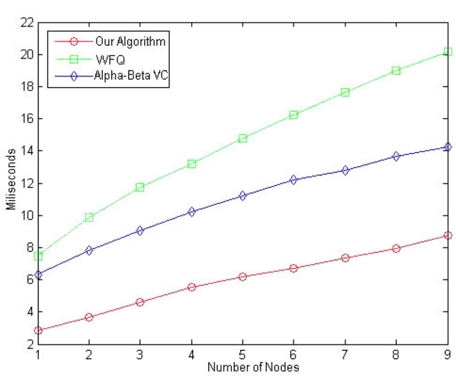

Figure 5. End-to-end mean packet delay variation for 3 algorithms

Furthermore, the performance of the proposed algorithm will be investigated on different wireless network topologies such as mesh networks.

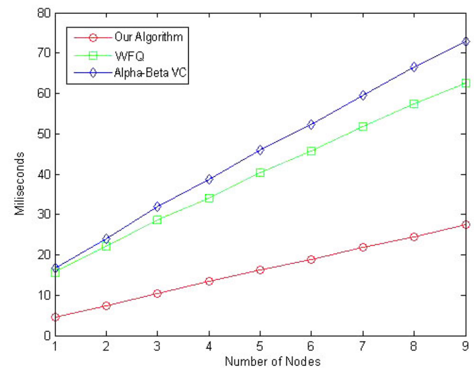

Figure 6. End-to-end mean delay for 3 algorithms

VII. REFERENCES

[1] VoIP Specifications, "http://www.ipdr.org/public/Service_Specifications/2.X/VoIP/VoIP2.5-A.0.pdf"

[2] RFC791, "http:// tools.ietf.org/html/rfc791", Sep. 1981

[3] J.F Kurose, K.W. Ross, Computer Networking:A Top Down Approach Fifth Ed., PEARSON, 2010

[4] A. Demers, S. Keshav, S. Shenker, "Analysis and Simulation of a Fair Queuing Algorithm", Internetworking: research and Experience, Vol 1, No1, 1990,pp 3-26

[5] M. Hosaagrahara, H. Sethu, 2001, Simulation-Based Analysis of a Novel Enhancement of Virtual Clock, *Proceedings of the Applied Telecommunication Symposium*

[6] L., Zhang, 1990, Virtual Clock: A New Traffic Control Algorithm for Packet Switched Networks, *Proceedings of ACM SIGCOMM'90*, Philadelphia

[7] Holt, C.C., "Forecasting seasonals and trends by exponentially weighted moving averages", International Journal of Forecasting Volume 20, Issue 1, January-March 2004, Pages 5-10