Journal of Computer Science, 9 (1): 63-73, 2013
ISSN 1549-3636

doi:10.3844/jcssp.2013.63.73 Published Online 9 (1) 2013 (http://www.thescipub.com/jcs.toc)

STATELESS AGGREGATE FAIR MARKING SCHEDULER FOR DIFFERENTIATED SERVICE NETWORKS

[1]Nandhini Sivasubramaniam and [2]Palaniammal Senniappan

[1]Department of Computer Science,
Garden City College of Science and Management Studies, Bangalore University,
16th KM, Old Madras Road, Bangalore-560 049, India
[2]Department of Science and Humanities,
VLB Janakiammal College of Engineering and Technology,
Anna University, Kovai Pudur, Coimbatore-641042, India

Received 2012-07-20, Revised 2013-01-03; Accepted 2013-02-09

ABSTRACT

In diffserv networks, as the traffic flows vary constantly, it is very difficult to maintain the per-flow state. The computation of rate information of the traffic flow also becomes complex. In this study, we propose a Stateless Aggregate Fair Marking (SAMQ) with Multiple Queue Priority Scheduler for Differentiated Service (DiffServ) networks. Initially, priority scheduler is applied to the flows entering the ingress edge router. If it is real time flow like Voice over IP (VoIP) or Video, then the packets are given higher priority else lower priority. In core router, for higher priority flows the Multiple Queue Fair Queuing (MQFQ) is applied that allows a flow to utilize multiple queues to transmit the packets. In case of lower priority, Stateless Aggregate Fair Marking technique is utilized. This technique applies Core Stateless Fair Queuing (CSFQ) technique for maintaining the rate information of packet flow and distributes the token to each incoming packet without maintaining the per-flow state. By simulation results, we show that this technique improves the throughput of non-real time flows.

Keywords: Stateless Aggregate Fair Marking (SAMQ), Voice over IP (VoIP), Core Stateless Fair Queuing (CSFQ), Multiple Queue Fair Queuing (MQFQ)

1. INTRODUCTION

1.1. Queuing Theory

Queuing scheme initially captured the imagination of researchers as a way to enforce fairness and provide traffic isolation required for applications such as Video-Conferencing, VoIP. Queuing theory is conceptually simple to understand which takes special care to handle variable packet sizes. Queuing scheme is a technique that control traffic congestion on the network by allowing each flow passing through a network device to have a fair share of network resources. The performance of the network and system can be enhanced through different queuing models (Mabayoje *et al*., 2011).

1.2. Queuing in Networking

Queuing schemes in networking provide Quality of Service (QoS) by controlling the forwarding capacity or bandwidth available to certain traffic flows. Queuing happens only when the interface is busy. Queues and queue-servicing algorithms are critical elements of traffic handling in providing network QoS. The queuing scheme has been applied in many applications in different fields like communication networks, computer systems and machine plants. Some examples of applications of queuing theory in networking are the dimensioning of buffers in routers or multiplexers, determining the number of trunks in a central office in POTS, calculating end-to-end throughput in networks and

Corresponding Author: Sivasubramaniam Nandhini, Department of Computer Science, Garden City College of Science and Management Studies, 16th KM, Old Madras Road, Bangalore-560 049, India

so forth. Each queuing algorithm was designed to solve a specific network traffic problem and has a particular effect on network performance (Nandhini and Palaniammal, 2012).

1.3. Fair Queuing Techniques

In fair queuing technique we have many techniques; some are Active Queue Management Technique, Deficit Round-Robin and MQFQ Technique, BR, SCFQ.

Active Queue Management Technique such as Random Early Detection (RED), drop or mark packets before the queue is full. Typically, they operate by maintaining one or more drop/mark probabilities and probabilistically dropping or marking packets even when the queue is short (Olawoyin *et al*., 2011). Therefore the CSFQ and RED can use Diffserv packet marking using token bucket specifications since it is superior to the current markers in terms of throughput and fairness

Deficit Round-Robin (DRR) is a scheme that provides solution to the unfairness caused by possible different packets by size used by different flows (Mabayoje *et al*., 2011) Flows are assigned to queues such that each queue would be served in round robin arrangement. The only difference from the traditional round robin is that, if a queue was not able to send a packet in the previous round because its packet size was too large, the remainder from the previous quantum is added to the quantum for the next round.

The Multiple Queue Fair Queuing (MQFQ) technique allows a flow to use multiple queues. It utilizes multiple hash functions to determine a set of FIFO queues for a flow and serves all queues in the round robin order. MQFQ puts the packet into the queue with the soonest service and if one queue associated with a flow grows large, then the flow uses another of its queues and thereby bypasses the congestion (Nandhini and Palaniammal, 2012).

1.4. Bit-By-Bit Round Robin (BR)

In this model, the data packets are sent one bit at a time in round robin fashion. The packet is then inserted into a queue of packets sorted on departure times.

The main drawback of this method is that it is expensive to insert into a sorted queue. And the packet processing cost makes it hard to implement cheaply at high speed. And also amongst all backlogged queues, the variable packet sizes cause bandwidth shares to be uneven. Further, it requires *O* (log (n)) time to transmit a packet, where 'n' is the number of connections (Mabayoje *et al*., 2011; Lin and Hamdi, 2010).

1.5. Self-Clocked Fair Queuing (SCFQ)

The scheme Self-Clocked Fair Queuing is based on virtual time function that makes computation of the packet departure time from their respective queues to be simpler. The virtual function is evaluated for every packet in the head of the queuing which is simply extracted from the packet in the head of the queue.

And the main drawback of this method is that the cost associated with the sorting technique used in SCFQ which retains 0 (log (n)) sorting which makes it complexity (Mabayoje *et al*., 2011).

1.6. Priority Queuing

This method comes in scene when different traffic types share common network resources, such as transmission lines, router and so on, they may be given different service requirements and the traffic within a queue is processed using FIFO.

And the drawback of this method is that only the packets with High Priority are processed and then the packets with low priority are processed, if the resources are available (Mabayoje *et al*., 2011).

1.7. Core Stateless Fair Queuing (CSFQ) Technique

In CSFQ, only edge routers maintain per flow state, while core routers do not maintain per flow state, instead uses the per flow information carried through a label in each packet's header. This label contains an estimate of the flow's arrival rate. Based on per-flow information, it updates at each router along the path based only on aggregate information at that router. The Core routers, in turn, can use the labels to allocate bandwidth fairly among all incoming flows. And they 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. CSFQ does not conform to the DiffServ services for, it necessitates the core routers to keep track a flow granularity state (Nandhini and Palaniammal, 2012; Bouras and Sevasti, 2009).

1.8. Realization of CSFQ

The CSFQ protocol involves following mechanisms:

- Estimation of flow arrival rate
- Estimation of fair rate
- Packet dropping algorithms

However, CSFQ has an issue due to packet dropping algorithm which is explained in following section.

1.9. Issues in Packet Dropping Algorithms

In CSFQ the packets are dropped with the dropping probability P = max (0, 1- F/R). Here F/R denotes the ratio of fair share rate to the rate acquired in the packet header. This probability fits to UDP flows as they do not offer any congestion control and it forwards the packets at steady rate in spite of the congestion state of the network. But this dropping probability is more destructive for TCP flows. This is because the TCP flows offer congestion control and minimizes the sending rate proportional to the congestion state of the network (Nabeshima, 2003).

1.10. Differential Service Networks (Diffserv Networks)

The Diffserv architecture offers various service levels for fulfilling several service needs in an accessible way. In this framework, the IP flows are categorized and accumulated into various forwarding classes. At the edge of the network, these flows are exhibited with different priority levels and at the core of a network; the packets are dropped according to the different dropping schemes. This reveals that Diffserv networks offers better Quality of Service (QoS).

As per the Internet Engineering Task Force (IETF), the maximum widespread forwarding mechanisms include the following categories:

- Expedited Forwarding Per-Hop Behavior (EF PHB): Envisioned to maintain traffic flows necessitating short delay
- Assured Forwarding Per-Hop Behavior (AF PHB): Envisioned to guarantee minimum throughout level

In order to guarantee minimum throughput i.e., Committed Information Rate (CIR), two mechanisms introduced by AF PHB are as follows.

Packet marking: This scheme monitors and blots the packets as per the service profile at the edge of a network.

Queue management: This scheme is applied to packets possessing high priority. When congestion occurs, high priority packets are forwarded and low priority flows are dropped (Bouras and Sevasti, 2009).

In this study, we use Token Bucket based marker as profile meters and CSFQ as the queue management technique which is described in the following section.

1.11. Token Bucket Model

The policing task of traffic conditioning block of Diffserv is denoted by a token bucket. The definition of token bucket is given as (x, L), where x is the flow rate of the tokens and L is the depth of accumulated tokens (in bytes).

The token bucket model contains two components:

- Committed Information Rate (CIR) in bps: It represents the rate at which the bucket is filled
- Committed Burst Size (CBS) in Bytes: It represents the maximum capacity of bucket

When the incoming packet matches with the service profile it is blotted as high priority and it is admitted through queue-in, otherwise it is blotted as low-priority and admitted through queue-out. The service profile maintains the criteria to differentiate short and long flows and the core device executes the scheduling methodology to decide removal of packet from the queue (Oyetunji *et al*., 2012).

From **Fig. 1**, it is shown that the tokens are entering the bucket at rate of x tokens/sec with L bytes of token. In case the bucket gets filled, the entering tokens will be removed:

- The three parameters considered in the Token Bucket profile (TB) are as follows
- Average rate: It is defined as the average rate at which a packet can be forwarded in the network
- Peak rate: It is defined as the maximum rate at which packets can be sent in minimum duration
- Burst size: It is defined as the maximum number of packets that can be transmitted in minimum duration (http://www.hynet.com.ar/eng/productos/extreme/vpn/)

1.12. Token Bucket in Differentiated Services

The process involved in the token bucket in Diffserv is described using following steps:

- If there are minimum tokens in the bucket, the incoming packets are processed at once i.e., conforming nature
- If the existing cumulative tokens are less than the incoming packets, then they are non-conforming and the following actions may occur
  - Removal of packets
  - Re-blot of the packets in a specific manner
  - Buffering of packets and it is not freed until the arrival of adequate number of tokens in the bucket

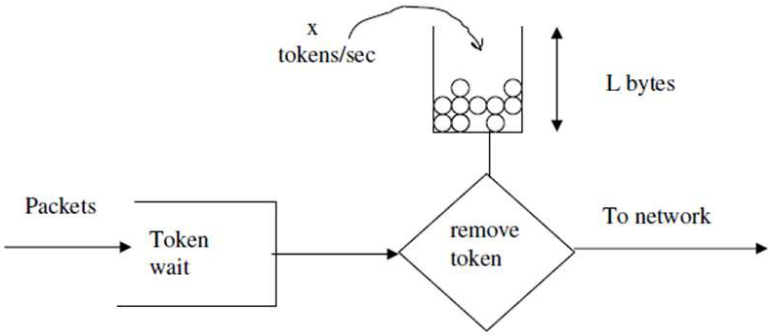

Fig. 1. Token bucket architecture

- Packets are permitted up to the average rate in bursts or up to burst size provided, that they are within the peak rate. When they exceed the peak rate, the bucket is exhausted
- If the packets are not available for transmission, tokens can be collected up to predefined size. The remaining tokens are removed

The conforming and non-conforming process in this technique is based on Service Level Agreement (SLA) (Oyetunji *et al*., 2012).

1.13. Related Works

Lu *et al*. (2012) have presented an Enhanced Weighted Fair Queuing scheme, known as EWFQ. Their scheme integrates the accuracy of scheduling algorithm namely WFQ to the decreased resource footprint of dropping-based active queue management schemes. EWFQ does not need the demand-specific buffer configuration and also it does not require the parameter adjustment, which is necessitated by Weighted Fair Queuing (WFQ) and some traditional active queue management schemes. Their mechanism drops the packet considering its flow weight and type.

Zhang and Ansari (2009) have proposed a utility max-min fair resource allocation for diversified applications in Ethernet Passive Optical Networks (EPONs). Initially they define application utility to illustrate the relationship among users Quality of Experience (QoE) and network-level QoS of each application. A bisection method is considered to get the optimal solution of the maximized minimum utility through bandwidth and queue management. This proposed scheme guarantees fairness among diversified applications.

Jin *et al*. (2009) have proposed a distributive flow control algorithm for networks with multiple paths among source-destination pairs. They employed a utility max-min fair resource allocation algorithm among competing users which is more appropriate for practical networks. The proposed approach removes typical oscillation performance in multipath networks by combining first order Lagrangian method and filtering mechanism. The factors such as delay and dynamic network behaviors such as stability are not considered in modeling the utility functions.

Vasiliadis *et al*. (2012) have introduced CBWFQ scheduling algorithm for a single-buffered, dual priority Multistage Interconnection Network (MIN). Their priority scheduling scheme has combined both class based and weighted fair queuing packet scheduling algorithms. While progressing algorithm, their scheme has considered previous and last state of switching element and thereby offered accuracy. Finally, their scheme is simulated and also analytical equations for modeling their scheme were also presented.

Yang *et al*. (2010) proposed a max-min fair share bandwidth allocation scheme that addresses challenges faced due to fairness during the allocation of the link bandwidth to competing users. They proved that the existence of the Nash Equilibrium (NE) in the Maximum Bandwidth Routing Problem (MAXBAR) game causes the players to be immobile from its chosen path. In order to compute NE, a game based algorithm is proposed. The network converges to NE only when all the users follow the natural game course.

1.14. Problem Identification

Nandhini and Palaniammal (2012), an Enhanced Core Stateless Fair Queuing (ECSFQ) with Multiple Queue Priority Scheduler is proposed. Initially priority scheduler is applied to the flows entering the ingress edge router. If it is real time flow i.e. VoIP or video flow, then the packets are given higher priority else lower priority. In core router, for higher priority flows the

MQFQ is applied that allows a flow to utilize multiple queues to transmit the packets. In case of lower priority, the normal max-min fairness criterion of CSFQ is applied to perform probabilistic packet dropping.

The probabilistic dropping function utilized for low priority flows is based on the average rate of a flow to which the packet belongs. This rate information, instead of being calculated at the queue using per-flow techniques is calculated near the source of the flow and inserted in every packet header.

As an alternative to normal max-min fairness technique, in this study we propose Stateless Aggregate Fair Marker technique (F-SAM) for differentiated service networks. This technique performs the probabilistic fair marking. The main idea is to apply the approximate fair queuing to the token bucket while distributing the tokens among the packets of the flows in the aggregate without maintaining any per- flow state.

1.15. Stateless Aggregate Fair Marking with Multiple Queue Priority Scheduler

1.15.1. Overview

In this study, we propose a Stateless Aggregate Fair Marking (SAMQ) with Multiple Queue Priority Scheduler for Diffserv networks.

When the packets enter into the ingress edge router, first the priority scheduler is applied to the flows. In case of VoIP and video flows, the packets are treated as higher priority whereas for the best effort traffic the packets are treated as lower priority. These priority values are marked along with flow arrival rate and transmitted to core router.

In core router, for higher priority flows the MQFQ is applied that allows a flow to utilize multiple queues to transmit the packets. In case of lower priority, a stateless aggregate fair marking technique is utilized that applies CSFQ technique to the packet flow and the token distribution to each incoming packet of the flow without maintaining the per-flow state.

1.16. Flow Classifier

The flow classifier identifies the ingress traffic flow as best effort or real-time based on the estimated delay and loss.

1.17. Delay Estimation

At all ingress routers the real time flows are sampled. The path of a real time flow has been probed by the header of the sampled packet. As the user does not get altered frequently inside a network domain, the probe and user traffic travel in same path with the high probability. Thus a rough estimation of delay value experienced by the sampled flows in the network domain is evaluated.

In case of probing the delay, the ingress routers encode the current timestamp Tc into the payload and header is marked with a new protocol value. Those packets are recognized by egress router and removed from the network. Before that, the egress router computes edge-to-edge link delay for a packet. The link delay is the resultant of difference between the own time of packets and Tc. The egress classifies the probe packets as belonging to flow i and updates the average packet delay, PDavi for delay sample Di (t) at time t using an Exponential Weighted Moving Average (EWMA) Equation 1:

$$PD_{avi}(t) = \mu * PD_{avi}(t-1) + (1-\mu) * D_i(t) \quad (1)$$

where, μ is a small fraction $0 \leq \mu \leq 1$to emphasize recent history rather than the current sample alone.

1.18. Loss Estimation

The detection algorithm runs as follows.

The edge-to-edge probing investigates excessive packet loss within a network domain. The back to back probe packets for a small sample interval of T seconds are utilized to deduce link loss. This is done by computing the correlation of a packet loss within a set of probe packets at different destination. In this technique, source forwards a series of probe packets along a path P_1 to the destination, with no delay during the transmissions of successive packets. The loss ratio (L_i) at a node N_i along the path P_1 at the interval T can be calculated as Equation 2:

$$L_i^T = P_{Lo} / R_a \quad (2)$$

where, P_{Lo} is the number of packets lost and R_a is the estimated arrival rate of the packet.

Then the total loss ratio at destination can be calculated as Equation 3:

$$L^T = \sum L_i^T \quad (3)$$

Now the actual traffic flows are transmitted for the same sample interval of T seconds through the ingress router which marks the flow arrival rate as label according to CSFQ. The actual loss ratio (L_{act}) at each node along P1 at the interval T can be estimated similarly as (1).

Then the total actual loss ratio L_{act} at destination can be calculated as Equation 4:

$$L_{act}{}^{T} = \sum L_{acti}{}^{T} \quad (4)$$

At egress router, the difference in loss ratios can be then estimated as Equation 5:

$$D = L_{act}^{T} - L^{T} \quad (5)$$

1.19. Flow Identification by Ingress Nodes

The links possessing high losses and egress router through which the flows are exiting are found. The flows that consume high bandwidth are isolated. These rates are forwarded to ingress routers through which the flows enter into the domain. The rate at which the flow is entering and exiting the network domain is compared by ingress router.

The real time flows can be reported either in per flow or aggregate fashion. If the flow value is greater than the threshold, then the feedback is done by aggregate manner for each ingress router. The aggregation is performed based on the traffic class.

The real time flows with high bandwidth are reported to the egress router. From the CSFQ labels, the identity of the ingress router is obtained. This identification code is used to relate a flow and its entry point else the egress does not know through which ingress routers the flow is entering into the domain.

The flow arrival rates and the corresponding source ids are collected from the labels of the packets which are marked by ingress node.

If the value of D (as per Equation 5) exceeds to a threshold T_1 and if the delay (as per Equation 1) exceeds a threshold T_2, then the flows are marked as real time flows by the egress node, otherwise they are considered as best effort traffic. Then the flow arrival rate and the flow id are sent to the source by the egress router.

1.20. Priority Scheduler

We apply priority scheduler to the above identified flow categories as per following condition:

1. If flow is VoIP or Video, Then
   Flow is marked as higher priority in Flow label
   Else
2. If the flow is best effort traffic, then
   Flow is marked as lower priority in Flow label
   End if

These priority values are marked along each flow and passed to the core router. The core router checks the priority values. For higher priority flows, MQFQ (Nandhini and Palaniammal, 2012) is applied and for lower priority flows, a Stateless Aggregate Fair Marking based CSFQ technique is applied. Flow chart that specifies priority scheduling is shown in **Fig. 2** and the functions of ingress and core routers are depicted in **Fig. 3**.

1.21. Stateless Aggregate Fair Marking

This technique employs the CSFQ technique for maintaining the rate information of a flow in the packet header and distributes the token among the packets of every flow. Through this approach, every incoming packet receives the fair probabilities of token devoid of upholding the per-flow state. The steps involved in the stateless aggregate fair marking technique are given by the following algorithm.

Step 1:

The rate information in each packet header is calculated and filled by the ingress node when the flow enters the domain. Since each ingress node is responsible for maintaining the rate of only the flow that enters through it, there is no scalability issue involved in the per-flow rate calculation that is needed (Stoica *et al*., 1998; Azath *et al*., 2009).

When a flow 'i' enters the network, the ingress node computes the arrival rate of that flow (x_i (t)) at time t. Then the cumulative arrival rate of all the flows at time 't' is given by Equation 6:

$$SX(t) = \sum_{i-1}^{n} x_i(t) \quad (6)$$

Step 2:

The token bucket rate corresponds to the output link speed of routers.

The token bucket rate at time 't' is calculated using Equation 7:

$$TBR(t) = \sum_{i=1}^{n} \min(x_i(t), FR(t)) \quad (7)$$

where, FR(t) is the fair rate of flows at time 't' which is same for all the flows that are bottlenecked by this router as per max-min fair bandwidth principle.

For simplicity of notation, we can represent TBR (t) as TBR.

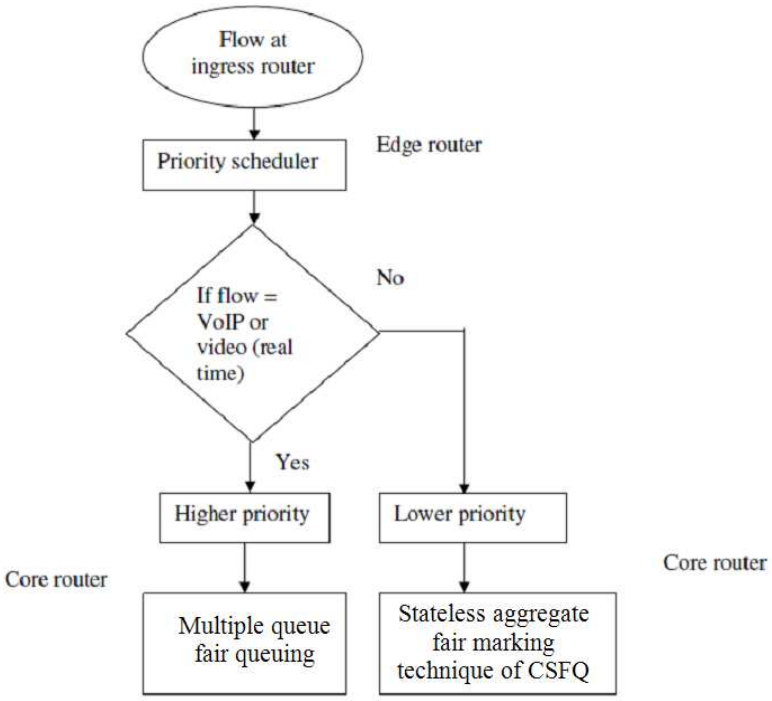

Fig. 2. Flow chart of the proposed technique

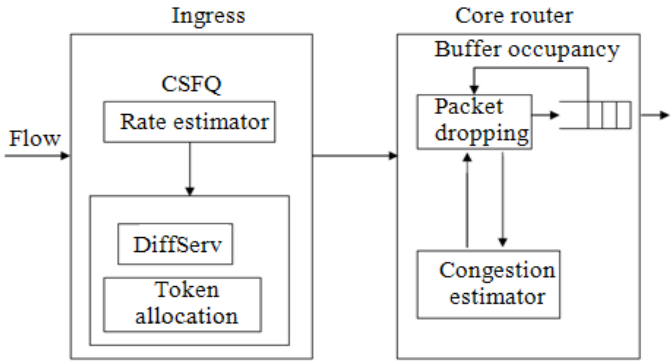

Fig. 3. Functions of Edge and Core Routers

Step 3:

Predicted token allocation rate corresponds to the marking probability of the packets of arriving flows.

The predicted token allocation rate (Pr_t) to a packet appropriate to a flow 'i' of rate x, at time 't' is given by Equation 8:

$$Pr_t = \min(FR'(t), x_i(t)) \quad (8)$$

Based on (8), the cumulative token allocation rate of all flows at time t is given by Equation 9:

$$T_c = \sum_i^n \min(FR'(t), x_i(t)) \quad (9)$$

Here FR'(t) selected such that it is a unique solution to Tc = TBR

Step 4:

At the ingress node, the edge marker calculates the Fair Rate (FR (t)) allocated to the flows based on SX (t), T_c and TBR which is given by Equation (10). Then it computes the token allocation probability (P_t) of a packet which is given by Equation 11:

$$FR_{new}(t) = FR_{old}(t) * (TBR / T_c) \quad (10)$$

$$P_t = \min(1, FR / x) \quad (11)$$

where, x is the rate of the corresponding flow.

Step 5:

At every ingress marker, the rate of flow is estimated by exponential averaging scheme. Each time when a new packet is received, flow rate is updated. Thus the new updated rate information is given by Equation 12 (Stoica *et al*., 1998; Azath *et al*., 2009).

$$x_i^{new} = (1 - e^{-T_i^n / \alpha}) \frac{L_i^n}{T_i^n} + e^{T_i^n / \alpha} x_i^{old} \quad (12)$$

Where:

L_i^n = Length of n^{th} packet of flow i.

T_i^n = Arrival time given by $t_i^n - t_i^{n-1}$

To filter out the estimation inaccuracies due to exponential smoothing, we use a window of size α.

Step 6:

To detect the congestion at a link, both the cumulative arrival rate and the token bucket rate are compared. (i.e.,) for any time interval of α, if the cumulative arrival rate SX (t) is greater than the token bucket rate TBR, the link is assumed to congested and the fair share rate FR is updated according to equation 10. If the link is not congested, FR is set to the maximum rate (Stoica *et al*., 1998; Azath *et al*., 2009):

If SX (t) > TBR Then
FR (t) updated as per equation (10)
Else
FR (t) is set to maximum rate that is observed on any incoming flow. i.e.,
FR(t) = max $\sum x_i(t)$
End if

1.22. Simulation Results

1.22.1. Simulation Model and Parameters

We examine the performance of our Stateless Aggregate Fair Marking (SAMQ) with an extensive simulation study based upon the ns-2 network simulator (http://www.isi.edu/nsnam/ns). We compare our results with our previous technique Enhanced CSFQ (ECSFQ) (Nandhini and Palaniammal, 2012). The topology used in the simulation is depicted in **Fig. 4**. As we can see from the figure, the DiffServ architecture consists of 3 pairs of ingress and egress routers indicated by IE1-EE1, IE2-EE2 and IE3-EE3 with 2 core routers C1 and C2.

We use a mixture of Video, CBR and TCP traffic flows. The packet size is 512 bytes and there are totally 10 flows. The link bandwidth and link delay is set as 10Mb and 10ms respectively. The bottleneck bandwidth for the links (3, 4), (9, 14), (18, 4), (22, 9), (5, 4) and (10, 9) is set as 5Mb initially.

1.23. Performance Metrics

In the simulation experiments, we vary the bottleneck bandwidth and traffic rate. We measure the following metrics for the non-real time traffic (UDP) flows only, since SAMQ enhances the performance of non-real time lower priority traffic:

- Packet Loss
- Throughput
- Delay

1.24. Results

1.24.1. Effect of Varying Bottleneck Bandwidth

In our first experiment, we vary the bottleneck bandwidth 2Mb, 4Mb… 8Mb in order to calculate the packet loss, delay and throughput. In our experiment, we use TCP for background traffic and UDP for non-real time traffic.

Figure 5 gives the TCP Throughput occurred for varying the bottleneck bandwidth. When the bottleneck bandwidth is increased from 2 to 8, the TCP throughput slightly decreases as UDP and Video flows tend to use more bandwidth. It shows that the TCP Throughput is more in the case of SAMQ when compared with ECSFQ.

Figure 6 shows the Delay variation. The delay increases linearly when the bottleneck bandwidth is increased. This is because of the fact that, increase in bottleneck bandwidth allows more traffic flows. **Figure 6** shows that out proposed SAMQ has lower delay than the ECSFQ.

In **Fig. 7**, the packet loss tends to decrease, as the bottleneck bandwidth increases. **Figure 7** shows that the packet loss is less for SAMQ at lower bandwidths as compared to ECSFQ.

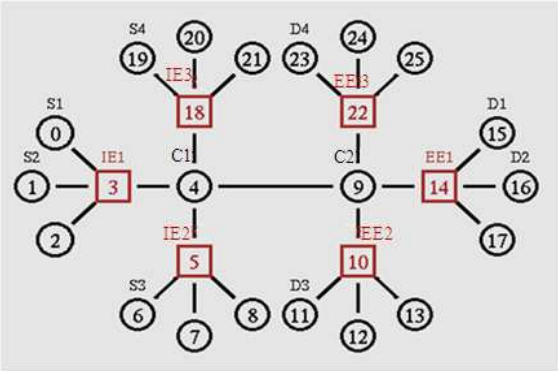

Fig. 4. Simulation topology

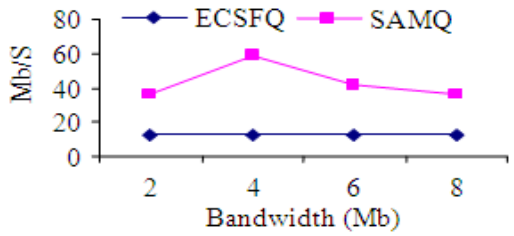

Fig. 5. Bottleneck BW Vs TCP-throughput

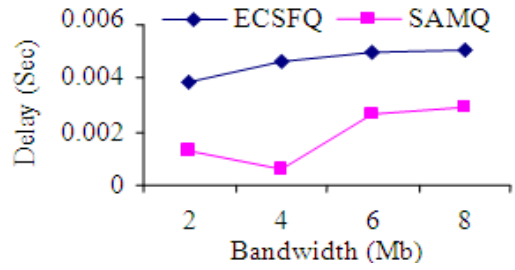

Fig. 6. Bottleneck BW vs Delay

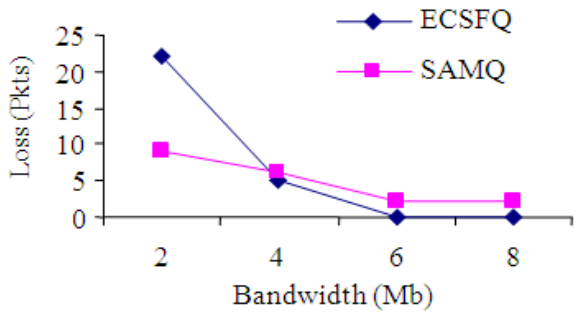

Fig. 7. Bottleneck BW Vs Loss

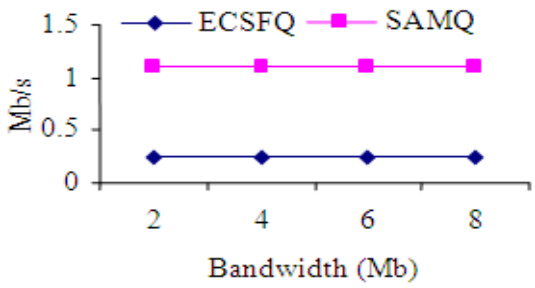

Fig. 8. Bottleneck BW Vs UDP-throughput

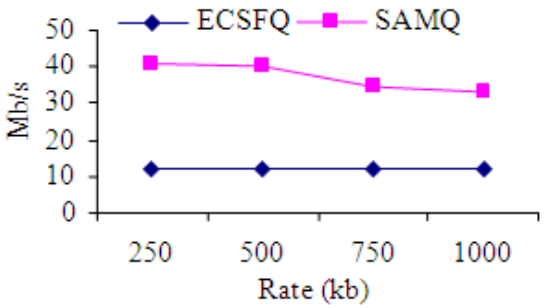

Fig. 9. Rate Vs TCP-throughput

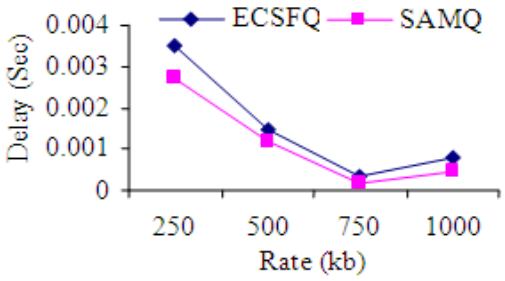

Fig. 10. Rate Vs Delay

Figure 8 gives the UDP Throughput occurred for varying the bottleneck bandwidth. As we can see from the figure, the UDP Throughput is more in the case of SAMQ when compared with ECSFQ.

1.25. Effect of Varying Rates

In our second experiment, we vary the traffic rate as 250, 500 …1000Kb in order to calculate the packet loss, delay and throughput. The bottleneck bandwidth is fixed as 5Mb. We use TCP for background traffic and UDP for non-real time traffic.

Figure 9 gives the TCP Throughput occurred for varying the Rate. It shows that the TCP Throughput is more in the case of SAMQ when compared with ECSFQ.

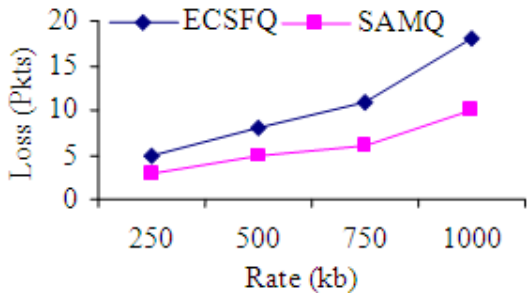

Fig. 11. Rate Vs Loss

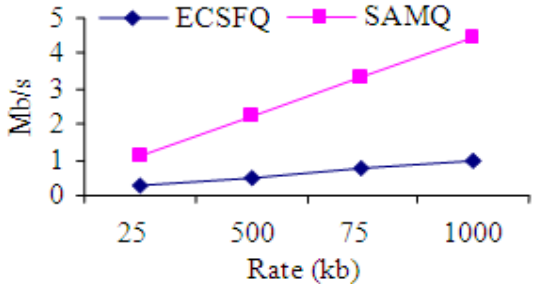

Fig. 12. Rate Vs UDP-throughput

Figure 10 shows the Delay variation. It shows that out proposed SAMQ has less delay than the ECSFQ when varying the Rate.

Figure 11 shows that, the packet loss is high in ECSFQ when compared with SAMQ by varying the Rate.

Figure 12 gives the UDP Throughput occurred for varying the Rate. As we can see from the figure, the UDP Throughput is more in the case of SAMQ when compared with ECSFQ.

2. CONCLUSION

In this study, we have proposed a Stateless Aggregate Fair Marking with Multiple Queue Priority Scheduler (SAMQ) for Differentiated service (DiffServ) networks. The flows are classified as higher-priority or lower priority by a priority scheduler. The priority is marked by the CSFQ at edge routers and transmitted to the core router. In core router, the higher priority flows are handled by MQFQ technique and lower priority flows are handled by Stateless Aggregate Fair Marking technique. The proposed technique is simulated in NS-2. We have varied the bottle-neck bandwidth and transmission rate and measured throughput, delay and packet loss. The results are in favor of the proposed technique when compared with the ECSFQ technique.

3. REFERENCES

Azath, M., A.N. Madheswari and R.S.D.W. Banu, 2009. Core-stateless fair queuing for fair bandwidth allocations in internet. Proceedings of the IEEE International Conference on Advances in Computing, Control and Telecommunication Technologies, Dec. 28-29, IEEE Xplore Press, Trivandrum, Kerala, pp: 596-598. DOI: 10.1109/ACT.2009.151

Bouras, C. and A. Sevasti, 2009. Performance analysis of Relative service using TCP-aware marking and dynamic WRED. Int. J. Commun. Syst., 22: 277-305. DOI: 10.1002/dac.v22:3

Jin, J., W.H. Wang and M. Palaniswami, 2009. Utility max-min fair resource allocation for communication networks with multipath routing. Comput. Commun., 32: 1802-1809. DOI: 10.1016/j.comcom.2009.06.014

Lin, D. and M. Hamdi, 2010. Two-stage fair queuing using budget round-robin. Proceedings of the IEEE International Conference on Communications (ICC), May 23-27, IEEE Xplore Press, Cape Town, pp: 1-5. DOI: 10.1109/ICC.2010.5502015

Lu, F., G.M. Voelker and A.C. Snoeren, 2012. Weighted fair queuing with differential dropping. Proceedings of the IEEE INFOCOM, Mar. 25-30, IEEE Xplore Press, Orlando, FL., pp: 2981-2985. DOI: 10.1109/INFCOM.2012.6195742

Mabayoje, M.A., S.O. Olabiyisi, A.O. Ameen, R. Muhammed and O.C. Abikoye, 2011. An efficient fair queuing model for data communication networks. Int. J. Comput. Sci. Inform. Security, 9: 206-216.

Nabeshima, M., 2003. Adaptive CSFQ: Determining the packet dropping probability adaptively for achieving fair bandwidth allocations in SCORE networks. Comput. Commun., 26: 579-590. DOI: 10.1016/S0140-3664(02)00188-3

Nandhini, S. and S. Palaniammal, 2012. Enhanced core stateless fair queuing with multiple queue priority scheduler. Int. Arab J. Inform. Technol.

Olawoyin, L.A., N. Faruk and L.A. Akanbi, 2011. Queue management in network performance analysis. Int. J. Sci. Technol., 1: 215-218.

Oyetunji, M.O., F.O. Oladeji and O.J. Emuoyinbofarhe, 2012. Performance evaluation of traffic meters: Token bucket marker and two rate Three Color Marker (trTCM) QoS admission control. Int. J. Comput. Appli.. 56: 40-44.

Stoica, I., S. Shenkar and H. Zhang, 1998. Core-stateless fair queueing: Achieving approximately fair bandwidth allocations in high speed networks. Proceedings of the ACM SIGCOMM '98 Conference on Applications Technologies, Architectures and Protocols for Computer Communication, Aug. 31-Sept. 04, ACM Press, New York, USA., pp: 118-130. DOI: 10.1145/285243.285273

Vasiliadis, D.C., G.E. Rizos and C. Vassilakis, 2012. Class-based weighted fair queuing scheduling on dual-priority delta networks. J. Comput. Netw. Commun. DOI: 10.1155/2012/859694

Yang, D., G. Xue, X. Fang, S. Misra and J. Zhang, 2010. Routing in max-min fair networks: A game theoretic approach. Proceedings of the 18th IEEE International Conference on Network Protocols (ICNP), Oct. 5-8, IEEE Xplore Press, Kyoto, pp: 1-10. DOI: 10.1109/ICNP.2010.5762749

Zhang, J. and N. Ansari, 2009. Utility max-min fair resource allocation for diversified applications in EPON. Proceeding of the 4th International Conference on Access Networks, Nov. 1-3, Springer Berlin Heidelberg, Hong Kong, China, pp: 14-24. DOI: 10.1007/978-3-642-11664-3_2