PerNEWQUE: An Active Queue Management Algorithm for Controlling Unresponsive Flows

SANTHI V, DR. A.M. NATARAJAN
Assistant Professor, Department of Computer Science and Engineering
Anna University of Technology Coimbatore
PSG College of Technology, Coimbatore, Tamil Nadu,
INDIA
sannthi@yahoo.com, amn@bannari.co.in

Abstract: - In order to design the congestion control scheme in the routers, an Active Queue Management (AQM) is proposed. This is used to control congestion at the router, where packets are dropped before queue become full. A new framework of AQM, namely NEWQUE with Per-flow Scheduling (PerNEWQUE) active queue management algorithm supporting explicit congestion notification (ECN), is proposed by extending scheduling nature in NEWQUE AQM. It is developed with the aim of strengthen the robustness of Internet against unresponsive flows. The objective of the new algorithm is to detect and penalize the unresponsive flows like UDP flows from responsive flows like TCP flows. The PerNEWQUE AQM is implemented with help of NS2 simulator. The simulation shows that the proposed design outperforms other AQM schemes in terms of reduction of throughput, link utilization and increasing percentage packet loss of unresponsive flows.

Key-Words: - Active Queue Management, Congestion control, Explicit Congestion Notification (ECN), Unresponsive flow, TCP, UDP

1 Introduction

Day by day people are using wider range of internet applications, requiring Quality of Service (QoS) in transmission of data. QoS in the existing and emerging applications in the Internet has been a big challenge to the Internet providers. Traditional applications such as File Transfer Protocol and E-mail, obey end-to-end flow and congestion control of TCP, require high throughput, less packet drop rate, and minimum queuing delay. However new applications such as audio/video applications are being deployed which do not use TCP congestion control and are unresponsive to the congestion signals given by the network. Such applications are potentially dangerous because they drive up the packet loss rates in the network and can eventually cause congestion collapse [8]. TCP congestion avoidance algorithms [12, 13] alone are not sufficient for controlling the congestion in all circumstance in the Internet. Some mechanisms [4, 5] are needed in the routers to give best performance to control congestion collapse. This has led some researchers to conclude that the routers must participate in avoidance of congestion collapse.

There are two classes of mechanisms proposed by B. Braden et al. to congestion avoidance at the router: "Queue management" and "Scheduling" algorithms [2]. In queue management algorithms manage the queue length by dropping packets when needed or appropriate, while scheduling algorithms determine which packet to send from the queue. Moreover, different types of application, like multimedia and audio/video applications use User Datagram Protocol (UDP), which does not obey end-to-end flow and congestion control. Generally, the sending rates of such flows are set by application and are not responsive to congestion signals given by TCP congestion avoidance algorithms [10, 12] in the networks. As a result, UDP flows send data rate aggressively, and makes use of more link bandwidth than TCP responsive flow. To overcome this problem, it is necessary to have router mechanisms that protect responsive flows from unresponsive flows.

At present, the traditional technique, "tail-drop", is used in the today Internet for dropping most recently added packet when the buffer is full. But it does not provide complete solution for detecting unresponsive flows. Different AQM schemes [1, 3] are proposed for addressing the inherent problems of unresponsive flows such as Flow Random Early Detection (FRED) [13], Stochastic Fair Blue (SFB) [7], Stochastic Fair Queuing (SFQ) [16], and Deficit Round Robin (DRR) [20] and Core-Stateless Fair Queuing (CSFQ) [11].

The aim of this paper to design a NEWQUE with Per-flow Scheduling AQM (PerNEWQUE) with Explicit Congestion Notification (ECN) based on total flow arrival rate, and link capacity. This algorithm includes the per-flow scheduling in

NEWQUE AQM [18]. This algorithm is rate-based scheme to predict the congestion and take actions based on the packet arrival rate. It is developed in order to strengthen the robustness of Internet against unresponsive flows. The objective of the new algorithm is to detect and penalize the unresponsive flows from responsive flows. It is evaluated by using NS2 simulator. The simulation shows that the PerNEWQUE AQM outperforms other active queue management techniques like SFB, SFQ, DRR, NEWQUE and CSFQ in terms of reducing unresponsive flow throughput, link utilization and increasing unresponsive flow packet loss ratio under different scenarios.

The Internet Engineering Task Force (IETF) introduces Explicit Congestion Notification (ECN) mechanism [9, 17] in IP header to indicate the congestion from receiver to sender by packet marking instead of packet dropping. Any AQM router supports ECN reduces packet loss rate comparatively AQM router without ECN.

The rest of the paper is planned as follows. Section 2 describes PerNEWQUE AQM and provides a detailed analysis. Section 3 describes evaluation of its performances based on simulations. Section 4 gives an explanation of some of the AQMs such as SFQ, SFB, DRR, NEWQUE and CSFQ. It shows how the related AQMs are controlling unresponsive flows rate.

2 NEWQUE with Per-Flow Scheduling (PerNEWQUE) AQM

For the purpose of detecting restraining unresponsive flows effectively, a new AQM algorithm named as NEWQUE AQM with Per-flow Scheduling (PerNEWQUE) [19] is proposed. The motivation behind PerNEWQUE is to modify NEWQUE AQM by introducing scheduling nature in queue management. In our AQM scheme, packets are dropped before buffer full based on individual probability is maintained for each active flow. According to the unresponsive flow feature of high data sending rate, it accumulates more number of packets into the queue. The probability of active flow with longest queue length is increased when the bottleneck router link is congested.

2.1 Proposed Mechanism

PerNEWQUE uses incoming total flow arrival rate, the link capacity to manage congestion. It uses deficit round robin hashing mechanism to assign flow into individual queue. Also, it maintains probability for each active flow. Note that a flow is as active if it has at least one packet in its queue. When the total flow arrival rate is less than the link capacity, the probability of current incoming active flow is decremented. When the total flow arrival rate is greater than or equal to the link capacity and the current queue length is greater than or equal to the buffer size, the probability of which active flow having longest queue is incremented. This marking probability effectively allows PerNEWQUE to learn the correct rate it needs to send back congestion notification. At the same time, the speed of updating of the marking probability depends on a parameter minTIVL. Fig.1 shows pseudo code for PerNEWQUE algorithm. The following parameters are defined for the PerNEWQUE AQM.

B	Buffer size
Q	Total queue length of active flows
C	Link Capacity
r_{new} (t)	Current Total Flow arrival rate at router
r_{old} (t)	Previous Total Flow arrival rate at router
P[i]	Packet dropping or marking probability of flow i
prevTime[i]	Time when the previous update of P[i] occurred
minTIVL	Minimum time interval between two successive updates of P[i]
now	Current time in seconds
queLen[i]	Queue length of flow i
N	Number of active flows

The PerNEWQUE AQM takes the following steps:

a) When a new packet of flow i arrives at a router, the router calculates the value of EQ that represents Q plus the size of the arriving packet.

b) At particular time t, the router calculates the incoming total flow arrival rate using fixed weight exponential averaging method. The following method shows the computation of total flow arrival rate.

Computation of Total Flow Arrival rate: The rates r_{new} (t) are estimated at each router [11]. At each router, use exponential averaging with the parameter $e^{-T/K}$ to estimate the rate of flows. Equation (1) shows computation of total flow arrival rate. Let t and l be the arrival time and length of the packet of flow. The estimated rate r_{new} (t), is updated every time a new packet is received.

$$r_{new}(t) = (1 - e^{-T/K})\, l / T + e^{-T/K}\, r_{old}(t) \quad (1)$$

where,

T is the inter-arrival time between the current and the previous packet.

K is constant.

c) If r_{new} (t) < C then decrement the marking probability P[i] of current flow i and if r_{new} (t) >= C then increment the marking probability P[i] of longest queue flow i and drop the front packet from that longest queue active flow i.

The speed of updating of the marking probability depends on a parameter minTIVL. Marking Probability P is updated as follows:

- Upon total flow arrival rate (r_{new} (t)) >= link capacity (C): Increment the Marking Probability P[i] of longest queue active flow i.
  P[i] is calculated as:
  If ((now – prevTime[i]) > minTIVL)
  P[i] = P[i] + α;
  prevTime[i] = now;
- Upon total flow arrival rate (r_{new} (t)) < link capacity (C): Decrement the Marking Probability P[i] for current active flow i.
  P[i] is calculated as:
  If (P[i] > 0 and and (Q/N > queLen[i]))
  If ((now– prevTime[i]) > minTIVL)
  P[i] = P[i] - β
  prevTime[i] = now

The logic behind the update is as follows. Flows whose accumulates more number of packets in to the queue tend to be chosen as the ones with the longest queue length (generally UDP flows makes longest queue because of high sending rate of its feature). Thus, it is reasonable to increase the packet dropping probability of such flows. Q/N represents the average queue occupancy of each active flow. If Q/N > queLen[i], the current queue length of flow i is less than the average. Thus, it is desirable to decrease the packet dropping probability of flow i.

d) Generally if (EQ >= B), the router chooses the active flow with longest queue length and increments the marking probability P[i] of that flow. Then the router drops the front packet from the buffer of that flow. If (EQ < B), the router chooses longest queue active flow. If the chosen flow is ECN-capable, router marks the first unmarked packet with marking probability P[i]. If flow is non-ECN-capable, the router drops the front packet from the flow with marking probability P[i].

The designed PerNEWQUE AQM can support Explicit Congestion Notification (ECN) (Floyd, 94) flows because ECN allows end-to-end notification of network congestion instead of dropping packets. If the packet is ECN capable then it reduces packet loss rate in the network. The ECN marking informs to senders to control sending rate when the buffer becomes full at the router. The advantages of using ECN are bandwidth up to bottleneck not wasted and no delay enforced by retransmission. If the flow i is set with ECN-capable transport (ECT) bit set, the PerNEWQUE AQM mark the first unmarked packet with probability P[i] and also set Congestion Experience (CE) bit in the IP header. If the CE bit is set by the router AQM then the TCP sinks react with setting of ECN-Echo (ECE) flag in the TCP header in its next acknowledge packet sent to TCP senders. After receiving acknowledge packet with ECN-Echo flag set, the TCP sender reduces its incoming rate by setting the Congestion window Reduced (CWR) flag in the TCP header of the next packet sent to the TCP sinks [17].

```
Arriving a new packet at time t;
Enqueue packet into the corresponding flow queue;
   Computation of total flow arrival rate r new (t);
    If (r new (t) < C)
        Decrement the marking probability of the
        current flow;
    else {
        Increment the marking probability of
        longest queue flow;
        Drop front packet from that flow;
    }
EQ=total queue length + size of arriving packet
    If (EQ < Buffersize){
        Choose longest queue active flow;
            If  (Does the flow belong to
                ECN-capable)
                Mark first unmarked packet
                with marking probability;
            else
                Drop front packet from that
                flow with marking probability;
     }
    else {
        Increment the marking probability of
        longest queue flow;
            While (EQ >= Buffersize)
                Drop the front packet from that
                flow;
        }
```

Fig.1 The PerNEWQUE Algorithm

e) When a packet of flow i is send out for transmission, if queLen[i] is 0 the router eliminates the state of flow i. That is, routers maintain the states of only active flows.

3 SIMULATIONS AND RESULTS

This section presents the performance of NEWQUE with Per-flow Scheduling AQM (PerNEWQUE) in penalizing unresponsive flows. This algorithm is evaluated by using NS2 [15] under different network parameters over a dumb-bell network topology

shown in Fig.2. The performance of PerNEWQUE AQM is validated in terms of throughput, percentage link utilization, and percentage packet loss in different scenarios. Some representative AQM schemes, namely, DRR [20], SFQ [16], SFB [6, 7], CSFQ [11] and NEWQUE [18], are also simulated and compared to this scheme, PerNEWQUE AQM.

3.1 Simulation Setup

The network topology configuration is shown in Fig.2. The links between the sources (Ss) and the router (R) are 100 Mbps links with 1 ms propagation delay, which are the same as those between the sinks (Ds) and the router (R). Router is connected to through a 10 Mbps 100 ms delay link, which is the bottleneck link since this link bandwidth is 10 times lesser than active sources. This link is shared between 1 UDP flow and (n-1) TCP flows. The TCP flows derived from FTP applications, which send bulk data transfer. The UDP flow sends packets at a Constant Bit Rate (CBR) of different fixed data rate. The sizes of all packets are set to 1000bytes. All sources are enabled with ECN support [17] and randomly started within the first one second of simulation.

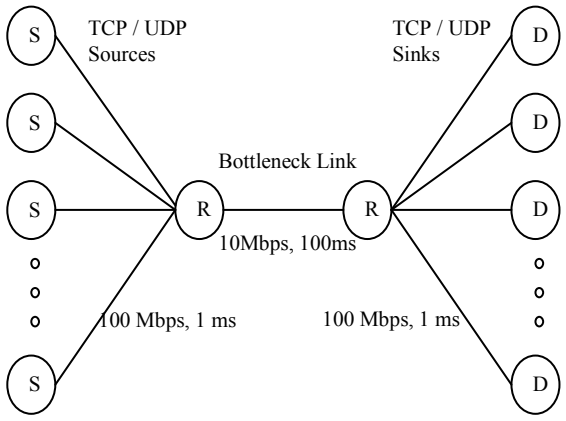

Fig.2 Dumb-Bell Network topology

The configuration parameters used in each of AQMs are: The queue capacity of the congested link between the routers is set as 300 packets, and it is shared by 99 TCP Sources and 1 UDP source (n=100). In DRR AQM: quantum = 1000 bytes [20]. In SFB AQM: $\delta 1 = 0.005$, $\delta 2 = 0.001$, freeze-time = 100ms and hash-interval = 20s, which are recommended in [6, 7]. In CSFQ: qsizethreshold = 67% of queue size, K = 0.1 [11]. In NEWQUE AQM: $\alpha = 0.01$, $\beta = 0.0001$, minTIVL = 100ms and K = 0.1. In PerNEWQUE AQM: $\alpha = 0.01$, $\beta = 0.0001$, minTIVL = 100ms and K = 0.1. The simulation time is set to 100 seconds. Throughput, packet loss statistics and link utilization are measured after 100 seconds.

3.2 Performance of PerNEWQUE Under Different Number of Sources

In this simulation, the total numbers of active sources or flows, 'n' are varied from 100 to 300. Here, 1 UDP source and n-1 FTP sources (Flow 1 to Flow (n-1)) are transmitted. The UDP source sends packets at the fixed rate of 0.1MB (0.8Mbps arrival rate). The buffer size is fixed at 300 packets.

3.2.1 Throughput of Different AQM Schemes

Fig.3 shows throughput of UDP flows (in MB) and Fig.4 shows throughput of TCP flows (in MB) under number of different number of flows. Under PerNEWQUE AQM, the unresponsive UDP flow is penalized heavily since its throughput is lesser than its arrival rate. Under SFB, the UDP flow is penalized in constant way even if the number of flows is increased. Under SFQ, there is a fluctuation for penalizing UDP flows even if the number of flows is increased. Under DRR and CSFQ, the UDP throughput is higher than the PerNEWQUE AQM. From Fig.4, PerNEWQUE AQM provides high TCP throughput than the other AQMs since most of UDP flows are penalized.

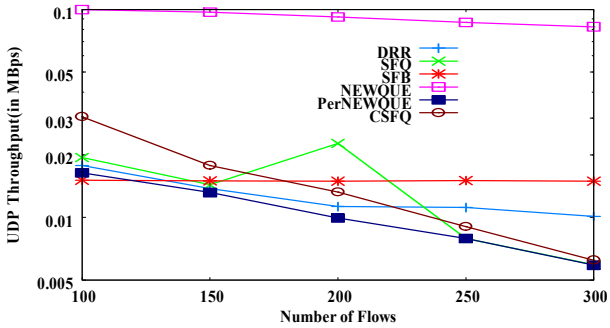

Fig.3 UDP Throughput w.r.to Number of Flows

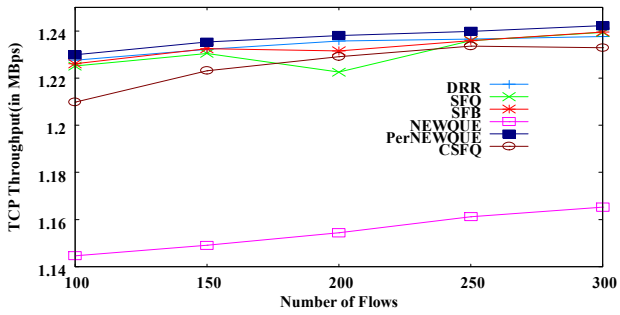

Fig.4 TCP Throughput w.r.to Number of Flows

3.2.2 Percentage Packet Loss statistics of Different AQM Schemes

Fig.5 shows UDP percentage packet loss and Fig.6 shows TCP percentage packet loss under the number of flows. It is observed that PerNEWQUE AQM has higher UDP drops rate than the other AQMs since it drops most of the unresponsive UDP flows and has less TCP drops rate than DRR, SFQ, SFB and CSFQ. Even if NEWQUE AQM has less TCP drops rate but it has less TCP throughput than PerNEWQUE AQM as shown in Fig.4. Table 1 shows the performance of PerNEWQUE AQM with NEWQUE AQM.

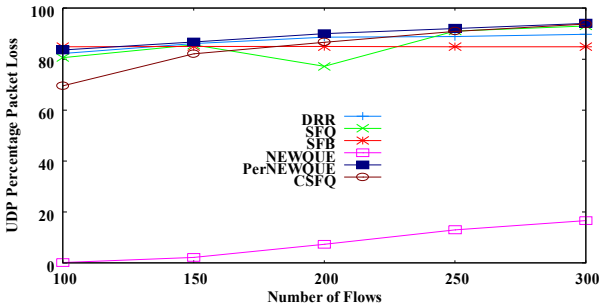

Fig.5 UDP Percentage Packet Loss w.r.to Number of Flows

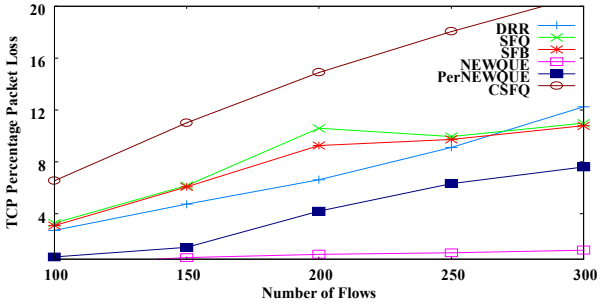

Fig.6 TCP Percentage Packet Loss w.r.to Number of Flows

3.2.3 Percentage Link Utilization of Different AQM Schemes

Fig.7 shows TCP percentage link utilization over the number of flows. Percentage link utilization is normalized by the bottleneck link capacity 10Mbps. It is observed that PerNEWQUE AQM has better TCP link utilization if the number of flows is increased.

3.3 Performance of PerNEWQUE Under Different Size of Buffers

In this simulation, the capacity of buffer between the congested routers (R's) is varied from 100 packets to 300 packets. The active flows are fixed at 200, in which 1 UDP flow and 199 TCP flows are transmitted. The UDP sources send packets at the fixed CBR rate of 0.1MB (arrival rate). The results of these simulations are shown in Table 2, Fig.8, Fig.9, Fig.10, Fig.11 and Fig.12.

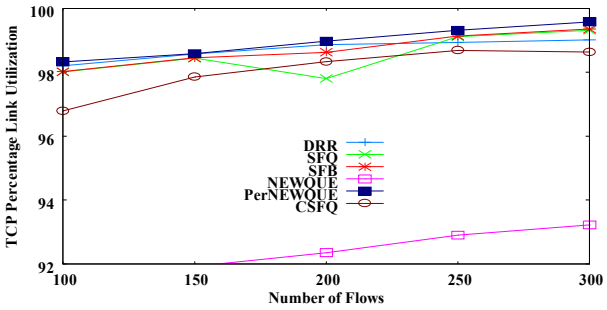

Fig.7 TCP Percentage Link Utilization w.r.to Number of Flows

Table 2 shows performance of PerNEWQUE AQM is compared with NEWQUE AQM. From Table 2, it is seen that NEWQUE AQM has less TCP throughput; highest UDP throughput and least UDP drop rates, compared to PerNEWQUE AQM. Due to these reason, NEWQUE AQM is not considered hereafter.

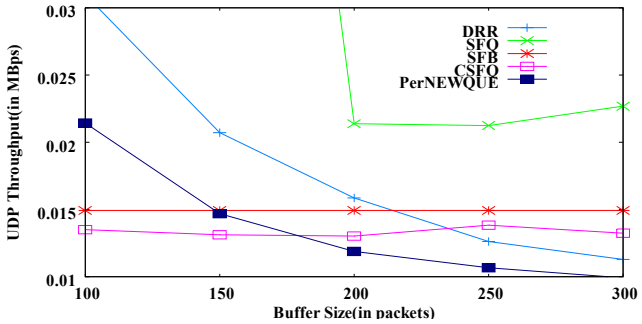

Fig.8 UDP Throughput Vs Buffer Size

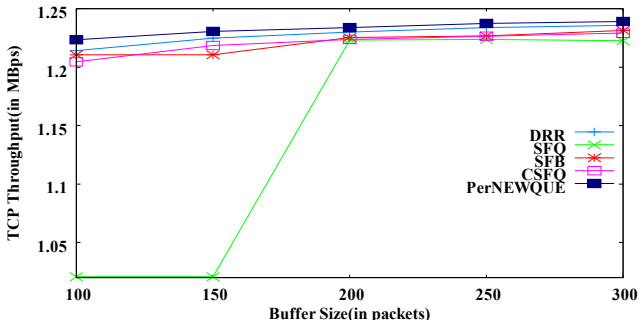

Fig.9 TCP Throughput Vs Buffer Size

Table 1 PerNEWQUE Vs NEWQUE

Number of Flows			100	150	200	250	300
PerNEWQUE AQM	Average Throughput (in Mbps)	TCP	9.83	9.87	9.9	9.91	9.93
		1UDP	0.13	0.11	0.08	0.06	0.05
	%Packet Loss	TCP	0.67	1.41	4.20	6.33	7.6
		1UDP	83.62	86.73	90	92.1	94
NEWQUE AQM	Average Throughput (in Mbps)	TCP	9.16	9.19	9.24	9.29	9.32
		1UDP	0.8	0.78	0.74	0.69	0.66
	%Packet Loss	TCP	0.05	0.63	0.87	1.00	1.20
		1UDP	0.16	2.16	7.30	13.02	16.62

Table 2 PerNEWQUE Vs NEWQUE

Buffer Size (in packets)			100	150	200	250	300
PerNEWQUE AQM	Average Throughput (in Mbps)	TCP	9.79	9.85	9.87	9.9	9.91
		1UDP	0.17	0.12	0.1	0.09	0.08
	% Packet Loss	TCP	5.07	3.79	3.79	3.62	4.19
		1UDP	78.41	85.20	88.03	89.2	90
NEWQUE AQM	Average Throughput (in Mbps)	TCP	9.31	9.29	9.26	9.25	9.24
		1UDP	0.65	0.68	0.7	0.72	0.74
	% Packet Loss	TCP	1.04	1.00	0.96	0.90	0.87
		1UDP	17.79	14.67	11.38	9.04	7.3

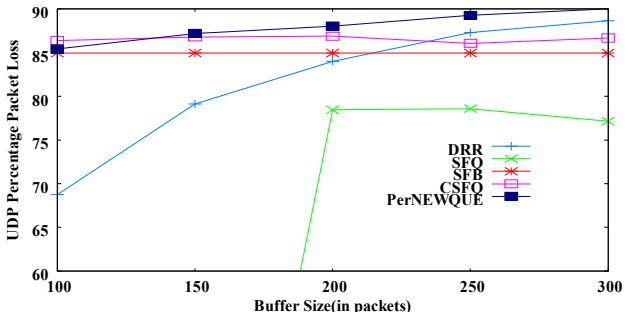

Fig.10 UDP Percentage Packet Loss Vs Buffer Size

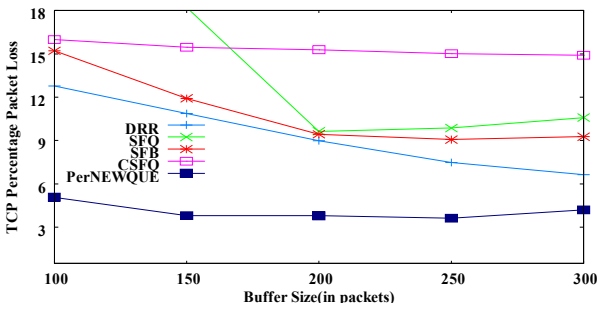

Fig.11 TCP Percentage Packet Loss Vs Buffer Size

From Fig.8, PerNEWQUE AQM has lesser UDP throughput than the other AQMs because its penalize most of unresponsive UDP packets. From Fig.9, PerNEWQUE AQM has high TCP throughput than the other AQMs used in comparison. Therefore it sends most of TCP packets through the bottleneck connection.

From Fig.10, PerNEWQUE AQM has high UDP drops rate than the other AQMs. The percentage UDP packet loss statistics of SFB is stable even different number of buffer size. Fig.11 shows PerNEWQUE AQM has much better performance than the other AQMs in terms of TCP percentage packet loss.

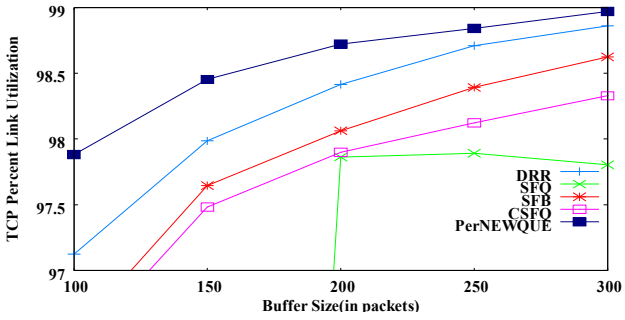

Fig.12 TCP Percentage Link Utilization Vs Buffer Size

From Fig.12, PerNEWQUE AQM has better link utilization than the other AQMs. It sends most of TCP packets through the bottleneck link (bottleneck link capacity 10Mbps).

3.4 Performance of PerNEWQUE under Different Arrival Rate ff UDP Sources

In this simulation, the capacity of buffer between the congested routers (R's) is fixed at 300 packets. The active flows are fixed at 200, in which 1 UDP flow and 199 FTP flows are transmitted. The UDP sources send CBR packets at the rate from 0.1MBps to 0.5MBps.

3.4.1 Throughput of Different AQM Schemes

Fig.13 shows throughput of UDP flows (in MB) under number of different arrival rate of UDP sources. Here, it is viewed that the PerNEWQUE AQM has lesser UDP throughput than the other AQMs. Fig.14 shows throughput of TCP flows (in MB) under number of different arrival rate of UDP sources. It is observed that PerNEWQUE has high TCP throughput than the other AQMs used in comparison.

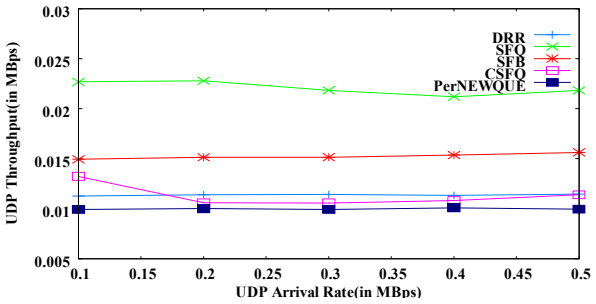

Fig.13 UDP Throughput Vs UDP Arrival Rate

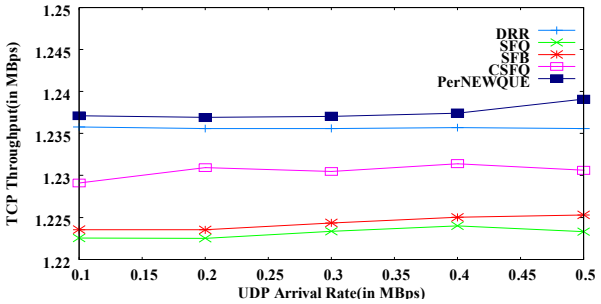

Fig.14 TCP Throughput Vs UDP Arrival Rate

3.4.2 Percentage Packet Loss Statistics of Different AQM Schemes

Fig.15 shows UDP percentage packet loss and Fig.16 shows TCP percentage packet loss under the number of different arrival rate of UDP sources. Here it is examined that PerNEWQUE AQM has high UDP drop rate and low TCP drop rate in which it does penalize the UDP packets.

3.4.3 Percentage Link Utilization of Different AQM Schemes

Fig.17 shows TCP percentage link utilization over the number of different arrival rate of UDP sources. Percentage Link Utilization is normalized by the bottleneck link capacity 10Mbps. It is examined that PerNEWQUE AQM has better TCP link utilization than the other AQMs.

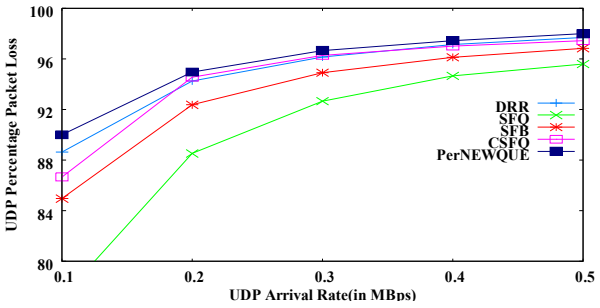

Fig.15 UDP Percentage Packet Loss Vs UDP Arrival Rate

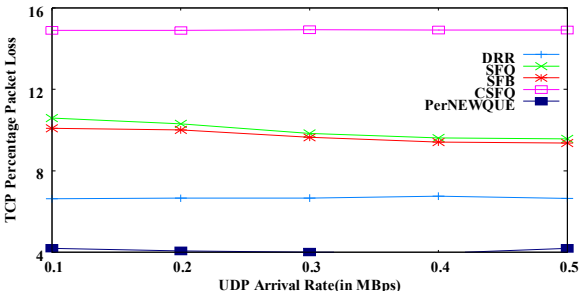

Fig.16 TCP Percentage Packet Loss Vs UDP Arrival Rate

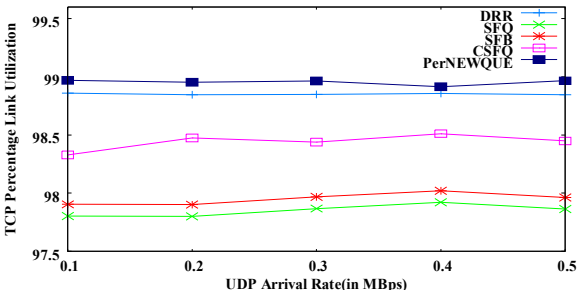

Fig.17 TCP Percentage Link Utilization Vs UDP Arrival Rate

3.5 Performance of PerNEWQUE under Multiple Unresponsive Flows

In this simulation, the traffic model mentioned in Fig.2 is followed. The flows are in an increasing order such that UDP flows are from 2 to 5 and TCP flows are 100 minus number of UDP flows. The first traffic model includes 2 UDP flows (Flow1 to

Flow2) and remaining 98 TCP flows (Flow3 to Flow100). The remaining models are defined in the same way based on number of UDP flows. Here any variables involved in all schemes are not changed except the number of sources with TCP or UDP flows. The sending rates of all UDP flows are 0.1MBps (0.8Mbps). The result of this traffic simulation is shown in Table 3.

From Table 3, TCP average throughput of PerNEWQUE is higher than all the other AQM schemes and UDP average throughput of PerNEWQUE is lower than all the other AQM schemes. This implies that the PerNEWQUE AQM penalize more number of UDP flows comparatively other schemes.

The next simulation is taken with different sending rates of UDP flows. Here the first traffic model includes 197 TCP flows (Flow1 to Flow197) and 3 UDP flows (the sending rate of Flow 198 is 0.2MBps, Flow199 is 0.1MBps and Flow200 is 0.025MBps). There are 196 TCP flows (Flow1 to Flow196) and 4 UDP flows (Flow197 to Flow200) in the second traffic model. The rates of the UDP flows are 0.2MBps, 0.1MBps, 0.05MBps and 0.025MBps, respectively. The results of these two traffic models are shown in Table 4, where Thr is throughput (Mbps) of a TCP flow and Pdrop is dropping probability of UDP flow, respectively. From Table 4, it is viewed that UDP dropping probability of PerNEWQUE AQM is higher than all the other AQMs and also average TCP throughput of PerNEWQUE is higher than all other schemes.

3.6 Performance of PerNEWQUE under Different Round-Trip Time (RTT)

In this simulation, the same network topology mentioned in Fig.2 is followed, but there is a change in propagation delay. In the first traffic model, the propagation delay is set between first and second router to 1ms and the propagation delays between the routers and end hosts 1ms, which corresponds to smaller RTT. Here this model includes 196 TCP flows (Flow1 to Flow196) and 4 UDP flows (the sending rate of Flow 197 is 0.2MBps, Flow 198 is 0.1MBps, Flow199 is 0.05MBps and Flow200 is 0.025MBps). Meanwhile, much larger RTT is considered with the propagation delay between first and second router to 200ms and the propagation delays between the routers and end hosts 1ms in the second traffic model. Here, the traffic flows are same as traffic model 1. The results of these simulations are shown in Table 5, where Thr is throughput (Mbps) of a TCP flow and Tot. Pdrop is total dropping probability of UDP flow, respectively.

From Table 5, it is observed that PerNEWQUE AQM has higher TCP throughput, lesser UDP throughput and higher UDP dropping probability than the other AQM schemes. In case of change in round trip time, the PerNEWQUE AQM penalize same amount of UDP flow packets(total dropping probability is same as both traffic models such as nearly 89%). But the other AQM schemes show instability in penalizing UDP flow packets. As it can observe that PerNEWQUE AQM gives good stability for restraining UDP flows, which outperforms other AQM schemes.

4 Overview of Other AQM Schemes

In this section provides brief introduction about different AQM schemes supports per-flow scheduling, which are compared with PerNEWQUE AQM.

4.1 Stochastic Fair Queuing (SFQ) AQM

SFQ [16] is queuing algorithm, detect and limit the unresponsive flows. It suggests that the number of queues to be considerably less than the number of flows. From that, some of the flows are belonging to same queue are treated equivalently. This allows reduce the processing of hashing computation of identifying the flow belongs to which queue and reduce the number of queues needed, but makes the disadvantage is that the flows collide with other flows will be treated unequally. One more disadvantage, queues are treated without considering individual packet length. But PerNEWQUE AQM maintain individual queue for each flows and queues are managed with byte length.

4.2 Stochastic Fair BLUE (SFB) AQM

SFB [6,7] is a FIFO queuing algorithm that identifies and rate-limits unresponsive flows based on accounting mechanisms similar to those used with BLUE [6]. SFB maintains N × L accounting bins. The bins are organized in L levels with N bins in each level. The accounting bins are used to keep track of queue occupancy statistics of packets belonging to a particular bin. Each bin in SFB keeps a marking/dropping probability Pm as in BLUE, which is updated based on bin occupancy. The observation which drives SFB is that a unresponsive flow quickly drives Pm to 1 in all of the L bins it is hashed into. If is Pm 1, the packet is identified as belonging to a unresponsive flow and is then rate-limited. As the number of unresponsive flows

Table 3. Average Throughput of Different AQM schemes

AQMs	Average Throughput of TCP and UDP Flows(in Mbps)							
	Traffic Model 1		Traffic Model 2		Traffic Model 3		Traffic Model 4	
	TCP	2UDP	TCP	3UDP	TCP	4UDP	TCP	5UDP
PerNEWQUE	9.71	0.25	9.58	0.38	9.46	0.51	9.33	0.64
DRR	9.69	0.27	9.56	0.41	9.43	0.54	9.3	0.67
CSFQ	9.44	0.49	9.25	0.67	9.08	0.86	8.93	0.98
SFB	9.67	0.31	9.52	0.47	9.4	0.6	9.23	0.77
SFQ	9.66	0.3	9.51	0.45	9.37	0.6	9.22	0.74

Table 4. Average Throughput of TCP and Dropping Probability of Individual UDP Flows

AQMs	Traffic Model 1				Traffic Model 2				
	Avg.TCP Thr (in Mbps)	3UDP P_{drop} %			Avg.TCP Thr (in Mbps)	4UDP P_{drop} %			
		UDP 1	UDP 2	UDP 3		UDP 1	UDP 2	UDP 3	UDP 4
PerNEWQUE	9.74	94.5%	89.9%	60.3%	9.66	94.5%	89.5%	79.9%	62.4%
DRR	9.7	93.9%	88.5%	55%	9.61	94.3%	88.2%	77.2%	54.9%
CSFQ	9.6	92.2%	84.3%	53.2%	9.55	93.7%	87.4%	74.8%	54.2%
SFB	9.47	90.2%	80.3%	21.4%	9.3	90.5%	80.9%	59.9%	18.9%
SFQ	9.47	89.2%	79.3%	19.4%	9.3	89.5%	78.9%	58.6%	18.5%

Table 5. Average Throughput of TCP, UDP Flows and Total Dropping Probability of UDP Flows

AQMs	Traffic Model 1			Traffic Model 2		
	Avg.TCP Thr (in Mbps)	Avg.4UDP Thr (in Mbps)	Tot. P_{drop} %	Avg.TCP Thr (in Mbps)	Avg.4UDP Thr (in Mbps)	Tot. P_{drop} %
PerNEWQUE	9.68	0.31	89.1%	9.7	0.35	88.6%
DRR	9.66	0.34	88.1%	9.54	0.42	86.1%
CSFQ	9.41	0.43	85.7%	9.24	0.67	77.6%
SFB	9.6	0.42	86.1%	8.11	1.83	38.6%
SFQ	9.58	0.41	86.1%	8.11	1.83	38.6%

increases, the number of bins which become "polluted" or have Pm values of 1 increase. Consequently, the probability that a responsive flow gets hashed into bins which are all polluted, and thus becomes misclassified, increases. Clearly, misclassification limits the ability of SFB to protect well-behaved TCP flows.

4.3 Deficit Round Robin (DRR) AQM

DRR [20] uses stochastic fair queuing to assign flows to queue. To service queues, it uses round-robin servicing with a quantum of service assigned to each queue; the only difference from traditional round-robin is that if the queue is not able to send a packet in a previous round because its packet size is too large, the remainder from previous quantum is added to the quantum for the next round. Thus, deficits are kept track off; queues are short changed in a round are compensated in the next round. It doesn't support Explicit Congestion Notification (ECN). PerNEWQUE AQM uses same hashing mechanism as in DRR. It is probability based scheme with supports of ECN.

4.4 Core-Stateless Fair Queuing (CSFQ) AQM

CSFQ [11], is a rate based scheme, uses a distributed algorithm in which only edge routers maintain per-flow state, while core routers do not maintain per-flow state but instead utilize the per-flow information carried via a label in each packet's header. This label contains an estimate of the flow's

rate; it is initialized by the edge router based on per-flow information, and then updated at each router along the path based only on aggregate information at that router. It uses FIFO queuing scheduling with probabilistic dropping on input. The probability of dropping a packet as it arrives to the queue is a function of the rate estimate carried in the label and of the fair share rate at that router, which is estimated based on measurements of the aggregate traffic. The simulation results detail how in an environment with a wide variation in number of sources and finite buffers, the performance suffers.

4.5 NEWQUE AQM

NEWQUE AQM [18] uses the flow arrival rate, the link capacity and link utilization history to manage congestion. Here the flow arrival rate is calculated using fixed weight exponential averaging method. Equation (2) is used to estimate the incoming total flow arrival rate.

$r_{new}(t) = (1 - e^{-T/K})\, l/T + e^{-T/K}\, r_{old}(t)$ (2)

where, T is the inter-arrival time between the current and the previous packet and K=0.1. It maintain single marking probability p, when the total flow arrival rate is greater than or equal to the link capacity, this probability is incremented, and when the total flow arrival rate is less than the link capacity, this probability is decremented and also when the link is idle, it is decremented. This AQM not effectively detect and penalize unresponsive flows in the Internet

5 CONCLUSIONS

In this paper, NEWQUE with Per-flow Scheduling (PerNEWQUE) AQM scheme supporting ECN is proposed. It is simple active queue management; it penalizes most of the unresponsive UDP flow packets. PerNEWQUE is compared with other AQM schemes, SFB, SFQ, DRR, CSFQ and NEWQUE. The performance metrics used for the comparison are throughput, percentage link utilization and packet loss. The simulation experiments showed that the planned AQM method maintains better throughput and less percentage packet loss for TCP flows and also reduces rate of unresponsive UDP flows. Finally, there are different areas in which such as uncertain routing topologies and different TCP versions, the method presented here could be extended.

References:

[1] Anjum F.M. and Tassiulas L., Fair bandwidth sharing among adaptive and non adaptive flows in the internet, in Proc. IEEE INFOCOM, New York, 1999, pp. 1412–1420.

[2] Braden B., Clark D., Crowcroft .J, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L.Peterson, K. Ramakrishnan, S.Shenker, J. Wroclawski, and L. Zhang, Recommendations on queue management and congestion avoidance in the Internet, RFC 2309, Apr 1998.

[3] Chatranon G., Labrador M.A., and Banerjee S., A survey of TCP-friendly router-based AQM schemes, Comput. Commun., 27,(15), 2004, pp. 1424–1440.

[4] Cidon I., Guerin R., and Khamisy A.,. Protective Buffer Management Policies, IEEE/ACM Transactions on Networking, 2(3), 1994, pp.36-37.

[5] Fall K. and Floyd S. (1997), Router Mechanisms to Support End-to-End Congestion Control, ftp://ftp.ee.lbl.gov/papers/ collapse.ps.

[6] Feng W., Kandlur D., Saha D., and Shin K.G., The Blue active queue management, IEEE/ACM transactions on Networking, vol 10, No.4, August 2002.

[7] Feng W,Shin K, Kandlur D and Saha D., Stochastic Fair Blue: A Queue Management Algoirthm for Enforing Fairness, Proceedings of INFOCOM 2001, April 2001.

[8] Feng W., Kandlur D., Saha D., and Shin K.G., Techniques for eliminating packet loss in congested TCP/IP networks, Univ. Michigan, Ann Arbor, MI, Tech. Rep. UM CSE-TR-349-97, Oct. 1997.

[9] Floyd S., TCP and explicit congestion notification, Comput. Commun. Rev., vol. 24, no. 5, pp. 10-23, Oct. 1994.

[10] Floyd S., Congestion Control Principles, RFC 2914, September 2000.

[11] Ion Stoica, Scott Shenkaer and Fellow, A Scalable Architecture to approximate fair bandwidth allocation in high speed networks, IEEE/ACM Transactions on networking, vol. 11, No. 1 February 2003.

[12] Jacobson V., Congestion avoidance and control, in Proc. ACM SIGCOMM, Aug 1998, pp. 314-329.

[13] Lin D., and Morris R., Dynamics of random early detection, in Proc. ACM SIGCOM, Sep 1997, pp-127-137.

[14] Mascolo S., Congestion control in high-speed communication networks, Automatica, vol. 35, no. 12, 1999, pp. 1921–1935.

[15] McCanne S. and Floyd, The network simulator NS2 (2000). [Online]. Available: http://www.isi.edu/nsnam/ns/

[16] McKenny P, Stochastic fairness queuing in Internetworking: Research and Experience, vol.2, Jan. 1991, pp. 113-131.
[17] Ramakrishnan K., S. Floyd and D. Black, The addition of Explicit Congestion Notification (ECN) to IP, RFC 3168, Sep. 2001.
[18] Santhi V., Dr. A.M. Natarajan, A New Approach to Active Queue Management for TCP with ECN, in Proc. IEEE International conference on Advanced Computing (ICAC 2009), Dec 2009, pp. 76-81.
[19] Santhi V., Dr. A.M. Natarajan, NEWQUE with Per-flow scheduling: Performance improvement of Active Queue Management, Pro. IEEE International conference on Computational Intelligence and Computing Research, Dec 2010, pp. 609-615.
[20] Shreedhar M and Varghese G, Efficient fair queuing using deficit round robin, IEEE/ACM Transactions on Networking, 1996, 4, pp. 375-385.

Biographies

V. Santhi received the B.E. degree in Computer Science and Engineering from Madras University with distinction. She received M.E. degree in Computer Science and Engineering from Anna University Chennai with distinction. She has 7 years of teaching experience. She is currently pursuing doctoral research in computer networks management. At present she is working as Assistant Professor in Computer Science and Engineering Department at PSG College of Technology, Coimbatore, India.

Dr. A. M. Natarajan is the CEO & Professor at Bannari Amman Institute of Technology, Coimbatore, India. He received the B.E. degree in Electrical Engineering, and M.Sc.(Engg.) in Applied Electronics and Servo Mechanism from Madras University and Ph.D degree in System Engineering from Madras University. He has 45 years of teaching experience. He received the “Best Engineering college principal award in India for the Year 2000” from Indian Society for Technical Education, New Delhi. He has published more than 100 papers in International and National journals and conference proceedings. He has published 10 books. His areas of research include systems engineering and computer networks.