FLoadAutoRED: An Active Queue Management Scheme to Prevent Congestion in a Dynamically Varying Traffic in IP Networks

CHAPTER 3

METHODOLOGY – PHASE I

3.1 METHODOLOGY

3.2 FEATURES OF FAutoREDwithRED – METHODOLOGY

3.3 EXPERIMENTATION

3.4 CONCLUSION

3.1 METHODOLOGY

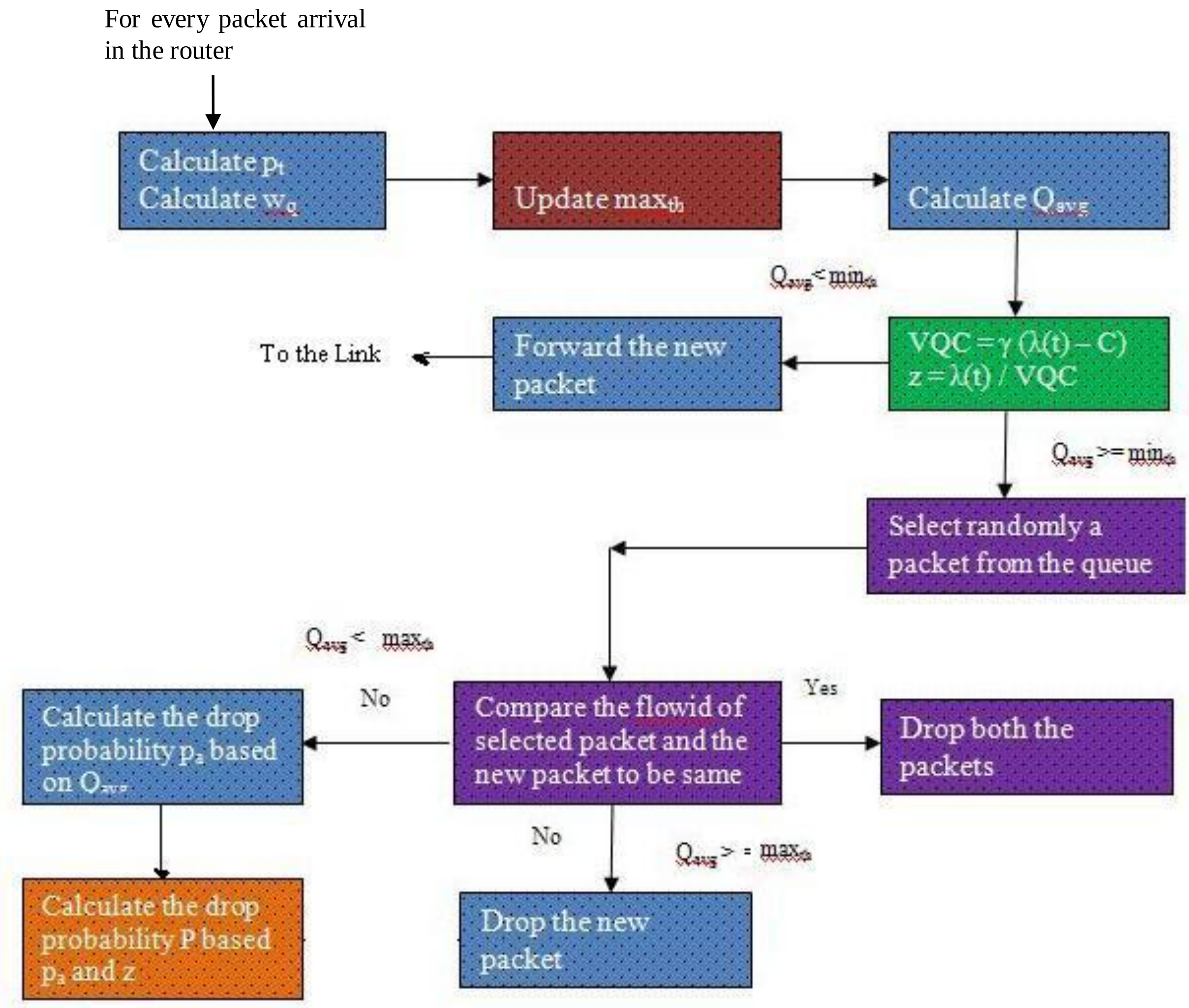
As discussed in the previous section, AutoREDwithRED does not achieve the objective of the work are to provide better Quality of Service in terms of, fairness, high link utilization with minimum delay, minimum queue loss, stabilized and moderate network queues in a dynamically varying traffic in IP networks. Therefore, to achieve these objectives, improvements in AutoREDwithRED are done in four stages as in Figure 3.1.

In the first stage, the methodology brings in fairness in IP network with a varying traffic flow. This is attained by using the flow information while maintaining the queue size within a limit. The flow information takes care of the unresponsive flows and brings in fair queuing. The methodology helped to remove bias but does not project a reduced packet drop rate.

In the second stage, the methodology uses a dynamic max_{th} to remove the parameter tuning problem. The packet drop rate is controlled by dynamically varying the maximum queue threshold max_{th}. However, this queue based AQM still introduced a slower adaptability. Therefore, this mechanism controlled the packet drop rate but does not attain a moderate average queue size and does not show a better adaptability.

Hence, in the third stage, input rate is used as a secondary metric which results in a moderate queue size and the system exhibits better adaptability (Responsive Performance).

The fourth stage, packet drop probability is updated to improve the overall utilisation and minimizes the queuing delay. Hence, the goal of this methodology is to maximize the link utilization and minimize queuing delay. This also reduces packet drop rate.



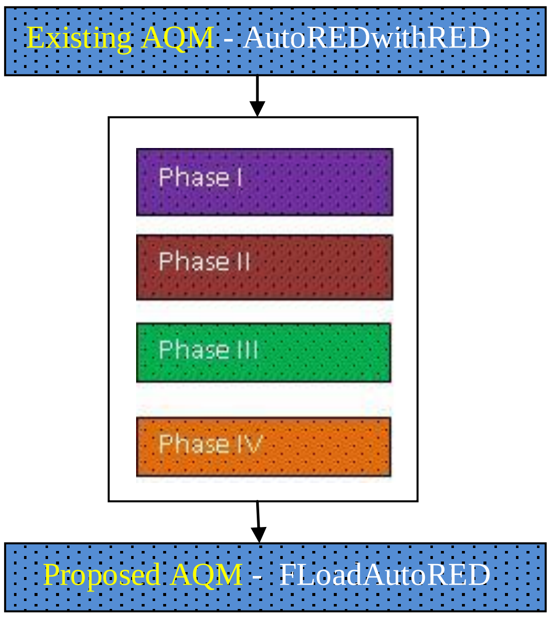


Figure 3.1 Block diagram of the Methodology - Four Phases

3.2 FEATURES OF FAutoREDwithRED (AutoREDwithRED WITH FAIRNESS) ALGORITHM – METHODOLOGY

AutoREDwithRED AQM could not easily identify which flows have received a significant fraction of the recently dropped packets. AutoREDwithRED AQM accepts a packet from one flow causes higher drop probability for future packets from other flows, even if the other flows consume less bandwidth. Therefore, additional mechanism is proposed to control the throughput of such flows during periods of congestion. Therefore, FAutoREDwithRED AQM provides a mechanism to identify flows using a large share of the total bandwidth rather than just identifying the level of congestion. This mechanism avoids a low-bandwidth TCP from not receiving its fair share and improves the fairness among the flows.

FAutoREDwithRED methodology identifies a flow that has received a large share of the bandwidth. The proposed methodology drops packets from flows that use the large share of the bandwidth. The proposed algorithm - FAutoREDWithRED enforces the concept of queue-based and uses the flow information to bring in fairness among the varying flows. The flow information of the packets helps in identifying the misbehaving flows receiving a large share of the bandwidth and brings in fair queuing.

The misbehaving flow is identified by simple comparison of the flow id of the packet from the incoming traffic and flowid of one of the packets in the queue. The same flow id indicates that packets belong to the misbehaving flows. The misbehaving flows are more likely to arrive more numerously and more likely to be chosen for comparisons than the other flows. Therefore, in a dynamically varying traffic, misbehaving flow packets are dropped more often than the other flows. So the packets belonging to the same flow and the packets that occupy the buffer very often and continuously are not allowed.

Fairness metrics is used to determine whether users or applications are receiving a fair share of system resources like buffer and bandwidth. Generally, while sharing resources that are characterized by low level of fairness with high average throughput will result in low stability in the service quality of other users. If this instability is severe, it may result in unhappy users. Fairness is an important factor to be improved in the performance efficient AQM AutoREDwithRED.

FAutoREDwithRED initialises the parameters w_q, min_{th}, max_{th} and max_p with user given values. w_q is a weight parameter that averages the queue size by filtering out transient congestion. Two thresholds on the buffer is defined in terms of, a minimum threshold min_{th} and a maximum threshold max_{th}. The two thresholds min_{th} and max_{th} are expressed as number of packets. max_p is a maximum packet drop probability expressed in decimal value.

As the packet arrive at the router, the proposed algorithm calculates the weight parameter w_q dynamically with the congestion characteristics, traffic characteristics and queue normalisation as follows:

$$w_{q,t} = p_t \cdot (1 - p_t) \times \frac{2(5.923 + |Q_t - Q_{avg,t-1}|)}{\ln(5.923 + |Q_t - Q_{avg,t-1}|)} \times \frac{1}{bs}$$

w_q is calculated using p_t the probability of congestion at time t, Q_t is the instantaneous queue size and $Q_{avg,t-1}$ is the average queue size at time t-1.

Then it calculates the average queue size using the dynamic parameter w_q, average queue size Q_{avg} and instantaneous queue size Q_t.

$$Q_{avg} = (1 - w_q) \cdot Q_{avg} + w_q \cdot Q_t$$

As in Table 3.1 and Figure 3.2, average queue size is compared with the queue thresholds for every arriving packet. If average queue size is less than min_{th}, every arriving packet is queued. If average queue size is greater than max_{th}, every arriving packet is dropped. This results in queue size below max_{th}.

When the average queue size is greater than min_{th} and less than max_{th}, every arriving packet is compared with a randomly selected packet from the queue for their flow id. If they have the same flow id, both are dropped otherwise the arriving packet is dropped with a probability based on the average queue size.

The AQM drops each packet that arrives when the average queue size Q_{avg} is between min_{th} and max_{th} is as follows:

$$p_b = \frac{Q_{avg} - min_{th}}{max_{th} - min_{th}} \cdot max_p$$

$$p_a = \frac{p_b}{1 - count \cdot p_b}$$

Table 3.1 Pseudocode of FAutoREDwithRED

```
For every packet arrival {

Calculate w_q
Calculate Q_avg
if (Q_avg< min_th)
          Forward the new packet
Else
     Select randomly a packet from the queue for their flow id
     Compare arriving packet with a randomly selected packet.
     If they have the same flow id
         Drop both the packets
     Else
       if (Q_avg <  max_th)
                    Calculate p_b = max_p . (Q_avg - min_th) / (max_th - min_th)
                    Calculate p_a = p_b / (1 – count . p_b)
                    Drop arriving packet with a probability p_a
       Else
                    Drop the new packet
}
```

As average queue size Q_{avg} varies from min_{th} to max_{th}, the packet-dropping probability p_b varies linearly from 0 to max_p, final packet-dropping probability p_a increases slowly as the count increases since the last dropped packet. This algorithm works fine as the parameters are well tuned automatically and parameterized based on the current traffic and congestion of the network.

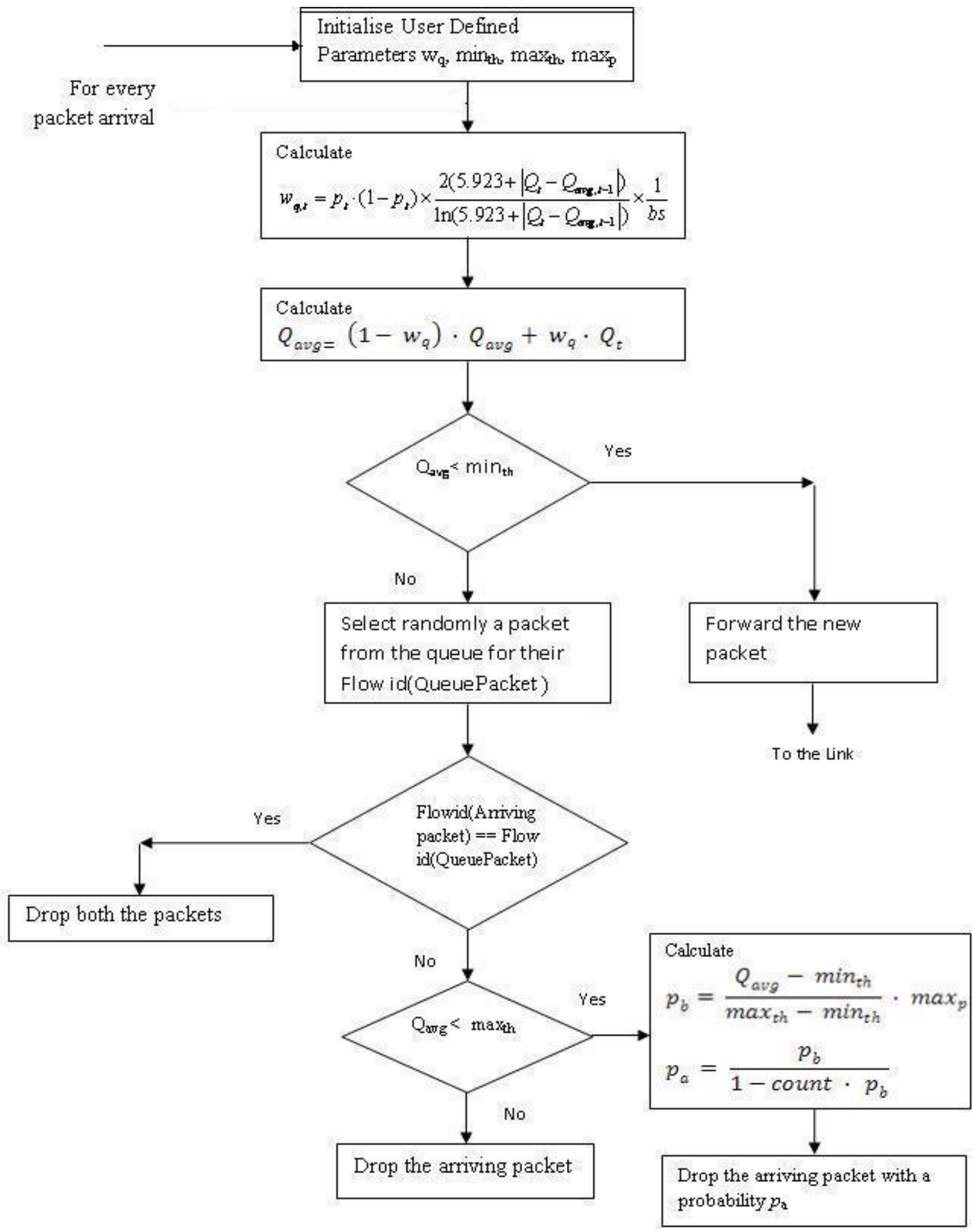


Figure 3.2 Flowchart of FAutoREDwithRED

3.3 EXPERIMENTATION

In this section, the packet-simulator ns-2 is used to simulate the FAutoREDWithRED algorithm. The simple network topology used for the experimentation is shown in Figure 3.3.

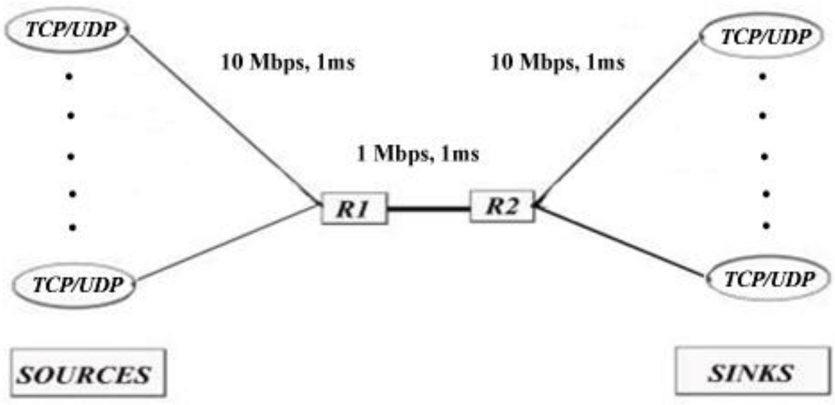


Figure 3.3 Simple Network Topology

It uses a single link of capacity 1Mbps that drops packet according to the AQM algorithm. As in a real-time scenario, the UDP hosts send packets at a constant bit rate of 2 Mbps. 32 TCP flows and 1 UDP flow are considered in the network with parameters set as in Table 3.2.

Table 3.2 Parameter Setting of FAutoREDwithRED

Type of Sources	Buffer Size	Link Capacity	Link Delay	Packet Size	Minimum Threshold min_{th}	Maximum Threshold max_{th}	max_p	w_q
TCP (FTP) UDP (CBR)	300 Packets	1Mbps	1ms	1 Kbytes	100 Packets	200 Packets	0.02	0.002

3.3.1 PERFORMANCE METRICS

The performance metrics considered for evaluation are

i) Fairness

ii) Link Utilisation

iii) Packet Dropping Rate

iv) Average Queue Size

v) Average Queue Size Stability

i. Fairness

Jain's fairness rates the fairness of a set of values where there are *n* users and x_i is the throughput for the i^{th} connection. The result ranges from $\frac{1}{n}$ (worst case) to 1 (best case), and it is maximum when all users receive the same allocation. This index is $\frac{k}{n}$ when *k* users equally share the resource, and the other $n - k$ users receive zero allocation.

This metric identifies underutilized channels and is not unduly sensitive to a typical network flow patterns. It is bounded between 0 and 1. The relationship is that Higher index indicates a More Fair and Higher variance indicates a Less fair .

$$Fairness\ Index = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2}$$

where x_i is the normalized throughput (in Kbps) of the i–th flow and *n* is the number of connections.

Normalised throughput x_i is calculated as follows:

$$x_i = \frac{t_i}{o_i}$$

Where t_i is given throughput and o_i is fair throughput.

ii. Link Utilization

The channel utilization, also known as bandwidth utilization efficiency, in percentage is the achieved throughput related to the net bitrate in bit/s of a digital communication channel. Network utilization is in percentage of current network traffic to the maximum traffic that the link can handle.

$$Utilisation = \frac{TotalNo.of PacketsReceived}{SimulationTime} \times \frac{8}{1000 \times Bandwidth} \%$$

iii. Packet Dropping Rate

Packet drop rate is the indication of the percentage of packets that are not received at their destination. Packet loss is possible in IP networks because packets from many sources are temporarily stored in a queue prior to transmission over an outgoing link in a router. Since the queue is finite, an arriving packet is lost (or dropped) in the network if there is no space left in the queue when the network is congested. For real-time traffic, loss also occurs when a packet arrives at its destination but is too late to be played out as part of a continuous bit stream. Packet loss is also expressed as a probability value, i.e. the probability that an individual packet will be discarded by the network. Packet loss can cause severe damage to QoS for real-time audio and video traffic.

The packet drop rate is expressed as,

$$Packet\ Drop\ Rate = \frac{DroppedPackets}{TransmittedPackets} \times 100$$

iv. Average Queue Size

The average queue size is defined as the ratio between instantaneous queue size and time.

$$AverageQueueSize = \frac{Sum\ of\ Instantaneous\ Queue\ Size}{No.of InstantaneousQueueSize}\ packets$$

v. Average Queue Size Stability

The stability is measured in terms of standard deviation of queue size and it is expressed as,

$$StdDev.AvgQueueSize = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(QSize_i - AvgQSize)^2}\ packets$$

The proposed algorithm FAutoREDwithRED is compared with other performance efficient algorithm AutoREDwithRED. The comparison of average queue size is shown in Table 3.3. AutoREDwithRED maintains a higher queue size. However AutoREDwithRED keeps the average queue at a stable point with a higher packet drop rate. The proposed algorithm FAutoREDwithRED shows a lower queue size compared to AutoREDwithRED as in Table 3.3. The queue stability of FAutoREDwithRED AQM is slightly improved compared to AutoREDwithRED.

Table 3.3 – Comparison of Avg. Queue Size and Std. Dev. of AQMs

AQMS	Average Queue Size (# Packets)	Std. Dev. Avg. Queue Size
AutoREDwithRED	190.4	3.5
FAutoREDwithRED	**90**	**3.2**

Queue based AQM AutoREDwithRED is unable to penalize unresponsive flows as the packets dropped from each flow over a period of time is almost the same. In a real time scenario, the misbehaving traffic like UDP takes up a large percentage of the link bandwidth and starves out TCP friendly flows as in Table 3.4**.** FAutoREDWithRED identifies and penalizes misbehaving flows and UDP uses only 21% of the link. For the existing AQM, the TCP throughput is reduced when UDP flow is introduced in the traffic.

Table 3.4 – Comparison of Utilisation of AQMs

AQMs	In %		
	CBRutilisation	TCPutilisation	OverallUtilisation
AutoREDwithRED	97.29	1.70	99.00
FAutoREDwithRED	**21.25**	**75.78**	**97.04**

The proposed AQM indicates the improvement of TCP throughput by limiting the UDP throughput to 212 Kbps which is around 21 % of the link capacity. The total TCP throughput is increased from 2 Kbps to 750 Kbps in FAutoREDwithRED. To enforce the degree to which FAutoREDwithRED gains fair bandwidth allocation, fairness and variation are introduced and the results are shown in the Table 3.5. In the proposed AQM, most of the TCP packet dropping is caused by random discard while matches are responsible for UDP packet drop.

Table 3.5 Comparison of Fairness of AQMs

AQM	Fairness	FAutoREDwithRED Improvement over AutoREDwithRED (%)	Variation	FAutoREDwithRED Improvement over AutoREDwithRED (%)
AutoREDwithRED	0.5	16.6	0.22	95.45
FAutoREDwithRED	**0.6**	**-**	**0.01**	**-**

In this real time scenario, a misbehaving flow with a high arrival rate and high buffer occupancy incurs high packet dropping mostly due to matches. The packets that belong to responsive flow or a normal traffic flow do not match to a higher level and therefore do not get dropped due to mismatch.

Inspite of the existence of UDP flows, FAutoREDwithRED outperforms the AutoREDwithRED AQM for link utilization of TCP traffic.

The AutoREDwithRED AQM almost takes up the entire bandwidth for UDP flow compared to its actual CBR fair share. In the proposed algorithm, CBR throughput is only 21% of the bandwidth. In the existing AQM, TCP utilization is almost very less but this algorithm shows a good fair utilization of 75%. In case of packet drop rate shown in Table 3.6, the FAutoREDwithRED gives a lower packet drop rate compared to other AQM.

Table 3.6 - Comparison of Packet Drop Rate of AQMs

AQMs	Packet Drop Rate	FAutoREDwithRED Improvement over AutoREDwithRED (%)
AutoREDwithRED	9.74	0.05
FAutoREDwithRED	**9.69**	-

3.4 CONCLUSION

The FAutoREDwithRED algorithm achieves an improvement as a simple, stateless algorithm and achieves flow isolation and/or approximate fair bandwidth allocation in the context of the Internet. Thus, an improvised AutoREDwithRED differentially penalizes "unfriendly" or "unresponsive" flows and brings in fairness in an IP network of dynamically varying traffic.

The adaptive flows are protected from non-adaptive flows in a mixture of traffic. It is obtained with controlled packet loss. It is achieved with well dynamically tuned parameter and flow information. The packet dropping scheme discriminates against the unresponsive flows resulting in fair congestion indication. The proposed FLoadAutoRED AQM scheme inherits the advantages of the queue length based AQM and uses flow information to satisfy the QoS requirements of the network in a varying traffic.

The proposed AQM achieves a reduced packet loss compared to AutoREDwithRED. However, it still requires a lower packet loss. The proposed algorithm maintains a lower queue size rather than a moderate queue size. FAutoREDwithRED still faces the problem of parameter tuning problem. Therefore, the proposed AQM requires an improvement in achieving better QoS. The next section considers the problem and the improvements are suggested.

Publication

- **International Conferences**

a. "FAutoREDWithRED: An Algorithm to Increase the Fairness in Internet Routers", Proceedings of International Conference on Control, Robotics and Cybernetics (ICCRC 2011), pp. 39-43, 2011 and Advanced Materials Research Journal, ISSN: 1662-8985, Vols. 403-408, pp. 3946-3952, November 2011.

b. "FAutoREDWithRED: An Algorithm to Reduce Queue Oscillation in Internet Routers", Proceedings of 4th Annual Conference of ACM Bangalore Chapter, ACM Compute'11, ISBN: 978-1-4503-07505, Article 20, 2011.

c. "FAutoREDWithRED: To Increase the Overall Performance of Internet Routers", Proceedings of the International Conference on Advances in Information Technology and Mobile Communication, AIM 2011, ISBN: 978-3642205729, Vol. 147, pp. 196 – 202, Springer LNCS-CCIS.