DOI: 10.21917/ijct.2010.0019

COMPARISON OF RED AND E-RED: A STABILIZING AQM SCHEME FOR CONGESTION CONTROL AND AVOIDANCE

T. Revathi[1] and M.Sushmera[2]
Department of Information Technology, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India
Email: trevathi@mepcoeng.ac.in[1], Sushmera@gmail.com[2]

Abstract
In today's globally integrated Emerging technologies, the current Internet protocols uses packet-level mechanisms having end to end links to detect and react to congestion. But the regulation of best-effort traffic and congestion control is largely left to the Queue Management. Drop-tail, Random Early Detection (RED) and Random Exponential Marking are quite effective in reducing packet losses under certain level of congestion and are insufficient to ensure overall network stability. At times of high congestion, not only does the end user experience very poor performance, but also a lot of network resource is wasted due to retransmissions and aborted connections. Hence, it is required to propose a new efficient AQM (Active Queue Management) scheme known as Exponential Random Early Detection (E-RED) to reduce the packet loss even at more congestion time. The proposed probability calculation approach for congestion control in an ERED framework is implemented with probability marking capabilities. We had implemented ERED algorithm and compared with the RED algorithm during packet loss in congestion control at various time intervals. The experimental result shows that our proposed approach E-RED outperforms RED. The experimental result also shows the packet loss for the nodes at various time intervals, Queue sizes and bandwidth.

Keywords:
Congestion Control, RED, Early Congestion Notification, E-RED, AQM Schemes.

1. INTRODUCTION

The rapid growth of the Internet and today's increased demand to use the Internet technologies for wide variety of real time applications are mostly time-sensitive voice and video applications which necessitates the user's to newly design and utilization of new Internet architectures. More effective congestion control algorithms, generally called as Active Queue Management Schemes (AQMS) has been an active area of research in the Internet community for facilitating Quality of Service (QoS) support for the congestion to the traffic sources. Designing the AQM Scheme is mainly to handle congestion for boosting system performance, while providing better stability, fairness and responsiveness to dynamically changing workloads and also this scheme operates in the interior of the network.

Congestion control algorithm in an distributed network can be classified based on the type and amount of feedback received from the network i.e. whether the link in the network is due to Loss, delay, single-bit or multi-bit explicit signals, high bandwidth-delay product networks, lossy links, fairness advantage to short flows and min-max potential delay. Congestion control can be viewed as a primal algorithm, Dual algorithm and primal-dual algorithm [1]. In primal algorithm, the users adapts x_r the source rates dynamically based on the route prices q_r, and the links select a static law to determine the link prices p_l directly from the arrival rates at the links with respect to y_l . In dual algorithm, the links adapt the link prices dynamically based on the link rates, and the users select a static law to determine the source rates directly from the route prices and the source parameters. Third congestion control algorithm is the primal-dual algorithm, which is the combination of both the primal and dual algorithm in which the user end adapts for the primal algorithm and link end adapts for the dual algorithm.

In our proposed method, we had successfully implemented AQM scheme based algorithm namely the RED and ERED algorithm which makes the real time applications more effective, robust, scalable and flexible to all users. ERED proposes a solution along with the support of ECN for detecting incipient congestion in the network. The proposed probability calculation approach for congestion control in a ERED framework is implemented with marking capabilities. The design of ERED algorithm avoids the necessity of any parameterization or tuning, apart from linguistic interpretation of the system behavior. We have successfully used the reported strength of virtual queue and exponential packet marking and have addressed the limitations of existing RED algorithms implementation in an ERED framework. The experimental result shows E-RED outperforms RED and also shows the various packet loss and packet delay for each and every node at various time intervals. In Business Context, our proposed algorithm can be used in the mail server and internet to provide a queuing method for the incoming packets.

This paper has been formulated as follows, followed by this section, it discusses in detail about the Literature Review. Section 3 enumerates the implementation of the proposed methodology. In section 4 the paper explains about the experimental results of our approach. Finally Section 5 discusses the conclusion of our research approach.

2. LITERATURE REVIEW

Most of the AQM algorithms, like RED, ERED relate the price to the queue length or queuing delay, and thus are dynamic link algorithms. This relationship makes the motivation to do research work on the primal-dual algorithms ie to directly relate the used end and link end as xr to qr and pl to yl. The detailed study on these algorithms can be viewed at [2].

The traditional congestion notification mechanism whereby routers drop packets to signal congestion, with ECN, routers have the capability to mark packets to indicate congestion [3]. Here marking the packets refers to a bit in the packet header which ranges from zero to one when the router detects incipient congestion. Each receiver echoes the marks to its source and the source is expected to respond to each mark by reducing its transmission rate. Here in our approach, we focus on the mechanism by which marking is performed at the routers.

Specifically, we compare the AQM schemes where a router marks packets based on the real queue length to schemes where the router marks packets based on the queue length of a virtual queue (VQ). *The AVQ [6]* algorithm maintains a virtual queue whose capacity is less than the actual capacity of the link. When a packet arrives in the real queue, the virtual queue is also updated to reflect the new arrival. Packets in the real queue are marked/dropped when the virtual buffer overflows. The virtual capacity at each link is then adapted to ensure that the total flow entering each link achieves a desired utilization of the link. In the virtual queue there is no actual enqueueing and dequeueing of packets is necessary just the virtual queue length is tracked.

The proposed Exponential-RED (E-RED) mechanism is mainly aimed to drop connection packets at severe congestion. The main characteristics of this scheme are to improve the throughput, maximize the queue size and resource utilization at times of intense network congestion. Under this framework, the packet marking probability will be an exponential function of the length of a virtual queue whose capacity is slightly smaller than the link capacity.

The two main criteria for assessing the performance of the network are the Queuing delay which should be maintained at a small fraction of the propagation delay in the network and second is Robustness, to maintain a closed loop performance in the network.

3. IMPLEMENTATION OF THE PROPOSED METHODOLOGY

Our proposed approach is implemented using network simulator, NS-2.28, Tool Command Language (TCL) and C++ applications [4]. Using TCL as a simulation tool, the network configuration, nodes and the packets loss and delay in the network environment are simulated.

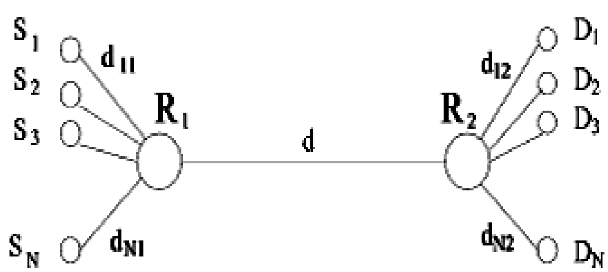


Fig.1. Proposed Network Topology used for packet simulations in ns-2

We use a packet model with round-trip delays and the network consists of SN source nodes and DN Destination nodes as shown in Fig.1. The source nodes from SN and DN are connected by a 2 Mbps link with a one-way propagation delay of 10 msec. The dN1 and d N2 is the delay between the user end and link end of the source node and the destination node. R1 and R2 are the two routers which marks the packets based on the real queue length to schemes where the router marks packets based on the queue length of a virtual queue.

The implementation of our approach consists of following steps:

1. Creation of Nodes as a Simulation environment
2. Attaching the application to the created Nodes
3. Implementation of Existing RED algorithm
4. Implementation of Proposed ERED algorithm
5. Comparison of RED and ERED algorithm
6. Performance Evaluation

Step 1 and Step 2: Creation of Nodes as a Simulation environment

The necessary source and destination node information are passed thorough the applications and transfers the applications using any of the file transfer protocols namely FTP or a CBR applications.

Step 3: Implementation of Existing RED Algorithm

RED algorithm was first proposed by Sally Floyd and Van Jacobson in Active Queue Management (AQM) whose main goal is to provide congestion avoidance [5] by controlling the average queue size even in the absence of cooperation from transport-layer protocols [6]. A router that implements RED uses two threshold values to mark positions in the queue: min, the minimum threshold and max, the maximum threshold. The algorithm for the existing RED algorithm is of the following steps.

Let avg indicates the discard probability

minth: minimum threshold for queue

maxth: maximum threshold for queue

Maintain the running average of queue length by following the set of conditions

If avg < minth do nothing

- ✓ low queuing, send the packets through the queue
- ✓ Set Marking probability as zero

If avg > maxth, drop the packet

- ✓ Protection from misbehaving sources
- ✓ Marking probability ranges linearly from 0 to maxp.

Else drop the packets with a probability proportional to queue length

- ✓ Notify sources of incipient Congestion

Some of the Limitations of the existing system –RED achieves lesser throughputs and it will not support heavy load of networks [7] [10]. Also based on the implementation work, it was observed that, at times of high congestion, not only do end users experience very poor performance, but a lot of network resource is wasted due to retransmission of packets. Thus it cannot handle high load congestion and it has some limitations in terms of throughput, delay, and utilization. RED also has several parameters which are hard to tune, needs adjustment according to the traffic flow and needs more memory allocation at RED routers.

Step 3.1: Packet Marking Probability Calculations

The most complex part of RED is the computation of discard probability, p. A new value of p is to be calculated for each packet and the value depends on current queue size and threshold values. Instead of using the actual queue size at any instant, RED computes a weighted average queue size, avg to determine the discard probability. The value of avg is an exponential weighted average, updated each time a packet arrives according to following equation:

avg = (1- b) * old_avg + b * current_queue_size (1)

where b has the values ranges between 0 and 1.

The marking probability is focused as a piecewise linear function of the average queue length, which is estimated using exponential averaging with weight factor.

Step 4: Implementation of Proposed ERED Algorithm

A Queue Management Mechanism known as Exponential Random Early Detection (E-RED) is built to control the network at times of heavy congestion in which Virtual Queue (AVQ) would improve the throughput, utilization and delay resource characteristics [8] [11]. This algorithm introduces and analyses a decentralized network congestion control algorithm which has dynamic adaptation at both user ends and link ends, a so called general primal-dual algorithm. In this scheme, the incoming packets are marked based on exponential function of the length of the virtual queue whose capacity is lesser than the link capacity. We used the reported strength of virtual queue and exponential packet marking and have addressed the limitations of existing RED algorithms implementation in an ERED framework.

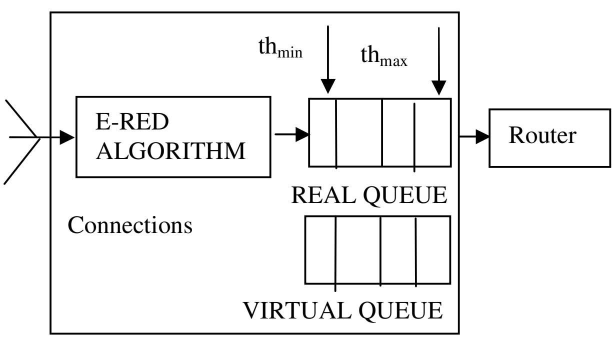


Fig.2. Architectural Design of ERED Algorithm

The Fig.2 shows the Architectural Design of ERED Algorithm which consists of incoming packets which are marked based on the exponential function of the length of the virtual queue and Real queue with two thresholds as th_{min} and th_{max}. The ERED Algorithm is mainly designed for avoiding and controlling the packet congestion during low and high traffic which measures the maximum and minimum queue size. Depending upon the virtual queue size, the algorithm will enqueue the packet, randomly drop the packet or dequeue the packet.

The algorithm for the ERED is explained in detail below.

Initialize the virtual queue size bl to 0

Initialize the real queue size qreal to 0

pl: the packet marking probability
q – capacity of link
bl : virtual queue length
thmax: maximum queue length threshold
thmin: minimum queue length threshold
pmin: marking threshold when bl = thmin
pmax: marking threshold when bl = thmax

If 0<=bl <=thmin

sends the incoming packets and enqued in the network.
The packets won't be dropped
Set packet marking probability as 0

If bl >=thmax

drop the packet
The packets are ECN marked and sent to destination.
Else if $th_{min,l} <= b_l <= th_{max,l}$

Drop the incoming packet with a probability proportional to virtual queue length

$$p_{min}\, e^{\frac{\beta}{q}(b_1 - th_{min})} \qquad (2)$$

Notify sources of incipient congestion

In this algorithm, the probability marking scheme is done between th_{min} and th_{max}.

Step 5: Comparison of RED and ERED algorithm

For comparing the existing and the proposed approach, the Adaptive Virtual Queue (AVQ) is proposed to design an AQM scheme that results in a low-loss, low-delay and high utilization operation at the link [9] [12]. In this approach replace the packet marking probability calculation method with the computation of the capacity of a virtual queue. AVQ scheme needs to maintain two important variables namely virtual capacity and virtual queue with a capacity lesser than the actual capacity of the link [13]. We compared the schemes where a router marks packets based on the real queue length to schemes where the router marks packets based on the queue length of the virtual queue.

The virtual capacity takes values smaller than the actual capacity of the link. When a packet arrives, it is queued in the real queue and further the virtual queue is updated to reflect a new arrival. Packets in the real queue are marked or dropped when the virtual buffer overflows [14].

The virtual capacity C_1 is updated according to the following equations:

$$C_1 \equiv \alpha(\gamma C - \lambda) \qquad (3)$$

where C is the link capacity¸ λ is the input traffic arrival rate, γ is a constant set to 0:98, and α is a constant whose value is chosen to achieve some stability requirements.

Step 6: Performance Evaluation

For the performance evaluation of our approach we used Xgraph as a plotting program in a network for both the algorithms by taking the number of packets at x-axis and time taken for the packet loss at y-axis.

4 EXPERIMENTAL RESULTS

The experimental results show the source nodes that are sending the packets to the destination node with the number of iterations and various time intervals. Each and every packet transfer in the network is based on the FTP applications. The maximum window size is set to be 1000 instead of 64. The bandwidth of the user link and link end between the two nodes are measured in packets per milli seconds with a delay from 1ms to 5 ms. Thmin is set to be 1/5 of the buffer limit and thmax to be 3*thmin and all other parameters are set to default settings.

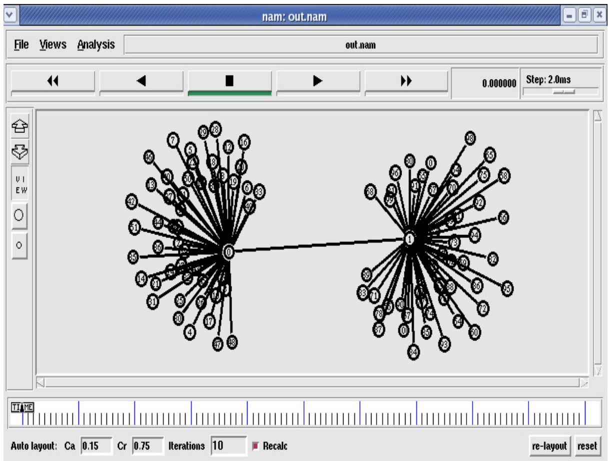


Fig 3 Simulation Network for the ERED Algorithm

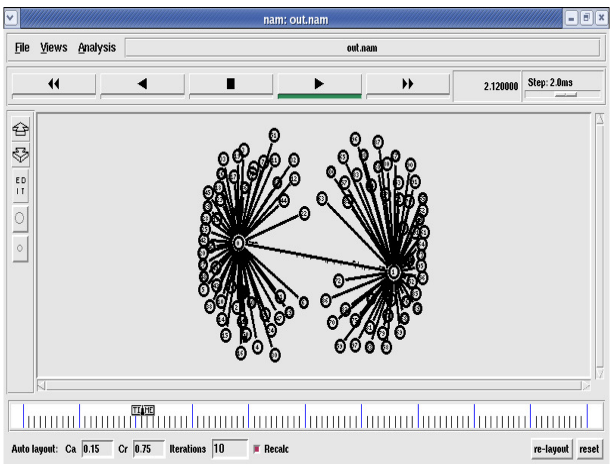


Fig 4 Simulation Results for ERED algorithm having 50 nodes at Source and Destination

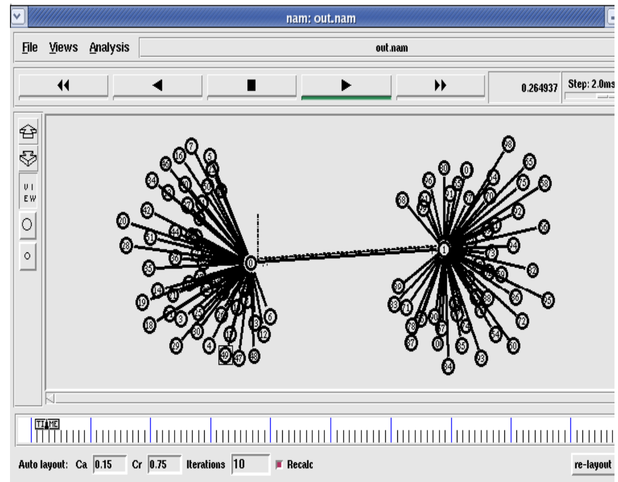


Fig 5 Simulation Results for the Queue Building having N nodes at Source and Destination

The following graphs shows the comparison results of RED and ERED algorithm with the packet loss and delay during the packet transfer. We plotted the graph by taking Time Intervals (in secs) as x-axis and number of packets on y-axis. The red color flow in the graph indicates the packet lost for ERED and green color flow of packets indicates the packet lost for RED approach.

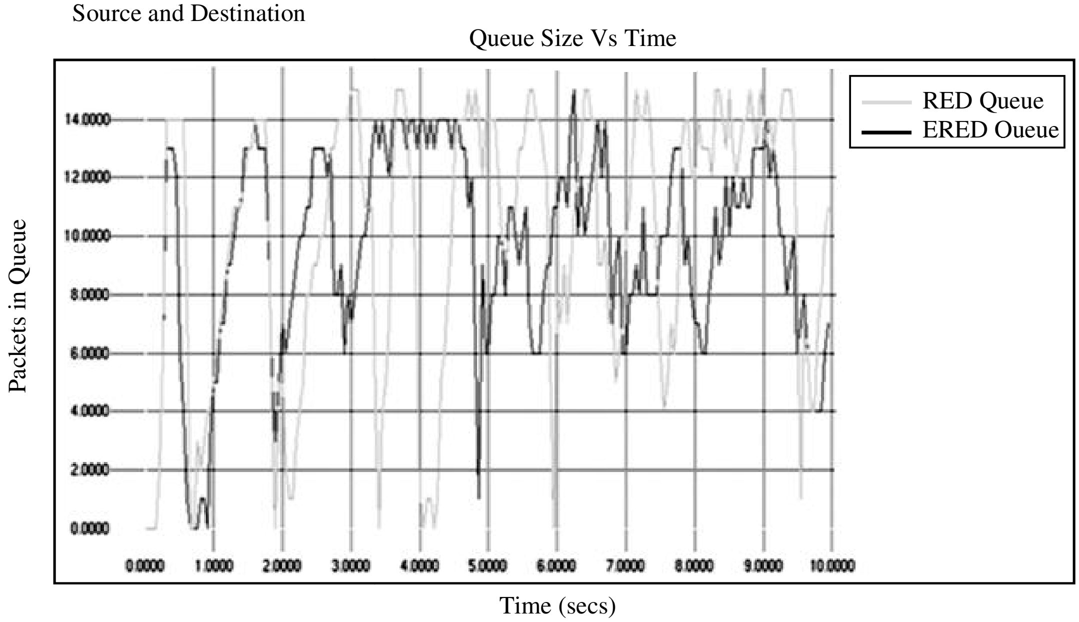
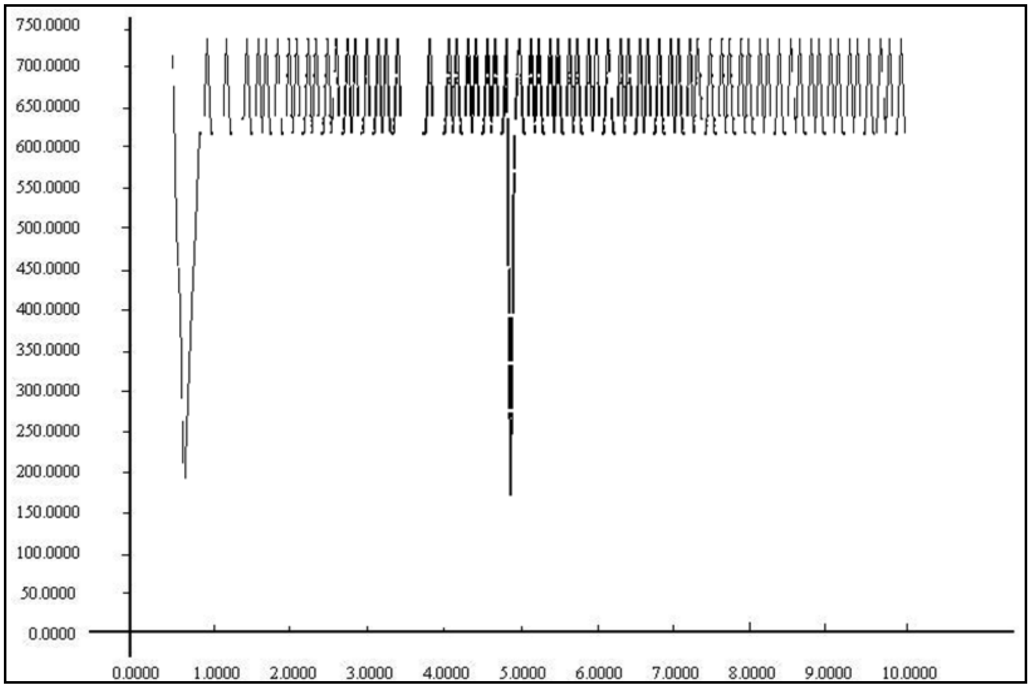


Fig 6 Queue size for RED and ERED algorithm at time interval t

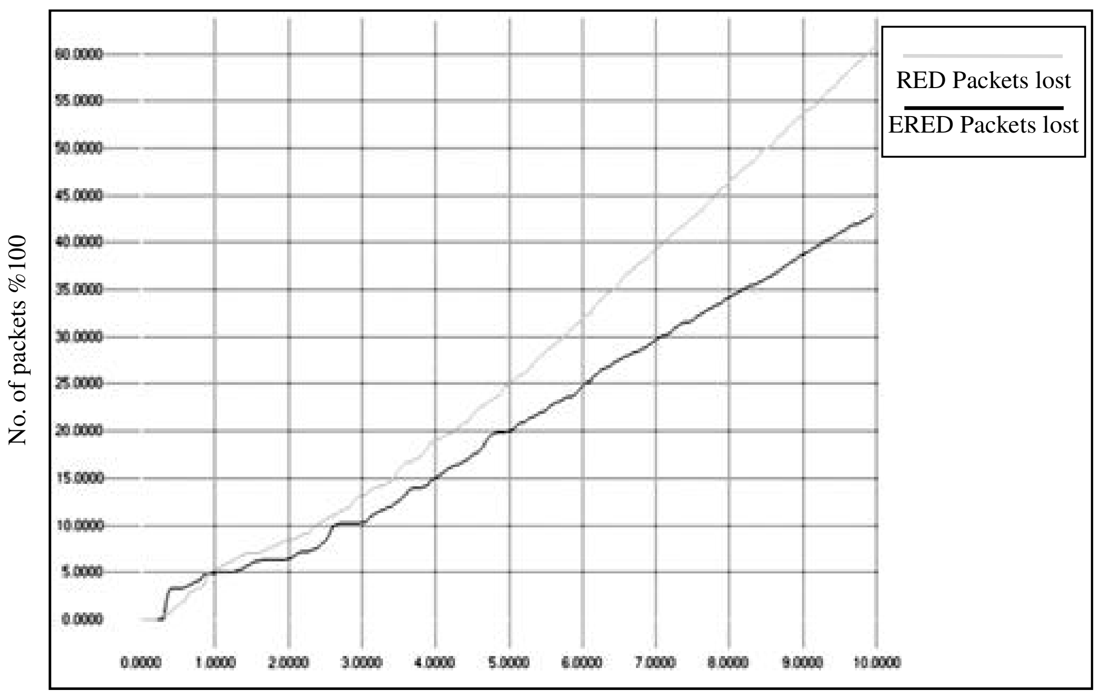
Bandwidth



Time (secs)

Fig 7 Bandwidth usage of RED and ERED algorithm for 100 Nodes

No. of Packets lost Vs Time



Time (secs)

Fig 8 Packet loss of RED and ERED algorithm for 100 Nodes

Table.1 shows the Comparison of RED and ERED algorithm for the packet arrival and packet loss of various nodes at different time intervals.

Table.1. Comparison of RED and ERED

TIME INTERVAL (in secs)	PACKETS ARRIVAL	PACKETS LOST	
		RED	ERED
0.5	11760	2300	3620
1.0	25640	3820	4700
1.5	36920	5800	5200
2.0	51360	7100	7060
2.5	66980	8540	8280
3.0	86220	12120	9080
3.5	106060	13780	10040
4.0	125820	15200	10520
4.5	145740	16680	12020
5.0	167380	18640	13000
5.5	186580	20120	14440
6.0	207360	22040	14900
6.5	229720	24440	16200
7.0	249480	26220	16780
7.5	271520	28000	18200
8.0	293600	30120	19020

5. CONCLUSION

Current TCP/IP congestion control algorithms cannot efficiently support new and emerging services needed by the Internet community. ERED proposes a solution along with the support of ECN for detecting incipient congestion in the network. Our proposed approach is more effective, robust, scalable and more flexible. The proposed probability calculation approach for congestion control in an ERED framework is implemented with packet marking capabilities. We had successfully implemented ERED algorithm and compared with the RED algorithm during packet loss in congestion control at various time intervals. The experimental result shows that our proposed approach E-RED outperforms RED and have successfully used the reported strength of virtual queue and also have addressed the limitations of existing RED algorithms implementation in a ERED framework. The experimental result also shows the packet loss for various nodes at various time intervals, Queue sizes and bandwidth for the transfer of packets from source node to destination node.

REFERENCES

[1] L. Shao and R.Srikant, Exponential-RED: A Stabilizing AQM Scheme for Low and High Speed TCP Protocols, IEEE/ACM Transactions on Networking, Vol. 13, No. 5, October 2005.

[2] R. Srikant and S. Srisankar Kunniyur, End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks, IEEE Transactions on Networking, Vol. 9, No.6, December 2001.

[3] T. John Wen and A. Murat, A Unifying Passivity Framework for Network Flow Control, IEEE Transactions on Automatic control, Vol. 49, No. 2, February 2004.

[4] M.Paul and D. Declan, An Introduction to NS, Nam and Otcl scripting. 2004-2005.

[5] D. Chiu and R. Jain, "Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks," Journal of Computer Networks and ISDN, Vol. 17, No. 1, June 1989, pp. 1-14.

[6] S. Srisankar and R. Srikant, An Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management, Member, University of Pennsylvania, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.

[7] J.Ramesh and T. David Kim Hong Tan, End-to-End Congestion Control for the Internet: Delays and Stability, IEEE Transactions on Networking Vol. 9, No. 6, December 2001.

[8] K.Srisankar and R. Srikant, End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks, Coordinated Science Laboratory University of Illinois.

[9] L.Ashvin, C. L. Beck and R. Srikant, Robustness of Real and Virtual Queue based Active Queue Management Schemes.

[10] C. V. Hollot, M.Vishal, T. Donald and G. Weibo, Analysis and Design of Controllers for AQM Routers Supporting TCP Flows, IEEE Transactions on Automatic control Vol. 47, No. 6, June 2002.

[11] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” IEEE/ACM Trans. Netw., Vol. 1, No. 4, pp. 397–413, Aug. 1993.

[12] C. V. Hollot, V. Misra, D. Towsley, and W. Gong, “A control theoretic analysis of RED,” presented at the IEEE INFOCOM, Anchorage, AK, Apr. 2001.LIU et al.: Exponential-Red: A Stabilizing Aqm Scheme For Low-And High-Speed Tcp Protocols 1081.

[13] H. Han, C. V. Hollot, Y. Chait, and V. Misra, “TCP network stabilized by buffer-based AQMs,” presented at the IEEE INFOCOM, Hong Kong, March, 2004.

[14] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. A Stochastic Model of TCP Reno Congestion Avoidance and Control. Technical Report CMPSCI TR 99-02, Univ. of Massachusetts, Amherst, 1999.