



Random early detection-techniques for
Differentiated services networks based on
packet droppings

Gorre.Narsimhulu	Associate professor in ECE dept&Vaageswari college of engineering,karimnagar ProfessorElectronics & Communication EngineeringJNTUH College of Engineering Hyderabad (Autonomous)
Dr.D.Srinivasa Rao	Associate professor in ECE dept&Vaageswari college of engineering,karimnagar ProfessorElectronics & Communication EngineeringJNTUH College of Engineering Hyderabad (Autonomous)

ABSTRACT

Differentiated Services architecture offers services at different levels. Packets are classified and treated differently, as defined in the Service Level Agreement (SLA). A traffic policy is created which defines the shaping, dropping and marking of packets. For Congestion Control in Differentiated services networks different variants of Random Early Detection (RED) Scheme are used. RED Scheme is an Active Queue Management Scheme, which can sense congestion before its occurrence and can take appropriate preventive action. In this paper the performance of three variants of RED scheme RED with In/Out with Coupled Queues (RIO-C), RED with In/Out with Decoupled Queues (RIO-D) and Weighted RED (WRED) are compared in terms of drop of packets.

KEYWORDS

Differentiated Services Networks, Congestion Control, Random Early Detection (RED), and Network Simulator2.

Introduction

Congestion is caused by shortage of network resources. If the source transmits at rate higher than the service rate of the queue, the queue size will grow and due to this, incoming packets will face delays and thus congestion will be caused. Congestion Control is the mechanism, which allocates the resources of the network such that the performance is acceptable, when network is operating near capacity. Without the proper control mechanism the throughput will be less under heavy load conditions [1,2].

The diverse and changing nature of service requirements among Internet applications mandates a network architecture that is both flexible and capable of differentiating between the needs of different applications. The traditional Internet architecture, however, offers best-effort service to all traffic. In an attempt to enrich this service model, the Internet Engineering Task Force (IETF) is considering a number of architectural extensions that permit service discrimination.

While the resource reservation setup protocol (RSVP) and its associated service classes may provide a solid foundation for providing different service guarantees, previous efforts in this direction [3] show that it requires complex changes in the Internet architecture.

That has led the IETF to consider simpler alternatives to service differentiation. A promising approach, known as Differentiated Services (DiffServ), allows the packets to be classified (marked) by an appropriate class or type of service (ToS) at the edges of the network, while the queues at the core simply support priority handling of packets based on their ToS. Another Internet-related quality of service (QoS) issue is the queue's packet drop mechanism.

Active queue management (AQM) has been recently proposed as a means to alleviate some congestion control problems as well as provide a notion of quality of service. A promising class is based on randomized packet drop or marking. In view of the IETF informational RFC [4], Random Early Detection (RED) [5] is expected to be widely deployed in the Internet.

Inter-Serv require per flow traffic handling at every hop along an

application's end-to-end path and RSV signalling Protocol. RSVP signalling protocol is used for each flows explicit signalling. Inter-Serv Architecture suffers from scalability problems [6]. To avoid scalability problems the Diff-Serv Architecture was proposed. In Diff-Serv networks, complex decision making is done at network edge routers. The edge router marks the packets with different code points according to service level agreement and core routers implement the dropping decision [5,6]. The outline of this paper is as follows, Section 2 illustrates the Architectural concepts of Diff-Serv Networks with three precedence levels, Section 3 states the RED and its variants, Section 4 explains the Simulation topology used in the study, results and observations are given in Section 5 and Section 6, concludes this paper. Architecture for differentiated services networks with three precedence levels.

The Diff-Serv architecture achieves scalability by maintaining class state information based on aggregates of traffic flows. Packets are classified and marked at the boundary of the network by implementing classification, marking, policing and shaping. The marking of packets decides the particular forwarding behaviour at nodes along the path. Network resources are assigned to the traffic streams according to service provisioning policies that decide how traffic would be marked and to be conditioned upon entering the DS network. The service determines the set of characteristics of packet transmission across a set of one path or more paths within the network [7,8].

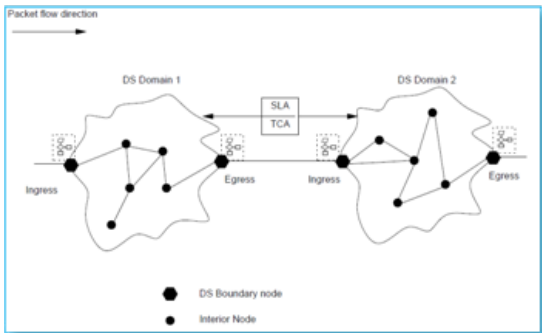


Fig. 1- Two Diff-Serv Domains [8].

Differentiated Services Domain

In [Fig-1], two different Diff-Serv domains are shown. Packets enter the DS domain 1 at the boundary ingress node, where packets are classified, conditioned and assigned to Behavioural Aggregates (BA). All the packets that belong to a particular BA are marked with same value in the DS field in IP header. This DS field is also called Differentiated Services Code Point (DSCP). A BA is treated in a special manner in a domain. Diff-Serv domain is a set of DS nodes with service provisioning policy and set of Per-Hop behaviours (PHBs). Traffic rules are applied on the entering packets by ingress nodes. The nodes through which traffic leaves a domain are called egress nodes. The boundary nodes connect two different DS do-mains, whereas the internal nodes connect to either other interior nodes or ingress/egress boundary nodes. Boundary nodes enforce the policy on the traffic entering the domain by conditioning [7,8].

Differentiated Services with Three Precedence Levels

Differentiated Services with three precedence levels is an extension to two-drop precedence. The edge devices monitor and mark packets of aggregated flows into one of green, yellow or red de-pending on the reservation rate and temporal sending rate of each colour. Usually the reservation rate of yellow is set greater than the reservation rate for green. If the rate is greater than reservation rate for green but, less than reservation rate for yellow, the packet is marked as yellow. Otherwise, the packet is marked as red. The differentiation is provided by dropping first red packets, the yellows second and then green packets [6]. In this paper, we have implemented Time Sliding Window Three Colour Marker (TSW3CM). It is implemented in edge ingress/egress routers for policing the traffic entering and leaving a particular domain. TSW3CM edge router takes two parameters Committed Information rate (CIR) and Peak Information rate (PIR) [6,11].The Random Early Detection (RED) queue which is associated with DS maintains 3 Virtual queues, each for Class A, B & C traffic in every physical queue. The work in this paper is focused on comparison between different variants of Multi RED (MRED) queues.

Random Early Detection and its Variants in Differentiated Ser-vices Networks

As proposed by Floyd et al. [12] Random Early Detection (RED) is an Active Queue Management (AQM) Scheme. RED employs intelligent admission decision for every packet based on local algorithm. The RED gateways detect congestion by computing the average queue size and taking dropping/marking decisions based on this average queue size. The gateways upon sensing the con-gestion either drops packets or mark the packets by setting a bit in IP header. This dropping/marking decision is taken by RED gate-way by comparing calculated average queue size against the mini-mum (MinTh) and maximum (MaxTh) thresholds according the following criteria [6,8,12]:

- When $avg < MinTh$, The Packet is Dropped.
- When $Minth < avg < MaxTh$, The packet is Dropped with Prob-ability $P(a)$
- When $MaxTh < avg$, Packet is Dropped with Probability 1

Where $avg = q * \text{measured queue length} + (1-q) * avg$

$$p(b) = MaxP * (avg - MinTh) / (MaxTh - MinTh),$$

$$p(b) = p(b) * (packetsize / MaximumPacketSize), \text{ and}$$

$$p(a) = p(b) / (1 - count * p(b)).$$

An estimation of the average queue size is computed for every incoming packet and updated at every packet arrival [12].

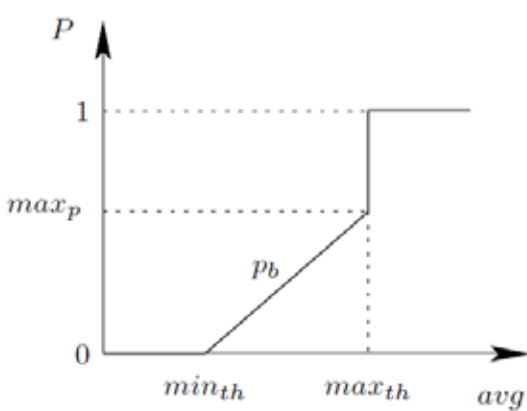


Fig. 2- RED Algorithm [8].

[Fig-2] shows the graphical representation of the RED scheme. The RED can be extended to Multi-Level RED (MRED) algo-rithm that can be used in differentiated services networks.

Multi-Level Red (MRED)

MRED is an extension of RED for differentiated services net-works. In MRED, for different drop precedence's, the drop prob-ability for packets is calculated independently. Multiple set of RED parameters are maintained for multiple drop precedence. The average queue used for making decision can also be cal-culated using differ-ent methods [13]. These different methods of calculating average queue yield following variants of MRED:

RIO-C (RED With In/Out with Coupled Virtual Queues)

RIO-C is configured with three sets of parameters for the all three drop precedence, one set for each precedence. The aver-age queue for packets with different colours is calculated separately, such that average queue length of a particular col-our is calculated by adding its average queue to the average queue of the lower drop precedence i.e. for yellow packets, average queue size is computed using green and yellow pack-ets. But, for green packets, the queue is calculated using only green colour packets [14]. RIO-D is similar to RIO-C, the only difference is that in RIO-D, average queue for each colour is calculated independently. The aver-age queue length is cal-culated using number of packets of that particular queue. i.e. The average queue length for green, yellow & red packets is calculated using number of green, yellow and red packets in that respective queue [8,14].

WRED (Weighted RED)

In WRED only a single average queue is maintained for all ar-riving packets of all colors. For the arrival of green, red or yel-low packet only a single average queue based on total number of packets of all precedence (red, yellow and green). However, multiple RED parameters are maintained for each colour.

The RED parameters which were implemented for all RED var-iants during the whole simulation process in all scenarios are given as under in Table1:

Table 1- MRED Parameters used in all simulation scenarios

Class	MinTh	MaxTh	MaxP
A (Green)	25	45	0.01
B (Yellow)	10	25	0.2
C (Red)	3	5	0.7

Simulation Setup

In this section the simulation setup used during this study is explained. The simulations were carried out over widely used simulator for network related simulation, network simulator 2 (Version 2.34) on Ubuntu 11.10 machine. The experimental setup used is shown in [Fig-3]:

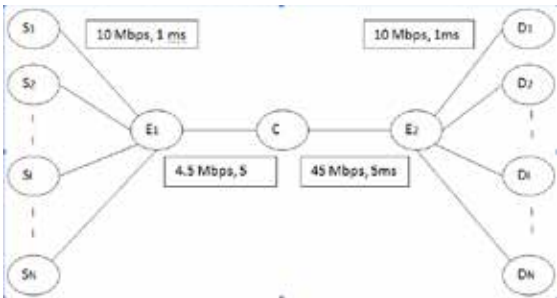


Fig. 3- Simulation Setup

In the experimental setup, there were N number of source nodes and destinations nodes. Each source node was connected to edge router 1(E1), whereas the destination nodes were connected to edge router 2 (E2). The link between edge router 1(E1) and core router (C) was considered as bottleneck link, which had capacity 4.5 Mbps and latency 5ms. All links had physical buffer which can hold 50 packets. All the source nodes were configured to generate a number of TCP or UDP based flows depending on type of scenario being considered. Each flow was configured to send packets having packet size of 512 bytes. In this study, two types of scenarios were considered. In first type of scenario, number of nodes was increased and number of flows per source was kept constant and number of packets dropped was tabulated, whereas in the second type of scenario the setup has only single source node and destination node and number of flows was varied and number of packets dropped was tabulated. The % of packets dropped were calculated as per following formula [15]:

$$\%age\ Packets\ dropped = \frac{Total\ number\ of\ Packets\ dropped}{total\ number\ of\ packets\ sent}$$

The % of packets dropped indicate the relative performance of RED variants in various scenarios

The % of packets dropped indicate the relative performance of RED variants in various scenarios

Results and Discussions

Several simulation scenarios were considered to find appropriate results. Firstly only FTP running over TCP flows were considered in both cases i.e. by increasing number of nodes and by number of increasing number of flows for a single source and destination network. After this, Constant Bit Rate (CBR) traffic was used to conduct simulations. Dropped packets at bottleneck link were tabulated and % of packets dropped or % Packet Loss was calculated for all precedence traffic. % Packet loss data was plotted in graphical form. An example of Simulation results generated in tabular form is shown below in [Table-2]:

Table 2- Example of simulation results generated in tabular form

Class	Total Packets sent	Transmitted Packets	ldrops	edrops	Total drops	%age of pack-ets Dropped
A(Green)	122	120	2	0	2	1.64
B(Yellow)	833	796	32	5	37	4.44
C (Red)	4117	2709	357	1051	1408	34.19
All	5072	3625	391	1056	1447	28.53

As shown in [Table-2], the results were generated in the tabular form. Packet drops for each class traffic was presented. Here ldrops (link drops) meant the dropping due to physical queue length exceeding the physical buffer. The edrops (early drops) were droppings due to RED mechanism. The statistics in All Class were combined statistics of all Traffic Classes (A, B and C). Total drops were sum of ldrops and edrops [16]. % of Packets Dropped for All Traffic Classes were calculated and used as metric in this work. Here node means both source and destination nodes. The results obtained during the study are given as under:

FTP/TCP Traffic

In this case the simulation was carried out by nodes running FTP applications over TCP flows:

Increasing Number of Source Nodes

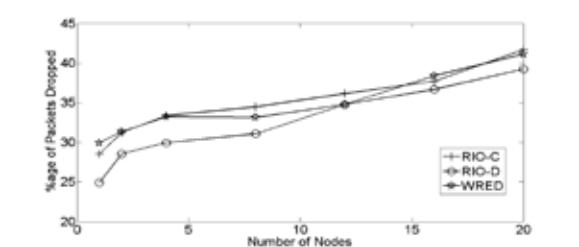


Fig. 4- % of Packets Drop Vs. No. of Nodes for FTP/TCP Traffic

Number of flows per node was kept constant at 80. The number of nodes was increased from 1 to 20. % of Packets dropped for All class traffic was noted. The simulation time was 5 seconds. Simulation results obtained during this scenario have been presented here in graphical form in [Fig-4].In [Fig-4], when number of nodes was less RIO-D outperforms both RIO-C and WRED, as it dropped less number of packets. There were some fluctuations observed at higher number of nodes due to variable nature of TCP. It was observed from this scenario that RIO-D performs better as compared to WRED and RIO-C.

Increasing Number of Flows

For a single source and destination network the number of FTP/TCP flows was increased from 1 to 80. The simulation time was 50 seconds. Simulation results obtained during this scenario have been presented here in graphical form in [Fig-5]:

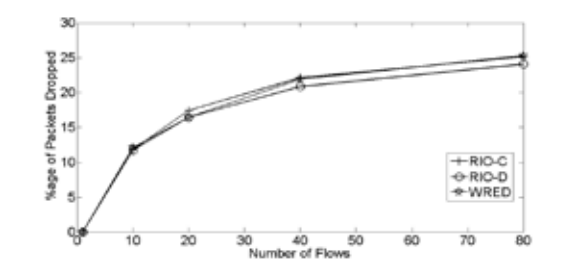


Fig. 5- % of Packets Dropped Vs. Number of Flows for FTP/TCP Traffic

As shown in [Fig-5], as the number of FTP/TCP flows were increased RIO-D dropped less packets as compared to RIO-C and WRED. This behaviour was mainly due to increase of Class C packets in All class traffic at large number of flows. For 80 number of flows, the RIO-D dropped 1.21% less packets as compared to WRED and RIO-C.

CBR/UDP Traffic

In this case sources were configured to run CBR/UDP flows. The CBR traffic sends data at a constant rate without any interruption in between [16]. The results obtained were as under:

Increasing Number of Sources

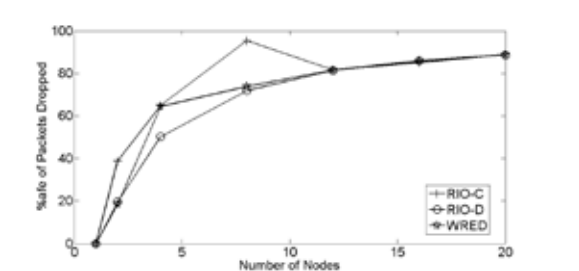


Fig. 6- % of Packets Dropped Vs. Number of Nodes for CBR/UDP Traffic

In this scenario, number of nodes was increased from 0 to 20 and number of flows was kept constant at 20 per node. Simulation was conducted for 5 seconds. Each CBR/UDP flow sends data at rate 12.207 Kbps. Simulation results obtained during this scenario have been presented here in graphical form in [Fig-6]:

In [Fig-6], as the numbers of nodes were increased RIO-D started performing better as compared to other two variants. When Number of nodes were increased beyond 10 nodes Idrops started in-creasing and thus after this there was very less difference in performance.

Increasing Number of Flows

Number of CBR/UDP flows was increased from 1 to 20 for a single source and destination network in this scenario. Each flow send date at rate 12.207 Kbps. Simulation was carried out for 50 seconds. Simulation results obtained during this scenario have been presented here in graphical form in[Fig-7]:

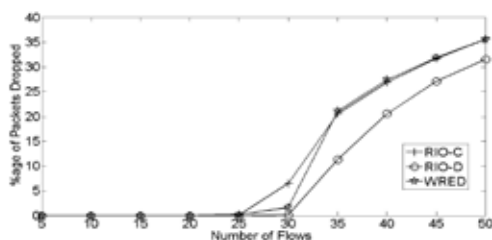


Fig. 7- % of Packets Dropped Vs. Number of Flows for CBR/UDP Traffic

As shown in [Fig-7], when there was less number of flows no congestion occurred and thus, no packets were dropped. When number of flows was increased beyond 25 RIO-D scheme had lesser %age of Packets Dropped as compared to other two schemes. Thus, RIO-D performs better as compared to other two variants. It was observed from the study that the RED with In/Out with decoupled queues (RIO-D) performed better as compared to two other modes for all the simulation scenarios conducted for FTP/TCP and CBR/UDP traffics , as it dropped less number of packets.

Conclusion

In this paper the performance of different variants of Random Early Detection (RED) such as RIO-C, RIO-D and WRED was compared on the basis of packet droppings. The RED parameters for all three variants were kept same and various simulation scenarios were created using a variety of traffic including FTP/TCP and CBR/UDP at different transmission rates. It was observed from the study that RIO-D drops less number of packets as compared to other two variants RIO-C and WRED in most of the cases. It can be concluded from the study that RIO-D performs better as compared to other two variants in terms of percentage of packets dropped. There is always a trade-off between packet loss and latency. In This work only packet loss was considered, the latency was not analyzed. Work based on latency is under process.

REFERENCES

- [1] Jain R. (1990) IEEE Network Magazine, 24-30. | [2] Pitsillides A. and Sekercioglu A. (2012) Fuzzy | Logic Based Congestion Control. | [3] J. Wroclawski. The Use of RSVP with IETF Integrated | Services. RFC Draft, IETF, September 1997. | [4] B. Braden et al. Recommendations on queue management and | congestion avoidance in the Internet. RFC 2309, IETF, April 1998 | [5] S. Floyd and V. Jacobson. Random early detection gateways for | congestion avoidance. IEEE/ACM Transactions onNetworking, 1(4):397-413, 1993. | [6] Raghavan S. (2011) Master of Science Thesis, Virginia Poly- | technique Institute & State University, Blacksburg, Virginia, 1-107. | [6] Revathi T., Muneeswaran K. and Ramar K. (2011) Applied | SofComputing, 5457-5462. | [7] Blake S., Black D., Carlson M., Davies E., Wang Z. and Weiss W. | (1998) Network Working Group, RFC 2475. | [8] Strauss M.D. (2005) Master of Science Thesis, University of | Pretoria, Pretoria, 1-98. | [9] Hernanen J., Baker F., Weiss W. and Wroclawski J. (1999) Internet | Engineering Task Force. | [10] Dave B. et al. (2002) Network Working Group, RFC 3246. | [11] Feng W., Seddigh B. and Nandy B. (2000) Network Working | Group, RFC 2859. | [12] Floyd S. and Jacobson V. (1993) IEEE/ACM Transactions on | Networking, 397-413. | [13] Durressi G.M., Jain A.R. and Lei C. (1999) Effect of Drop Prec- | edence in Assured Forwarding. | [14] Elloumi O. et al. (1999) Internet Engineering Task Force. | [15] Makkar R.S. et al. (2000) Proceedings of ICCCN, Las Vegas, | [16] Fall K. and Varadhan K. (2011) The VINT Project, 1-433 |