Soc. Netw. Anal. Min. (2016) 6:70
DOI 10.1007/s13278-016-0382-5

ORIGINAL ARTICLE

Comparative analysis of queuing mechanisms: Droptail, RED and NLRED

Shubhangi Rastogi[1] · Hira Zaheer[2]

Received: 7 April 2016 / Revised: 30 June 2016 / Accepted: 24 August 2016 / Published online: 3 September 2016

Abstract Congestion control in today's Internet is an important issue. Congestion control algorithms have been extensively studied for managing the traffic and maintaining the stability in the network. In traffic management system, queuing plays an important role. This paper presents a comparison of three queuing mechanisms, namely Drop-tail, random early detection (RED) and nonlinear random early detection (NLRED) in wired network on the basis of different performance metrics such as end-to-end delay, throughput, packet drop and packet delivery ratio using NS2 simulator. The simulation results show that in high congestion, NLRED performs best while in low cohesive network Droptail gives good result. Also, we analyzed these queuing mechanisms in real audio traffic; again, all the experiments show that in congested network NLRED and RED are better while in low congested network Drop-tail is better because in heavy congested network congestion avoidance mechanism will help the network to achieve better performance. But in low congested network, the unnecessary computation avoidance mechanisms will degrade the network performance. However, if parameters are set effectively in RED, then it will be the best queuing mechanism for that particular network.

Keywords Droptail · Random early detection · NS2 simulator · NLRED · RED · Queuing mechanisms

✉ Hira Zaheer
hirazaheeriitr@gmail.com

Shubhangi Rastogi
shubhangi05.rastogi@gmail.com

[1] Ajay Kumar Garg Engineering College Ghaziabad, Ghaziabad, India

[2] Indian Institute of Technology Roorkee, Roorkee, India

1 Introduction

Congestion control in today's Internet is an important issue. Traffic on the Internet keeps on fluctuating. So, for overseeing the traffic and keeping the network stable, congestion control algorithms have been extensively used. Some queuing discipline must be implemented by each router in the network that oversees how packets are cradled while holding up to be transmitted. In this method, packets are dropped only if the buffer is full (Huebner 2004). It contains some algorithms such as Drop-tail, head drop and push out. These are known as passive queue management algorithms which are simpler to implement than active queue management algorithms, but have some disadvantages also. First, here end-to-end delay is more because the average queue size is large for long duration. Second, passive algorithms control the congestion instead of avoiding it. When queue is full, the source reduces the rate of transmission of data. The packets which are transmitted before this reduction process are dropped out and have to be retransmitted. Third, passive queue management leads to lock-out problem. In this case, one queue transmits its data with normal rate, but others decrease their rate. So handling fairness is an issue here. While in the case of active queue management, average queue length is decreased. This in turn decreases the end-to-end latency as well. However, active queue management algorithms are complex. So choosing a suitable queue management for a particular network is important. Mohamed (2010) submitted his dissertation in which the three mechanisms droptail (DT), early random drop (ERD) and multithreshold adaptive early random drop (MTAERD) are compared using a Java framework and outcomes are showing the total improvement in the quality of service that can be obtained by the mechanisms over their non-adaptive supplements.

One approach in this area was given by Hui et al. (2009), in which they replaces RED based on linear packet dropping function by NLRED which is based on nonlinear quadratic function. NLRED is gentler than RED which improves the performance. Jiang et al. (2004) compare the performance of RED and adaptive RED from the viewpoint of nonlinear dynamics. Their simulation results confirm that adaptive RED performs better than RED. Chen et al. (2010) proposed a new improved algorithm ARED to stabilize the queue length and avoid oscillation. Katiyar and Jain (2014) presented a survey of RED, GRED, ARED and DRED for congestion avoidance mechanism and measured their performances on the basis of different metrics such as delay, throughput, packet loss and average queue length. Kaur et al. (2013) discussed the performances of various queuing algorithms in order to fight with the low-rate distributed denial of service attacks (LDDoS). Chhabra et al. (2013) introduced a congestion avoidance scheme RED in a way in which packet dropping is according to the rate of input as there are some aggressive packets and some fragile so dropping depends on the type of packets too. Also, they analyzed the performance on different simulation parameters. Soni and Mishra (2013) discussed an active queue management technique RED detects congestion according to current length of average queue. It keeps throughput high and average queue sizes low without packet loss.

Zhou et al. (2006) proposed a nonlinear RED active queue management scheme. With the proposed nonlinear packet dropping function, packet dropping changes according to the type of the load, i.e., heavy or light load. By simulation, it can be concluded that NLRED achieves a more stable throughput than any other queuing mechanism, viz. RED, REM or a variant of RED. Abdel-jaber’s (2015) proposal suggested the various variants of RED, i.e., adaptive GRED, REDD and GRED-linear analytical model. Several performance measures are used to evaluate the effectiveness of these algorithms to identify which algorithm gives more satisfactory results in which scenario (congested and non-congested). Jain et al. (2014) evaluated the performance of a recently proposed algorithm controlled delay (CoDel) in wired-cum-wireless networks with other RED and droptail. Simulations are carried out by using ns2. Joshi et al. (Joshi2005) developed a technique named as multiple average–multiple threshold (MAMT) as a solution for giving available and dependable service to traffic from emergency users after disasters. Also, their work considered the non-responsive flows since this is getting the most attention from emergency management organizations.

Yahia and Bíró (2005) investigated the interactions between the preventive admission control and reactive queue management algorithms such as DropTail and RED in terms of transaction times of data, and the performance of admission control is analyzed. Xu and Sun (2014) proposed Straightforward AQM (SFAQM) algorithm and compared with PI, RaQ and CARED and show that SFAQM is effective in various scenarios. Khademi and Othman (2010) introduced the Least Attained Service (LAS) and Threshold-based LAS (TLAS) queue management and validated them by different simulation networks using ns2. Some applications were shown by Zaheer et al. (2014) and Zaheer and Pant (2015).

2 Different queuing mechanisms

In this paper, we have considered the following three mechanisms because droptail is the most simplest and basic queue mechanism commonly used in almost every network scenario, but has some drawbacks. These drawbacks are removed by our second queue mechanisms, i.e., RED, and the third, NLRED is improvement over RED.

2.1 DropTail queuing mechanisms

Droptail is of passive queue management type. It stores the packet till buffer gets full, and when the buffer gets full, i.e., there is no vacant space left in the queue, it starts dropping each and every packet. The dropping probability of packets is shown in Fig. 1. There are two possible dropping probabilities, i.e., 0 or 1. The packet dropping probability is 0 if the number of arrived packets is lesser than the number of buffered packets, and it will be 1 in the reverse case.

2.2 RED queuing mechanism

RED is an active queue management technique. RED stands for random early detection. The main purpose of why this algorithm has been proposed is to reduce the

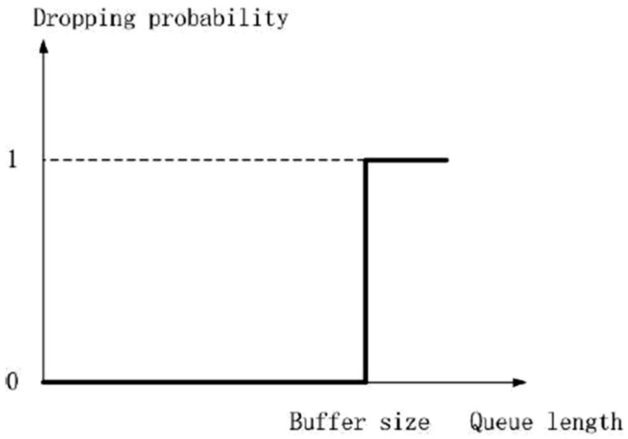

Fig. 1 Dropping probability of droptail queue mechanism

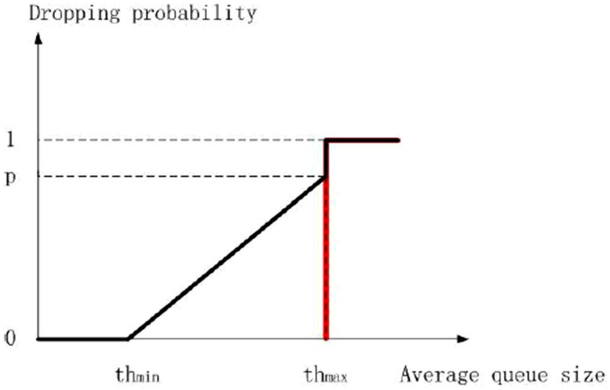

Fig. 2 Dropping probability of RED queuing mechanisms

limitations involved in droptail queue mechanism. Following are the objectives of RED:

1. Attenuate packet loss and queuing delay
2. Reduce the need of global synchronization of sources
3. Provide high link utilization
4. Remove biases against busty sources

The dropping probability varies linearly with average queue size. When the average queue length lies between minimum threshold $T_{\min}$ and maximum threshold $T_{\max}$, the dropping probability lies from 0 to p within the range. In starting, the dropping probability is 0 and packets start dropping as the average queue size increases. All packets are dropped as the average queue size crosses the maximum threshold and the dropping probability becomes 1. It will be more clear in Fig. 2 of dropping probability of RED (red line shows the droptail).

2.3 NLRED queuing mechanism

RED does not maintain the state of each flow, that is data is placed from all the flows into one queue and concentrates on their performance. It originates the problems caused by non-responsive flows. Nonlinear random early detection (NLRED) congestion control algorithm has been introduced to tackle with problem. In NLRED, there is nonlinear quadratic function where in RED the dropping function is linear. With the proposed nonlinear packet dropping function, at light traffic load packet drops in gentler fashion and once average queue size approaches the maximum threshold $T_{\max}$, this is used as an indicator that the queue size could soon be full, so NLRED will first drop more aggressive packets, i.e., more heavy packets. Also, it is less parameter sensitive. NLRED obtains much stable throughput as compared to RED. Also, the packet dropping probability of RED will be always greater than that of NLRED due to which at same $P_{\max}$ value NLRED is gentler than RED for all traffic load (Fig. 3).

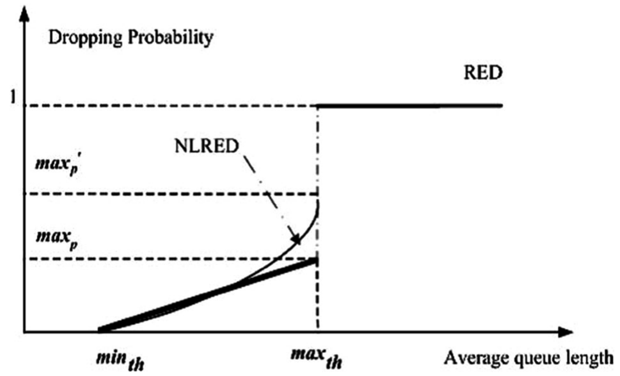

Fig. 3 Dropping probability of NLRED and RED queuing mechanisms

3 Performance metrics

Throughput, end-to-end delay, packet drop and packet delivery ratio are the primary metrics in our simulations because these can work as basic building blocks for calculating other metrics.

3.1 Throughput

It is the main characteristic of performance measure and most commonly used. This measures how quick the beneficiary can get a particular measure of information sent by the sender. It is the proportion of the aggregate information got to the end-to-end delay. With the help of throughput, the fairness among different flows is illustrated, and total throughput is basically dependent on bottleneck bandwidth.

3.2 Delay

Delay is time slipped when a packet moves from sender to receiver. As large as the value of delay, the bandwidth will be high and it will be more difficult for transport layer to maintain. This characteristic can be indicated in various diverse ways, including average delay, delay bound and variance of delay (jitter). Here, end-to-end delay is being calculated.

3.3 Packet drop

Packet loss means packet starts dropping when queue size increases with the maximum capacity. When queue is in the network node, if surpasses, then packets also loses in network. One major property of congestion control is the quantity of packet drops during steady state. It will be more difficult to sustain high bandwidths, sensitivity to loss of packets for higher rate of packet drop.

3.4 Packet delivery

Packet delivery ratio is the proportion of the total packet sent to the total packet received. Mathematically, it can be defined as: PDR = T1 ÷ T2 where T1 is the Total number of data packets received from each destination and T2 is the total data packets formed by each source.

4 Simulation

The considered network topology is a classic dumb-bell as shown in Fig. 4. This is a common scenario in which different types of traffic share a bottleneck link. TCP (FTP application in particular), UDP flows (CBR application in particular) and real audio traffic are chosen as typical traffic patterns.

In this paper, we considered three different mechanisms which have different behavior for different network configuration and traffic pattern. The most important task in designing the simulation is to select parameters (bandwidth, queue limit, packet size, etc.) and a typical set of network topology. A simple topology is used in our simulation where different flows share a bottleneck between the two routers (3.4). The packets sent from sources queue to the queue of router 3 and wait for transmitting. If the sender keeps sending and the queue overloaded, then congestion occurs. This paper mainly focuses on congestion, and by this simple topology we can easily create congestion by setting different parameters accordingly and analyze the behavior of all the three controlling mechanisms. If we increase the network size or change its topology, then results may change depending on the amount of congestion and bandwidth at any bottleneck link of the network. If bandwidth is large enough to allow the transmission of all the flows, then no need of applying any active complex mechanism. Simple drop-tail would give the best result. Else if bandwidth is low compared to the amount of congestion, then RED or NLRED would give good performance.

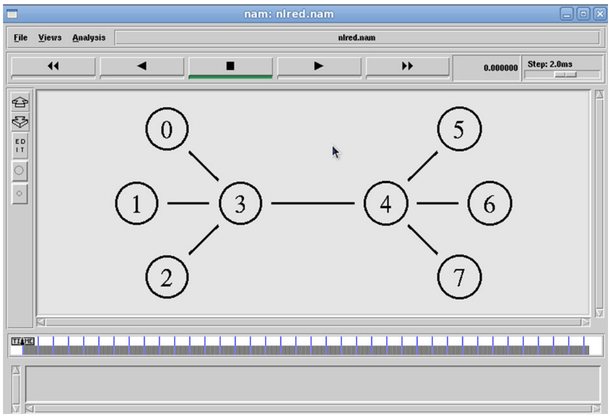

Fig. 4 Classic dumb-bell network topology

In this research work, there are total four experiments that have been performed, and then, the simulation results have been analyzed through a thorough study.

4.1 Experiment 1

In this experiment, there are three nodes at each side of bottleneck link 3–4 where node 0 is acting as a UDP source to send CBR traffic and node 6 as null, while node 2 is acting as TCP source to send FTP traffic and node 6 as sink. Bandwidth at bottleneck link 3–4 is 1.7 MB. Here, in this experiment we take CBR flow rate as 1Mbps and size of packet as 1000 bytes. Queue limit is 15 packets. We mimic this network on ns2 for diverse queuing instruments Drop-Tail, RED and NLRED. Parameter setting in Droptail is easy. For RED, we have to select values for $\min_{th}$ and $\max_{th}$. Also, some other remaining parameters such as maximum probability of drop are taken as 0.5 bytes; exponential weighted moving normal consistent is 0.001. Calculations of average queue size are in bytes. Setting queue in bytes to false demonstrate average queue size is in packets (not in bytes). Gentle RED mode is set to be false indicates gentle mode is OFF. Average size of a packet touching the router is likewise made equivalents to 1000. We have chosen such values of parameters just to create a congested network as 1.7 MB is very low bandwidth in comparison with packet size and queue size (Figs. 5, 6, 7, 8; Tables 1, 2, 3, 4).

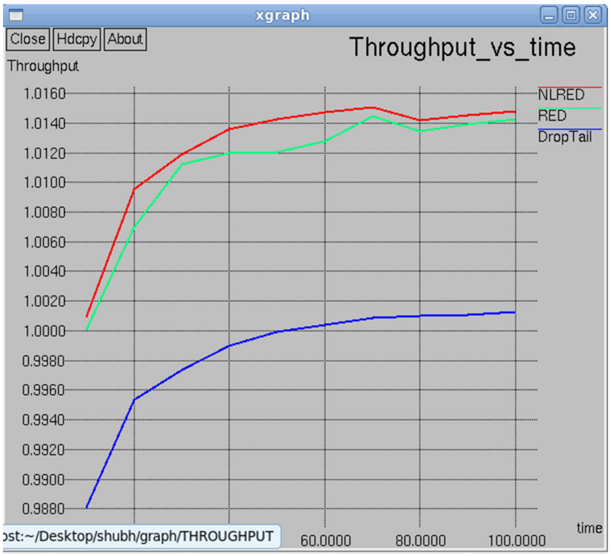

Fig. 5 Comparison of variation of throughput for Droptail, RED and NLRED queue

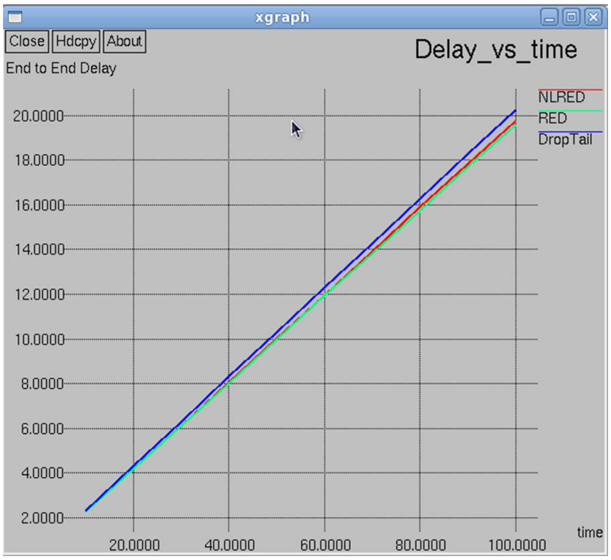

Fig. 6 Comparison of variation of end-to-end delay for Droptail, RED and NLRED queue

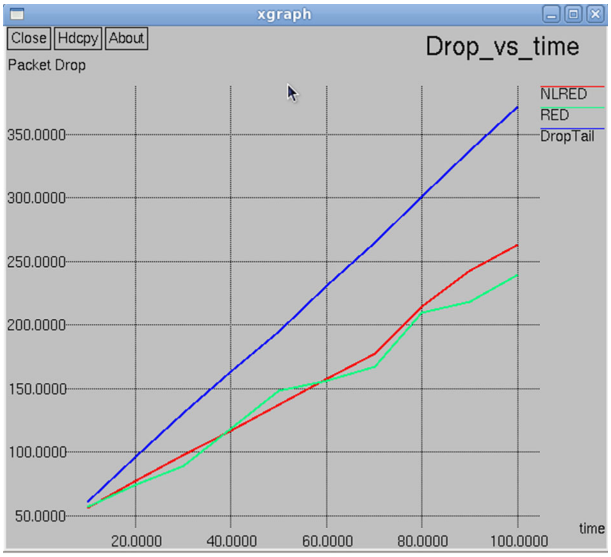

Fig. 7 Comparison of variation of packet drop for Droptail, RED and NLRED queue

4.2 Experiment 2

In this experiment, the same simple dumb-bell topology in wired network is being used. But, bandwidth at bottleneck link 3–4 is being increased from 1.7 to 10 MB. It means congestion in the network is decreased. Remaining parameters such as CBR flow rate, packet size and queue limit are kept same. Also, parameters of RED and NLRED (such as minimum threshold, maximum threshold, weight, etc.) remain unchanged. Here, all other parameter values

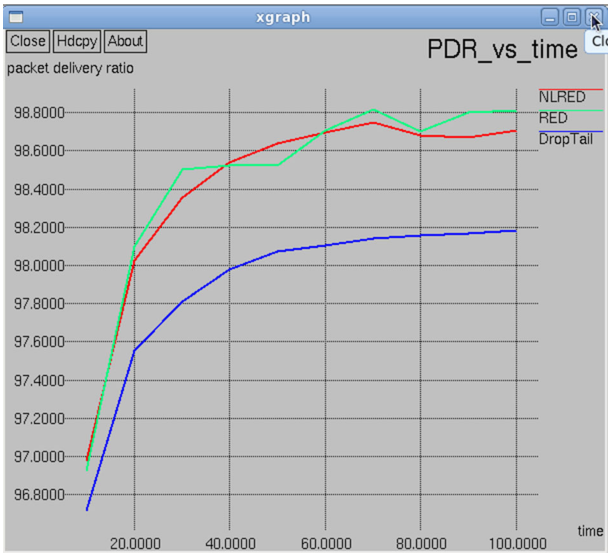

Fig. 8 Comparison of variation of PDR for Droptail, RED and NLRED queue

Table 1 Throughput analysis and results

Simulation time	DropTail	RED	NLRED
10	0.988089	1.00004	1.00092
20	0.995385	1.00699	1.00952
30	0.997385	1.01123	1.01191
40	0.998974	1.01197	1.01361
50	0.999915	1.01202	1.01428

Table 2 End-to-end delay analysis and results

Simulation time	DropTail	RED	NLRED
10	2.30797	2.25459	2.25353
20	4.30649	4.20411	4.1902
30	6.30824	6.09862	6.12759
40	8.30739	8.01948	8.0669

Table 3 Packet drop analysis and results

Simulation time	DropTail	RED	NLRED
10	61	57	56
20	96	74	77
30	131	89	98
40	163	118	117
50	163	118	117

Table 4 Packet delivery ratio (PDR) analysis and results

Simulation time	DropTail	RED	NLRED
10	96.7187	96.9272	96.9811
20	97.5541	98.1006	98.0282
30	97.8134	98.4997	98.354
40	97.9767	98.5207	98.5382
50	98.0735	98.5249	98.6374

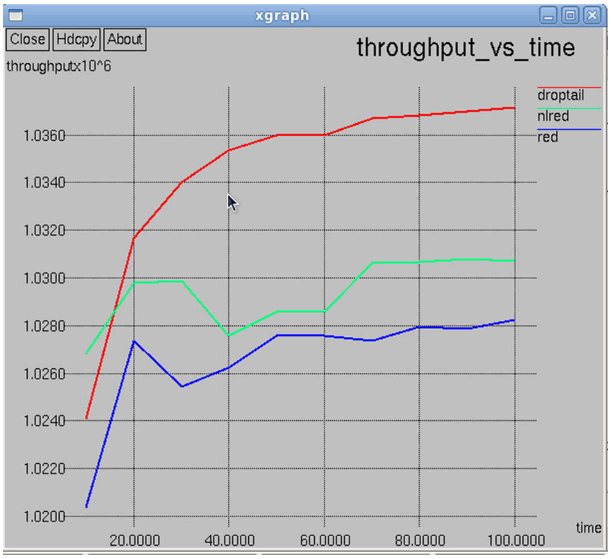

Fig. 9 Comparison of variation of throughput for Droptail, RED and NLRED queue

are same except the bandwidth just to check the performance of all the three mechanisms in non-congested network (Figs. 9, 10, 11, 12; Tables 5, 6, 7, 8).

4.3 Experiment 3

Simple dumb-bell topology is used in wired network. There are three nodes at each side of bottleneck link 3–4. Node 0 is acting as a RA source to send real audio traffic and node 6 as dump, while node 2 is acting as TCP source to send FTP traffic and node 6 as sink. Bandwidth at bottleneck link 3–4 is 1.7 MB. Packet size is 1000 bytes, CBR flow rate is 1Mbps, whereas queue limit is 15 packets.

All four end-to-end delay, performance metrics throughput, PDR and packet drop are same for all the three queues, as the bandwidth of the bottleneck link is so small 1.7 as compared to audio traffic. So we are getting same readings for all queues Drop-Tail, RED and NLRED.

In this experiment, the values are chosen same as in experiment-1, but instead of CBR and FTP traffic we are using real audio traffic just to check the performance in a network where packets are highly aggressive and bandwidth is not sufficient to send the packets.

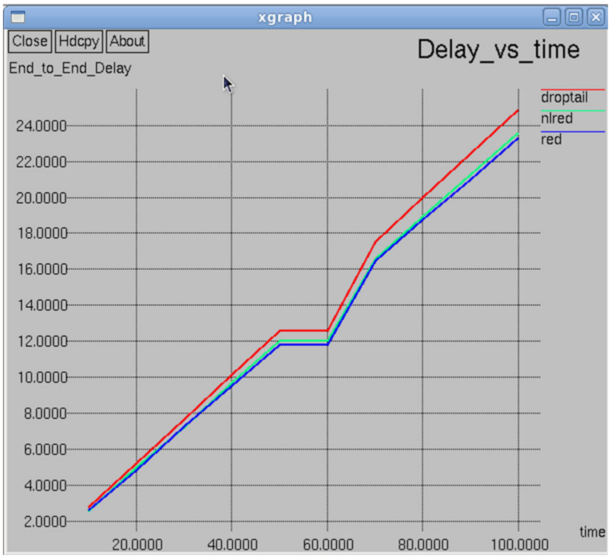

Fig. 10 Comparison of variation of end-to-end delay for Droptail, RED and NLRED queue

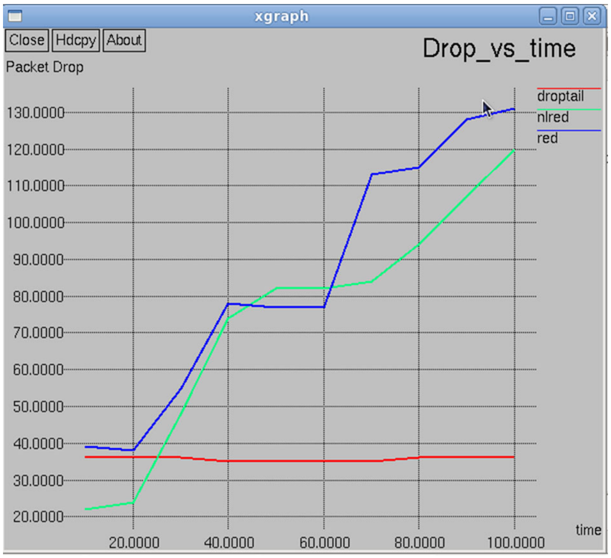

Fig. 11 Comparison of variation of packet drop for Droptail, RED and NLRED queue

4.4 Experiment 4

Simple dumb-bell topology is used in wired network. There are three nodes at each side of bottleneck link 3–4 where node 0 is acting as a RA source to send real audio traffic and node 6 as RA dump, while node 2 is acting as TCP source to send FTP traffic and node 6 as sink. Bandwidth at bottleneck link 3–4 is now increased to 10 MB. CBR flow rate, packet size and queue limit are kept same as in experiment 3. Here, all other parameters are same as in experiment 3, but the bandwidth is chosen in such a way to provide a sufficient amount of width to the packets to pass on (Figs. 13, 14, 15, 16; Tables 9, 10, 11, 12).

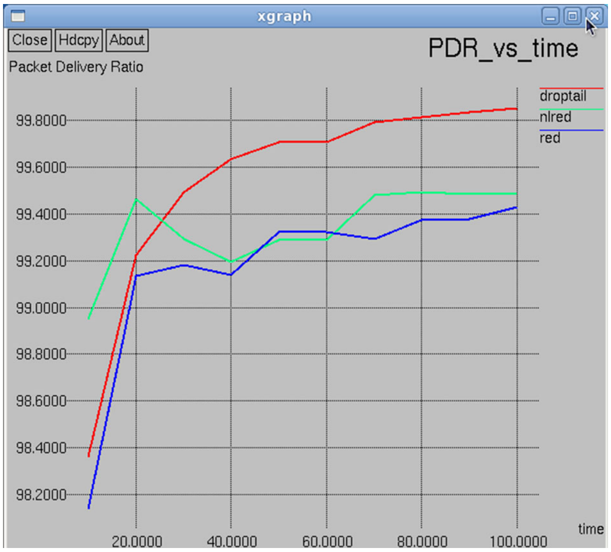

Fig. 12 Comparison of variation of PDR for Droptail, RED and NLRED queue

Table 5 Throughput analysis and results

Simulation time	DropTail	RED	NLRED
10	1.02411	1.02037	1.0268
20	1.03166	1.02735	1.02978
30	1.03401	1.02545	1.02986
40	1.03535	1.02624	1.02756
50	1.03598	1.02758	1.02856

Table 6 End-to-end delay analysis and results

Simulation time	DropTail	RED	NLRED
10	2.7992	2.60668	2.57713
20	5.25303	4.87861	4.99981
30	7.70504	7.27161	7.2506
40	10.1535	9.51449	9.69648
50	12.605	11.7855	12.0177

Table 7 Packet drop analysis and results

Simulation time	DropTail	RED	NLRED
10	36	39	22
20	36	38	24
30	36	55	48
40	35	78	74
50	35	77	82

Table 8 Packet delivery ratio (PDR) analysis and results

Simulation time	DropTail	RED	NLRED
10	98.3629	98.1402	98.9499
20	99.226	99.1358	99.4613
30	99.4932	99.1828	99.2928
40	99.6336	99.1371	99.1925
50	99.7085	99.3210	99.2873

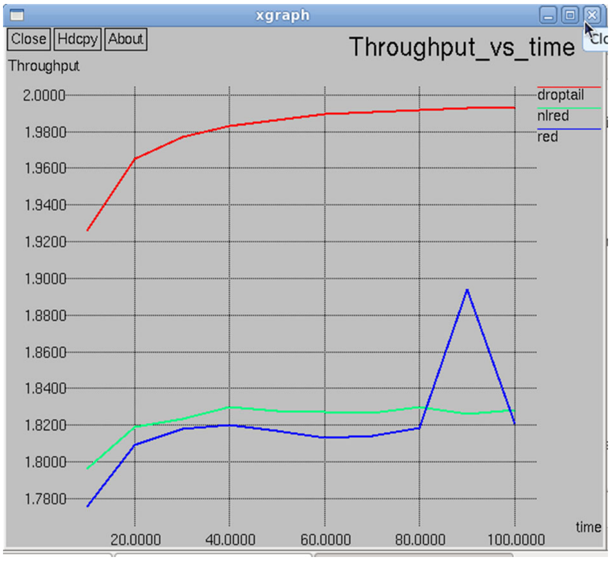

Fig. 13 Comparison of variation of throughput for Droptail, RED and NLRED queue

5 Conclusion

In order to trim the growing packet drop rates caused by epidemic inflation in network traffic, various queuing techniques come into picture. However, it is important to know which queuing mechanism is suitable in which network and traffic. Here, we analyzed the working of three queuing mechanisms, viz. Drop-tail, RED and NLRED on the basis of simulation results. It can be concluded that when there is congestion in the network, NLRED maintains good link utilization and small queue size so its performance is better than RED and Droptail. Also, it maintains high throughput, low packet drop, low delay and high packet delivery ratio than others. Fairness between flows achieved with NLRED is significantly better than that achieved from RED and Drop-tail. NLRED is not much sensitive to parameter such as max_p. Also, performance of Drop-tail is better than RED. But, in congested networks performance of NLRED holds better than both RED and

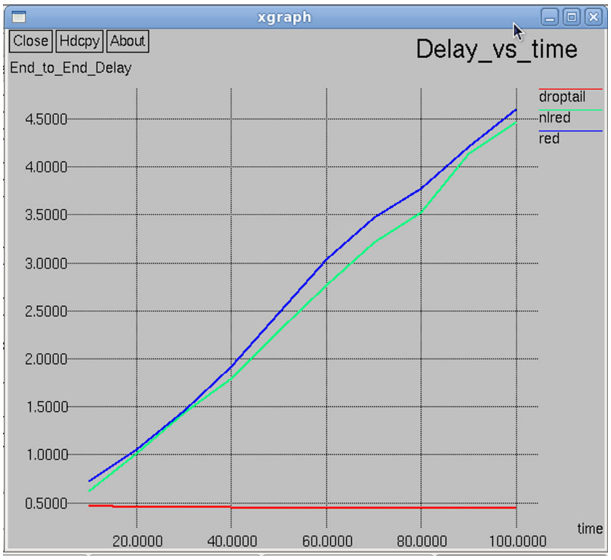

Fig. 14 Comparison of variation of end-to-end delay for Droptail, RED and NLRED queue

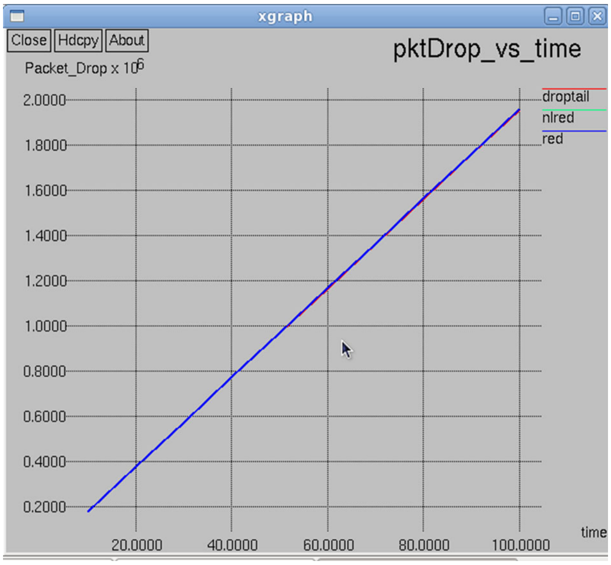

Fig. 15 Comparison of variation of packet drop for Droptail, RED and NLRED queue

Drop-tail. As we can clearly see in our experimental results, in experiment 1 there is a congested network so NLRED performed best, but on increasing the bandwidth at the bottleneck link, i.e., decreasing the network congestion in the experiment 2, we see that Drop-Tail performs better than RED and NLRED.

In experiment 3, where real audio packets are used (i.e., heavy) and bandwidth chosen was not sufficient to pass any of the packet, all the queues performed equally. But, on increasing the bandwidth here NLRED and RED performed better because here packets are heavy so this much bandwidth is not sufficient to remove the congestion. The

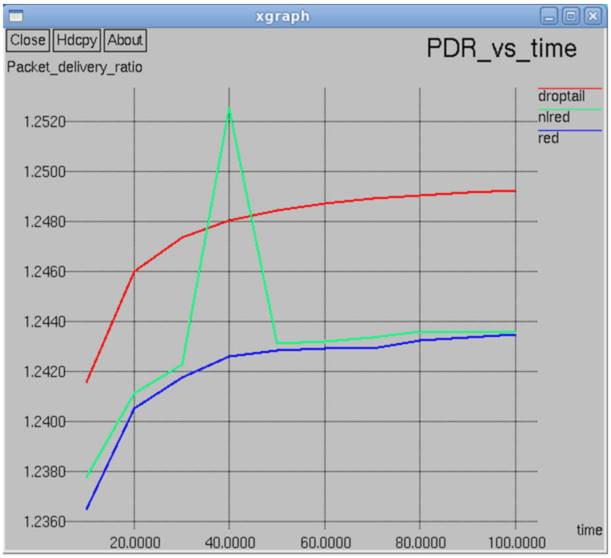

Fig. 16 Comparison of variation of PDR for Droptail, RED and NLRED queue

Table 9 Throughput analysis and results

Simulation time	DropTail	RED	NLRED
10	1.92612	1.77568	1.79623
20	1.96501	1.80912	1.81885
30	1.97707	1.81817	1.82350
40	1.98295	1.82042	1.82987
50	1.98643	1.81689	1.82803

Table 10 End-to-end delay analysis and results

Simulation time	DropTail	RED	NLRED
10	0.475924	0.72372	0.61625
20	0.457026	1.05407	1.01802
30	0.451199	1.46259	1.44238
40	0.448364	1.91779	1.79257
50	0.446688	2.47312	2.2949

Table 11 Packet drop analysis and results

Simulation time	DropTail	RED	NLRED
10	177862	178042	178017
20	375362	375767	375748
30	572863	573507	573492
40	770363	770948	771210
50	967864	969031	968966

Table 12 Packet delivery ratio (PDR) analysis and results

Simulation time	DropTail	RED	NLRED
10	1.24155	1.23647	1.23774
20	1.24599	1.24052	1.24110
30	1.24737	1.24174	1.24228
40	1.24805	1.24259	1.25252
50	1.24844	1.24283	1.24312

network is still congested and it gives results similar to experiment number 1. Therefore, when congestion is low, simple Drop-Tail mechanism achieves better throughput as compared to RED and NLRED because of the unnecessary computations involved in RED and NLRED to avoid congestion will decrease the overall performance. By comparing the network packet drop, delay and packet delivery ratio in both the traffics, it could be stated that RED uses the network bandwidth less effectively than Drop-Tail. When congestion is high, NLRED and RED perform better than droptail. However, in every case NLRED performed better than RED. This is because the uniform packet drop distribution from the random early packet drop behavior of RED will result in more application packet losses.

6 Future work and limitations

- This paper can be further built up for the analysis of various other queuing mechanisms such as fair queue (FQ), weighted fair queue (WFQ), round-robin queue (RR), weighted RR, and more versions of RED queues.
- In this paper, we analyzed the performance of three mechanisms on a small wired dumbbell network topology. Results may not be generalize to cases where network is much larger with multiple flows and also wireless.
- More performance metrics can also be introduced to variation of congestion window with time, queue size with time, any metrics with packet size, etc. Also, simulator parameters can be changed to get more information about queues specially parameter sensitive queues such as RED.

References

Abdel-jaber H (2015) Performance study of active queue management methods: adaptive GRED, REDD, and GRED-linear analytical model. J King Saud Univ Comput Inform Sci 27(4):416–429

Chen J, Hu C, Ji Z (2010) An improved ARED algorithm for congestion control of network transmission. Math Probl Eng 2010:329035. doi:10.1155/2010/329035

Chhabra K, Kshirsagar M, Zadgaonkar AS (2013) Effective congestion indication for performance improvement of random early detection. Int J Innov Technol Explor Eng 3:35–38

Huebner F (2004) Performance, quality of service, and control of next-generation communication networks ii. In: Proceedings of SPIE international society for optical engineering

Hui WANG, Xiao-Hui LIN, Kai-Yu ZHOU, Nin XIE, Hui LI (2009) On the scalable fairness and efficient active queue management 399 of RED. Int J Commun Netw Syst Sci 2(01):73–83. doi:10.4236/ijcns.2009.21009

Jain T, Annappa B, Tahiliani MP (2014). Performance evaluation of CoDel for active queue management in wired-cum-wireless networks. In: 2014 fourth international conference on advanced computing and communication technologies (ACCT). IEEE, pp 381–385

Jiang K, Wang X, Xi Y (2004) Nonlinear analysis of RED—a comparative study. Chaos Solitons Fractals 21(5):1153–1162

Joshi M, Mansata A, Talauliker S, Beard C (2005) Design and analysis of multi-level active queue management mechanisms for emergency traffic. Comput Commun 28(2):162–173

Katiyar V, Jain AC (2014) A survey on red and some it's varients incongestioncontrol mechanism. Int J Eng Manage Res 4(4):184–188

Kaur K, Kaur N, Singh G (2013) Performance comparison of queuing algorithms: a review. IOSR J Electron Commun Eng (IOSR-JECE) 8(4):46–48

Khademi N, Othman M (2010) Least attained service queue management for ns-2 network simulator. In: 2010 second international conference on computer research and development. IEEE, pp 317–321

Mohamed MHE (2010) Some active queue management methods for controlling packet queueing delay. Design and performance evaluation of some new versions of active queue management schemes for controlling packet queueing delay in a buffer to satisfy quality of service requirements for real-time multimedia applications. Doctoral dissertation, University of Bradford

Soni H, Mishra PK (2013) Reducing packet loss in active queue management. Int J Comput Appl 81(16):25–28

Xu Q, Sun J (2014) A simple active queue management based on the prediction of the packet arrival rate. J Netw Comput Appl 42:12–20

Yahia M, Bíró J (2005) Admission control for elastic traffic using RED and/or droptail queues. In Proceedings of the 8th international conference on telecommunications, 2005. ConTEL 2005, vol 2. IEEE, pp 391–396

Zaheer H, Pant M (2015) A differential evolution approach for solving integer programming problems. In: Proceedings of fourth international conference on soft computing for problem solving. Springer, India, pp 409–420

Zaheer H, Pant M, Kumar S, Monakhov O, Monakhova E, Deep K (2014) A new guiding force strategy for differential evolution. Int J Syst Assur Eng Manag. doi:10.1007/s13198-014-0322-6

Zhou K, Yeung KL, Li VO (2006) Nonlinear RED: a simple yet efficient active queue management scheme. Comput Netw 50(18):3784–3794