The Impact of the Degree of Self-Similarity on the NLREDwM Mechanism with Drop from Front Strategy

Adam Domański[1(✉)], Joanna Domańska[2], and Tadeusz Czachórski[2]

[1] Institute of Informatics, Silesian Technical University, Akademicka 16, 44-100 Gliwice, Poland
adamd@polsl.pl
[2] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Baltycka 5, 44-100 Gliwice, Poland
{joanna,tadek}@iitis.gliwice.pl

Abstract. This paper examines the impact of the degree of self-similarity on the selected AQM mechanisms. During the tests we analyzed the length of the queue, the number of rejected packets and waiting times in queues. We use fractional Gaussian noise as a self-similar traffic source. The quantitative analysis is based on simulation.

Keywords: Self-similarity · Active queue management · Non-linear RED · Dropping packets

1 Introduction

The development of the Internet is partially based on new solutions for traffic control to improve the Quality of Service (QoS) provided at the network layer. Among others, the studies are related to real-time applications such as Voice over IP or Video on Demand. To ensure QoS, the Internet Engineering Task Force (IETF) has proposed *Integrated Services* (IntServ) and *Differentiated Services* (DiffServ) architectures. They include a number of mechanisms, in particular for queue management in routers and the efficiency of the TCP protocol depends largely on them. Queue management may be passive or active. In passive solutions, packets coming to a buffer are rejected only if there is no space in the buffer to store them, hence the senders have no earlier warning on the danger of the increasing congestion and all packets coming during saturation of the buffer are lost.

To enhance the throughput and fairness of a link sharing, also to eliminate the synchronization, the IETF recommends active algorithms of buffer management (Active Queue Management, AQM) [1]. They incorporate mechanisms of preventive packet dropping already when there is still place to store packets, this way advertising that the queue is increasing and the danger of congestion is ahead. The packets are dropped randomly, hence only certain users are notified and the global synchronization of connections is avoided. The probability of packet rejection is increasing with the level of congestion.

© Springer International Publishing Switzerland 2016
P. Gaj et al. (Eds.): CN 2016, CCIS 608, pp. 192–203, 2016.
DOI: 10.1007/978-3-319-39207-3_17

The basic active queue management algorithm is Random Early Detection (RED) algorithm. It was primarily proposed in 1993 by Sally Floyd and Van Jacobson [2]. Its performance is based on a drop function giving probability that a packet is rejected. The argument avg of this function is a weighted moving average queue length determined at arrival of a packet:

$$avg = (1 - w_q)avg' + w_q q$$

where q is the current queue length, avg' is the previous value of avg and w_q is a weight parameter, typically $w_q << 1$, thus avg varies much more slowly than q. Therefore avg indicates long-term changes of q. If $avg < Min_{\text{th}}$, all packets are admitted. If $Min_{\text{th}} < avg < Max_{\text{th}}$, then dropping probability p is increasing linearly:

$$`p = p_{\max} \frac{avg - Min_{\text{th}}}{Max_{\text{th}} - Min_{\text{th}}}.$$

The value $p_{\max}$ corresponds to a probability of packet rejecting at $avg = Max_{\text{th}}$. If $avg > Max_{\text{th}}$ then all packets are dropped. Dropping probability p is thus dependent on network load.

Efficient operation of the RED mechanism depends on the proper selection of its parameters. There were several works on the impact of various parameters on the RED performance [3] and many variants of RED mechanism were developed to improve its performance [4–6]. They may be classified according to the dropping packet function and according to the parameters of the algorithm. Section 2 briefly reviews the modifications of the RED mechanism studied in this article.

Research related to the Internet traffic aims to provide a better understanding of the modern Internet, inter alia, by presenting the current characteristics of Internet traffic based on a large number of experimental data and introducing the internet traffic models. The understanding of the traffic nature of the modern Internet is important to the Internet community. It supports optimization and development of protocols and network devices, improves the network applications security and the protection of network users.

Measurements and statistical analysis (performed already in the 90s) of packet network traffic show that this traffic displays a complex statistical nature including self-similarity, long-range dependence and burstiness [7–10].

Self-similarity of a process means that the change of time scale does not influence the statistical characteristics of the process. It results in long-distance auto-correlation and makes possible the occurrence of very long periods of high (or low) traffic intensity. These features have a great impact on a network performance [11]. They enlarge mean queue lengths at buffers and increase the probability of packet loss, deteriorating this way the quality of services provided by a network.

In consequence, it is needed to propose new or to adapt known types of stochastic processes able to model these negative phenomena in network traffic. Several models have been introduced to imitate self-similar processes in the network traffic. They use fractional Brownian Motion, chaotic maps, fractional Autoregressive Integrated Moving Average (fARIMA), wavelets and multifractals, and processes based on Markov chains: SSMP (Special Semi-Markov

Process), MMPP (Markov-Modulated Poisson Process) [12], HMM (Hidden Markov Model) [13]. Section 3 briefly describes the self-similar traffic source used in this article.

2 Our Modifications of the RED Mechanism

Our previous works [14–16] presented a study of the influence of RED modifications on its performance in the presence of self-similar traffic.

In classic RED and its variations described in the literature a packet to be dropped is taken usually from the end of the queue. As Sally Floyd wrote: "when RED is working right, the average queue size should be small, and it shouldn't make too much difference one way or another whether you drop a packet at the front of the queue or at the tail". Our article [14] reconsiders the problem of choosing tail or front packets in presence of self-similar traffic.

It was shown that in the case of light non-self-similar traffic the obtained results confirmed the opinion of S. Floyd. If the mean queue length is relatively low, the influence of dropping scheme on queueing time is negligible: the introduction of drop-front strategy gives less then 1 % shorter mean queueing time. In the case of heavy traffic, drop from front strategy gives two times shorter mean queueing times. However, when the Poisson traffic is replaced by self-similar one with the same intensity and the same parameters of RED are preserved, the length of the queue increases and the influence of the dropping scheme is more visible: drop from front strategy reduces mean queueing time by about 16 % even in the case of light load. This fact confirms the advantage of drop from front strategy if the traffic exhibits the self-similarity.

In [15] we investigated the influence of the self-similarity on the non-linear packet dropping function in a special case of NLRED queues. In the NLRED mechanism the linear packet dropping function is usually replaced by a quadratic function. We introduced a linear combination of independent polynomials of 3rd degree:

$$p(x, a_1, a_2, p_{\max}) = \begin{cases} 0 & \text{for} \quad x < Min_{\text{th}} \\ \varphi_0(x) + a_1\varphi_1(x) + a_2\varphi_2(x) & \text{for} \quad Min_{\text{th}} \le x \le Max_{\text{th}} \\ 1 & \text{for} \quad x > Max_{\text{th}} \end{cases}$$

where the set of basis function is defined as follows:

$$\begin{aligned} \varphi_0(x) &= p_{\max}\frac{x - Min_{\text{th}}}{Max_{\text{th}} - Min_{\text{th}}}, \\ \varphi_1(x) &= (x - Min_{\text{th}})(Max_{\text{th}} - x), \\ \varphi_2(x) &= (x - Min_{\text{th}})^2(Max_{\text{th}} - x). \end{aligned}$$

The process of finding the best values of $p_{\max}$, a_1 and a_2 for a given type of traffic may be considered as optimization problem in 3-dimensional space. The experimental results show the existence of one optimal set of values of parameters; self-similarity of network traffic and traffic load have no influence on the choice of the optimal dropping packet function. The results obtained for this optimal set of parameter values demonstrate that the mean waiting time is two and half times shorter compared to the RED mechanism in the case of non-self-similar traffic and it is four times shorter in the case of self-similar traffic.

Then in [16] we investigated the impact of the way how the weighted moving average is defined on the performance of the RED mechanism in the presence of self-similar traffic. The proposed approach, named REDwM, is an extension of RED where the average queue length $A(n)$ at a moment n is given by the first order difference equation

$$\begin{aligned} A(n) = a_1 A(n-1) + a_2 A(n-2) + \ldots + a_k A(n-k) \\ + b_0 Q(n) + b_1 Q(n-1) + \ldots + b_m Q(n-m) \end{aligned}$$

where a_j $(j = 1, \ldots, k)$ and b_i $(i = 0, \ldots, m)$ are constant, $A(l)$ is the average queue length at the l-th moment of time, and $Q(l)$ is the current length of the packet queue at the l-th moment; a_j and b_i are subject to constraints:

$$\sum_{j=1}^{k} a_j + \sum_{i=0}^{m} b_i = 1 \ \wedge \ a_j \geq 0 \ \wedge \ b_i \geq 0.$$

The optimal values of equation coefficients were found during minimization of the score function. The improvements, following numerical experiments, are over 5 % if we refer to results given by the classic RED approach (for the assumed score function based on the mean waiting time).

The improvements of the RED mechanism described above may be combined making NLREDwM mechanism. The primary goal of this article is to study its performance.

3 Self-Similar Traffic Source

Previously in [14–16] we used the SSMP markovian traffic source to represent the self-similar traffic. Such Markov based model can generate a self-similar traffic over a finite number of time scales. Here we use fractional Gaussian noise which is an exactly self-similar traffic source.

Fractional Gaussian noise (fGn) has been proposed in [17] as a model for the long-range dependence postulated to occur in a variety of hydrological and geophysical time series. Nowadays, fGn is one of the most commonly used self-similar processes in network performance evaluation [18] and the only stationary Gaussian process being exactly self-similar.

The autocorrelation function of fGn process [7]

$$\rho^{(m)}(k) = \rho(k) = \frac{1}{2}\left[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\right]$$

assures second-order self-similarity.

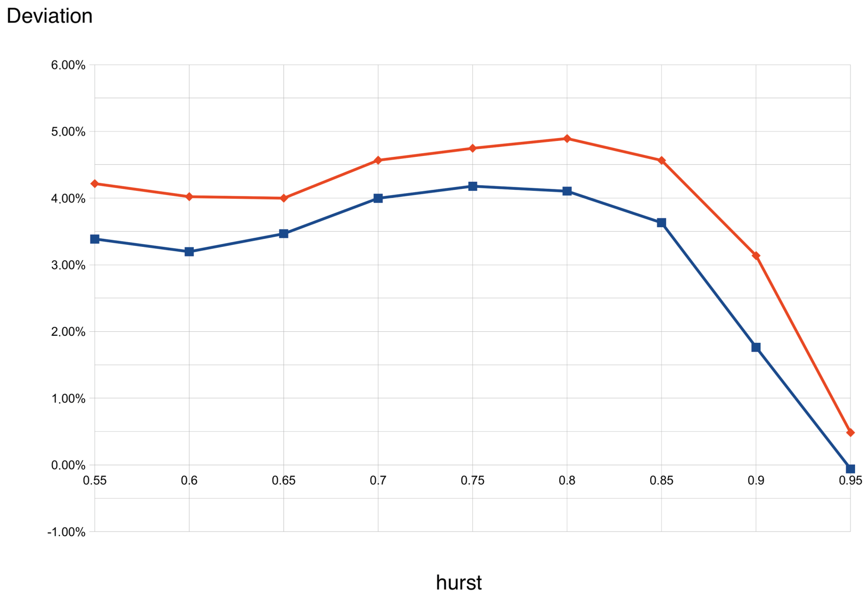

Fig. 1. Maximum and minimum difference between assumed and estimated Hurst parameter

The synthetic generation of sample paths (traces) of self-similar processes is an important problem [18]. In this paper we use a fast algorithm for generating approximate sample paths for a fGn process, first introduced in [19].

The Hurst parameter H characterizes a process in terms of the degree of self-similarity, the degree increases with the increase of H. Thge value $H \leq 0.5$ denotes the lack of long-range dependence, but the process is still self-similar, [20]. We have generated the sample traces with the Hurst parameter with the range of 0.5 to 0.95. After each trace generation, the parameter was estimated with the use of aggregated variance method [21]. Table 1 presents results of this estimation for 10 generated traces with the Hurst parameter assumed to be equal to 0.7. These results show that the assumed and estimated Hurst parameters are not the same. This situation repeated for each value of Hurst parameter (see Fig. 1).

Table 1. Estimated Hurst parameters obtained for sample fGn traces generated for assumed Hurst parameter $H = 0.7$

Trace number	Estimated Hurst parameter	Trace number	Estimated Hurst parameter
1	0.7279822	6	0.73197
2	0.7299411	7	0.7311628
3	0.7288566	8	0.7291909
4	0.731594	9	0.7290085
5	0.7313482	10	0.7284157

Table 2. FIFO queue

	Hurst parameter	Mean queue length	Mean waiting time	Rejected packets	
Tail drop	0.50	299.099	119.380	249520	49.90 %
Front drop	0.50	299.089	119.223	249494	49.89 %
Tail drop	0.70	298.118	119.158	249879	49.97 %
Front drop	0.70	298.132	119.118	250034	50.01 %
Tail drop	0.80	296.878	118.883	250354	50.07 %
Front drop	0.80	296.870	118.772	250383	50.08 %
Tail drop	0.90	248.553	102.061	256587	51.32 %
Front drop	0.90	247.848	101.399	255954	51.19 %

fGn generator usually generates the traffic with a slightly greater Hurst parameter. The difference between assumed and estimated Hurst parameter decreases with the increase of the value of Hurst parameter. For our purpose we chose the samples with the smallest difference.

4 Obtained Results

The simulation model of an appropriate AQM mechanism was prepared with the use of SimPy. SimPy is a process-based discrete-event simulation framework based on the language Python. Its event dispatcher is based on Python's generators [22]. SimPy is released under the MIT License (free software license originating at the Massachusetts Institute of Technology).

We investigated the influence of combination of modifications described in Sect. 2 on RED performance. We also studied the impact of the degree of self-similarity on the examined AQM mechanisms. During the tests we analyzed the following parameters of the transmission with AQM: the length of the queue, queue waiting times and the number of rejected packets. The service time represented the time of a packet treatment and dispatching. The input process, following fGn was based on descrete time slots (1 or 0 arrivals in a time slot), the average interarrival time was 2 time slots. The size of this time slot was our symbolic time unit presented in figures. The service time was geometrically distributed with the average of 4 time slots. That means a heavy charge, leading to the link saturation, as our goal was to study the mechanisms performance at high load intervals. The type of the distributions and their mean values were the same for all Hurst parameters.

Table 2 shows the results obtained for the FIFO queue without any AQM mechanism. The introduction of *drop from front strategy* gives shorter mean waiting time compared to *drop tail strategy*. Additionally, an increase in the degree of self-similarity causes an increase in number of rejected packets. The

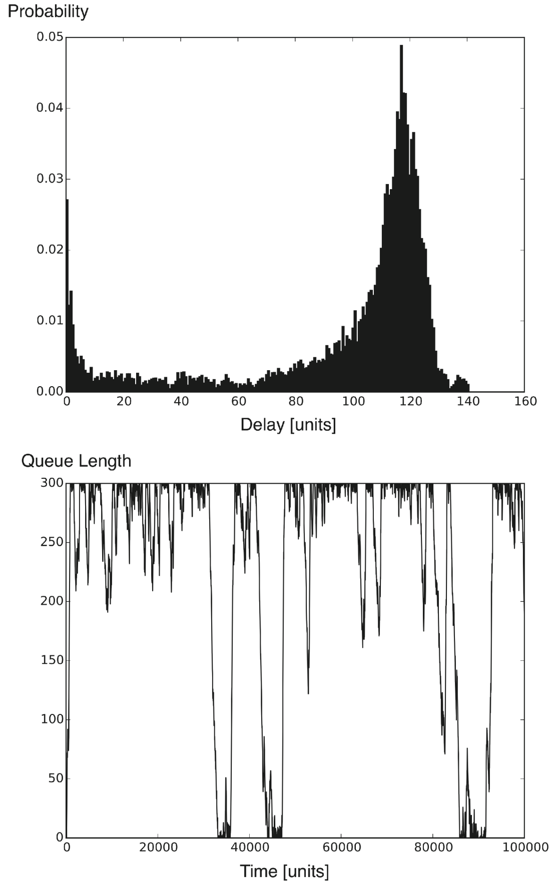

Fig. 2. FIFO queue – drop from front, waiting time distribution (top), fluctuations of queue length (bottom), Hurst parameter $H = 0.90$

reason of it is bursty nature of self-similar traffic. Figure 2 shows fluctuations of queue length in the case of $H = 0.9$.

The same results were obtained in case of the RED queue (see Table 3). The RED parameters were: buffer size 300 packets, threshold values $Min_{\text{th}} = 100$ and $Max_{\text{th}} = 200$, $p_{\max} = 0.1$, $w = 0.02$. We distinguish packets rejected when RED starts dropping packets and packets rejected when the walking average of the queue is at the maximum threshold. Table 3 shows that majority of packets were rejected when the average is at the maximum threshold. The number of packets rejected because of reaching the maximum threshold increased with the increase of Hurst parameter.

Tables 4 and 5 show the results obtained for two sets of parameter values of our NLRED mechanism (described in Sect. 2). The first set of values of parameters (Table 4) refers to the case with the minimal value of average waiting time

Table 3. RED queue

	Hurst parameter	Mean queue length	Mean waiting time	No. of rejected packets			
				$\leq Max_{\text{th}}$		$> Max_{\text{th}}$	
Tail drop	050	199.801	79.865	27616	5.52 %	222436	44.48 %
Front drop	050	199.841	79.477	27083	5.41 %	222318	44.46 %
Tail drop	070	198.549	79.485	26948	5.38 %	223461	44.69 %
Front drop	070	198.524	79.079	27230	5.44 %	222519	44.50 %
Tail drop	080	196.941	78.473	26805	5.36 %	222453	44.49 %
Front drop	080	196.864	78.604	26491	5.29 %	223819	44.76 %
Tail drop	090	158.818	66.119	19108	3.82 %	240991	48.19 %
Front drop	090	158.541	65.797	19487	3.89 %	240405	48.08 %

at the expense of the number of rejected packets (the best case). The second set (Table 5) refers to the case with the maximal value of average waiting time (the worst case). The results obtained for the worst case resemble those obtained for the classical RED mechanism. In the best case all packets are rejected when RED starts dropping packets. In this case the mean waiting time is 1.77 times shorter compared to the RED mechanism (in the case of $H = 0.9$). The results presented in both tables show the impact of degree of self-similarity on mean queue length, mean waiting time and number of rejected packets. Figures 3 and 4 compare the waiting time distributions and queue length fluctuations (the best case of NLRED) for non-self-similar traffic to the case of self-similar traffic with $H = 0.9$.

Table 4. NLRED queue; $a1 = 0.00042$, $a2 = -0.0000038$, $p_{\max} = 0.855$

	Hurst parameter	Mean queue length	Mean waiting time	No. of rejected packets		
				$\leq Max_{\text{th}}$		$> Max_{\text{th}}$
Tail drop	050	112.3725	44.8316	249846	49.96 %	0
Front drop	050	112.4435	44.6774	249943	49.98 %	0
Tail drop	070	111.3998	44.6255	250865	50.17 %	0
Front drop	070	111.3741	44.2123	249685	49.93 %	0
Tail drop	080	109.9958	43.8370	249588	49.91 %	0
Front drop	080	109.9371	43.6606	249788	49.95 %	0
Tail drop	090	87.7811	36.8922	264244	52.84 %	0
Front drop	090	87.7484	37.0343	264662	52.93 %	0

Table 5. NLRED queue; $a1 = -0.00008$, $a2 = -0.0000008$, $p_{\max} = 0.6$

	Hurst parameter	Mean queue length	Mean waiting time	No. of rejected packets			
				$\leq Max_{\text{th}}$		$> Max_{\text{th}}$	
Tail drop	050	194.1852	77.8825	227407	45.48 %	23477	4.69 %
Front drop	050	194.1163	77.3081	227516	45.50 %	22249	4.44 %
Tail drop	070	191.3899	76.4370	201347	40.26 %	48478	9.70 %
Front drop	070	191.3007	76.2139	200608	40.12 %	49243	9.85 %
Tail drop	080	188.8292	75.4878	183403	36.68 %	66689	13.34 %
Front drop	080	188.8145	75.1376	184170	36.83 %	65392	13.08 %
Tail drop	090	152.5569	63.7271	143043	28.61 %	117863	23.57 %
Front drop	090	152.4339	63.4312	142710	28.54 %	118333	23.67 %

Table 6. NLREDwM mechanism; $a1 = 0.00042$, $a2 = -0.0000038$, $p_{\max} = 0.855$

	Hurst parameter	Mean queue length	Mean waiting time	No. of rejected packets		
				$\leq Max_{\text{th}}$		$> Max_{\text{th}}$
Tail drop	050	112.4056	44.8221	249713	49.94 %	0
Front drop	050	112.4650	44.8019	250599	50.12 %	0
Tail drop	070	111.3513	44.3102	249209	49.84 %	0
Front drop	070	111.3267	44.1467	249418	49.88 %	0
Tail drop	080	109.9675	43.7743	249306	49.86 %	0
Front drop	080	110.1168	43.7970	250160	50.03 %	0
Tail drop	090	87.5870	37.1751	264962	52.99 %	0
Front drop	090	87.4502	36.8666	264395	52.88 %	0

Table 7. NLREDwM mechanism; $a1 = -0.00008$, $a2 = -0.0000008$, $p_{\max} = 0.6$

	Hurst parameter	Mean queue length	Mean waiting time	No. of rejected packets			
				$\leq Max_{\text{th}}$		$> Max_{\text{th}}$	
Tail drop	050	194.2196	77.5939	228135	45.63 %	21802	4.36 %
Front drop	050	194.1596	77.3485	226814	45.36 %	23049	4.61 %
Tail drop	070	191.4063	76.5439	201466	40.29 %	48687	9.74 %
Front drop	070	191.4217	76.4816	201918	40.38 %	48654	9.73 %
Tail drop	080	188.8828	75.3979	184848	36.97 %	64876	12.98 %
Front drop	080	188.8172	75.1325	184379	36.88 %	65160	13.03 %
Tail drop	090	152.4997	63.7525	141616	28.32 %	119465	23.89 %
Front drop	090	152.3294	63.4188	142388	28.48 %	118311	23.66 %

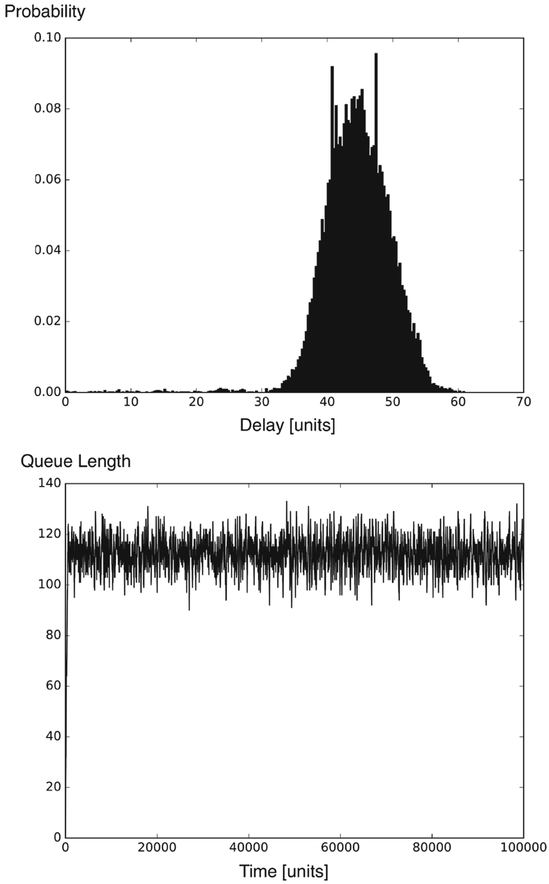

Fig. 3. NLRED – tail drop, waiting time distribution (top), fluctuations of queue length (bottom), $H = 0.50$, $a1 = 0.00042$, $a2 = -0.0000038$, $p_{\max} = 0.855$

Tables 6 and 7 show the results obtained for NLREDwM mechanism (best and worse case), which is a combination of our NLRED and REDwM mechanisms (described in Sect. 2). The introduction of modified weighted moving average function gives about 0.1 % of changes compared to NLRED. This improvement increases with the increase of Hurst parameter.

5 Conclusions

The article confirms the important impact of the degree of self-similarity (expressed in terms of Hurst parameter) on the following parameters of the transmission with AQM: the length of the queue, queue waiting times and the number of rejected packets. We discuss the problem of choosing the optimal shape of dropping packet function for NLRED algorithm and at the same time investigate the influence of the weighted moving average on packet waiting time reduction for this NLRED mechanism.

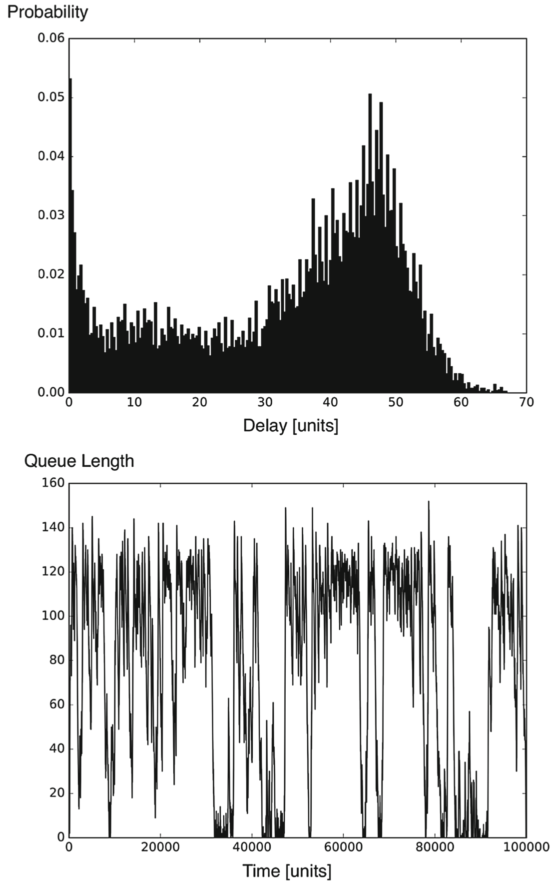

Fig. 4. NLRED – tail drop, waiting time distribution (top), fluctuations of queue length (bottom), $H = 0.90$, $a1 = 0.00042$, $a2 = -0.0000038$, $p_{\max} = 0.855$

Drop from front strategy, when applied in place of tail drop one, results in reduction of packet waiting time in examined AQM mechanism. Obtained results are closely related to the level of self-similarity. Hence the application of presented AQM mechanism may be recommended for bursty traffic connections with real-time requirements.

References

1. Braden, B., et al.: Recommendations on queue management and congestion avoidance in the internet. RFC 2309, IETF (1998)
2. Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. IEEE/ACM Trans. Network. **1**(4), 397–413 (1993)
3. May, M., Diot, C., Lyles, B., Bolot, J.: Influence of active queue management parameters on aggregate traffic performance. Research Report, INRIA (2000)
4. Ho, H.-J., Lin, W.-M.: AURED - Autonomous random early detection for TCP congestion control. In: 3rd International Conference on Systems and Networks Communications, Malta (2008)

5. Bhatnagar, S., Patro, R.: A proof of convergence of the B-RED and P-RED algorithms for random early detection. IEEE Commun. Lett. **13**(10), 809–811 (2009)
6. Domański, A., Domańska, J., Czachórski, T.: Comparison of CHOKe and gCHOKe active queues management algorithms with the use of fluid flow approximation. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2013. CCIS, vol. 370, pp. 363–371. Springer, Heidelberg (2013)
7. Karagiannis, T., Molle, M., Faloutsos, M.: Long-range dependence: ten years of internet traffic modeling. IEEE Internet Comput. **8**(5), 57–64 (2004)
8. Domański, A., Domańska, J., Czachórski, T.: The impact of self-similarity on traffic shaping in wireless LAN. In: Balandin, S., Moltchanov, D., Koucheryavy, Y. (eds.) NEW2AN 2008. LNCS, vol. 5174, pp. 156–168. Springer, Heidelberg (2008)
9. Domańska, J., Domański, A., Czachórski, T.: A few investigation of long-range dependence in network traffic. In: Czachórski, T., Gelenbe, E., Lent, R. (eds.) Information Science and Systems 2014, pp. 137–144. Springer, Heidelberg (2014)
10. Domańska, J., Domański, A., Czachórski, T.: Estimating the intensity of long-range dependence in real and synthetic traffic traces. In: Gaj, P., Kwiecień, A., Stera, P. (eds.) CN 2015. CCIS, vol. 522, pp. 11–22. Springer, Heidelberg (2015)
11. Domańska, J., Domański, A.: The influence of traffic self-similarity on QoS mechanism. In: International Symposium on Applications and the Internet, SAINT, Trento, Italy (2005)
12. Domańska, J., Domański, A., Czachórski, T.: Modeling packet traffic with the use of superpositions of two-state MMPPs. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2014. CCIS, vol. 431, pp. 24–36. Springer, Heidelberg (2014)
13. Domańska, J., Domański, A., Czachórski, T.: Internet traffic source based on hidden markov model. In: Balandin, S., Koucheryavy, Y., Hu, H. (eds.) NEW2AN 2011 and ruSMART 2011. LNCS, vol. 6869, pp. 395–404. Springer, Heidelberg (2011)
14. Domańska, J., Domański, A., Czachórski, T.: The drop-from-front strategy in AQM. In: Koucheryavy, Y., Harju, J., Sayenko, A. (eds.) NEW2AN 2007. LNCS, vol. 4712, pp. 61–72. Springer, Heidelberg (2007)
15. Domańska, J., Augustyn, D.R., Domański, A.: The choice of optimal 3rd order polynomial packet dropping function for NLRED in the presence of self-similar traffic. Bull. Polish Acad. Sci., Tech. Sci. **60**(4), 779–786 (2012)
16. Domańska, J., Domański, A., Augustyn, D.R., Klamka, J.: A RED modified weighted moving average for soft real-time application. Int. J. Appl. Math. Comput. Sci. **24**(3), 697–707 (2014)
17. Mandelbrot, B.B., Ness, J.V.: Fractional brownian motions, fractional noises and applications. SIAM Rev. **10**, 422–437 (1968)
18. Lopez-Ardao, J.C., Lopez-Garcia, C., Suarez-Gonzalez, A., Fernandez-Veiga, M., Rodriguez-Rubio, R.: On the use of self-similar processes in network simulation. ACM Trans. Model. Comput. Simul. **10**(2), 125–151 (2000)
19. Paxson, V.: Fast, approximate synthesis of fractional Gaussian noise for generating self-similar network traffic. ACM SIGCOMM Comput. Commun. Rev. **27**(5), 5–18 (1997)
20. Samorodnitsky, G., Taqqu, M.S.: Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance. Chapman and Hall, New York (1994)
21. Clegg, R.G.: A practical guide to measuring the Hurst parameter. Int. J. Simul. **7**(2), 3–14 (2006)
22. http://simpy.readthedocs.org