The Drop-From-Front Strategy in AQM

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

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

Abstract. The article investigates the influence of the way packets are chosen to be dropped (end of the tail, head of the tail) on the performance, i.e. response time for in case of RED and DSRED queues - two representative active queue management mechanisms used in IP routers. In particular, the self-similar traffic is considered. The quantitative analysis is based on simulation and Markov chain models solved numerically.

1 Introduction

The algorithms of queue management at IP routers determine which packet should be deleted when necessary. The active queue management, recommended now by IETF, enhances the efficiency of transfers and cooperate with TCP congestion window mechanism in adapting the flows intensity to the congestion at a network [1]. In classic RED the incoming packet is considered to be dropped or marked. In [2] S. Floyd wrote: *"when RED is working right the average queue size should be small, and it shouldn't make too much different one way or another whether you drop a packet at the front of the queue or at the tail"*. Here, we reconsider the problem of choosing either tail or front packets in presence of self-similar traffic. Sections 2 gives basic notions on active queue management, Section 3 presents briefly a self-similar model used in the article. Section 4 gives Markov chain and simulation models of the considered two active queue management schemes: RED and Double-Slope RED (DSRED). Section 5 discusses numerical results, some conclusions are given in Section 6.

2 Active Queue Management

In *passive* queue management, 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 growing congestion. In this case all packets coming during saturation of the buffer are lost. The existing schemes may differ on the

Y. Koucheryavy, J. Harju, and A. Sayenko (Eds.): NEW2AN 2007, LNCS 4712, pp. 61–72, 2007.

choice of packet to be deleted (end of the tail, head of the tail, random). During a saturation period all connections are affected and all react in the same way, hence they become synchronised. To enhance the throughput and fairness of the link sharing, also to eliminate the synchronisation, the Internet Engineering Task Force (IETF) recommends *active* algorithms of buffer management. They incorporate mechanisms of preventive packet dropping when there is still place to store some packets, to advertise that the queue is growing and the danger of congestion is ahead. The probability of packet rejection is growing together with the level of congestion. The packets are dropped randomly, hence only chosen users are notified and the global synchronisation of connections is avoided. A detailed discussion of the active queue management goals may be found in [1].

The RED (Random Early Detection) algorithm was proposed by IETF to enhance the transmission via IP routers. It was primarily described by Sally Floyd and Van Jacobson in [3]. 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, acting as a low-pass filter and calculated at the arrival of each packet as

$$avg = (1 - w)avg' + wq$$

where avg' is the previous value of avg, q is the current queue length and w is a weight determining the importance of the instantaneous queue length, typically $w \ll 1$. If w is too small, the reaction on arising congestion is too slow, if w is too large, the algorithm is too sensitive on ephemeral changes of the queue (noise). Articles [3,4] recommend $w = 0.001$ or $w = 0.002$, and [5] shows the efficiency of $w = 0.05$ and $w = 0.07$. Article [6] analyses the influence of w on queueing time fluctuations, obviously the larger w, the higher fluctuations. In RED drop function there are two thresholds Min_{th} and Max_{th}. If $avg < Min_{th}$ all packets are admitted, if $Min_{th} < avg < Max_{th}$ then dropping probability p is growing linearly from 0 to p_{max} :

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

and if $avg > Max_{th}$ then all packets are dropped. The value of p_{max} has also a strong influence on the RED performance: if it is too large, the overall throughput is unnecessarily choked and if it's too small the danger of synchronisation arises; [4] recommends $p_{max} = 0.1$. The problem of the choice of parameters is still discussed, see e.g. [7,8]. The mean *avg* may be also determined in other way, see [9] for discussion. Despite of evident highlights, RED has also such drawbacks as low throughput, unfair bandwidth sharing, introduction of variable latency, deterioration of network stability. Therefore numerous propositions of basic algorithms improvements appear, their comparison may be found e.g. in [10].

DSRED (double-slope RED) introduced in [11] and developed in [12] is one of these modifications. Three thresholds K_l, K_m and K_h (usually $K_m = (K_l + K_h)/2$) and parameter γ determine two slopes of the DSRED drop function:

$$p(avg) = \begin{cases} 0 & \text{if } avg < K_l \\ \alpha(avg - K_l) & \text{if } K_l \leq avg < K_m \\ 1 - \gamma + \beta(avg - K_m) & \text{if } K_m \leq avg < K_h \\ 1 & \text{if } K_h \leq avg \leq N \end{cases}$$

where

$$\alpha = \frac{2(1-\gamma)}{K_h - K_l}, \qquad \beta = \frac{2\gamma}{K_h - K_l}$$

The double slope function makes the algorithm more elastic (more parameters to fix); gentle at the beginning (for low congestion) drop function enhances throughput and reduces queue waiting times.

In this article, we present analytical (based on Markov chain) and simulation models of RED and DSRED. We assume either Poisson or self-similar traffic. Because of the difficulty in analyzing RED mathematically [13], RED and DSRED are studied in an open-loop scenario.

3 Self-similarity of Network Traffic

Measurements and statistical analysis of network traffic, e.g. [14,15] show that it displays a self-similar character. It is observed on various protocol layers and in different network structures. Self-similarity of a process means that the change of time scales does not affect the statistical characteristics of the process. It results in long-range dependence 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. They enlarge the mean queue lengths at buffers and increase the probability of packet losses, reducing this way the quality of services provided by a network. Also TCP/IP traffic is characterised by burstiness and long-term correlation, [16], its features are additionally influenced by the performance of congestion avoidance and congestion management mechanisms, [17,18].

Let a process X_k represent the traffic intensity measured in fixed time intervals and let the aggregated process $X_k^{(m)}$ be the average of the basic process over a group of m consecutive samples: $X_k^{(m)} = \frac{1}{m}(X_{k \cdot m - m + 1} + ... + X_{k \cdot m})$, where $k \geq 1$. There are several methods used to check if a process is self-similar. The easiest one is a visual test: one can observe the behaviour of the basic process X_t and the aggregated process $X_k^{(m)}$. If these processes have the same character - the increase of m does not smooth the process, the process is self-similar. More formally, the difference between short-range dependent and long-range dependent (self-similar) process is as follows [15]: for the first process the sum of covariance $\sum_{k=0}^{k=\infty} cov(k)$ is convergent, the spectrum of the process $S(\omega) = \sum_{k=-\infty}^{k=\infty} R(k) e^{-j\omega k}$, where $R(k)$ is the autocorrelation function of the process, is finite at $\omega = 0$, and the variance $var(X_k^{(m)})$ tends asymptotically for large m to the function $\frac{var(X)}{m}$. In the case of long-range dependent process, the sum of covariance $\sum_{k=0}^{k=\infty} cov(k)$ is divergent, $S(0)$ is singular, and $var(X_k^{(m)})$

tends asymptotically to $\frac{var(X)}{m^{\beta}}$, where $0 < \beta < 1$. The parameter β is related to the Hurst parameter H (often used to characterise the self-similarity of a process): $H = 1 - \frac{\beta}{2}$ [15]. For $0.5 < H \leq 1$ process is self-similar; the closer H is to 1, the greater is the degree of persistence of long-range dependence.

To represent the self-similar traffic we use here a model introduced by S. Robert [19,20]. The time of the model is discrete and divided into unit length slots. Only one packet can arrive during each time-slot. In the case of memoryless, geometrical source, the packet comes into system with fixed probability α_1. In the case of self-similar traffic, packet arrivals are determined by a n-state discrete time Markov chain called modulator. It was assumed that modulator has $n = 5$ states ($i = 0, 1, \ldots 4$) and packets arrive only when the modulator is in state $i = 0$. The elements of the modulator transition probability matrix depend only on two parameters: q and a – therefore only two parameters should be fitted to match the mean value and Hurst parameter of the process. If p_{ij} denotes the modulator transition probability from state i to state j, then it was assumed that $p_{0j} = 1/a^j$, $p_{j0} = (q/a)^j$, $p_{jj} = 1 - (q/a)^j$ where $j = 1, \ldots, 4$, $p_{00} = 1 - 1/a - \ldots - 1/a^4$, and remaining probabilities are equal to zero. The passages from the state 0 to one of other states determine the process behaviour on one time scale, hence the number of these states corresponds to the number of time-scales where the process may by considered as self-similar.

The model was fitted to real data [21]. This model enables us to represent, with the use of few parameters, a network traffic which is self-similar over several time-scales.

4 Analytical and Simulation Models of RED and DSRED

The RED or DSRED queue mechanisms are represented by a single-server model based either on discrete-time Markov chain or simulation. The service time represents the time of a packet treatment and dispatching. Its distribution is geometric. The model of incoming traffic was presented above. For both considered in comparisons cases, i.e. for geometric interarrival time distribution (which corresponds to Poisson traffic in case of continuous time models) and self-similar traffic, the considered traffic intensities are the same. A detailed discussion of the choice of model parameters is also presented in [22].

In Markov model, the Markov chain state is defined by the number of packets in the queue, the integer part of the *avg* value and by four flags u_1, u_2, u_3, u_4 approximating the rest of this value (as *avg* is a real number, it is impossible to attribute a state to each of the infinite number of its possible values) in the following way:

$$\frac{[(i-1) * 0.25] + (i * 0.25)}{2}$$

where i is the number of non-zero flag. If all flags are null, we assume the integer value of *avg*. In case of self-similar traffic this state definition is supplemented by a variable denoting the state of the modulator.

The vector p of state probabilities is given by a system of linear equations

$$p = p * P$$

where P is the transition probability matrix which is generally large (the number of states, hence the order of the matrix P may be hundreds of thousands or millions), sparse and ill conditioned, and the use of well known and broadly used numerical algorithms for algebraic and differential equation systems gives poor results. That is why a projection method using Krylov subspaces, as recommended in [23] was chosen.

The method of Arnoldi is an orthogonal projection process onto the Krylov subspace. It may be used to compute approximations to the unit eigenvalue and the corresponding eigenvector of the matrix P. The matrix H_m (upper Hessenberg matrix) represents the restriction of the linear transformation P to the subspace K_m. Approximation of the eigenvalue of P can be obtained from the eigenvalue of H_m. We often use the so-called Rayleigh-Ritz procedure for extracting eigenvalue and eigenvector approximations from a given subspace. If λ_i is an eigenvalue of H_m and p_i the corresponding eigenvector, i.e.,

$$H_m p_i = \lambda_i p_i,$$

then λ_i is taken as an approximation to an eigenvalue of P, and $V_m p_i$ as an approximation to the corresponding eigenvector of P.

To simplify the notation, we denote: $v = p(t_i)$, and $w = p(t_{i+1})$. The solution should have the form $w = e^P v$ (for simplicity, we omit here the constant $\tau_i = t_{i+1} - t_i$).

Following the observation that a truncated series of order $m-1$ (or, more generally, that approximating $e^P v$ by a polynomial of degree $m-1$, notice that this polynomial is a linear combination of the vectors $v, Pv, ..., P^{m-1}v$), will yield an element of the Krylov subspace:

$$K_m(P, v) \equiv Span\{v, Pv, ..., P^{m-1}v\}$$

The method is reduced to find such as element $K_m(P, v)$ of this space, that best approximates $w = e^P v$. The set of base vectors of these subspace is denoted by $V_m = [v_1, v_2, ..., v_m]$, $v_1 = v/\beta$ where $\beta = \| v \|_2$, $v = \beta V_m e_1$ and:

$$w \approx V_m \left[(V_m^T V_m)^{-1} V_m^T e^P V_m \right] \beta e_1 \tag{1}$$

In vector e_i the i-th element is equal 1 and the others are null. The set of base vectors V_m is obtained via Arnoldi's procedure [23]:

1. $v_1 = v / \| v \|_2$
2. For j=1,2,...,m do
   $z = P v_j$
   For $i = 1, 2, ..., j$ do
      $h_{ij} = v_i^T z$
      $z = z - h_{ij} v_i$
   $h_{j+1,j} = \| z \|_2$

$$v_{j+1} = z/h_{j+1,j}$$

The above algorithm is the modified Gram-Schmidt orthogonalization procedure, the obtained vectors v_i are orthonormal, and the upper Hessenberg matrix H_m (its dimension is $m \times m$) which is composed of coefficients h_{ij} holds the equation:

$$PV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T \tag{2}$$

The set of vectors V_m is orthonormal, hence we can simplify the eq. (1):

$$w \approx \beta V_m (V_m^T e^P V_m) e_1$$

If we approximate $V_m^T e^P V_m$ by $e^{V_m^T P V_m}$ the sought vector w will become:

$$w \approx \beta V_m e^{V_m^T P V_m} e_1$$

Also, because the set of vectors is orthonormal V_m we may rewrite (1) as:

$$H_m = V_m^T P V_m \tag{3}$$

and the solution may be expressed as:

$$w \approx \beta V_m e^{H_m} e_1$$

This solution still needs the calculation of matrix exponential but the size of the matrix is considerably smaller (m - the dimension of Krylov subspace is significantly smaller than n - the number of states of the considered system). Hence we can use any method advised for small systems. e.g. Padé approximation.

In the above description the constant τ was omitted. It may be easily put to the obtained solution because $V_m^T (P\tau) V_m = H_m$, and Krylov subspaces related to P and $P\tau$ are indentical.

Hence, the use of the Krylov subspaces for transient states consists in:

- the use of Arnoldi procedure to obtain the orthonormal set of base vectors V_m and Hessenberg matrix H_m.
- the use of Padé approximation to obtain $e^{H_m \tau}$.
- calculation of the state probability vector approximated by $\beta V_m e^{H_m \tau} e_1$.

To validate the Markovian results, we used a simulation packet OMNET++, written in C++ by A. Varga [http://www.omnetpp.org/]. Below we present some numerical results.

5 Numerical Results

Our goal is to capture the influence of the way a packet is chosen to be deleted (end of the tail, head of the tail) on the RED and DSRED queueing times. Input traffic intensity (for geometric and self-similar traffic) was chosen as $\alpha = 0.5$,

and due to the modulator characteristics, the Hurst parameter of self-similar traffic was fixed to $H = 0.78$.

The RED parameters had the following values: buffer size 250 packets, threshold values $Min_{th} = 100$ and $Max_{th} = 200$, $p_{max} = 0.1$, $w = 0.002$ or $w = 0.07$. Parameter μ of geometric distribution of service times (probability of the end of service within a current time-slot) was $\mu = 0.25$ or $\mu = 0.5$. Due to the changes of μ, two different traffic loads (low and high) were considered.

In case of DSRED policy, the traffic pattern and the buffer size are the same, parameters $K_l = Min_{th} = 100$ and $K_h = Max_{th} = 200$, intermediate threshold $K_m = 150$. The shaping parameter γ had three values $\gamma = 0.15, 0.5, 0.85$.

Fig. 1 displays a comparison of analytical and simulation results. They are almost identical if probabilities are greater then 10^{-10}, for smaller values the simulation results are not significant (the simulation run involved 250 millions of packets) while Markov model is able to give probabilities of very rare events.

If the mean queue length is relatively low, the influence of dropping scheme on queueing time is negligible: the introduction of drop-from-front strategy gives 0.7% shorter mean queueing time in case of RED and 0.8% shorter mean queueing time in case of DSRED, see Fig. 2.

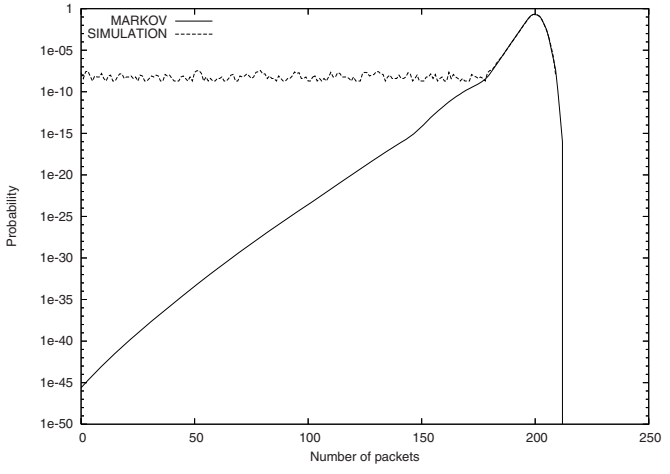

Fig. 1. Queue distribution for RED queue: geometric source, $\alpha = 0.5$, $\mu = 0.25$, $w = 0.07$, analytic and simulation results

Naturally, the introduction of DSRED gives shorter mean queue length and shorter mean queueing time compared to RED. However, when the Poisson traffic is replaced by self-similar one with the same intensity and preserving the same parameters of RED, the length of the queue grows and the influence of the dropping scheme is more visible: drop-from-front strategy reduces mean queueing time by 16.4%. A comparison of response time distributions for RED queue, for both strategies is presented in Fig. 3 (left). The same comparison in case of

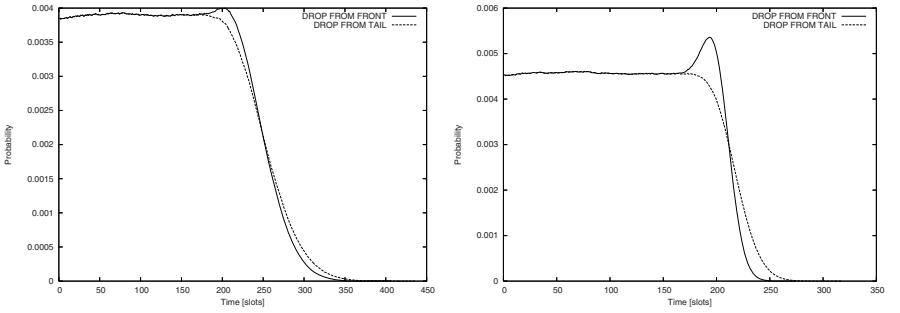

Fig. 2. Waiting times for RED (left) DSRED (right) queues: drop-from-front and drop-from-tail strategies, geometric source, $\alpha = 0.5$, $\mu = 0.5$, $w = 0.07$, $\gamma = 0.5$.

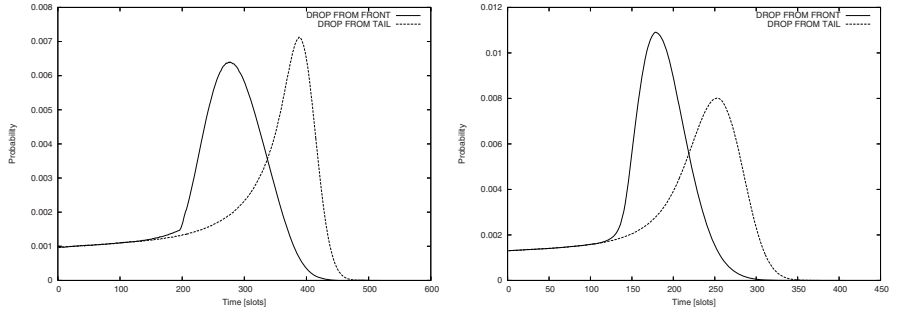

Fig. 3. Waiting times for RED (left) and DSRED (right) queues: drop-from-front and drop-from-tail strategies, self-similar source, $\alpha = 0.5$, $\mu = 0.5$, $w = 0.07$, $\gamma = 0.5$.

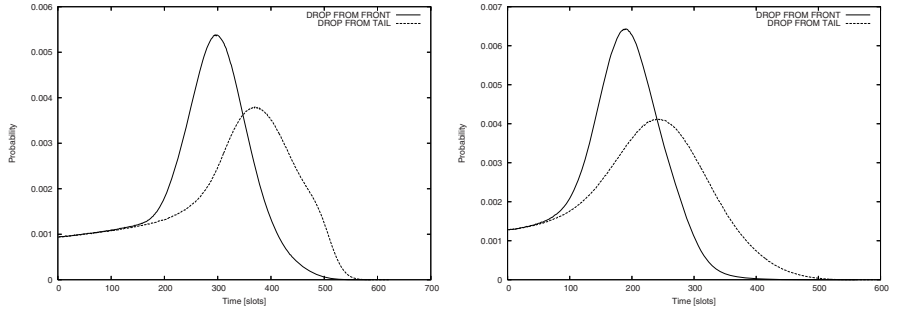

Fig. 4. Waiting times for RED (left) and DSRED (right) queues: drop-from-front and drop-from-tail strategies, self-similar source, $\alpha = 0.5$, $\mu = 0.5$, $w = 0.002$, $\gamma = 0.5$.

DSRED queue is presented in Fig. 3 (right). In this case the response time with drop-from-front strategy is 18.1% shorter then for tail-drop mechanism.

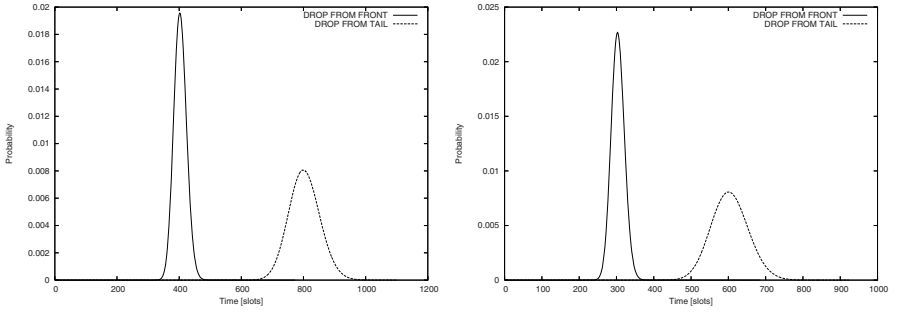

Fig. 5. Waiting times for RED (left) and DSRED (right) queues: drop-from-front and drop-from-tail strategies, geometric source, $\alpha = 0.5$, $\mu = 0.25$, $w = 0.07$, $\gamma = 0.5$.

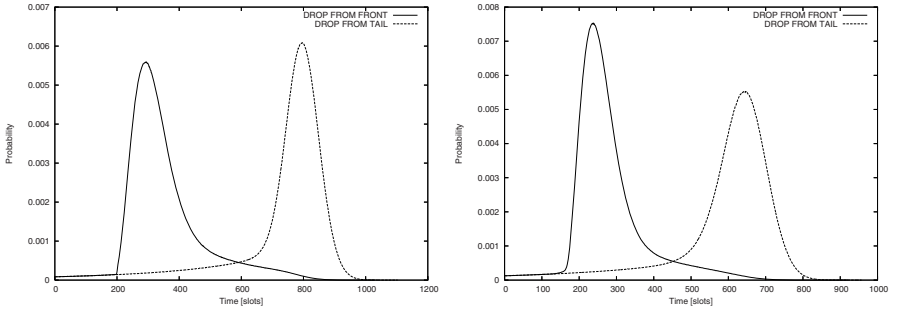

Fig. 6. Waiting times for RED (left) and DSRED (right) queues: drop-from-front and drop-from-tail strategies, self-similar source, $\alpha = 0.5$, $\mu = 0.25$, $w = 0.07$, $\gamma = 0.5$.

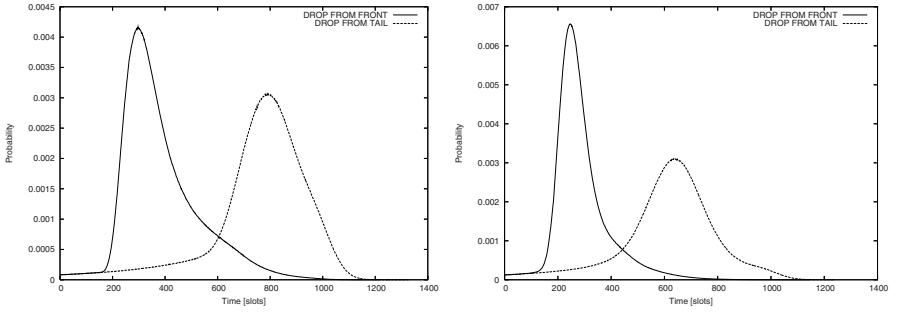

Fig. 7. Waiting times for RED (left) and DSRED (right) queues: drop-from-front and drop-from-tail strategies, self-similar source, $\alpha = 0.5$, $\mu = 0.25$, $w = 0.002$, $\gamma = 0.5$.

Table 1. Comparison of RED and DSRED (geometric and self-similar traffic)

		Mean queue length	Variation of queue length	Loss probability	Mean waiting time	Variance of waiting time
RED $\mu = 0.5$ $w = 0.002$	GEO	64.92	1562.07	0.00389652	132.34	6340.05
	SELF-S	130.61	6484.46	0.150939	308.49	16866.7
DSRED $\gamma = 0.5$ $\mu = 0.5$ $w = 0.002$	GEO	54.65	1077.13	0.00455001	111.8	4385.54
	SELF-S	89.53	3703.73	0.168627	218.26	10063.2
RED $\mu = 0.25$ $w = 0.002$	GEO	199.84	82.33	0.500013	803.35	3731.58
	SELF-S	169.86	5056.31	0.551741	760.58	34059.4
DSRED $\gamma = 0.5$ $\mu = 0.25$ $w = 0.002$	GEO	150.01	131.036	0.500066	604.09	3910.51
	SELF-S	136.21	3962.95	0.55552	617.071	31355
RED $\mu = 0.5$ $w = 0.07$	GEO	64.43	1504.8	0.00390818	131.375	6109.14
	SELF-S	123.79	5570.37	0.151645	293.89	13634.1
DSRED $\gamma = 0.15$ $\mu = 0.5$ $w = 0.07$	GEO	53.07	977.58	0.004675	108.63	3985.63
	SELF-S	76.1	2202.89	0.175138	186.39	4924.46
DSRED $\gamma = 0.85$ $\mu = 0.5$ $w = 0.07$	GEO	54.87	1018.47	0.00457423	110.67	4150.01
	SELF-S	83.01	2628.08	0.170974	202.25	6025.25
DSRED $\gamma = 0.85$ $\mu = 0.5$ $w = 0.07$	GEO	57.91	1182.4	0.00426957	118.314	4811.09
	SELF-S	100.03	3731.86	0.162186	240.73	8821.86
RED $\mu = 0.25$ $w = 0.07$	GEO	199.75	3.47	0.500006	803	2467.85
	SELF-S	163.2	4497.22	0.55187	735.69	25134.7
DSRED $\gamma = 0.15$ $\mu = 0.25$ $w = 0.07$	GEO	129.41	24.57	0.499999	521.63	1958.88
	SELF-S	109.52	2313.9	0.559453	501.39	14030.4
DSRED $\gamma = 0.85$ $\mu = 0.25$ $w = 0.07$	GEO	150	40.05	0.499999	603.98	2554.41
	SELF-S	130.26	3100.71	0.555498	590.9	18972.3
DSRED $\gamma = 0.85$ $\mu = 0.25$ $w = 0.07$	GEO	170.59	24.58	0.499999	686.33	2453.79
	SELF-S	144.43	3642.48	0.554022	652.57	21408.3

The change of w_q value (from 0.07 to 0.002) in computation of moving average results in longer response time and mean queue – see the Table – but the introduction of drop-from-front in place of tail-drop gives about 1% of changes. A comparison of queueing time distributions in these cases is given in Fig. 4 (left - RED) and (right - DSRED).

In case of heavy traffic, for both mechanisms RED/DSRED, irrespective of the w_q value and of the traffic self-similarity, drop-from-front strategy gives two times shorter mean queueing times. Queueing time distributions for all considered cases are presented in Figs. 5, 6, 7.

6 Conclusions

Drop-from-front strategy, when applied in place of tail-drop one, results in reduction of mean queueing time in RED/DSRED mechanisms of active queue management. In case of light load, the difference is more visible for self-similar traffic. In case of heavy load, the difference is also substantial for short-dependent traffic. Hence the application of drop-from-front strategy in AQM mechanisms may be recommended for connections with real-time requirements, even if the quantitative results depend on the distribution of the packet size and thus may differ slightly from presented here.

Acknowledgements

This research was financed by Polish Ministry of Science and Higher Education project no. N517 025 31/2997 and supported by European Network of Excellence EuroFGI (Future Generation Internet).

References

1. Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., Zhang, L.: Recommendations on queue management and congestion avoidance in the internet, RFC 2309, IETF (1998)
2. Floyd, S.: Red with drop from front (1998), `ftp://ftp.ee.lbl.gov/email/sf.98mar11.txt`
3. Floyd, S., Jacobson, V.: Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking 1(4), 397–413 (1993)
4. Floyd, S.: Discussions of setting parameters (1997), `http://www.icir.org/floyd/REDparameters.txt`
5. Zheng, B., Atiquzzaman, M.: A framework to determine the optimal weight parameter of red in next generation internet routers. Technical report, The University of Dayton, Department of Electrical and Computer Engineering (2000)
6. May, M., Bonald, T., Bolot, J.: Analytic evaluation of red performance. IEEE Infocom 2000, Tel-Aviv, Izrael (2000)
7. Feng, W.C., Kandlur, D.D., Saha, D.: Adaptive packet marking for maintaining end to end throughput in a differentiated service internet. IEEE/ACM Transactions on Networking 7(5), 685–697 (1999)
8. May, M., Diot, C., Lyles, B., Bolot, J.: Influence of active queue management parameters on aggregate traffic performance. Technical report, Research Report, Institut de Recherche en Informatique et en Automatique (2000)
9. Zheng, B., Atiquzzaman, M.: Low pass filter/over drop avoidance (lpf/oda): An algorithm to improve the response time of red gateways. Int. Journal of Communication Systems 15(10), 899–906 (2002)
10. Hassan, M., Jain, R.: High Performance TCP/IP Networking. Pearson Education Inc., London (2004)
11. Zheng, B., Atiquzzaman, M.: Dsred: A new queue management scheme for next generation networks. In: The 25th Annual IEEE Conference on Local Computer Networks, pp. 242–251. IEEE Computer Society Press, Los Alamitos (2000)

12. Zheng, B., Atiquzzaman, M.: Improving performance of active queue management over heterogeneous networks. In: ICC 2001: International Conference on Communications, pp. 2375–2379 (2001)
13. Liu, C., Jain, R.: Improving explicit congestion notification with the mark-front strategy. Computer Networks (Amsterdam, Netherlands: 1999) 35(2–3), 185–201 (2001)
14. Stallings, W.: High Speed Networks, TCP/IP and ATM Design Principles. Prentice Hall, Upper Saddle River, N.J (1998)
15. Willinger, W., Leland, W.E., Taqqu, M.S.: On the self-similar nature of ethernet traffic. IEEE/ACM Transactions on Networking (1994)
16. Abry, P., Flandrin, P., Taqqu, M., Veitch, D.: Wavelets for the analysis, estimation and synthesis of scaling data. In: Park, K., Willinger, i W. (eds.) Self-similar Network Traffic Analysis and Performance Evaluation (1999)
17. Paxson, V., Floyd, S.: Wide area traffic: the failure of poisson modeling. IEEE/ACM Transactions on Networking 3 (1995)
18. Feldman, A., Gilbert, A., Huang, P., Willinger, W.: Dynamics of ip traffic: Study of the role of variability and the impact of control. In: ACM SIGCOMM'99, Cambridge (1999)
19. Robert, S.: Modélisation Markovienne du Trafic dans les Réseaux de Communication. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Nr 1479 (1996)
20. Robert, S., Boudec, J.-Y.L.: New models for pseudo self-similar traffic. Performance Evaluation 30(1-2), 57–68 (1997)
21. Czachórski, T., Domańska, J.: Markovian models for long-range dependent traffic. Archiwum Informatyki Teoretycznej I Stosowanej 13(3), 297–308 (2001)
22. Domańska, J.: Procesy Markowa w modelowaniu natężenia ruchu w sieciach komputerowych.PhD thesis, IITiS PAN, Gliwice (2005)
23. Stewart, W.: An Introduction to the Numerical Solution of Markov Chains. Princeton Academic Press, London (1994)