INTERNATIONAL JOURNAL OF COMMUNICATIONS
Issue 1, Volume 7, 2013

Waiting Time Distribution for an Active Queue Management Algorithm

Andrzej Chydzinski

Abstract— The active queue management algorithms are designed for the Internet routers and their goal is to drop incoming packets before the actual buffer overflow occurs – in this way they notify the TCP senders about the necessity to reduce their sending rates and prevent the link from congestion. Most of the known active queue management algorithms drop incoming packets randomly, with probability that depends on the queue size observed upon packet arrival. In this paper we present calculations of the distribution of the waiting time for such active queue management algorithms, i.e. algorithms whose dropping probability is a function of the queue size. In particular, we give a formula for the transform of the waiting time distribution as well as a formula for the average waiting time. The mathematical results are accompanied with examples of numerical calculations.

Keywords— active queue management, single-server queue, packet dropping, waiting time distribution, queueing delay

I. INTRODUCTION

In the architecture of the current Internet there exists an incoherence between the congestion control mechanism in the transport layer and the packet buffering mechanism in the network layer, [2]. The buffers in the routers are designed basically for storing random bursts of traffic caused by the statistical multiplexing. On the other hand, the congestion control mechanism in TCP probes the maximum available bitrate by increasing the sending rate until a router buffer overflows and a packet (or packets) is lost. In other words, TCP tries to fill the buffers all the time. This causes several negative effects, like too long queues, too high queueing delays, synchronization of TCP sources, just to mention a few. The solution to this problem is a cross-layer optimization idea, namely the active queue management (AQM), which is based on the assumption that the router can drop packets preventively, before the buffer overflow occurs. In this way the router notifies the TCP sender about the forthcoming congestion and he can reduce his rate before the queue gets long.

The research activity on AQM has been initiated by the famous RED algorithm, [3], followed by several its modified and improved versions, e. g. [4], [5]. Some other well-known AQM algorithms include BLUE, [6], REM, [7], PI, [8] and AVQ, [9]. In order to avoid the synchronization of TCP sources, most of the AQMs assume that the arriving packets are dropped randomly, with the probability that is a function of some statistics collected by the router. In the most popular AQMs, the dropping probability is a function of the queue size – this is also the case studied herein.

Most research done so far on active queue management is based on simulation experiments (e.g. [10]-[29]) and usage of discrete-event network simulators, like ns2, Opnet Modeler or Omnet++ (see [30], [31], [32], respectively). In [33], a wide set of tests of active queue management algorithms, designed for simulation purposes, is presented.

Although the active queue management has been around for a while, there are not many papers devoted to its analysis via queueing theory methods. In [34], the RED queue model with batch Poisson arrivals and exponential service time distribution is considered. In [35], a queue-size based packet dropping mechanism is analyzed by exploiting an extension of the $GI^X/M/1$ queue and a technique based on thinning of the arrival process. In [36], an analysis of the single-server queue with Poisson arrivals, general service time distribution and a queue-size based packet dropping has been carried out, resulting in formulas for the queue size distribution, loss ratio and throughput. Using these results, the stability of the queue size in an AQM router has been studied in [37]. Finally, in [38], the usage of dropping functions to control the performance parameters of a single-server queue has been demonstrated.

In this papers we deal again with the AQM model with Poisson arrivals, general service time distribution and the dropping probability depending on the queue size observed upon packet arrival. This time, the analysis of the waiting time (queueing delay), which is one of the most important performance characteristics, is carried out. In particular, the joint distribution of the queue size and the remaining service time upon packet arrival in the steady state is computed first. Basing on this result, a formula for the Laplace transform of the steady-state waiting time is shown.

The remaining part of the paper is structured in the following way. In section 2, a formal description of the queueing model is given. In section 3, the actual analysis of the waiting time, preceded by some preliminary studies, is carried out. Section 4 presents numerical examples. Finally, in section 5 remarks concluding the paper are presented.

II. THE QUEUEING MODEL

The model considered herein is the classic single-server, finite-buffer queueing system of $M/G/1/b$ type, extended by a packet dropping functionality based on so called dropping function.

Namely, the packets arrive to the queue according to the Poisson process with average rate λ, the service time is

This is extended version of the paper [1] presented during DNCOCO'12 conference, Sliema, Malta, September 7-9, 2012.

The author is with the Silesian University of Technology, Institute of Informatics. Address: Akademicka 16, 44-100 Gliwice, Poland, e-mail: Andrzej.Chydzinski@polsl.pl

distributed according to a distribution function $F(x)$, which is not further specified. There is a finite buffer of size b for storing the packets. In other words, the number of packets in the system, including the one being serviced, cannot exceed b. Every packet arriving when there are already b packets present in the system is dropped. An arriving packet can be dropped also even if the system is not full. This happens with probability $d(n)$, where n is the queue length observed upon the arrival of this packet. The function $d(n)$ is called a dropping function. The dropping function must be must fulfill the following two assumptions:

$$d(n) \in [0,1], \quad \text{for} \quad 0 \le n \le b-1,$$

$$d(n) = 1, \quad \text{for} \quad n \ge b.$$

Besides that, it can have any form. The following notation will be used:

$\mathbf{P}$ – the probability,
$X(t)$ – the queue size at time t (including service position),
X_k – the queue size left by the k-th departing packet (again, including service position).

The load offered to the system will be denoted by ρ, i.e.:

$$\rho = \lambda m,$$

where m denotes the mean service time.

III. Waiting time distribution

Before the actual waiting time can be computed, we have to repeat some results on the queue size distribution, obtained in [36], which are necessary for computation of the waiting time.

In particular, in [36] it is shown that the steady-state distribution of the queue length is equal to:

$$P_k = \frac{\pi_k/(1-d(k))}{\pi_0/(1-d(0))+\rho}, \qquad 0 \le k \le b-1, \tag{1}$$

$$P_b = 1 - \frac{\sum_{i=0}^{b-1} \pi_i/(1-d(i))}{\pi_0/(1-d(0))+\rho}, \tag{2}$$

where

$$P_k = \lim_{t\to\infty} \mathbf{P}(X(t) = k), \qquad 0 \le k \le b, \tag{3}$$

and π_k is the steady-state distribution of the queue length left behind by the n-th departing packet, i.e.:

$$\pi_k = \lim_{n\to\infty} \mathbf{P}(X_n = k), \qquad 0 \le k \le b-1. \tag{4}$$

Thus, to calculate P_k, the distribution π_k for $0 \le k \le b-1$ have to be calculated first. It can be obtained by solving the following system of linear equations:

$$\begin{cases} \pi_k = \sum_{j=0}^{b-1} \pi_j p_{j,k}, & 0 \le k \le b-1, \\ \sum_{j=0}^{b-1} \pi_j = 1. \end{cases} \tag{5}$$

with

$$p_{j,k} =$$
$$= \begin{cases} a_{1,k}, & \text{if} \quad j=0,\ 0 \le k \le b–1, \\ a_{j,k-j+1} & \text{if} \quad 1 \le j \le b–1,\ j-1 \le k \le b–1, \\ 0 & \text{otherwise}, \end{cases} \tag{6}$$

$$a_{n,k} = \int_0^\infty Q_{n,k}(u) dF(u). \tag{7}$$

$Q_{n,k}(u)$, which appears in (7), is the conditional probability that the first departure time is after u and k packets are allowed into the system in interval $(0,u]$, given that $X(0) = n$. $Q_{n,k}(u)$ can be computed effectively by inverting its Laplace transform:

$$q_{n,k}(s) = \int_0^\infty e^{-su} Q_{n,k}(u) du, \tag{8}$$

which has the following form (see [36]):

$$q_{n,k}(s) = \frac{\prod_{i=0}^{k-1} \lambda(1-d(n+i))}{\prod_{i=0}^{k}(s+\lambda(1-d(n+i)))}, \qquad n \ge 0, \quad k \ge 0. \tag{9}$$

With help of the queue length distribution we can calculate the loss ratio, LR, which is defined as the long-run fraction of dropped packets. Namely, it is equal to

$$LR = 1 - \frac{1-P_0}{\rho}. \tag{10}$$

For the waiting time analysis we will use a method similar to presented in [39], p. 214 (used there for analysis of classic, finite-buffer models). In the preliminary steps, this method exploits the remaining and elapsed service times. Namely, we denote:

X_+ – the remaining service time upon packet arrival (in the steady-state),

X_- – the elapsed service time upon packet arrival (in the steady-state),

$A(X_-)$ – the number of packets accepted during the elapsed service time X_-,

$\psi_{n,k}(s)$ – the joint distribution for the length of the remaining service time X_+ and the number of packets accepted during the elapsed service time X_-, given that at the beginning of the last service period the queue size was n. Namely, we have:

$$\psi_{n,k}(s) = \mathbf{E}_n(e^{-sX_+}|A(X_-) = k)\mathbf{P}(A(X_-) = k),$$
$$k = 0, 1, 2, \ldots. \qquad (11)$$

We have

$$\psi_{n,k}(s) = \frac{1}{m}\int_0^\infty Q_{n,k}(x)dx \int_0^\infty e^{-sy}dF_x(y),$$
$$k = 0, 1, 2, \ldots, \qquad (12)$$

with

$$F_x(y) = F(x + y).$$

Let X be the queue size in the steady state and let $\Pi_k(s)$ be the joint distribution of the queue size and the remaining service time upon packet arrival in the steady state:

$$\Pi_k(s) = \int_0^\infty e^{-sy}\mathbf{P}(X = k, X_+ \in dy), \quad 1 \le k \le b. \qquad (13)$$

For $1 \le k \le b$ we have:

$$\begin{aligned}\Pi_k(s) &= \\ &= \rho'\pi_0\mathbf{E}_1(e^{-sX_+}|A(X_-) = k-1)\mathbf{P}(A(X_-) = k-1) \\ &+\rho'\sum_{j=1}^{k}\pi_j\mathbf{E}_j(e^{-sX_+}|A(X_-)=k-j)\mathbf{P}(A(X_-)=k-j). \qquad (14)\end{aligned}$$

where

$$\rho' = \rho(1 - LR) \qquad (15)$$

is the carried load of the system (or, equivalently, the probability that the server is busy at an arbitrary time).

Therefore, we have:

$$\Pi_k(s) = \rho'\pi_0\psi_{1,k-1}(s) + \rho'\sum_{j=1}^{k}\pi_j\psi_{j,k-j}(s). \qquad (16)$$

Now, we introduce the following notation:

$W(t)$ – distribution function of the real waiting time (queueing delay), i. e. the time spent by an accepted packet in the queue, before processing, in the steady state (rejected packets are not taken into account).

Denoting

$$w(s) = \int_0^\infty e^{-st}dW(t), \qquad (17)$$

and

$$f(s) = \int_0^\infty e^{-st}dF(t), \qquad (18)$$

we can obtain the Laplace transform of the waiting distribution:

$$\begin{aligned}w(s) &= \\ &= \frac{1}{1-LR}\left(P_0 + \sum_{k=1}^{b-1}(1-d(k))\Pi_k(s)(f(s))^{k-1}\right), \qquad (19)\end{aligned}$$

Now we can use (19) either for obtaining the shape of the waiting time distribution (a method for inverting the Laplace transform is needed then – see, for instance, [40]), or for obtaining directly the average waiting time by means of the formula:

$$\mathbf{E}W = -w'(0). \qquad (20)$$

If we have given a particular form of the service time distribution, we may try to simplify the expression for $\psi_{n,k}(s)$. For instance, assuming that the service time is constant and equal to c, we get:

$$f(s) = e^{-sc},$$

$$\psi_{n,k}(s) = \frac{e^{-sc}}{c} \int_0^c Q_{n,k}(x) e^{sx} dx.$$

IV. Numerical examples

Naturally, the queueing delay observed at the router is negative effect of statistical multiplexing, therefore we want to be as small as possible. However, a short delay should n be obtained by increasing the loss ratio - this would decreas the router throughput. Therefore, when testing the impact (different dropping functions on the queueing delay we hav to keep the same loss ratio in every case.

In all examples we assume $b = 10$, $\lambda = 1$ and a consta service time equal to 1 (which gives also the offered loa $\rho = 1$).

A. Example 1

In this example we consider four shapes of the dropping function, namely:

- RED dropping function (see Fig. 1):

$$d(n) = \begin{cases} 0 & \text{if} \quad n \leq 3, \\ 0.0820928n – 0.164185 & \text{if} \;\; 3 < n < 10, \\ 1 & \text{if} \quad n \geq 10. \end{cases}$$ (2

- REM dropping function (see Fig. 2):

$$d(n) = \begin{cases} 0 & \text{if} \quad n \leq 4, \\ 1 - e^{1.309757(4-n)} & \text{if} \quad 4 < n < 10, \\ 1 & \text{if} \quad n \geq 10. \end{cases}$$ (2

- constant dropping function (see Fig. 3):

$$d(n) = \begin{cases} 0.08032 & \text{if} \quad n < 10, \\ 1 & \text{if} \quad n \geq 10. \end{cases} \tag{23}$$

- decreasing dropping function (see Fig. 4):

$$d(n) = \begin{cases} 0.44043 - 0.35n & \text{if} \quad n < 3, \\ 0 & \text{if} \quad 3 < n < 10, \\ 1 & \text{if} \quad n \geq 10. \end{cases} \tag{24}$$

Each of these dropping functions has been parameterized so that the resulting los ratio is exactly 10% (which means that the carried load is $\rho' = 90\%$).

Tab. I presents the results, namely the average queueing delay for four considered dropping functions. It is worth noticing that even though the system throughput is the same in every case, the queueing delay may differ significantly, depending on the shape of the dropping function – in our worst case the delay is more than two times larger than in the best case.

Secondly, as one can notice, more favourable are dropping functions with steep, increasing shapes. For flat or decreasing dropping functions we obtain larger delays.

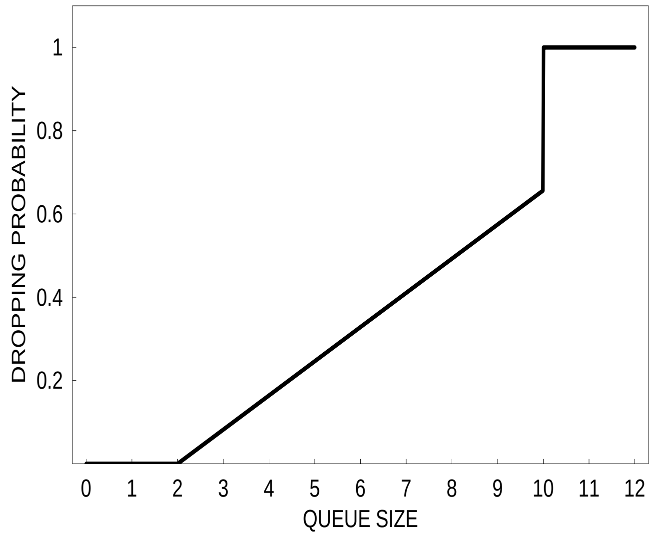

Fig. 1. The RED dropping function used in Example 1.

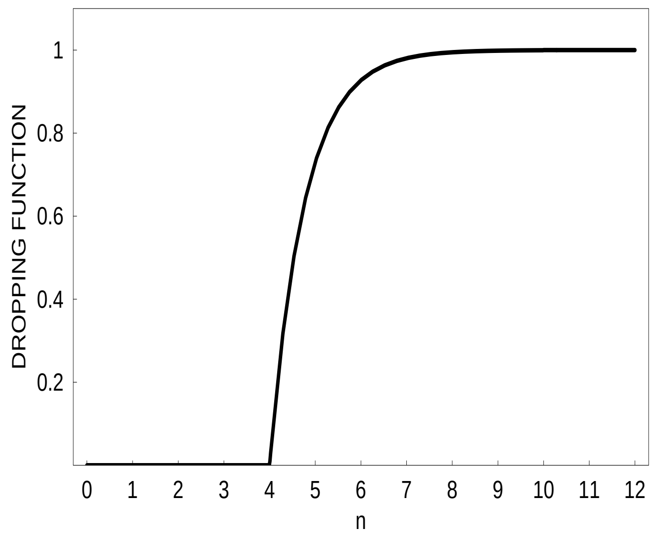

Fig. 2. The REM dropping function used in Example 1.

dropping function	loss ratio	queueing delay
RED, (21)	0.10000	2.1606
REM, (22)	0.10000	1.9349
constant, (23)	0.10000	3.1850
decreasing, (24)	0.10000	4.2908

TABLE I

The loss ratio and the mean queueing delay for four dropping functions.

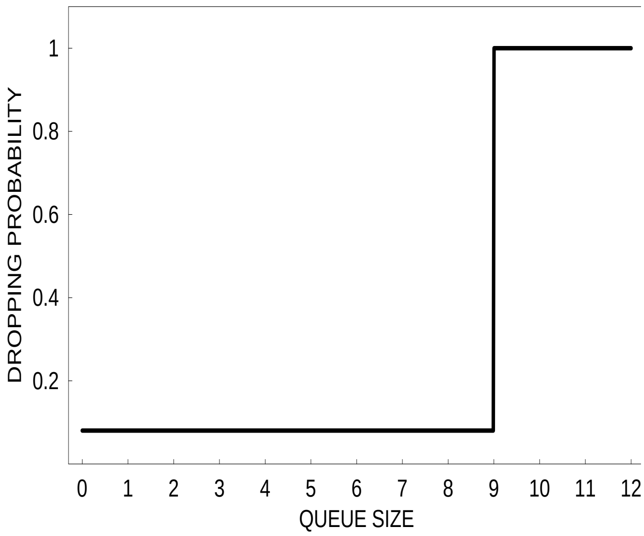

Fig. 3. The constant dropping function used in Example 1.

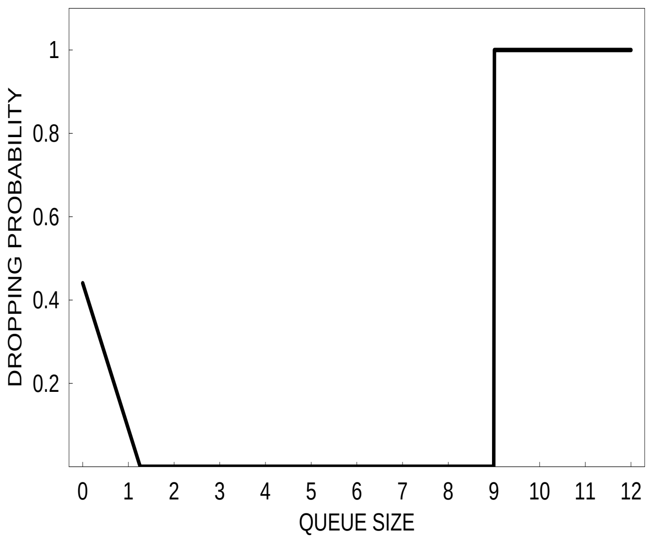

Fig. 4. The decreasing dropping function used in Example 1.

B. Example 2

In the second example we will consider the following class of linear dropping functions (see Fig. 5):

$$d(n) = \begin{cases} a \cdot n & \text{if} \quad n < 10, \\ 1 & \text{if} \quad n \geq 10, \end{cases} \tag{25}$$

where a is a parameter. We will check the queueing performance depending on a.

In Fig. 6 the dependence of the average waiting time on a is presented, in Fig. 7 the dependence of the loss ratio on a is presented while in Fig. 8 the dependence of the standard deviation of the waiting time on a is depicted.

It is not a surprise that the waiting decreases with a while the loss ratio increases with a. Now, interesting problems arise when we want to optimize the performance of the queueing

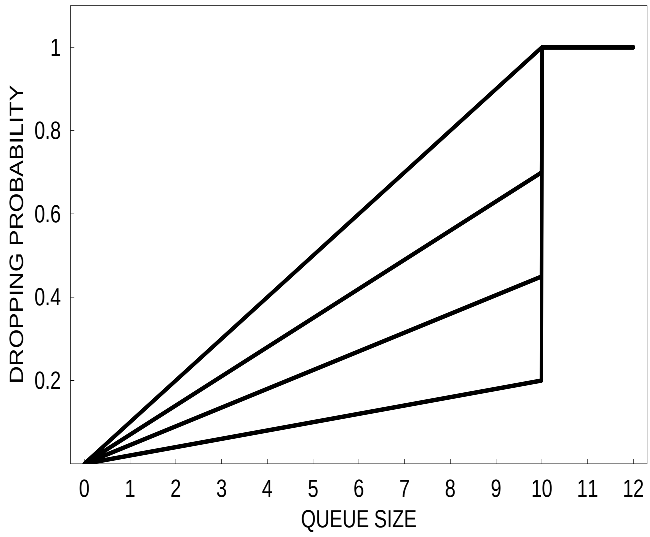

Fig. 5. Dropping functions used in Example 2.

system with respect to both of these characteristics. For instance, assume we want to minimize the following objective function:

$$f_0(a) = \mathbf{E}W \cdot LR,$$

under constrain:

$$LR \leq 15\%. \tag{26}$$

To solve this optimization problem, we first have to solve the equation:

$$LR(a) = 0.15. \tag{27}$$

Using (10) and (1) we obtain

$$a = 0.06972,$$

(see also Fig. 7 for comparison).

In Fig. 9 function $f_0(a)$ is depicted in interval $[0, 0.06972]$. As we can see, $f_0(a)$ is not monotonic and reaches a maximum around 0.025. Under constrain (26), the minimum is reached for $a = 0$. Naturally, for other constrains the optimal a may be different.

V. Conclusions

In this paper we dealt with a queueing model in which a packet can be dropped with probability that depends on the queue size observed upon arrival of this packet. The study was motivated by the active queue management algorithms for Internet routers, which often exploits such packet dropping mechanism. In particular, the Laplace transform of the distribution of the waiting time has been calculated and illustrated via numerical examples.

Acknowledgements: The material is based upon work supported by the Polish National Science Centre under Grant No. N N516 479240.

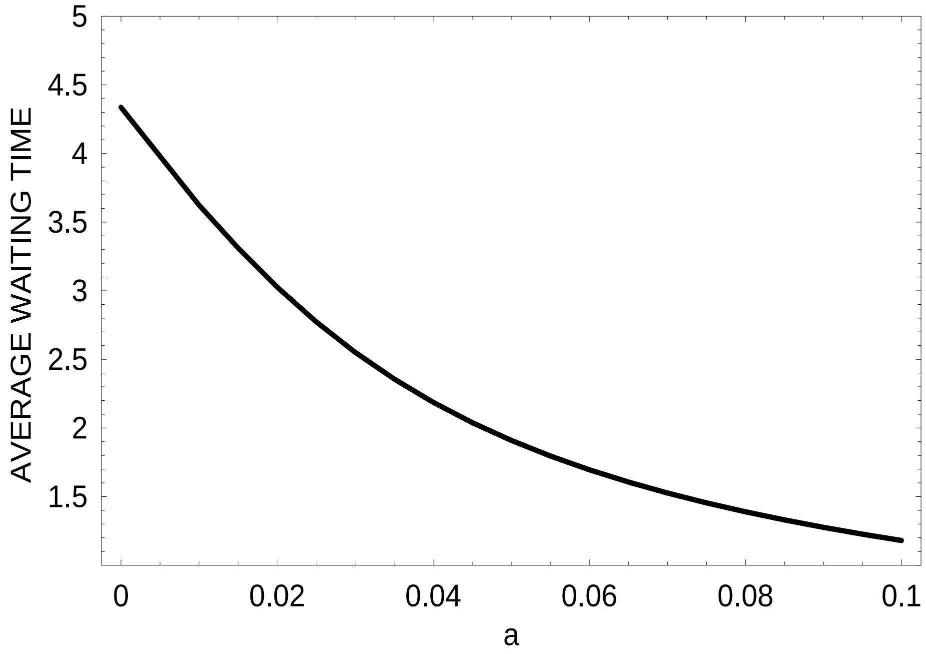

Fig. 6. The average waiting time as a function of a in Example 2.

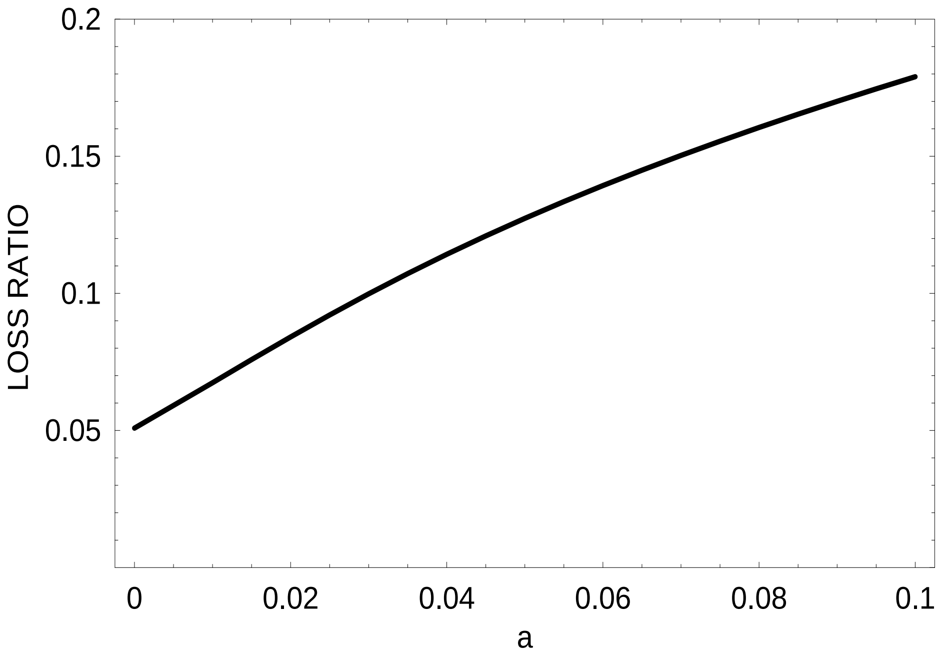

Fig. 7. The loss ratio as a function of a in Example 2.

REFERENCES

[1] Chydzinski, A. Delay Analysis for an AQM Queue with Dropping Function. Proc. of International Conference on Data Networks, Communications, Computers, Sliema, Malta, September 7-9, pp. 53-57, 2012.

[2] Bohacek, S.; Shah, K.; Arce, G.R.; Davis, M. Signal processing challenges in active queue management. IEEE Signal Processing Magazine, Vol. 21, Iss. 5, Pages: 69–79, Sept. 2004.

[3] Floyd, S.; Jacobson, V. Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking, Volume 1, Issue 4, Page(s): 397 – 413, 1993.

[4] Rosolen, V.; Bonaventure, O., and Leduc, G. A RED discard strategy for ATM networks and its performance evaluation with TCP/IP traffic. SIGCOMM Comput. Commun. Rev. 29, 3, Jul. 1999.

[5] Floyd, S.; Gummadi, R. and Shenker, S. Adaptive RED: An algorithm for increasing the robustness of RED, Tech. Rep.: ACIRI, 2001, http://icir.org/floyd/red.html

[6] Feng, W.; Shin, K. G.; Kandlur, D. D.; Saha, D. The BLUE active queue management algorithms. IEEE/ACM Transactions on Network-

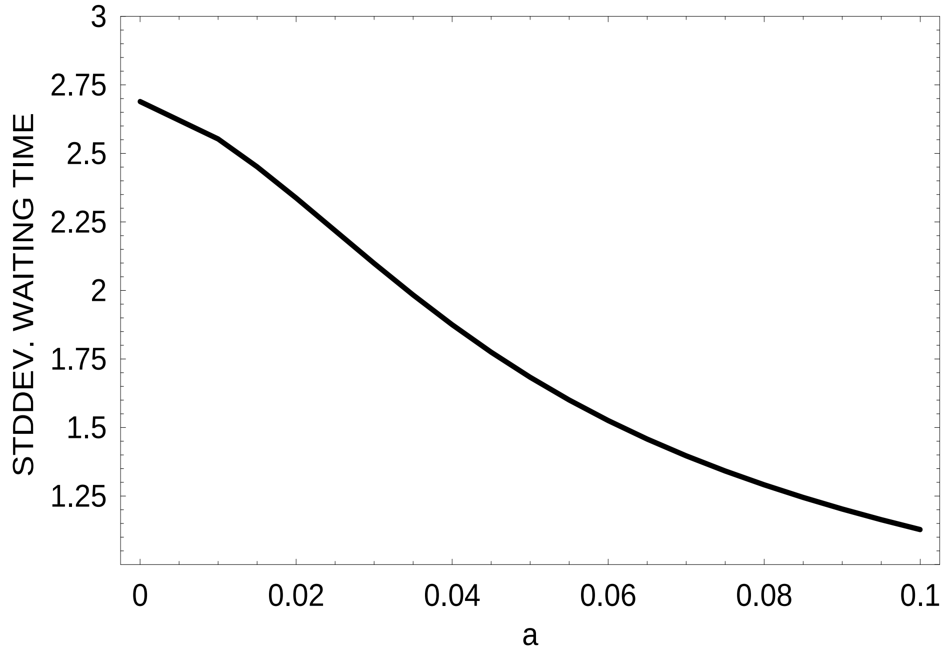

Fig. 8. The standard deviation of the waiting time as a function of a in Example 2.

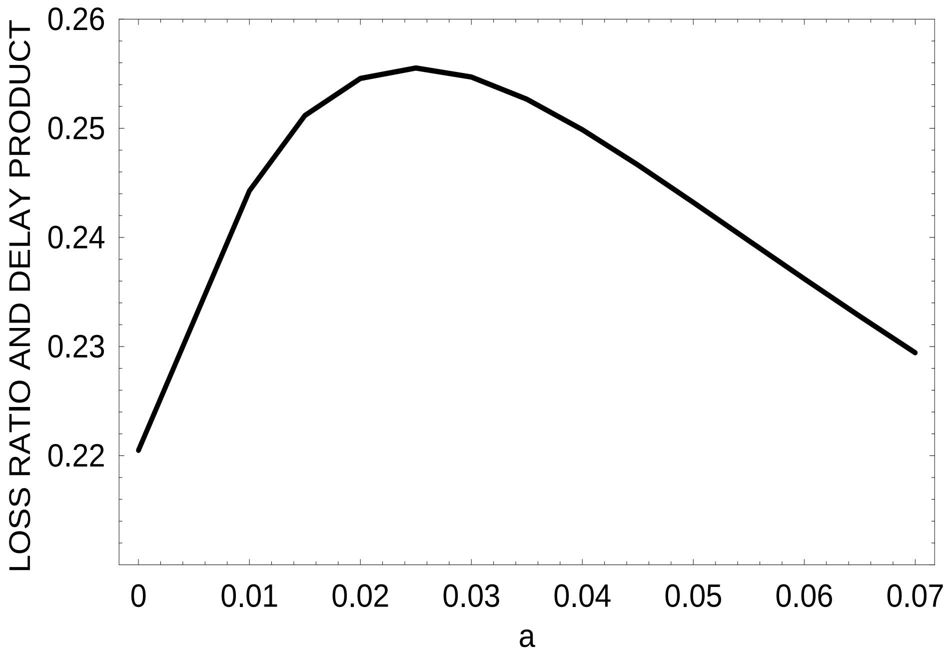

Fig. 9. The loss ratio and delay product as a function of a in Example 2.

ing. Page(s): 513 – 528, Volume: 10, Issue: 4, Aug 2002.

[7] Athuraliya, S.; Low, S. H.; Li, V. H.; Qinghe Yin. REM: active queue management, IEEE Network. On page(s): 48-53, Volume: 15, Issue: 3, May 2001.

[8] Hollot, C. V.; Misra, V.; Towsley, D.; Weibo Gong. Analysis and design of controllers for AQM routers supporting TCP flows, IEEE Transactions on Automatic Control. On page(s): 945–959, Volume: 47, Issue: 6, Jun 2002.

[9] Kunniyur, S. S.; Srikant, R. An adaptive virtual queue (AVQ) algorithm for active queue management. IEEE/ACM Transactions on Networking. Page(s): 286–299, Volume: 12, Issue: 2, April 2004.

[10] Mrozowski, P. and Chydzinski, A. Stability of AQM algorithms in low congestion scenarios, International Journal of Applied Mathematics and Informatics, Volume 3, Issue 1, pp. 9-16, Nov 2009.

[11] Mrozowski, P. and Chydzinski, A. Interactions between AQM routers in the Internet, International Journal of Systems Applications, Engineering & Development, Vol. 3, Issue 3, pp. 85-92, Nov 2009.

[12] Chrost, L., Brachman. A and Chydzinski, A. On the performance of AQM algorithms with small buffers, Communications in Computer and Information Science, Vol. 39, pp. 168-173, (2009)

[13] Chydzinski, A. and Brachman, A. Performance of AQM Routers in the Presence of New TCP Variants, Proc. of International Conference on Advances in Future Internet, pp. 88- 93, Venice, July (2010)

[14] Mrozowski, P. and Chydzinski, A. On the Stability of AQM Algorithms, Proc. of Applied Computing Conference (ACC'09), pp. 282-287, Athens, September 2009.

[15] Mrozowski, P. and Chydzinski, A. On the Deployment of AQM Algorithms in the Internet, Proc. of Applied Computing Conference (ACC'09), pp. 276-281, Athens, September 2009.

[16] Adamczyk, B. and Chydzinski, A. On the optimization of the inter-flow fairness in the Internet, Theoretical and Applied Informatics, in press, scheduled for Vol. 3/2012, 2012.

[17] Wu, W., Ren, Y., Shan, X. A self-configuring PI controller for active queue management. In Asia-Pacific Conference on Communications (APCC), Japan, 2001.

[18] Wydrowski, B. and Zukerman, M. GREEN: an active queue management algorithm for a self managed Internet, in: Proceeding of IEEE International Conference on Communications ICC2002, vol. 4, pp. 2368-2372, April, 2002.

[19] Heying, Z., Liu, B. and Wenhua, D. Design of a robust active queue management algorithm based on feedback compensation, in: Proceedings of ACM/SIGCOMM 2003, pp. 277-285. 2002.

[20] Aweya J. I, Ouellette M., Montuno D.Y. Multi-level active queue management with dynamic thresholds. Computer Communications. Volume 25, Number 8, pp. 756–771, 2002.

[21] Fatta, G., Hoffmann, F., Re, G. L., Urso, A. A genetic algorithm for the design of a fuzzy controller for active queue management, IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews. On page(s): 313-324, Volume: 33, Issue: 3, Aug. 2003.

[22] Chatranona, G., Labradorb, M.A. and Banerjee, S. A survey of TCP-friendly router-based AQM schemes. Computer Communications. v27. pp. 1424–1440, 2004.

[23] Ren, F., Lin, C. and Wei, B. A robust active queue management algorithm in large delay networks. Computer Communications 28(5): Pages: 485–493, 2005.

[24] Lakshmikantha, A., Beck, C. L., Srikant, R. Robustness of real and virtual queue-based active queue management schemes, IEEE/ACM Transactions on Networking. On page(s): 81– 93, Volume: 13, Issue: 1, Feb. 2005.

[25] Hong, Y., Yang, O. W. W. Adaptive AQM controllers for IP routers with a heuristic monitor on TCP flows: Research Articles, International Journal of Communication Systems, v.19 n.1, pp.17–38, 2006.

[26] Chang, X., Muppala, J. K. A stable queue-based adaptive controller for improving AQM performance, Computer Networks: The International Journal of Computer and Telecommunications Networking, v.50 n.13, pp. 2204–2224, 2006.

[27] Wang, C., Liu, J., Li, B. Sohraby, K, Hou, T. LRED: A Robust and Responsive AQM Algorithm Using Packet Loss Ratio Measurement. IEEE Transactions on Parallel and Distributed Systems, v.18 n.1, pp.29–43, 2007.

[28] Hariri, B., Sadati, N. NN-RED: An AQM mechanism based on neural networks. Electronics Letters. Volume: 43, Issue: 19, On page(s): 1053–1055, 2007.

[29] Sun, J. and Zukerman, M. An Adaptive Neuron AQM for a Stable Internet, Proc. Networking'07, LNCS 4479, 2007.

[30] http://www.isi.edu/nsnam/ns/

[31] http://www.opnet.com/solutions/network_rd/modeler.html

[32] http://www.omnetpp.org/

[33] Chrost, L. and Chydzinski, A. On the Evaluation of the Active Queue Management Mechanisms. Proc. of International Conference on Evolving Internet (INTERNET'09), pp. 113-118, Cannes, August 2009.

[34] Bonald, T., May, M. and Bolot, J.-C. Analytic evaluation of RED performance. Page(s): 1415–1424, Proc. INFOCOM, 2000.

[35] Hao, W. and Wei, Y. An Extended $GI^X/M/1/N$ Queueing Model for Evaluating the Performance of AQM Algorithms with Aggregate Traffic. Page(s): 395–414, Proc. ICCNMC, LNCS 3619, 2005.

[36] Chydzinski, A., Chrost, L. Analysis of AQM queues with queue-size based packet dropping, International Journal of Applied Mathematics and Computer Science, Vol. 21, No. 3, 567577, 2011.

[37] Chydzinski, A. Towards a Stable AQM via Dropping Function Shaping, Proc. of International Conference on Networks '10, pp. 93-97, Les Menuires, April 2010.

[38] Chydzinski, A. Optimization problems in the theory of queues with dropping functions, Proc. of HET-NETS'10 (Performance Modelling and Evaluation of Heterogeneous Networks), pp. 121-132, Zakopane, January 14th - 16th, 2010.

[39] Takagi, H. *Queueing analysis - Finite Systems*. North-Holland Amsterdam, 1993.

[40] Abate, J., Choudhury, G. L. and Whitt W. An introduction to numerical transform inversion and its application to probability models. Chapter in Computational Probability, pp. 257–323, W. Grassman (ed.), Kluwer, Boston, 2000.

Andrzej Chydzinski received his MS (in applied mathematics), PhD (in computer science) and DSc (in computer science) degrees from the Silesian University of Technology, Gliwice, Poland, in 1997, 2002 and 2008, respectively. He is currently a professor in the Institute of Informatics of this university. His academic and professional interests are with computer networking, in particular with performance evaluation of computer networks, Future Internet design, active queue management in Internet routers, mathematical modelling, queueing theory and discrete-event network simulators. Prof. Chydzinski authored and co-authored 3 books, about 40 conference papers and about 40 journal articles, including papers in Telecommunication Systems, Performance Evaluation, Pattern Recognition, Microprocessors and Microsystems, Queueing Systems, Stochastic Models, Mathematical Problems in Engineering, Applied Mathematical Modelling and other. He is also reviewing articles for several high-quality journals, including Annals of Operations Research, Applied Mathematical Modelling, Queueing Systems, Mathematical Problems in Engineering, International Journal of Applied Mathematics and Computer Science, Journal of Network and Computer Applications, Performance Evaluation and other. Prof. Chydzinski is a Technical Program Committee member for several conferences. He was (and is) a leader of several scientific projects founded by Polish state and European Union. Since January 2011 he has been an IEEE Senior Member.