Int. J. Appl. Math. Comput. Sci., *2014, Vol. 24, No. 3, 697–707*
DOI: 10.2478/amcs-2014-0051

A RED MODIFIED WEIGHTED MOVING AVERAGE FOR SOFT REAL–TIME APPLICATION

JOANNA DOMAŃSKA [*], ADAM DOMAŃSKI [**], DARIUSZ R. AUGUSTYN [**], JERZY KLAMKA [*]

[*] Institute of Theoretical and Applied Informatics
Polish Academy of Sciences, ul. Bałtycka 5, 44-100 Gliwice, Poland
e-mail: {joanna,jerzy.klamka}@iitis.pl

[**] Institute of Informatics
Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland
e-mail: {adamd,draugustyn}@polsl.pl

The popularity of TCP/IP has resulted in an increase in usage of best-effort networks for real-time communication. Much effort has been spent to ensure quality of service for soft real-time traffic over IP networks. The Internet Engineering Task Force has proposed some architecture components, such as Active Queue Management (AQM). The paper investigates the influence of the weighted moving average on packet waiting time reduction for an AQM mechanism: the RED algorithm. The proposed method for computing the average queue length is based on a difference equation (a recursive equation). Depending on a particular optimality criterion, proper parameters of the modified weighted moving average function can be chosen. This change will allow reducing the number of violations of timing constraints and better use of this mechanism for soft real-time transmissions. The optimization problem is solved through simulations performed in OMNeT++ and later verified experimentally on a Linux implementation.

Keywords: active queue management, soft real-time, dropping packets.

1. Introduction

Real-time data transfer is normally used in dedicated industrial networks. The protocols used in such a network are usually incompatible with those used in Local Area Networks (LANs). LANs and TCP/IP apply the best-effort principle in providing communication services. It is quite difficult to satisfy real-time requirements by using best-effort networks.

The popularity of TCP/IP caused a significant research trend of using best-effort networks for real-time communication. This trend was also caused by technical developments such as an increasing network bandwidth and the development of new Quality of Service (QoS) mechanisms. Much effort has been put to ensure QoS for soft real-time traffic over IP networks. The IETF proposed two QoS architectures: Integrated Services (IntServ) (Braden *et al.*, 1997) and Differentiated Services (DiffServ) (Blake *et al.*, 1998). They also proposed some architecture components, such as Active Queue Management (AQM) (Braden *et al.*, 1998).

AQM replaces the traditional drop-tail queue management. This mechanism improves network performance by making the average queue size smaller. In this paper we propose a modification of one of the AQM algorithms: Random Early Detection (RED) (Floyd and Jacobson, 1993). We modify the RED weighted moving average by using a high order difference equation. Our modification can be used to improve the QoS for soft real-time traffic.

Real-time networked applications depend on network parameters such as bandwidth, delay, jitter (interpacket delay variation) and loss for their correct operation. The degree of tolerance or sensitivity to each of these parameters varies widely from one application to another (El-Gendy *et al.*, 2003). However, real-time traffic is not so sensitive to packet losses, depending on the codecs and loss recovery schemes used at the receiver (Wang, 2006).

1.1. Our contribution.

This paper presents the RED algorithm with a modified weighted moving average. We have already performed a fluid flow analysis of this type of the RED algorithm (Domańska *et al*., 2013). In fluid flow models the details are reduced using continuous flows instead of packet flows. This kind of model is especially suitable for TCP traffic modeling (Shah *et al*., 2012). Our paper describes the influence of the weighted moving average on packet waiting time reduction analysed by simulations performed in OMNeT++. A packet-level network simulator mimics the behavior of every packet in a network. Hence, packet-level models give a more detailed analysis of packet behavior. This paper shows that our RED algorithm makes the average queue size smaller and hence improves the link utilization and the packet loss rate with the compared basic RED.

There are many results concerning the improvement of the RED efficiency (Haider, 2004; Padhy and Sundaram, 2010; Haider and Nakao, 2008). Some researchers make an attempt to use the RED algorithm for achieving soft real-time guarantees (Wang, 2006; Reid, 2008). To the best of our knowledge, there are no published studies of modification of the RED moving average. Our modification of the RED moving average can be used in various RED algorithms and hence seems to be useful for achieving soft real-time guarantees.

The remainder of the paper is organized as follows. Section 2 describes the works related to this article. This section helps to place our work with regard to other scientific studies conducted in the area. Section 3 shows the proposed model of estimating the average queue length in the RED algorithm. Section 4 presents the method of obtaining optimal model parameters. Section 5 discusses the obtained numerical results. Section 6 describes a Linux IP router with AQM and discusses the obtained results. Conclusions are given in Section 7.

2. Related work

Some researchers have studied the problem of adapting real-time traffic to TCP/IP networks and proposed a number of solutions. Wang (2009) studied the methods of providing real-time data transmission over the best-effort Internet. Wijnants and Lamotte (2008) presented an alternative method for maintaining the network bandwidth for many client applications. These proposals did not provide any end-to-end timing guarantees for real-time traffic. Another kind of solutions was presented also by Wijnants and Lamotte (2006). The authors suggest using prioritization mechanisms which meet the requirements of the IEEE 802.1D standard. They conclude that a big part of message latency appears at the end nodes. Lu *et al*. (2006) and Abeni *et al*. (2002) combined reservation and feedback techniques to provide real-time guarantees for tasks that coexist in the same environment. Cena *et al.* (2010; 2007) suggest using the IEEE 802.11e network for achieving real-time guarantees.

The works on providing QoS services for networks, in which the communication infrastructure is an IP network itself, have already been carried out. The Internet Engineering Task Force (IETF) groups worked on proposals for providing better QoS control in IP networks. Their proposals included RSVP, IntServ and DiffServ. IntServ (Braden *et al*., 1997; Wroclawski, 1997) ensures end-to-end QoS by means of hop-by-hop resource reservation inside of the IP network. It was based on mathematical guarantees of bandwidth, delay, jitter and supported specific applications such as video streaming. IntServ provides individualized QoS guarantees to separate applications, but it needs serious changes in the IP network. This architecture requires an implementation of RSVP inside each core routers. This requirement caused an increase in the router's complexity. To make the complexity within each router lower, Measurement Based Admission Control (MBAC) schemes were proposed (Georgoula *et al*., 2005). According to these, each core router must contain load estimation algorithms. Totally different solutions were provided by DiffServ (Blake *et al*., 1998). This approach categorizes traffic into different classes and applies QoS parameters to these classes. It needs relatively small changes at the network and transport layers. The fault of DiffServ is the lack of strict QoS guarantees.

Real-time systems should be classified according to their timing requirements. The distinction between hard, firm and soft real-time systems is described in the literature (Stankovic and Rajkumar, 2004). In a soft real-time system a critical real-time task receives a higher priority than other tasks. Similarly to hard real-time systems, in a soft real-time system delays need to be bounded, but timing constraints can rarely be infringed without causing a system failure. Many researchers have studied the problem with the aim to achieve soft, or statistical—not only hard—real-time QoS guarantees. Many existing examples of soft real-time applications in the industrial application domain are evidences of these studies (Cucinotta *et al*., 2010). Control applications using a soft real-time environment are good known examples as well (Palopoli *et al*., 2000; Cervin and Eker, 2005). Reinemo *et al*. (2003) studied the method of providing soft real-time guarantees by using three different admission control schemes, in combination with DiffServ based QoS. Wang (2006) presented a method of improving the QoS of soft real-time applications over IP network infrastructure. This method has low service cost and requires no changes in the existing network. Although soft real-time applications are sensitive to end-to-end delay and delay jitter, they can put up with an occasional loss.

In general, there are some researchers who regard making serious changes in existing best-effort services as important. They just suggest adding more bandwidth to provide suitable QoS over IP networks. On the other hand, other researchers claim that additional bandwidth could be very expensive. An alternative approach shows the possibility of providing new service models and mechanisms such as the mentioned IntServ and DiffServ models, proposed by the IETF. Kurose and Ross (2007) claim that IntServ and DiffServ models are not generally used in real-time transmission, due to economic as well as legacy reasons rather than technical ones. These QoS networking architectures contain various mechanisms such as AQM, resource reservation signaling, scheduling mechanisms, etc. One of the above-mentioned mechanisms, i.e., RED (Floyd and Jacobson, 1993), is, for instance, very useful in keeping the average queue size and hence having the queuing delay under control. This control is helpful for soft real-time traffic.

Algorithms of queue management at IP routers determine which packet should be deleted, if necessary. AQM, recommended now by the IETF, enhances the efficiency of transfers and cooperates with the TCP congestion window mechanism in adapting flow intensity to the congestion at a network (Braden *et al.*, 1998).

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 of 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 choice of the 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, so that they become synchronized.

To enhance the throughput and fairness of link sharing, and also to eliminate synchronization, the 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 and hence only chosen users are notified and the global synchronization of connections is avoided.

A detailed discussion of the active queue management goals is given by Braden *et al.* (1998). The RED algorithm was proposed by the IETF to enhance the transmission via IP routers. It was primarily described by Floyd and Jacobson (1993). Its idea is based on a drop function giving the probability that a packet is rejected. Figure 1 shows dropping functions for RED. 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

$$avg = (1-w)avg' + wq, \tag{1}$$

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 to ephemeral changes in the queue (noise). Floyd (1997) recommends $w = 0.001$ or $w = 0.002$, while Zheng and Atiquzzaman (2000) show the efficiency of $w = 0.05$ and $w = 0.07$. According to May *et al.* (2000a), concerning the influence of w on queuing time fluctuations, the result is obvious: the larger w, the higher the fluctuations. In the RED drop function there are two thresholds, Min_{th} and Max_{th}. If $avg < \mathrm{Min}_{th}$, all packets are admitted, if $\mathrm{Min}_{th} < avg < \mathrm{Max}_{th}$, then dropping probability p is growing linearly from 0 to $p_{\max}$:

$$p = p_{\max}\frac{avg - \mathrm{Min}_{th}}{\mathrm{Max}_{th} - \mathrm{Min}_{th}}, \tag{2}$$

and if $avg > \mathrm{Max}_{th}$, then all packets are dropped.

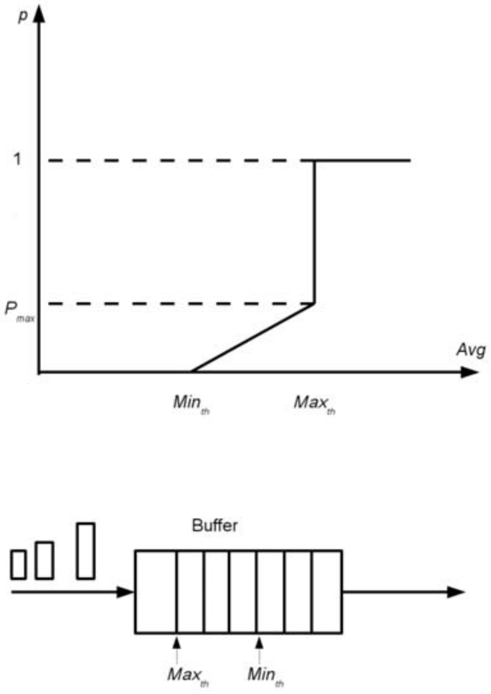

Fig. 1. Dropping functions for RED.

The value of $p_{\max}$ has also strong influence on the RED performance: if it is too large, the overall throughput is unnecessarily choked, and if it is too small, a danger of synchronization arises; Floyd (1997) recommends $p_{\max} = 0.1$. The problem of the choice of parameters is still being discussed (see, e.g., Chang Feng *et al.*, 1999; May *et al.*, 2000b). The mean avg may be also determined in another way (see Zheng and Atiquzzaman, 2002). Despite evident highlights, RED also shows drawbacks such as low throughput, unfair bandwidth sharing, introduction of variable latency, deterioration of network stability. Therefore, numerous propositions of basic algorithms improvements appear; their comparison can be found, e.g., in the work of Hassan and Jain (2004).

An alternative for the RED mechanism may be ECN (Explicit Congestion Notification) (Kinicki and Zheng, 2001). The ECN mechanism detects an overload between two network nodes and marks transmitted packets by a specific bit. The marked packets inform the receiver about congestion and the receiver requests a reduction of the transmission. Improving the RED mechanism is important regardless of whether packets are marked or dropped.

As already mentioned, the RED mechanism helps control the average queue size and hence is helpful for delay sensitive real-time traffic. This sort of real-time traffic mainly does not use TCP in the transport layer. Grinnemo *et al.* (2004) introduce a unified taxonomy of partially reliable transport protocols. Wang (2006) claimed that the RED mechanism could help control the queueing delay, no matter if the transport protocol remains cooperative or not.

3. Theoretical background: The assumed model of estimating the average queue length

The proposed average queue length calculating method is based on a difference equation (a recursive equation).

Such autoregressive moving average models are very well known in linear system identification problems (Box *et al.*, 1970). The proposed approach is some extension of the RED technique, where the average queue length is given by the first-order difference equation (1).

$A(n)$ (the average length at the n-th moment of time) may be expressed using the difference equation as follows:

$$\begin{aligned} A(n) &= a_1 A(n-1) + a_2 A(n-2) \\ &\quad + \cdots + a_k A(n-k) + b_0 Q(n) \\ &\quad + b_1 Q(n-1) + \cdots + b_m Q(n-m), \end{aligned} \tag{3}$$

where $a_j = \mathit{const}$ for $j = 1, \ldots, k$, $b_i = \mathit{const}$ for $i = 0, \ldots, m$, $A(l)$ is the average length at the l-th moment of time, $Q(l)$ is the current length of the packet queue at the l-th moment.

Constraints for coefficients a_j and b_i are

$$\sum_{j=1}^{k} a_j + \sum_{i=0}^{m} b_i = 1 \wedge a_j \geq 0 \wedge b_i \geq 0. \tag{4}$$

Such a model was named REDwM, i.e., RED with a modified weighted moving average. A score function (as defined by the cost function) based on, e.g., mean wait time, mean queue length, the number of dropped packets, the probability of packet dropping by the RED mechanism, may be used for obtaining the order of the model (m and k values) and concrete values of equation coefficients (a_j and b_i), but most of them are highly correlated. The minimum of the assumed score function determines an optimal model.

4. Proposed method

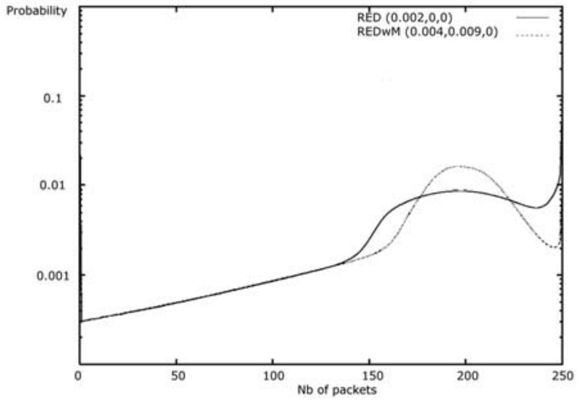

Fig. 2. Queue distribution for coefficient values $[a_1, a_2, b_1] = [0.004, 0.009, 0]$ and $[a_1, a_2, b_1] = [0.002, 0, 0]$.

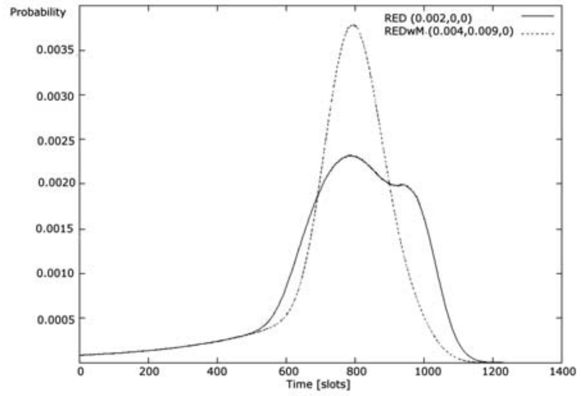

Fig. 3. Waiting time distribution for $[a_1, a_2, b_1] = [0.004, 0.009, 0]$ and $[a_1, a_2, b_1] = [0.002, 0, 0]$.

The classic RED approach (where the average length is given by Eqn. (1)) satisfies the equation of the model given by Eqn. (3), where only a_1 and b_0 are significant coefficients. Only one parameter, a_1, should be determined in the RED approach (because $b_0 = 1 - a_1$). This was named a one-dimensional model, which can be described by only one value, $[a_1]$ (i.e., $k = 1, m = 0$).

The score function based on the mean packet waiting time mw was used for finding the order of the model and its parameter values. The problem of obtaining an optimal

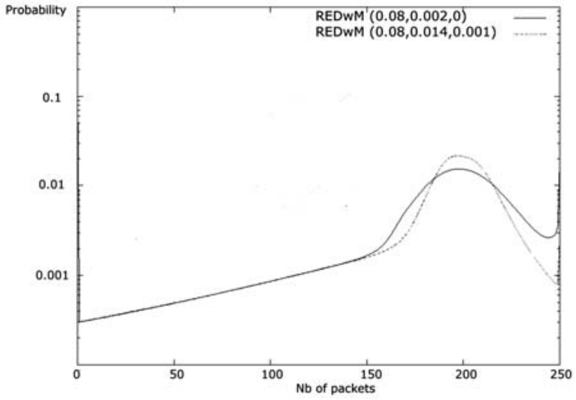

Fig. 4. Queue distribution for coefficient values $[a_1, a_2, b_1] = [0.08, 0.002, 0]$ and $[a_1, a_2, b_1] = [0.08, 0.0014, 0.001]$.

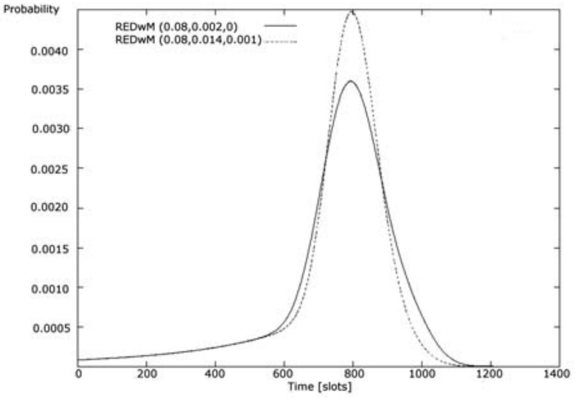

Fig. 5. Waiting times distribution for coefficient values $[a_1, a_2, b_1] = [0.08, 0.002, 0]$ and $[a_1, a_2, b_1] = [0.08, 0.0014, 0.001]$.

model is equivalent to that of finding a minimum of a multivariate score function. The optimization was made by using the well-known Hooke and Jeeves direct search method (Hooke and Jeeves, 1961; Kelley, 1999) applied for minimization of a multivariate score function.

We considered some set of multi-dimensional models for $k = 1, 2, 3$ and $m = 0, 1, 2$, and we found experimentally that only four parameters (a_1, a_2, b_0, b_1) are important.

Using a high dimensional model ($k > 2$ and $m > 1$) yields greater values of the score function (relative to the value of the score function for the optimal model based case with order $k = 2$ $m = 1$). We also obtained very small coefficient values of a_3 and b_2 (relative to the obtained values of a_1, a_2, b_0, b_1) for a high dimensional model.

Finally, we may say that only the three-dimensional model $[a_1, a_2, b_1]$ (where $k = 2, m = 1$ and $b_0 = 1 - a_1 - a_2 - b_1$) is based on significant coefficients and it is optimal subject to the assumed score function.

Using (3), we can calculate the average length for the assumed three-dimensional model as follows:

$$\begin{aligned} A(n) = a_1 A(n-1) &+ a_2 A(n-2) \\ &+ (1 - a_1 - a_2 - b_1) Q(n) \\ &+ b_1 Q(n-1). \end{aligned} \tag{5}$$

Especially for selected values of a_2 and b_1 does the proposed model $[a_1, a_2, b_1] = [a_1, 0, 0]$ become the classic RED model, i.e., model $[a1]$.

The process of finding the minimum of the score function was restarted many times for the set of initial value vectors $[a_1, a_2, b_1]$ defined over some discrete space size $10\times10\times10$, where $a_1 = 0.001, 0.001$+0.01,. . . , 0.091, and a_2 = 0, 0+0.001,. . . , 0.009, and b_1 = 0, 0+0.001,. . . , 0.009.

This multi-start procedure allows reducing the risk of finding only some local minimum of the score function.

For every initial vector, simulations were repeatedly restarted (with different seed values of pseudo-random generators) for evaluating the score function whose value is $\overline{mw}$. Here $\overline{mw}$ is the mean of mw. The number of simulations for achieving $\overline{mw}$ with acceptable accuracy was determined using the Central Limit Theorem and properties of the standard deviation distribution for the assumed confidence level equal to 0.95 and the confidence interval [$\overline{mw} - 0.05\overline{mw}$, $\overline{mw} + 0.05\overline{mw}$]. (The number of restarts was about 5.)

Every evaluation of the score function based on computation of mw for a given initial vector was performed after detection of the moment of entering a stationary state. For every simulation, the moment of time which determines when a stationary state of simulation begins was detected by observing the moving average of the last r values of the mean waiting time ($r = 4$). Here mw_i was taken into account at each i-th moment after sending some number of packets (about 1000). Let us assume that mw_i is the mean waiting time measured at the i-th moment, and the mean of the mean waiting time is defined as $mmw_j(r)$ = mean (mw_{j-r+1}, mw_{j-r+2},. . . , mw_j). The l-th moment determines the beginning of the stationary state when all new mw_j for $j = l + 1, \dots, l + r$ belong to the interval [$mmw_l(r) - 0.05mmw_l(r)$, $mmw_l(r) + 0.05mmw_l(r)$] (i.e., the values of mw_j are placed near those of $mmw_l(r)$). Every step of minimization with evaluating the waiting time score function is equivalent to performing a simulation, and it takes about one hour.

5. Results of simulation

The simulation evaluations were carried out with the use of the OMNeT++ simulation framework. It is a modular component-based simulator mainly designed for simulation of communication networks, queuing networks and performance evaluation. The framework is very popular in research and was established for academic purposes (OMNET++, 2014).

The service time represents the time of packet treatment and dispatching. In packet-switched networks it is the time required to transmit information. Bonald *et al.* (2000) consider the exponential service time distribution. We used a discrete-time model. Hence we assumed that the service-time distribution is geometric (which corresponds to a Poisson traffic in a continuous time model). The parameter μ of the geometric distribution of service times (the probability of the end of service within the current time-slot) was $\mu = 0.25$. A more accurate model should take into account the latest research on packet size distribution (Sinha *et al.*, 2007; CAIDA, 2014), but it is out of the scope of this paper.

To represent a self-similar traffic, we used an SMPP (Special Semi-Markov Process) model introduced by Robert (1996; 1997), Domańska and Domański (2005) as well as Domański *et al.* (2008). This model enables us to represent, with the use of few parameters, a network traffic which is self-similar over several time-scales (pseudo self-similar traffic (cf. Robert, 1997)). The time of the model is discrete and divided into unit-length slots. Packet arrivals are determined by an n-state discrete time Markov chain called a modulator. It was assumed that the modulator has $n = 5$ states ($i = 0, 1, \ldots, 4$) and packets arrive only when the modulator is in state $i = 0$. The input traffic intensity (the mean value of the Markov process) was chosen as $\alpha = 0.5$, and, due to the modulator characteristics, the “local" Hurst parameter of self-similar traffic was fixed to $H = 0.78$. For such a source model, one must fit only two parameters: expectation and the “local" Hurst parameter (plus the number of states in Markov chain n; it defines the number of time-scales on which the process has self-similar character).

The RED parameters had the following values: the buffer size of 250 packets, the threshold values of $\mathrm{Min}_{th} = 100$ and $\mathrm{Max}_{th} = 200$, $p_{\max} = 0.1$. A detailed discussion of the choice of model parameters is also presented by Domańska *et al.* (2012).

During the experiments we found two local minima of the score function based on the waiting time:

$$[a_1, a_2, b_1] = [0.004, 0.009, 0],$$

which yields WaitingTimeScore$(0.004, 0.009, 0) = 751.385$ and

$$[a_1, a_2, b_1] = [0.08, 0.0014, 0.001]$$

where WaitingTimeScore $(0.08, 0.0014, 0.001) = 738.232$.

Figure 2 displays the comparison of the queue distributions for coefficients values [0.004, 0.009, 0]. The comparison of the waiting times distributions for the same coefficient values is presented in Fig. 3. Figures 4 and 5 display the queue distributions and waiting times for coefficient values [0.08, 0.0014, 0.001].

Both results for [0.004, 0.009, 0] and [0.08, 0.0014, 0.001] are more convincing than the one based on the classical RED (Floyd and Jacobson, 1993), i.e., for $[a_1, a_2, b_1] = [0.002, 0, 0]$, in which WaitingTimeScore $(0.002, 0, 0) = 775.605$ was obtained.

Finally, the optimal parameters values are $[a_1, a_2, b_1] = [0.08, 0.0014, 0.001]$. For this case we obtained smaller average waiting times and a lower average queue length, but the loss probability was the same.

The above simulation parameters $\alpha = 0.50$, $\mu = 0.25$, correspond to a situation when the buffer is loaded. To show the behavior of the algorithm for the situation of an unloaded queue, the parameter $\mu = 0.50$ was increased. This corresponds to a situation when the buffer fills up not as fast as previously. This situation is presented in Figs. 6 and 7. For the case of an unloaded queue, the obtained waiting time is also better than that obtained for RED (WaitingTimeScore $(0.002, 0, 0) = 321.05$, Waiting TimeScore$(0.08, 0.0014, 0.001) = 308.536$).

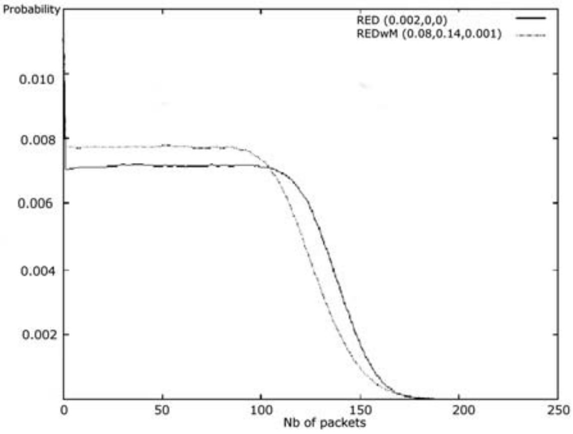

Fig. 6. Queue distribution for coefficient values $[a_1, a_2, b_1] = [0.08, 0.0014, 0.001]$ and $[a_1, a_2, b_1] = [0.002, 0, 0]$.

6. Results of real experiments using implementation of the REDwM algorithm for Linux

The simulation experiments allowed us to find some interesting coefficient values for the modified weighted moving average function. The next goal was the implementation of the modified RED algorithm in a real software router based on Linux. The main objective of this stage of our study was the confirmation of simulation results in a real working network. This section provides a

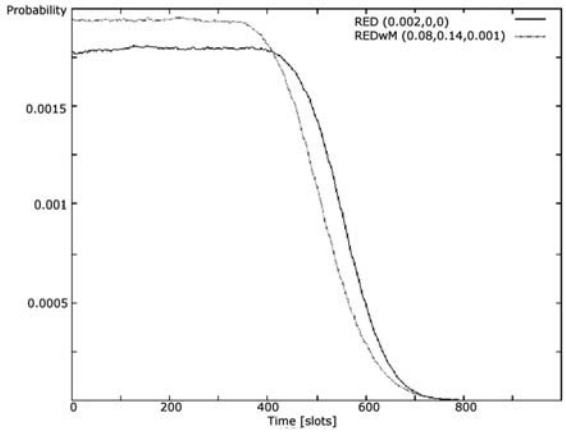

Fig. 7. Waiting times distribution for coefficient values $[a_1, a_2, b_1] = [0.08, 0.0014, 0.001]$ and $[a_1, a_2, b_1] = [0.002, 0, 0]$.

description of the created programs, a specification of the conducted experiments and the obtained results.

Figure 8 displays the experiment topology of the network. The server is the most important part of the network. It works as a router with AQM algorithms implemented inside. The router is running the Linux operating system and its main task is forwarding packets between two network segments. The connection between the server and the Internet is the bottleneck. As far as the router is concerned, its main task is to prevent overloads occurring in the network. For the purpose of the experiments the authors created an application implementing a standard RED algorithm and the RED algorithm with the modified weighted moving average.

The principle of the program is as follows. The packet routing (between networks) is possible thanks to the *iptables* package. To allow the application to function properly, the "ip_queue" module must be activated in the system. By applying appropriate rules, it is possible to redirect input traffic to the appropriate network output interface. At this point, we introduce the following modification: the corresponding rule places the whole transmitted traffic in a special user queue. Then the buffer is transferred from the kernel to the user space. In that space, the program removes the packet (or does not remove it) in accordance with the AQM rules. After the verdict, the packet comes back to the kernel space and after that information it is redirected to the output interface.

Such program construction brought some dangers into the system. In the case of packet processing, each incoming data portion must be shifted from the kernel space to the user space, for a heavy traffic system cannot keep up with packets support. In this case, the queue (in the kernel space) can be overflown by stored packets. To reduce the probability of the buffer overflows, the size of the buffer should be set to the maximum value.

Finally, the created software allowed us to evaluate the RED algorithm with modified weighted moving average behavior. The experiment environment consisted of the following components:

- Computer A—the traffic generator. The network traffic was generated by the *Nsasoft Network Traffic Emulator* program (Nsasoft, 2014).
- Computer B—the packet router (enabling to generate additional traffic).
- Server—the AQM packet router.
- link1—FastEthernet (100Mb/s) link RTT $= 0.5$ ms (Round Time Trip between Computer A and Computer B).
- link2—FastEthernet (100Mb/s) link RTT $= 5.6$ ms (Round Time Trip between Computer B and server).
- link to the Internet 516 kb/s (the connection between the server and the Internet is a bottleneck).

During the test we observed the behavior of the RED algorithm on the server.

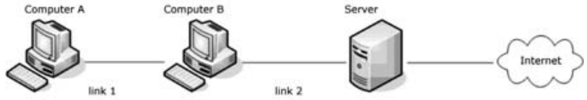

Fig. 8. Network used during our experiment.

The RED parameters had the following values:

- size of the queue $= 25$ packets,
- threshold $\mathrm{Min}_{th} = 5$ packets,
- threshold $\mathrm{Max}_{th} = 15$ packets,
- weight determining the importance of the instantaneous queue length $w = 0.002$.

The REDwM parameters had the following values: $a_1 = 0.08$, $a_2 = 0.0014$, $b_1 = 0.001$.

We observed the mean length of the queue, waiting times and the number of dropped packets for three kinds of experiments:

- AQM behavior for TCP flows,
- AQM behavior for UDP flows,
- AQM behavior for mixed (TCP and UDP) flows.

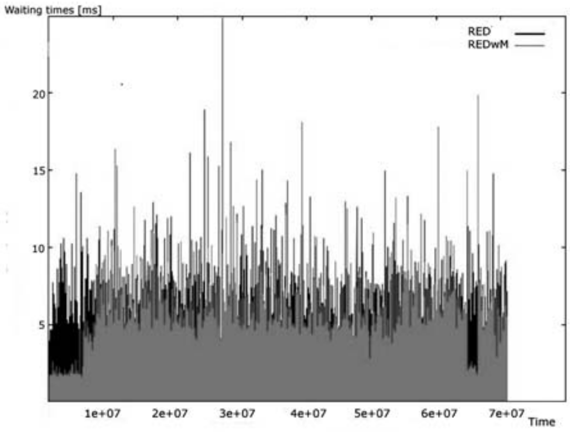

Fig. 9. Waiting times for the RED and REDwM algorithms and the TCP traffic.

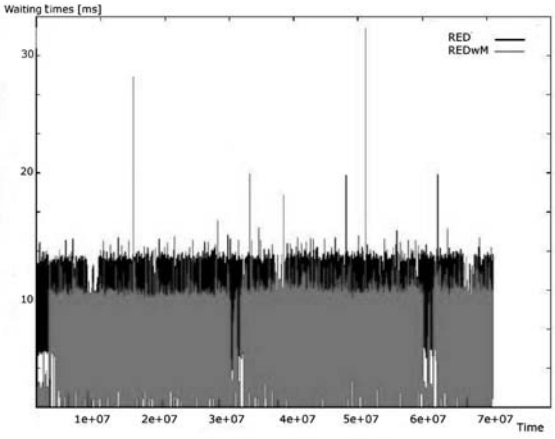

Fig. 10. Waiting times for the RED and REDwM algorithms and the UDP traffic.

For all analyzed cases the rate of packet generation was 2000 packets per second.

Figures 9–11 show the waiting times for RED and RED with modified weighted moving average algorithms. For each type of traffic the obtained waiting times were better for RED with modifications.

In the case of TCP traffic the difference between the results (number of dropped packets, average waiting times, average queue length) obtained for RED and REDwM is not significant (see Table 1). For the REDwM mechanism the probability of dropping packets is slightly bigger than for the RED one. The distributions of the queue length show that the buffer is not overloaded for TCP in contrast to the presence of UDP (see positively skewed histograms in Figs. 12 and 13, and the negatively skewed histogram in Fig. 14). Either the RED mechanism or the REDwM one works very well with TCP and therefore the losses are not significant (less than 1%).

For the mixed traffic and especially for the UDP one, the waiting times are greater (compared with the results for TCP traffic). The waiting times for the REDwM mechanism are smaller than for the RED one. For the UDP traffic the difference is about 1.5 ms, which gives 15% of the entire waiting time for RED.

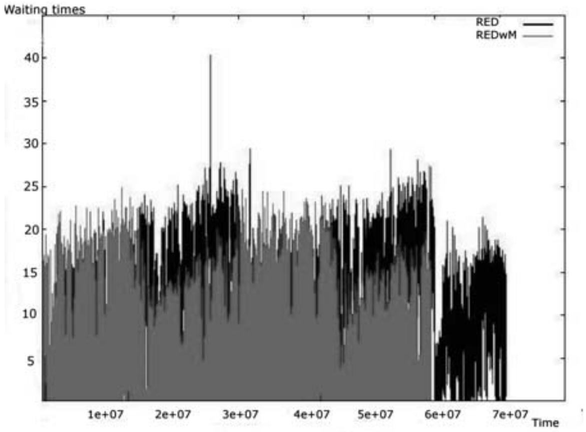

Fig. 11. Waiting times for the RED and REDwM algorithms and the mixed (TCP and UDP) traffic.

Table 1. Obtained results: number of dropped packets (%), average waiting times (ms), and average queue length.

Source	Algorithm	Drop.	Waiting	Length
TCP	RED	0.05	4.64	12.29
	REDwM	0.06	4.49	11.64
UDP	RED	18.81	10.51	24.80
	REDwM	18.95	8.95	21.40
UDP+TCP	RED	17.74	10.04	22.85
	REDwM	17.82	8.35	18.54

For TCP traffic AQM mechanisms reject not many packets, but in the case of mixed traffic and the UDP one the losses are significant.

7. Conclusions

Many extensions of the classic RED algorithm have been considered in the past. Some of them have been based on modifications of the probability of packed dropping function (Floyd *et al.*, 2001; Zhou *et al.*, 2006; Domańska and Domański, 2008; Augustyn *et al.*, 2010a; 2010b; Domańska *et al.*, 2012). The improvement proposed in this article is new and can be combined with the previously proposed alternatives.

This approach, named REDwM, is based on a proposed simple dynamical discrete model for obtaining the average packet queue length. The linear difference equation was proposed to compute the average length. The optimal values of equation coefficients were found during minimization of the score function. The achieved results obtained in OMNET-based numerical experiments

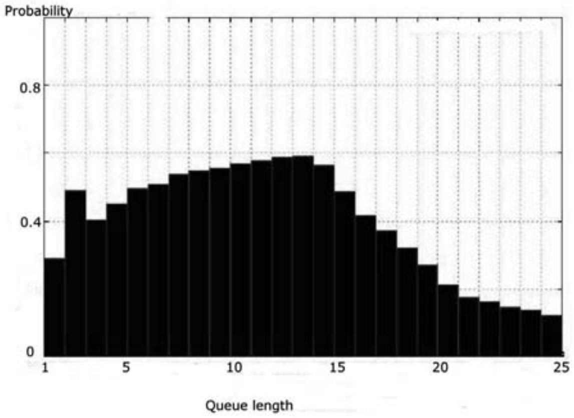

Fig. 12. Queue distribution for the RED algorithm and the TCP traffic.

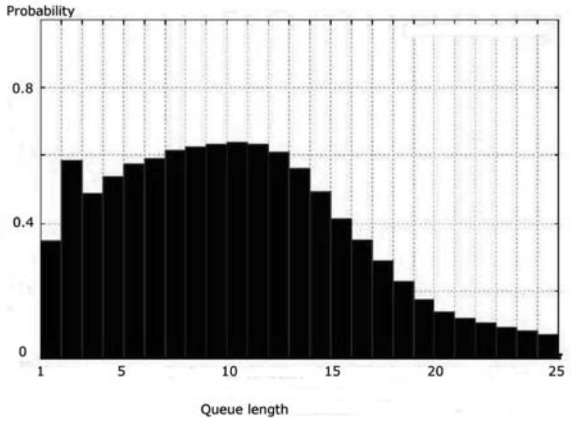

Fig. 13. Queue distribution for the REDwM algorithm and the TCP traffic.

are better than those from the classical RED approach (for the assumed score function based on the mean waiting time).

In this article we additionally presented the behavior of this mechanism involving real working routers. We presented advantages of active queue management for Linux-based routers. We implemented two variants of RED algorithms (classic RED and REDwM). We also demonstrated that the influence of our RED modifications upon the behavior of the router queue for TCP, UDP, and mixed (TCP, UDP) traffic was significant. The conducted experiments showed advantages of the proposed algorithm with a modified weighted moving average of the queue (REDwM). The proposed algorithm will make better usage of RED mechanisms for soft real-time transmissions.

The designed experimental environment allowed easy addition of AQM mechanisms in Linux. Unfortunately, the simplicity of the solution caused large delays (packets must be shifted between the kernel and user spaces). Adapting the presented mechanisms to soft real-time transmission will require making the packet discard a decision in the kernel.

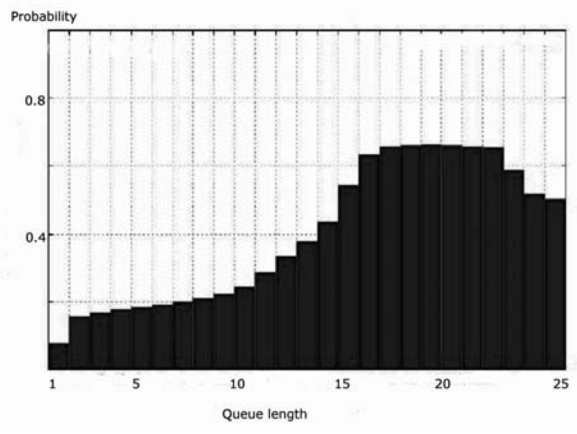

Fig. 14. Queue distribution for the REDwM algorithm and the TCP+UDP traffic.

Acknowledgment

This research was partially financed by the Polish Ministry of Science and Higher Education (project no. N N516441438).

References

Abeni, L., Palopoli, L., Lipari, G. and Walpole, J. (2002). Analysis of a reservation-based feedback scheduler, *Proceedings of the 23rd IEEE Real-Time System Symposium, Austin, TX, USA*, pp. 71–80.

Augustyn, D., Domański, A. and Domańska, J. (2010a). Active queue management with non linear packet dropping function, *Proceedings of the 6th International Conference on Performance Modelling and Evaluation of Heterogeneous Networks HET-NETs, Zakopane, Poland*, pp. 133–142.

Augustyn, D., Domański, A. and Domańska, J. (2010b). A choice of optimal packet dropping function for active queue management, *in* A. Kwiecień, P. Gaj and P. Stera (Eds.), *Computer Networks*, Communications in Computer and Information Science, Vol. 79, Springer-Verlag, Berlin/Heidelberg, pp. 199–206.

Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and Weiss, W. (1998). An architecture for differentiated services, *RFC 2475*, Network Working Group.

Bonald, T., May, M. and Bolot, J. (2000). Analytic evaluation of RED performance, *Proceedings of INFOCOM, Tel Aviv, Israel*, pp. 1415–1424.

Box, G., Jenkins, G. and Reinsel, G. (1970). *Time Series Analysis: Forecasting and Control*, John Wiley & Sons, Inc., Hoboken, NJ.

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. and Zhang, L. (1998). Recommendations on queue management and congestion avoidance in the internet, *RFC 2309*, Network Working Group.

Braden, R., Zhang, L., Berson, S., Herzog, S. and Jamin, S. (1997). Resource reservation protocol (RSVP)—version 1 functional specification, *RFC 2205*, Network Working Group.

CAIDA (2014). *Packet Size Distribution Comparison between Internet Links in 1998 and 2008,* `http://www.caida.org`.

Cena, G., Bertolotti, I.C. and Zunino, C. (2007). Evaluation of response times in industrial WLANs, *IEEE Transactions on Industrial Informatics* **3**(3): 191–201.

Cena, G., Seno, L., Valenzano, A. and Zunino, C. (2010). On the performance of IEEE 802.11e wireless infrastructures for soft-real-time industrial applications, *IEEE Transactions on Industrial Informatics* **6**(3): 425–437.

Cervin, A. and Eker, J. (2005). Control-scheduling codesign of real-time system: The control server approach, *Journal of Embedded Computing* **1**(2): 209–224.

Chang Feng, W., Kandlur, D. and Saha, D. (1999). Adaptive packet marking for maintaining end to end throughput in a differentiated service internet, *IEEE/ACM Transactions on Networking* **7**(5): 685–697.

Cucinotta, T., Palopoli, L., Abeni, L., Faggioli, D. and Lipari, G. (2010). On the integration of application level and resource level QoS control for real-time application, *IEEE Transaction on Industrial Informatics* **6**(4): 479–491.

Domańska, J., Augustyn, D. and Domański, A. (2012). The choice of optimal 3-rd order polynomial packet dropping function for NLRED in the presence of self-similar traffic, *Bulletin of the Polish Academy of Sciences: Technical Sciences* **60**(4): 779–786.

Domańska, J. and Domański, A. (2005). The influence of traffic self-similarity on QoS mechanism, *Proceedings of the International Symposium on Applications and the Internet, SAINT, Trento, Italy*, pp. 300–303.

Domańska, J. and Domański, A. (2008). Active queue management in linux based routers, *in* J. Klamka, J. Domańska and A. Kapczyński (Eds.), *Advanced Problems of Internet Technologies*, WSB, Dąbrowa Górnicza, pp. 63–73.

Domańska, J., Domański, A. and Czachórski, T. (2013). Fluid flow analysis of RED algorithm with modified weighted moving average, *in* A. Dudin, V. Klimenok, G. Tsarenkov and S. Dudin (Eds.), *Modern Probabilistic Methods for Analysis of Telecommunication Networks*, Communications in Computer and Information Science, Vol. 356, Springer-Verlag, Berlin/Heidelberg, pp. 50–58.

Domański, A., Domańska, J. and Czachórski, T. (2008). The impact of self-similarity on traffic shaping in wireless LAN, *in* S. Balandin, D. Moltchanov and Y. Koucheryavy (Eds.), *Next Generation Teletraffic and Wired/Wireless Advanced Networking*, Lecture Notes in Computer Science, Vol. 5174, Springer, Berlin/Heidelberg, pp. 156–168.

El-Gendy, M.A., Bose, A. and Shin, K.G. (2003). Evolution of the internet QoS and support for soft real-time applications, *Proceedings of the IEEE* **91**(7): 1086–1104.

Floyd, S. (1997). Discussions of setting parameters, `http://www.icir.org/floyd/redparameters.txt`.

Floyd, S., Gummadi, R. and Shenker, S. (2001). Adaptive RED: An algorithm for increasing the robustness of REDS active queue management, *Technical report*, ICSI, Berkeley, CA.

Floyd, S. and Jacobson, V. (1993). Random early detection gateways for congestion avoidance, *IEEE/ACM Transactions on Networking* **1**(4): 397–413.

Georgoula, S., Trimintzios, P., Pavlou, G. and Ho, K. (2005). Heterogeneous real-time traffic admission control in differentiated services domains, *Proceedings of the IEEE GLOBECOM, St. Louis, MO, USA*, Vol. 1, pp. 523–528.

Grinnemo, K., Garcia, J. and Brunstrom, A. (2004). Taxonomy and survey of retransmission-based partially reliable transport protocols, *Computer Communications* **27**(15): 1441–1452.

Haider, A. (2004). *Improved Congestion Control for Packet Switched Data Networks and the Internet*, Ph.D. thesis, University of Canterbury, Christchurch.

Haider, A. and Nakao, A. (2008). On path switching in overlay network, *Proceedings of the Australasian Telecommunication Networks and Applications Conference (ATNAC), Adelaide, Australia*, pp. 355–360.

Hassan, M. and Jain, R. (2004). *High Performance TCP/IP Networking*, Pearson Education Inc., Upper Saddle River, NJ.

Hooke, R. and Jeeves, T. (1961). Direct search solution of numerical and statistical problems, *Journal of the ACM* **8**(2): 212–229.

Kelley, C. (1999). *Iterative Methods for Optimization*, Frontiers in Applied Mathematics, Vol. 18, SIAM, Philadelphia, PA.

Kinicki, R. and Zheng, Z. (2001). A performance study of explicit congestion notification (ECN) with heterogeneous TCP flows, *Proceedings of the First International Conference on Networking, London, UK*, pp. 98–106.

Kurose, J. and Ross, K. (2007). *Computer Networking: A Top-Down Approach Featuring the Internet*, 4th Edn., Addison Wesley, Boston, MA.

Lu, C., Lu, Y., Abdelzaher, T., Stankovic, J. and Son, S. H. (2006). Feedback control architecture and design methodology for service delay guarantees in web servers, *IEEE Transactions on Parallel and Distributed Systems* **17**(9): 1014–1027.

May, M., Bonald, T. and Bolot, J. (2000a). Analytic evaluation of RED performance, *Proceedings of IEEE Infocom, Tel-Aviv, Israel*, pp. 1415–1424.

May, M., Diot, C., Lyles, B. and Bolot, J. (2000b). Influence of active queue management parameters on aggregate traffic performance, *Technical report*, Institut de Recherche en Informatique et en Automatique, Sophia Antipolis.

Nsasoft Traffic Emulator (2014). http://www.nsasys.com.

OMNET++ (2014). http://www.omnetpp.org.

Padhy, P. and Sundaram, R. (2010). Analysis and design of improved PI-PD controller for TCP AQM routers, *Proceedings of the International Conference on Power, Control and Embedded Systems (ICPCES), Allahabad, India*, pp. 1–5.

Palopoli, L., Conticelli, F., Natale, M.D. and Butazzo, G. (2000). Real-time control system analysis: An integrated approach, *Proceedings of the 21st IEEE Real-Time System Symposium, Orlando, FL, USA*, pp. 131–140.

Reid, C. (2008). *Achieving Soft Real-Time Guarantees for Interactive Applications in Wireless Mesh Networks*, Ph.D. thesis, University of Waterloo, Ontario.

Reinemo, S., Sem-Jacobsen, F., Skeie, T. and Lysne, O. (2003). Admission control for DiffServ based quality of service in cut-through networks, *in* T.M. Pinkston and V.K. Prasanna (Eds.), *High Performance Computing (HiPC 2003)*, Lecture Notes in Computer Science, Vol. 2913, Springer, Berlin/Heidelberg, pp. 118–129.

Robert, S. (1996). *Modélisation Markovienne du Trafic dans les Réseaux de Communication*, Ph.D. thesis, Ecole Polytechnique Fédérale de Lausanne, Lausanne.

Robert, S.J.B. (1997). New models for pseudo self-similar traffic, *Performance Evaluation* **30**: 57–68.

Shah, P., Yasmin, S., Asghar, S., Qayyum, A. and Hasbullah, H. (2012). A fluid flow model for SCTP traffic over the internet, *Proceedings of the International Conference on Emerging Technologies (ICET), Cairo, Egypt*, pp. 1–6.

Sinha, R., Papadopoulos, C. and Heidemann, J. (2007). Internet packet size distributions: Some observations, *Technical Report ISI-TR-2007-643*, USC, Los Angeles, CA.

Stankovic, J. and Rajkumar, R. (2004). Real-time operating systems, *Real-time Systems* **28**(2–3): 237–253.

Wang, B. (2009). *Priority and Realtime Data Transfer over the Best-effort Internet*, VDM Verlag, Saarbrücken.

Wang, X. (2006). *Active Queue Management for Real-time IP Traffic*, Ph.D. thesis, University of London, London.

Wijnants, M. and Lamotte, W. (2006). Timeliness of real-time IP communications in switched industrial ethernet networks, *IEEE Transaction on Industrial Informatics* **2**(1): 25–39.

Wijnants, M. and Lamotte, W. (2008). Managing client bandwidth in the presence of both real-time and non real-time network traffic, *Proceedings of the 3rd International Conference on Communication Systems Software and Middleware (COMSWARE 2008), Bangalore, India*, pp. 442–450.

Wroclawski, J. (1997). The use of RSVP with IETF integrated services, *RFC 2210*, Network Working Group.

Zheng, B. and Atiquzzaman, M. (2000). A framework to determine the optimal weight parameter of RED in next generation internet routers, *Technical report*, University of Dayton, Dayton, OH.

Zheng, B. and Atiquzzaman, M. (2002). Low pass filter/over drop avoidance (LPF/ODA): An algorithm to improve the response time of red gateways, *International Journal of Communication Systems* **10**(15): 899–906.

Zhou, K., Yeung, K. and Li, V. (2006). Nonlinear RED: A simple yet efficient active queue management scheme, *Computer Networks: The International Journal of Computer and Telecommunications Networking* **50**(18): 3784–3794.

Joanna Domańska, Ph.D., Eng., works in the Computer Systems Modelling and Performance Evaluation Group of the Institute of Theoretical and Applied Informatics, Polish Academy of Sciences. Her main areas of research include performance modelling methods for computer networks.

Adam Domański, Ph.D., Eng., works in the Computer Equipment Group of the Institute of Computer Science, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology. His main research interest in the computer network domain is congestion control in packet networks.

Dariusz R. Augustyn, Ph.D., Eng., received his M.Sc. and Ph.D. degrees from the Silesian University of Technology in Gliwice. He works in the Institute of Computer Science at the same university. His main research areas include database theory, mathematical modelling and software engineering.

Jerzy Klamka, Ph.D., Prof., full member of the Polish Academy of Sciences, works in the Quantum Systems of Informatics Group of the Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences. His main area of research are controllability and observability of linear and nonlinear dynamical systems, and mathematical foundations of quantum computations. He is the author of monographs and numerous papers published in international journals.

Received: 18 April 2013
Revised: 2 December 2013
Re-revised: 18 March 2014