Int. J. Appl. Math. Comput. Sci., 2016, Vol. 26, No. 4, 777–789
DOI: 10.1515/amcs-2016-0055

THE USE OF A NON–INTEGER ORDER PI CONTROLLER WITH AN ACTIVE QUEUE MANAGEMENT MECHANISM

ADAM DOMAŃSKI [b], JOANNA DOMAŃSKA [a], TADEUSZ CZACHÓRSKI [a], JERZY KLAMKA [a,*]

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

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

In this paper the performance of a fractional order PI controller is compared with that of RED, a well-known active queue management (AQM) mechanism. The article uses fluid flow approximation and discrete-event simulation to investigate the influence of the AQM policy on the packet loss probability, the queue length and its variability. The impact of self-similar traffic is also considered.

Keywords: active queue management, PI controller, dropping packets, fractional calculus.

1. Introduction

The congestion problem is frequently encountered in communication networks. Network routers transmit packets by links having a limited bandwidth and, if the number of incoming packets exceeds a link capacity, the link is congested and router buffers storing packets to be transmitted via this link may overflow.

To address this problem in TCP/IP networks, the Internet Engineering Task Force (IETF), which develops and promotes Internet standards, in particular those that comprise the Internet protocol suite (TCP/IP), recommends to use in IP routers *active queue management* (AQM), a mechanism of preventive packet dropping even if there is still place to store packets, which aims at advertising that the queue is increasing and the danger of congestion approaches. The probability of packet rejection is increasing together with the level of congestion. The packets are dropped randomly, hence only certain users are notified and the global synchronization of connections is avoided. AQM 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).

The most well-known AQM scheme is RED (random early detection), proposed by Floyd and Jacobson (1993). At each packet arrival, a moving queue average ν is determined as the current queue length taken with the weight w and the old moving average taken with the weight $1 - w$. In this way, the moving average acts as a low-pass filter. The probability of rejection p is a function of the moving average: if ν is smaller than a defined lower threshold $\mathrm{Min}_{\mathrm{th}}$, then $p = 0$; p increases linearly from 0 to a determined $P_{\max}$ within the lower and upper threshold $\mathrm{Max}_{\mathrm{th}}$, and then for greater ν all packets are rejected: $p = 1$.

This mechanism improves link utilization and is very useful in keeping the average queue size and queuing delay on a reasonable level. The choice of its parameters is not evident (Tan *et al*., 2006) and has been thoroughly discussed (see, e.g., Chang Feng *et al*., 1999; May *et al*., 2000). If they are not chosen correctly, RED performance can degrade and the TCP/RED system may become unstable (Unal *et al*., 2013). Tan *et al*. (2006) derived some explicit conditions under which the TCP/RED system is stable in terms of the average queue length. Despite its evident highlights, RED also shows such drawbacks as unfair bandwidth sharing, introduction of variable latency, or deterioration of network stability. The

*Corresponding author

TCP/RED system becomes unstable when the round-trip delay and link capacity increase, and the number of TCP sessions decreases (Tan *et al*., 2006). A number of studies how to improve the basic algorithm have appeared; their comparison may be found, e.g., in the work of Hassan and Jain (2004). We have also proposed and evaluated a few variants (cf. Domańska *et al.* 2014a; 2014c; 2012; 2007; 2008; 2013; Augustyn *et al.*, 2010; Domańska and Domański, 2008; Domański *et al.*, 2012).

The AQM mechanism may be seen as part of closed loop control of TCP/IP traffic. The feedback control system includes a TCP congestion window increasing or decreasing the TCP flow as a function of losses, the queue of packets at a bottleneck router reacting to the changes in the input flow, and AQM determining the loss probability, which in turn, after a certain delay, modifies the congestion window.

As has been discussed in many papers (e.g., Misra *et al*., 2000; Hollot *et al*., 2002), this feedback control system may be modeled and the model used to investigate the system stability with the use of control theory. The most common model is based on fluid flow approximation and is presented more formally in the next section. Each element of the control loop, after its linearization, is represented by its transfer function. The RED mechanism, seen as a control block with the queue length $q(t)$ as the input signal and loss probability $p_{\text{loss}}(t)$ as the output signal, turns out to be a first-order system (Hollot *et al.*, 2001b), i.e., having the Laplace transform of the transfer function of type $k/(sT+1)$, where k and T are constant.

We may also investigate the performance of this closed loop control if the RED mechanism is replaced by any other controller. Hollot *et al*. (2002) focused on the design of a proportional-integral controller on low-frequency dynamics. Quet and Ozbay (2004) described a robust controller based on a known technique for H^∞ control of systems with time delays. The H^∞ controller was designed for the original linear system proposed by Hollot *et al*. (2002) but without neglecting high-frequency dynamics that ensure robust stability and good performance for a wider range of network parameters. The AQM scheme proposed by Quet and Ozbay (2004) performs better than RED by obtaining shorter transients and less oscillatory responses, which gives higher link utilization, a low packet loss rate and smaller queue fluctuations. Chen and Yang (2007) proposed a robust AQM controller for IP routers based on modern robust μ-analysis and H_∞ optimal control theory. The authors approximated the μ-optimal control problem by the $H_\infty S/T/U$ mixed sensitivity problem. In the work of Manfredi *et al*. (2009) an H^∞ state feedback controller for time-delay systems was used to adjust the queue length against variations in the average round-trip time, the load and link capacity. Zheng and Nelson (2009) proposed an AQM congestion controller based on the H_∞ approach and studied the problem of the robustness of the congestion control algorithm against the disturbance on the available link bandwidth since it is often time-varying and cannot be exactly measured. The disadvantage of the above described controllers is their difficult implementation in real networks due to their computational complexities.

The PI AQM controller proposed by Hollot *et al.* (2001b; 2002) was designed following the small-gain theorem. Based on easy implementation of PI AQM controllers in real networks, a number of PI controllers have been proposed (Michiels *et al*., 2006; Melchor-Aquilar and Castillo-Tores, 2007; Ustebay *et al*., 2007; Melchor-Aquilar and Niculescu, 2009). Unal *et al*. (2013) compared the performance of several of them.

During the last few years, fractional order calculus has been used in many mathematical models of dynamical systems both continuous and discrete. The literature (Podlubny, 1999b; Chen *et al*., 2009) indicates that non-integer order controllers may have better performance than traditional integer order ones. Luo and Chen (2009) showed that, in many instances, fractional order controllers outperform the best integer order controllers. The first application of the fractional order PI controller as an AQM policy in the fluid flow model of a TCP connection was presented by Krajewski and Viaro (2014). The authors focused on determining the parameter regions, where the PI$^\alpha$ controller ensures a given modulus margin (inverse of the H_∞ norm of the sensitivity function).

Our article investigates the performance of a fractional order PI controller (PI$^\alpha$) applied to control Internet traffic supervised by TCP and UDP transport protocols. We investigate the influence of parameters of the controller on the packet loss probability, the queue length (hence also transmission time) and its variability (jitter), which are usual determinants of the quality of service for network transmissions. We also compare its performance with the RED mechanism, usually implemented in IP routers. The performance of TCP closed loop control is investigated with the use of fluid flow approximation, as proposed earlier (Misra *et al*., 2000; Hollot *et al*., 2002) and summarized in Section 2.1. The basic definition and properties of the theory of fractional calculus are summarized in Section 2.2. Section 2.3 gives features of the fractional order PI controller. The results of comparing the PI$^\alpha$ controller with the RED policy are presented in Section 3.

We also model open loop performance of an AQM router with the PI$^\alpha$ controller via discrete event simulation. In this case we include a more detailed Internet traffic model reflecting self-similarity, a feature frequently observed in traffic and having decisive impact on the quality of service. Section 4 provides simulation

results of open loop performance of the AQM router with the PI$^{\alpha}$ controller versus RED policy in the presence of self-similar traffic. Concluding remarks are presented in Section 5.

2. Theoretical background

2.1. Fluid flow analysis. This section presents a fluid flow model of a TCP connection having a bottleneck router with RED or an AQM policy. This model demonstrates the TCP protocol dynamics based on a model developed by Misra *et al*. (2000), as well as Hollot *et al.* (2001a), which uses fluid flow and stochastic differential equation analysis. The model ignores TCP timeout mechanisms and allows obtaining the average value of key network variables. In Misra *et al*. (2000), a differential-equation-based fluid model was presented to enable transient analysis of TCP/AQM networks. The authors described the behavior of TCP networks (flows and queues) using a set of stochastic differential equations. They also showed how to obtain ordinary differential equations by taking expectations of stochastic differential equations, and how to solve the resultant coupled ordinary differential equations numerically. These equations represent the expected or mean behavior of the system.

The dynamics of the TCP window for the i-th stream are approximated by (Hollot *et al.*, 2001a)

$$\frac{\mathrm{d}W_i(t)}{\mathrm{d}t} = \frac{1}{R_i(t)} - \frac{W_i(t)W_i(t-R_i(t))}{2R_i(t-R_i(t))}p(t-R_i(t)), \tag{1}$$

where $W_i(t)$ is the expected TCP sending window size (packets),

$$R_i(t) = \frac{q(t)}{C} + Tp_i$$

is the round-trip time [s], $q(t)$ is the queue length (packets), C is the link capacity [packets/s], Tp_i is the propagation delay [s], p is the probability of packet drop, i is the index of flow.

The first term on the right-hand side of Eqn. (1) represents the rate of increase in the congestion window due to incoming acknowledgments, while the second represents the rate with which the congestion window decreases due to packet losses. The model ignores the slow start phase in congestion window algorithm, where the increase of its size is non-linear. Each packet loss halves the congestion window size as it is done in Reno. The router transmits also UDP packets, assuming that there are n_2 UDP constant bit rate streams with intensities U_i.

The dynamics of a queue are given by

$$\frac{\mathrm{d}q(t)}{\mathrm{d}t} = \sum_{i=1}^{n_1}\frac{W_i(t)}{R_i(t)} + \sum_{i=1}^{n_2}U_i - C, \tag{2}$$

where n_1 is the number of TCP streams, n_2 is the number of UDP streams.

Figure 1 presents the control block diagram based on the differential equations presented above.

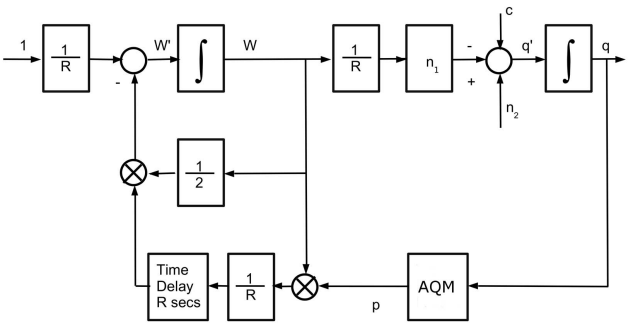

Fig. 1. Non-linear dynamic model for TCP/UDP/AQM connection.

The maximum values of q and W (queue length and congestion window size) depend on the buffer capacity and the maximum window size. The dropping probability p depends on the AQM queue algorithm. Equation (2) gives the speed of the bottleneck router queue changes due to incoming and leaving flows of packets.

The traffic composed of TCP and UDP streams was considered by Wang *et al*. (2005) and Domański *et al*. (2012). In these works, all TCP sources had the same window dynamics and UDP streams were permanently associated with the TCP stream. The pair taken into consideration was the TCP stream being limited by the UDP stream. In this article, the TCP and UDP streams are treated as separate streams. In the works of Kiddle *et al*. (2003) and Yung *et al*. (2001) a separate UDP stream was used; our work (described by Domańska *et al.* (2014c)) differs in this respect in that TCP can start from various initial window sizes. Additionally, we can set up a time after which the stream (TCP or UDP) will be started. The dynamics of the queue described by Eqn. (2) are influenced only by the streams that had started and have not yet sent a predetermined number of packets.

The model may be extended to any network topology and any number of TCP flows (e.g., Nycz *et al*., 2015) presents a numerical example for a real network topology with 134,023 nodes and 50,000 flows. In this case, for a

flow i,

$$\frac{\mathrm{d}W_i(t)}{\mathrm{dt}} = \frac{1}{R_i(\boldsymbol{q}(t))} - \frac{W_i(t)}{2}\frac{W_i(t-\tau)}{R_i(\boldsymbol{q}(t-\tau))} \times \Big(1 - \prod_{j\in V_i}(1-p_{ij})\Big). \tag{3}$$

A router allows reception of traffic from K TCP flows ($K \leq N$), where N is the entire number of flows in the network, p_{ij} denotes the loss probability at node j for packets of connection i, V_i is the set of nodes belonging to this connection, and $\boldsymbol{q}(t)$ is the vector of queues at these nodes. Delays R_i consist of queue delays at all nodes j, defined as $q_j(t)/C_j$ along this connection and propagation delay Tp_i:

$$R_i(\boldsymbol{q}(t)) = \sum_{j\in V_i}^{M}\frac{q_j(t)}{C_j} + Tp_i, \tag{4}$$

where M is the number of hops, i.e., $M-1$ is the number of intermediate routers in connection i. The drop probability p_{ij} is determined according to RED or another AQM mechanism, while the differential equations are solved numerically.

2.2. Fractional calculus. As was already mentioned in Introduction, fractional calculus is important in many areas, e.g., viscoelasticity, capacitor theory, electrical circuits, electro-analytical chemistry, neurology, diffusion, control theory and statistics (Podlubny, 1999a).

Fractional order derivatives and integrals (FODs/FOIs) are a natural extension of the well known integrals and derivatives. Differ-integrals of non-integer orders enable better and more precise control of physical processes. In this section, basic definition and properties of the fractional integral and fractional calculus are presented. These definitions unify the definition of a derivative and an integral to one differ-integral definition. The function is a fractional derivative (for $\alpha > 0$) or a fractional integral (for $\alpha < 0$). For $\alpha = 0$, a function is the function itself.

The most popular formulas to calculate differ-integral numerically are the Grunwald–Letnikov (GrLET) formula and the Riemann–Liouville (RL) formulas (Miller and Ross, 1993; Podlubny, 1999a; Ciesielski, 2006). The Grunwald–Letnikov formula is

$$\frac{\mathrm{d}^\alpha}{\mathrm{d}t^\alpha} = \lim_{h\to 0}\frac{1}{h^\alpha}\sum_{r=0}^{\infty}(-1)^r\binom{\alpha}{r}f(t-rh). \tag{5}$$

In active queue management, packet drop probabilities are determined at discrete moments of packet arrivals, so in simulations we consider the queue as a discrete system.

In the case of discrete systems, there is only one definition of differ-integrals of non-integer order. This definition is a generalization of the traditional definition of the difference of integer order to non-integer order, and it is analogous to a generalization used in GrLET formula. In discretization it is generally assumed that the step discretization $h > 0$ is given. The difference of a non-integer order is defined as follows:

$$\triangle^\alpha x_k = \sum_{j=0}^{k}(-1)^j\binom{\alpha}{j}x_{k-1}, \tag{6}$$

where $\alpha \in \mathbb{R}$ is generally a non-integer fractional order, x_x is a differentiated discrete function and the coefficient $\binom{\alpha}{j}$ is defined as follows:

$$\binom{\alpha}{j} = \begin{cases} 1 & \text{for } j = 0, \\ \dfrac{\alpha(\alpha-1)\ldots(\alpha-j+1)}{j!} & \text{for } j = 1,2,\ldots, \end{cases} \tag{7}$$

i.e., the difference of a non-integer order is the sum of all samples from point x_0 to point x_k with coefficients $\nu(j) = \binom{\alpha}{j}$ defined above.

2.3. Fractional order PI$^\alpha$ controller. Recall that the proportional-integral controller (PI controller) is a traditional mechanism used in feedback control systems. Earlier works show that the non-integer order controllers have better behavior than classic controllers (Podlubny, 1999b). PI$^\alpha$ may be used instead of the RED mechanism to determine the probability p of a packet drop in the following way:

$$p = \max\{0, -(K_p e + K_i\sum_{j\in V_i}\nu(j)e)\}, \tag{8}$$

where K_p, K_i are tuning parameters, $\nu(j)$ is a coefficient associated with the non-integer integral order, e is the error $e = q - q_d$ (q: actual queue size, q_d: desired queue size).

In this proposal the dropping probability depends on three parameters: the coefficients for the proportional and integral terms (K_p, K_i) and the integral order (α).

Figures 2–4 present the probability of packet dropping p given by Eqn. (8) and based on the PI$^\alpha$ response. In Figs. 2 and 3 we have $\alpha = -1$, so it is in fact a PI controller. The integral term in PI$^\alpha$ yields a strong correlation between the packet dropping probability and the history of queue occupancy and this probability is not the same as in the RED mechanism. The figures present how to change of the probability of packet dropping during a continuous increase in buffer occupancy as a result of the continuous packets incoming. Naturally, the response depends on the choice of parameters.

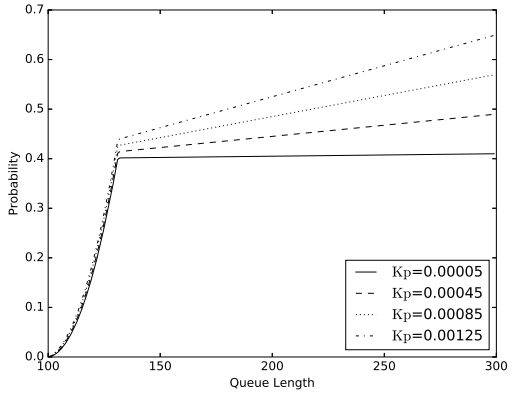

Fig. 2. Packet dropping probability based on a PI$^\alpha$ controller response (the influence of the parameter K_p): $\alpha = -1$, $K_i = 0.0008$.

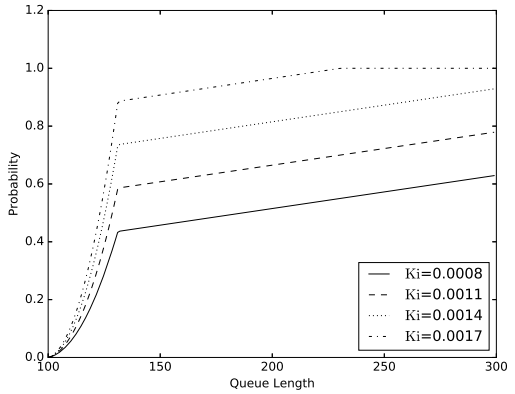

Fig. 3. Packet dropping probability based on a PI$^\alpha$ controller response (the influence of the parameter K_i): $\alpha = -1$, $K_p = 0.00115$.

3. Fluid flow analysis of PI$^\alpha$ controller performance

Our main goal is here to evaluate the PI$^\alpha$ controller as an active queue management mechanism taking into account the queue length and packet loss. For numerical fluid flow computations, we used software written in Python and presented previously (Domańska *et al.*, 2014c). During the tests we assumed the following TCP connection parameters:

- transmission capacity of AQM router: $C = 0.075$,
- propagation delay for the i-th flow: $T_{p_i} = 2$,
- initial congestion window size for the i-th flow (measured in packets): $W_i = 1, 2, 3, 4, \ldots$,
- starting time for the i-th flow (TCP and UDP),

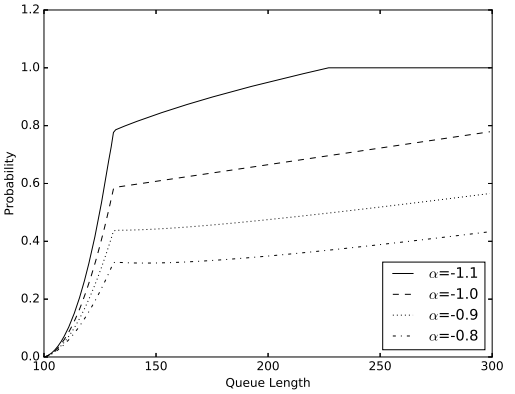

Fig. 4. Packet dropping probability on a PI$^\alpha$ controller response (the influence of the integral order α): $K_p = 0.00115$, $K_i = 0.0011$.

- the number of packets sent by the i-th flow (TCP and UDP).

The results were compared with the performance of the RED mechanism with the following parameters: $\mathrm{Min_{th}} = 10$, $\mathrm{Max_{th}} = 15$, buffer size (measured in packets) $= 20$, $P_{\max} = 0.1$, weight parameter $w = 0.007$, and PI$^\alpha$ behavior with setpoint $= 10$.

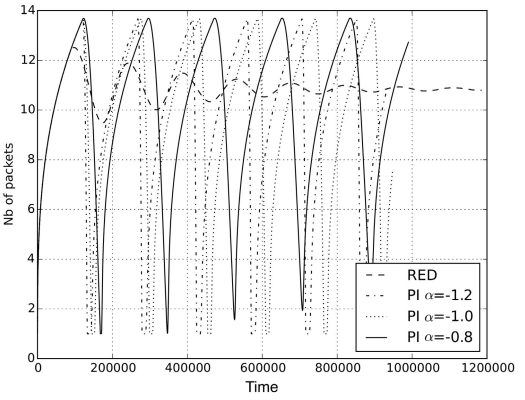

Fig. 5. TCP congestion window evolution. PI$^\alpha$ parameters: $K_p = 0.00115$, $K_i = 0.0011$.

Figure 5 displays the evolution of the congestion window W in the case of one TCP flow. The experiment was repeated for four different AQM mechanisms: RED and a PI$^\alpha$ controller with different integral orders: -0.8, -1.0, -1.2. The tuning parameters were $K_p = 0.00115$, $K_i = 0.0011$. The controller with the same set parameters was used later in an open loop scenario. In all cases, the size of the congestion window increases as long as the drop probability (given by the AQM mechanism based on the queue length) equals zero. When the AQM mechanism

Table 1. Average queue length.

AQM	Avg. queue length
RED	10.4067
$PI, \alpha = -0.8$	8.6814
$PI, \alpha = -1.0$	8.3312
$PI, \alpha = -1.2$	8.1807

starts dropping packets, the size of the congestion window decreases, causing a decrease in the queue length and this pattern is repeated periodically. For all graphs, time is represented in abstract simulation units.

We remark that congestion window changes have several properties.

The politics of packet dropping (compared with RED) are more aggressive for the PI$^{\alpha}$ controller (W decreases more). The reaction of the PI$^{\alpha}$ controller is also more delayed than in RED. This is due to a slightly different method of calculating packet drop probability. Contrary to RED, in the case of the PI$^{\alpha}$ controller, the length of the period when queue occupancy does not exceed the desired point influences the drop probability. In the case of a single TCP stream, the PI$^{\alpha}$ parameters are not as significant. The obtained average queue length (Table 1) differs slightly and does not exceed the setpoint. However, the trends are consistent with the results obtained earlier. The most aggressive controller reacts first on buffer overflow.

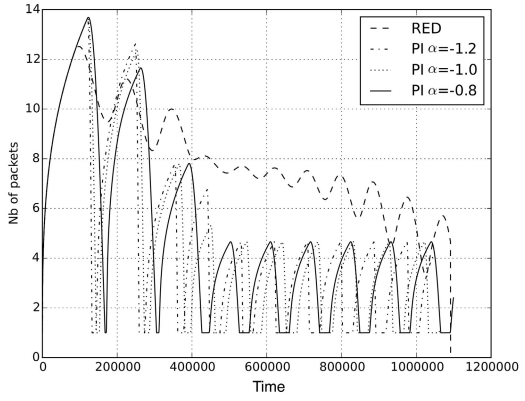

Fig. 6. TCP congestion window W evolution: 4 TCP streams.

The next example shows the cooperation of four TCP streams. The experiment is divided into phases. In the first part, only one stream transmits data. Then at $t = 200000$ the next TCP flow starts transmission and at $t = 400000$ two further streams become active. The first stream sends 60,000, the second stream 40,000, the third and fourth stream send 20,000 packets.

Figure 6 presents the congestion window for the first TCP flow depending on the AQM mechanism. The

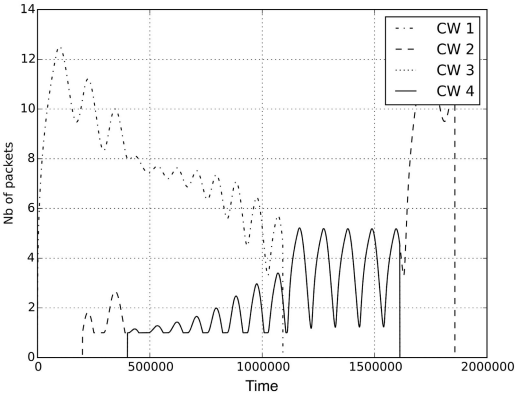

Fig. 7. TCP congestion window evolution: 4 TCP stream cooperation, RED.

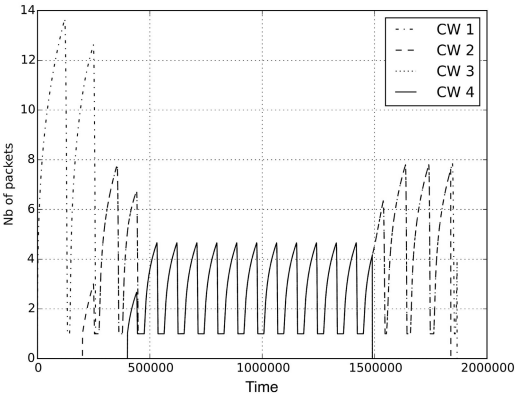

Fig. 8. TCP congestion window evolution: 4 TCP stream cooperation, PI.

changes in the first phase are similar to the ones presented in Eqn. (5). In the next phase the second TCP flow reduces the available bandwidth and the congestion window. This reduction is continued when the next two streams appear. We see how AQM effects TCP behavior and the speed of data transmission. First the transmission with the RED algorithm is ended—Fig. 7 explains it. The first stream in the first phase when it is alone accelerates its transmission and then appropriates the channel until it sends all data. In the case of the PI$^{\alpha}$ controller (Fig. 8) the streams are treated fairly. The congestion windows of the cooperating streams fluctuate identically. Figures 7 and 8 show the TCP streams cooperation. The third and fourth streams begin transmission at the same time and transmit the same number of packets. Therefore, the evolution of the window for these streams is identical.

4. PI$^\alpha$ controller under self-similar traffic: An open loop discrete-event simulation model

Over the last two decades, long-range dependence (LRD), self-similarity and heavy-tailed distributions have dominated Internet traffic analysis. Extensive measurements have revealed these phenomena in network traffic (Crovella and Bestavros, 1997; Karagiannis *et al.*, 2004; Domańska *et al.*, 2008; 2014b; 2015). These features have great impact on network performance (Domańska and Domański, 2005; Domański *et al.*, 2008; Domańska *et al.*, 2012). They enlarge mean queue lengths at buffers and increase the packet loss probability, thereby reducing the quality of the services provided by a network (Stallings, 1998).

A continuous time process $Y(t)$ is exactly self-similar with the Hurst parameter H if (Gong *et al.*, 2005)

$$Y(t) \stackrel{d}{=} a^{-H} Y(at)$$

for $t \geq 0$, $a \geq 0$ and $0 < H < 1$. The above equality is in the sense of finite dimensional distributions and the Hurst parameter expresses the degree of self-similarity (Bhattacharjee and Nandi, 2010). The process $Y(t)$ may be non-stationary (Nogueira *et al.*, 2011).

In the case of network traffic one usually has to deal with time series, rather than a continuous process. In that context the above definition can be summarized as follows. Let $X(t)$ be a stationary sequence representing an increment process (e.g., in bytes/second). The corresponding aggregated sequence having the level of aggregation m,

$$X^{(m)}(k) = \frac{1}{m} \sum_{i=1}^{m} X((k-1)m + i), \quad k = 1, 2, \ldots,$$

is obtained by averaging $X(t)$ over non-overlapping blocks of length m. The following condition is satisfied for a self-similar process:

$$X \stackrel{d}{=} m^{1-H} X^{(m)}$$

for all integers m. A stationary sequence X is second-order self-similar if $m^{1-H} X^{(m)}$ has the same variance and auto-correlation as X for all m. A stationary sequence X is asymptotically second-order self-similar if $m^{1-H} X^{(m)}$ has the same variance and auto-correlation as X as $m \to \infty$.

Asymptotically second-order self-similar processes are also called long-range dependent processes, and this is a property exhibited by network traffic (Gong *et al.*, 2005).

To take into account the self-similar and LRD characteristics of network traffic, we used in the experiments the fGn (fractional Gaussian noise) to simulate network traffic. The Hurst parameter was varied from 0.5 to 0.9.

Fractional Gaussian noise has been proposed as a model (Mandelbrot and Ness, 1968) for long-range dependence in a variety of hydrological and geophysical time series. Nowadays, fGn is one of the most commonly used self-similar processes in network performance evaluation (Lopez-Ardao *et al.*, 2000). Let $B_h(t)$ be a fractional Brownian motion process. Then the sequence of increments

$$X(t) = B_h(t) - B_h(t-1)$$

is an exactly self-similar stationary Gaussian process with zero mean, referred to as an fGn process.

The autocorrelation function of the fGn process is given by (Karagiannis *et al.*, 2004)

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

whose existence is the sufficient condition for second-order self-similarity. The fGn process is the only stationary Gaussian process that is exactly self-similar (Samorodnitsky and Taqqu, 1994).

For $0.5 < H < 1$, the auto-correlation decays hyperbolically (Cox, 1984):

$$\rho(k) \sim H(2H-1)k^{2H-2},$$

so the process exhibits long-range dependence.

The spectral density of the fGn process is given by (Lopez-Ardao *et al.*, 2000)

$$f(\lambda) = c|e^{J\lambda} - 1|^2 \sum_{i=-\infty}^{\infty} |2\pi i + \lambda|^{2H-1},$$

where $\lambda \in [-\pi, \pi]$, $0.5 < H < 1$ and c is a normalization constant such that $\int_{-\pi}^{\pi} f(\lambda)\,\mathrm{d}\lambda = \mathrm{Var}(X)$.

An important problem is synthetic generation of sample paths (traces) of self-similar processes (Lopez-Ardao *et al.*, 2000). Here we use a fast algorithm for generating approximate sample paths for an fGn process, first introduced by Paxson (1997). The simulations were done using the Simpy Python simulation packet.

Table 2. Obtained results for the FIFO queue.

Hurst	Avg. queue length	Dropped packets
0.50	299.0801	249621
0.60	298.9145	249740
0.70	298.1436	249829
0.80	296.9191	250367
0.90	248.2070	256447

Figures 9–12 present unstable behavior of the queue depending on the self-similarity factor of packet sources.

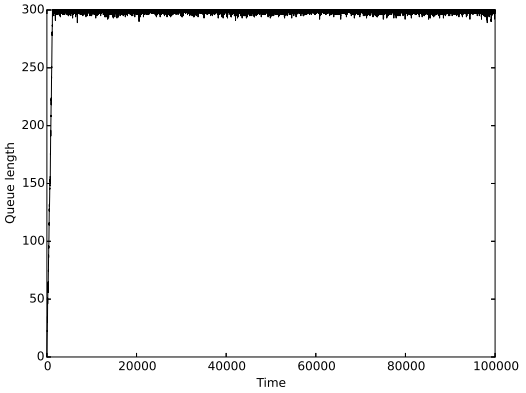

Fig. 9. Queue length: FIFO queue, $\alpha = 0.5, H = 0.50$, fGn source.

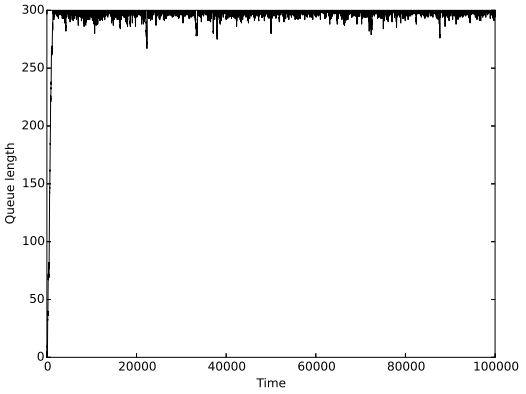

Fig. 10. Queue length: FIFO queue, $\alpha = 0.5, H = 0.70$, fGn source.

The stormy characteristic of network traffic causes extreme fluctuations in queue occupancy. Detailed results are presented Table 2. The increasing values of the Hurst parameter make the number of lost packets grow and decrease the average queue length. Similar correlations were obtained during simulations for the RED queue (Figs. 13–15). The parameters of the RED queue were selected as follows: $w = (0.02, 0.007), \mathrm{Min}_{\mathrm{th}} = 100, \mathrm{Min}_{\mathrm{th}} = 200, \mathrm{Max}_{\mathrm{th}} = 0.1$, maximum queue length$= 300$. Tables 3 and 4 present average queue length and packets loss obtained during the simulations. The column RED1 presents the number of packets dropped at the queue length shorter then $\mathrm{Max}_{\mathrm{th}}$. The column RED2 shows the number of packets dropped due to exceeding the parameter $\mathrm{Max}_{\mathrm{th}}$. There were no packets dropped due to exceeding the buffer size.

The absence of losses caused by exceeding the buffer size and the low average queue size in combination with a large number of losses caused by RED for $H = 0.90$

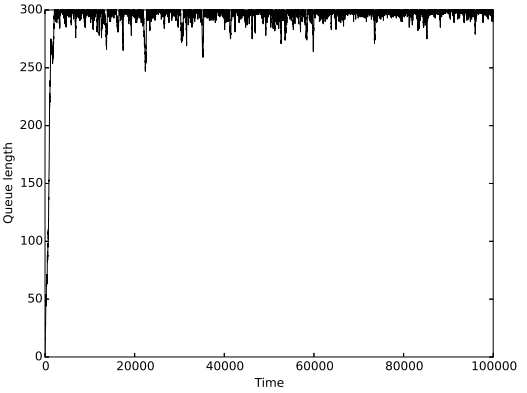

Fig. 11. Queue length: FIFO queue, $\alpha = 0.5, H = 0.80$, fGn source.

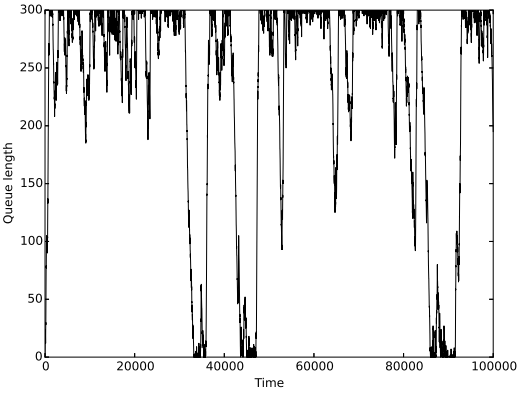

Fig. 12. Queue length: FIFO queue, $\alpha = 0.5, H = 0.90$, fGn source.

shows incorrect operation of the RED mechanism in the case of LRD traffic. The RED mechanism does not work correctly in the case of large fluctuations in traffic. Changing the w parameter and increasing the impact of historical data does not bring any positive effect.

Table 3. Obtained results for the RED queue $w = 0.007$.

		Packets dropped by	
Hurst	Avg. queue length	RED1	RED2
0.50	199.8494	27743	222034
0.60	199.5653	27200	222743
0.70	198.4269	27028	223165
0.80	196.6820	26562	223607
0.90	158.6728	19184	242195

Tables 5, 6, 7 present results obtained during testing the PI$^{\alpha}$ controller as the AQM mechanism. As the desired setpoint, 100 packets buffer occupancy (the same value as for the $\mathrm{Min}_{\mathrm{th}}$ RED parameter) was set. These results show how important controller parameters are for proper queue management, and display three completely

Table 4. Obtained results for the RED queue $w = 0.02$.

		Packet dropped by	
Hurst	Avg. queue length	RED1	RED2
0.50	199.8232	27461	222108
0.60	199.5370	27505	221995
0.70	198.5152	27038	222934
0.80	196.9210	26534	223337
0.90	158.2300	19074	240355

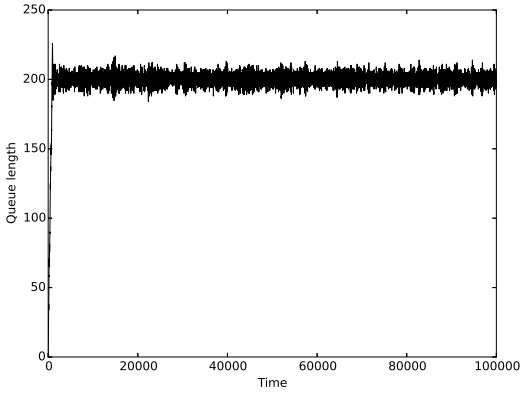

Fig. 13. Queue length: RED queue, $\alpha = 0.5, \mu = 0.25, H = 0.50, w = 0.02$, fGn source.

different types of controller behavior.

Table 5 presents ideal queue behavior. The number of dropped packets increases with the increase in the Hurst parameter. For the RED mechanism, we observe a simultaneous decrease in the average queue length. The PI controller allows an increase in the buffer occupancy. Figure 16 additionally shows the advantages of the PI$^\alpha$ mechanism. After periods of traffic stagnation (queue occupancy reaches minimal values), the management mechanism allows temporarily for an increased buffer occupancy (much larger than the desired point). This situation is unprecedented for traditional AQM mechanism.

Table 6 shows an intermediate situation. Albeit packet losses are similar, the lower queue length shows the mechanism overreaction. Figure 17 exactly shows controller behavior. The controller for such selected parameters perfectly stabilizes the queue around the desired point, but does not respond at all to the stormy traffic character.

The situation of a total failure of the mechanism is presented in Table 7. Most packets are removed because of exceeding the maximum queue size. Poorly chosen parameters make the mechanism completely useless. Figure 18 shows a total instability of queue behavior.

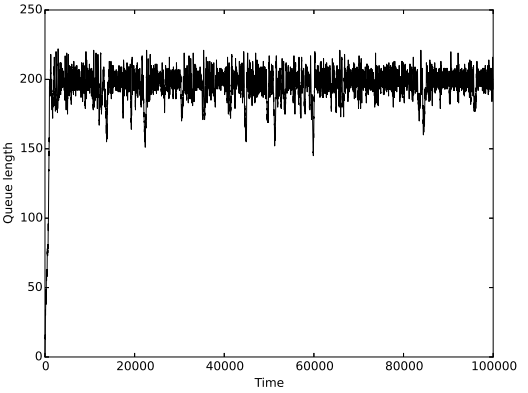

Fig. 14. Queue length: RED queue, $\alpha = 0.5, \mu = 0.25, H = 0.80, w = 0.02$, fGn source.

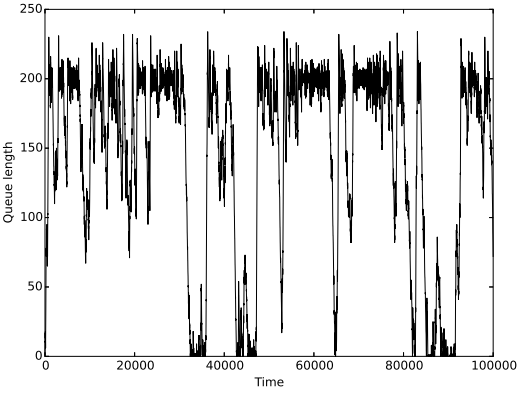

Fig. 15. Queue length: RED queue, $\alpha = 0.5, \mu = 0.25, H = 0.90, w = 0.02$, fGn source.

5. Conclusions

The article presents an evaluation of the fractional order PI$^\alpha$ controller used as an active queue management mechanism. Its quality highly depends on proper selection of parameters. Three sets of parameter were tested, and they determine completely different behavior of the controller. The three selected controllers have the same proportional and integral terms and differ only by the integral non-integer fractional order to study its influence on the controller performance.

Controllers behavior was also compared with RED, a well-known active queue management mechanism. The performance of the PI$^\alpha$ and RED controllers was investigated with the use of two methods: fluid flow approximation (closed loop control) and simulation (open loop scenario). In the open loop scenario, Internet traffic is considered the sum of a large number of streams, and the influence of the AQM mechanism on a single TCP stream is passed over. This influence can be assessed using fluid flow analysis. Both the approaches confirm the advantage

Table 5. Obtained results for the PI$^\alpha$ queue: $K_p = 0.00115$, $K_i = 0.0011$, $\alpha = -1.0$.

Hurst	Avg. queue length	Dropped
0.50	106.4579	249685
0.60	108.59444	249638
0.70	113.3477	249758
0.80	119.2418	249792
0.90	123.6004	263439

Table 6. Obtained results for the PI$^\alpha$ queue: $K_p = 0.00115$, $K_i = 0.0011$, $\alpha = -1.2$.

Hurst	Avg. queue length	Dropped packets
0.50	100.2944	250357
0.60	99.9524	250092
0.70	99.0882	249943
0.80	97.2899	249740
0.90	75.7385	265413

Table 7. Obtained results for the PI$^\alpha$ queue: $K_p = 0.00115$, $K_i = 0.0011$, $\alpha = -0.8$.

		Packet dropped by	
Hurst	Avg. queue length	PI^α	Queue
0.50	297.8812	96916	152562
0.60	297.3174	97317	151756
0.70	295.0205	95333	154864
0.80	291.0318	93749	155355
0.90	234.3215	71571	185790

of the PI$^\alpha$ controller over the RED mechanism. Open loop simulations display greater abilities of the PI$^\alpha$ controller to stabilize the queue. With a comparable number of losses, the PI$^\alpha$ controller stabilized the queue length at a lower level. An interesting property is obtained for the controller with fractional order $\alpha = 1$. After periods of low traffic, the controller temporarily allows higher queue occupancy. This is impossible for the standard RED mechanism.

Fluid flow analysis allows a less detailed study of the PI$^\alpha$ controller impact on the router buffer; however, it demonstrates a fairer treatment of TCP streams by the PI$^\alpha$ controller.

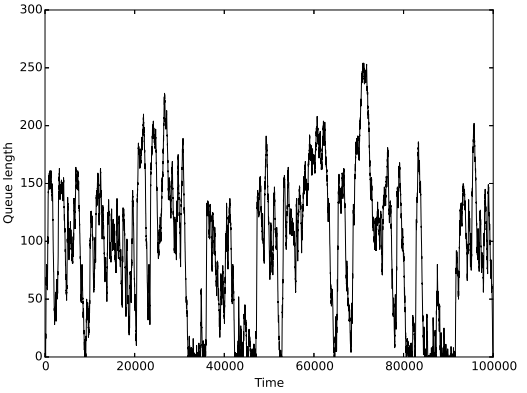

Fig. 16. Queue length: PI$^\alpha$ queue, $K_p = 0.00115$, $K_i = 0.0011$, $\alpha = -1.0$, $H = 0.90$.

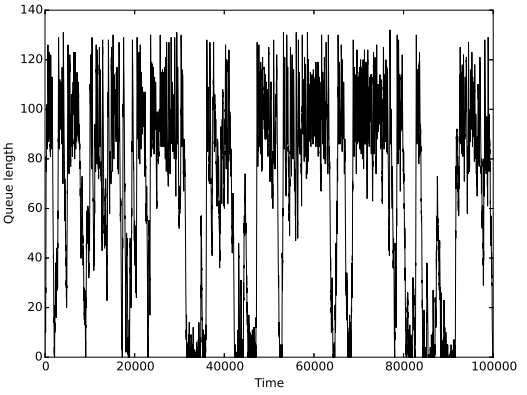

Fig. 17. Queue length: PI$^\alpha$ queue, $K_p = 0.00115$, $K_i = 0.0011$, $\alpha = -1.2$, $H = 0.90$.

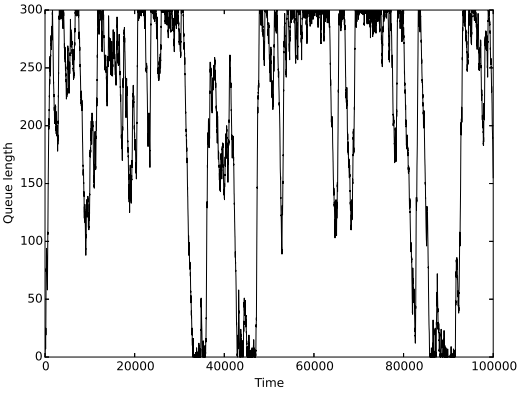

Fig. 18. Queue length: PI$^\alpha$ queue, $K_p = 0.00115$, $K_i = 0.0011$, $\alpha = -0.8$, $H = 0.90$.

References

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

Bhattacharjee, A. and Nandi, S. (2010). Statistical analysis of network traffic inter-arrival, *Proceedings of the 12th International Conference on Advanced Communication Technology, Gangwon-Do, South Korea*, pp. 1052–1057.

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, *Internet Performance Recommendations*, Network Working Group.

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.

Chen, Q. and Yang, Q. (2007). Robust controller design for AQM router, *IEEE Transactions on Automatic Control* **52**(5): 938–943.

Chen, Y., Petras, I. and Xue, D. (2009). Fractional order control—a tutorial, *American Control Conference, St. Louis, MO, USA*, pp. 1397–1411.

Ciesielski, J.L. (2006). A numerical method for solution of ordinary differential equations of fractional order, *in* R. Wyrzykowski (Eds.) *Parallel Processing and Applied Mathematics*, Lecture Notes in Computer Science, Vol. 2328, Springer, Berlin/Heidelberg, pp. 695–702.

Cox, D. (1984). Long-range dependance: A review, *in* W. Palma (Ed.), *Statistics: An Appraisal*, Iowa State University Press, Ames, IO, pp. 55–74.

Crovella, M. and Bestavros, A. (1997). Self-similarity in world wide web traffic: Evidence and possible causes, *IEEE/ACM Transactions on Networking* **5**(6): 835–846.

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 *et al.* (Eds.), *Advanced Problems of Internet Technologies*, WSB, Dąbrowa Górnicza, pp. 63–73.

Domańska, J., Domański, A., Augustyn, D.R. and Klamka, J. (2014a). A RED modified weighted moving average for soft real-time application, *International Journal of Applied Mathematics and Computer Science* **24**(3): 697–707, DOI:10.2478/amcs-2014-0051.

Domańska, J., Domański, A. and Czachórski, T. (2007). The drop-from-front strategy in AQM, *in* Y. Koucheyavy *et al.* (Eds.), *Next Generation Teletraffic and Wired/Wireless Advanced Networking*, Lecture Notes in Computer Science, Vol. 4712, Springer, Berlin/Heidelberg, pp. 61–72.

Domańska, J., Domański, A. and Czachórski, T. (2008). Implementation of modified AQM mechanisms in IP routers, *Journal of Communications Software and Systems* **4**(1): 61–70.

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

Domańska, J., Domański, A. and Czachórski, T. (2014b). A few investigation of long-range dependence in network traffic, *in* T. Czachórski *et al.* (Eds.), *Information Science and Systems 2014*, Springer International Publishing, Cham, pp. 137–144.

Domańska, J., Domański, A. and Czachórski, T. (2015). Estimating the intensity of long-range dependence in real and synthetic traffic traces, *in* P. Gaj *et al.* (Eds.), *Computer Networks*, Communications in Computer and Information Science, Vol. 522, Springer International Publishing, Cham, pp. 11–22.

Domański, A., Domańska, J. and Czachórski, T. (2008). The impact of self-similarity on traffic shaping in wireless LAN, *in* Y. Koucheryavy *et al.* (Eds.), *Next Generation Teletraffic and Wired/Wireless Advanced Networking*, Lecture Notes in Computer Science, Vol. 5174, Springer, Berlin/Heidelberg, pp. 166–168.

Domański, A., Domańska, J. and Czachórski, T. (2012). Comparison of AQM control systems with the use of fluid flow approximation, *in* P. Gaj *et al.* (Eds.), *Computer Networks*, Communications in Computer and Information Science, Vol. 291, Springer-Verlag, Berlin/Heidelberg, pp. 82–90.

Domańska, J., Domański, A., Czachórski, T. and Klamka, J. (2014c). Fluid flow approximation of time-limited TCP/UDP/XCP streams, *Bulletin of the Polish Academy of Sciences: Technical Sciences* **62**(2): 217–225.

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

Gong, W.-B., Liu, Y., Misra, V. and Towsley, D. (2005). Self-similarity and long range dependence on the internet: A second look at the evidence, origins and implications, *Computer Networks* **48**: 377–399.

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

Hollot, C., Misra, V. and Towsley, D. (2001a). A control theoretic analysis of RED, *IEEE/INFOCOM 2001, Anchorage, AK, USA*, pp. 1510–1519.

Hollot, C., Misra, V., Towsley, D. and Gong, W. (2001b). On designing improved controllers for AQM routers supporting TCP flows, *IEEE/INFOCOM 2001, Anchorage, AK, USA*, pp. 1726–1734.

Hollot, C., Misra, V., Towsley, D. and Gong, W. (2002). Analysis and design of controllers for AQM routers supporting TCP flows, *IEEE Transactions on Automatic Control* **47**(6): 945–959.

Karagiannis, T., Molle, M. and Faloutsos, M. (2004). Long-range dependence: Ten years of internet traffic modeling, *IEEE Internet Computing* **8**(5): 57–64.

Kiddle, C., Simmonds, R., Williamson, C. and Unger, B. (2003). Hybrid packet/fluid flow network simulation, *17th Workshop on Parallel and Distributed Simulation, San Diego, CA, USA*, pp. 143–152.

Krajewski, W. and Viaro, U. (2014). On robust fractional order PI controller for TCP packet flow, *BOS Conference: Systems and Operational Research, Warsaw, Poland*, pp. 493–505.

Lopez-Ardao, J., Lopez-Garcia, C., Suarez-Gonzalez, A., Fernandez-Veiga, M. and Rodriguez-Rubio, R. (2000). On the use of self-similar processes in network simulation, *ACM Transactions on Modeling and Computer Simulation* **10**(2): 125–151.

Luo, Y. and Chen, Y. (2009). Fractional-order proportional derivative controller for a class of fractional order systems, *Automatica* **45**(10): 2446–2450.

Mandelbrot, B. and Ness, J. (1968). Fractional Brownian motions, fractional noises and applications, *SIAM Review* **10**(4): 422–437.

Manfredi, S., Bernardo, M. and Garofalo, F. (2009). Design, validation and experimental testing of a robust AQM control, *Control Engineering Practice* **17**(3): 394–407.

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

Melchor-Aquilar, D. and Castillo-Tores, V. (2007). Stability analysis of proportional-integral AQM controllers supporting TCP flows, *Computacion y Sistemas* **10**(1): 401–414.

Melchor-Aquilar, D. and Niculescu, S. (2009). Computing non-fragile PI controllers for delay models of TCP/AQM networks, *International Journal of Control* **82**(12): 2249–2259.

Michiels, W., Melchor-Aquilar, D. and Niculescu, S. (2006). Stability analysis of some classes of TCP/AQM networks, *International Journal of Control* **79**(9): 1136–1144.

Miller, K. and Ross, B. (1993). *An Introduction to the Fractional Calculus and Fractional Differential Equations*, Wiley, New York, NY.

Misra, V., Gong, W. and Towsley, D. (2000). Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED, *Proceedings of ACM/SIGCOMM, Stockholm, Sweden*, pp. 151–160.

Nogueira, A., Salvador, P., Valadas, R. and Pacheco, A. (2011). Markovian modelling of internet traffic, *in* D. D. Kouvatsos (Ed.), *Network Performance Engineering*, Lecture Notes in Computer Science, Vol. 5233, Springer, Heidelberg, pp. 98–124.

Nycz, M., Nycz, T. and Czachorski, T. (2015). Modelling dynamics of TCP flows in very large network topologies, *30th International Symposium on Computer and Information Science, London, UK*, pp. 251–259.

Paxson, V. (1997). Fast, approximate synthesis of fractional Gaussian noise for generating self-similar network traffic, *ACM SIGCOMM Computer Communication Review* **27**(5): 5–18.

Podlubny, I. (1999a). *Fractional Differential Equations*, Academic Press, San Diego, CA.

Podlubny, I. (1999b). Fractional order systems and $PI^{\lambda}d^{\mu}$ controllers, *IEEE Transactions on Automatic Control* **44**(1): 208–214.

Quet, P. and Ozbay, H. (2004). On the design of AQM supporting TCP flows using robust control theory, *IEEE Transactions on Automatic Control* **49**(6): 1031–1036.

Samorodnitsky, G. and Taqqu, M. (1994). *Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance*, Chapman and Hall, New York, NY.

Stallings, W. (1998). *High-Speed Networks: TCP/IP and ATM Design Principles*, Prentice-Hall, New York, NY.

Tan, L., Zhang, W., Peng, G. and Chen, G. (2006). Stability of TCP/RED systems in AQM routers, *IEEE Transactions on Automatic Control* **51**(8): 1393–1398.

Unal, H., Melchor-Aguilar, D., Ustebay, D., Niculescu, S.-I. and Ozbay, H. (2013). Comparison of PI controllers designed for the delay model of TCP/AQM, *Computer Communications* **36**: 1225–1234.

Ustebay, D., and Ozbay, H. (2007). Switching resilient PI controllers for active queue management of TCP flows, *Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control, London, UK*, pp. 574–578.

Wang, L., Li, Z., Chen, Y.-P. and Xue, K. (2005). Fluid-based stability analysis of mixed TCP and UDP traffic under RED, *10th IEEE International Conference on Engineering of Complex Computer Systems, Shanghai, China*, pp. 341–348.

Yung, T.K., Martin, J., Takai, M., and Bagrodia, R. (2001). Integration of fluid-based analytical model with packet-level simulation for analysis of computer networks, *Proceedings of SPIE* **4523**: 130–143.

Zheng, F. and Nelson, J. (2009). An H_{∞} approach to the controller design of the AQM routers supporting TCP flows, *Automatica* **45**(3): 757–763.

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 networks domain is congestion control in packet networks.

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 modeling methods for computer networks.

Tadeusz Czachórski, Ph.D., Prof., the head of the Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences. His main areas of interest are mathematical and numerical methods for modeling and evaluation of computer networks.

Jerzy Klamka, Ph.D., Prof., a 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 areas of research are controllability and observability of linear and nonlinear dynamical systems, and mathematical foundations of quantum computations. He is an author of monographs and numerous papers published in international journals.

Received: 14 March 2016
Revised: 30 May 2016
Accepted: 8 July 2016