BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 62, No. 2, 2014
DOI: 10.2478/bpasts-2014-0021

Dedicated to Professor J. Klamka
on the occasion of his 70th birthday

Fluid flow approximation of time-limited TCP/UDP/XCP streams

J. DOMAŃSKA[1]*, A. DOMAŃSKI[2], T. CZACHÓRSKI[1], and J. KLAMKA[1]

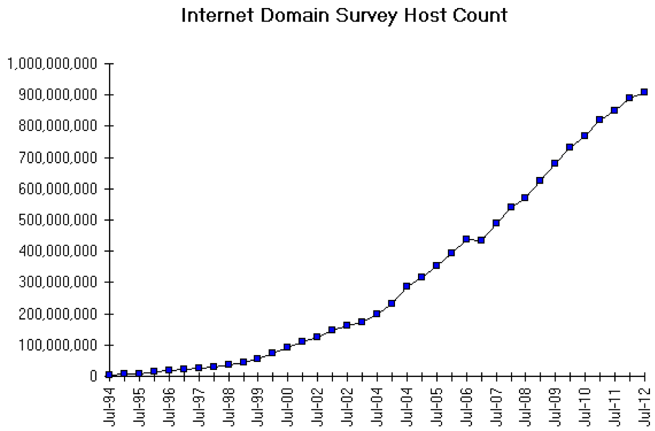
[1] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, 5 Baltycka St., 44–100 Gliwice, Poland
[2] Institute of Informatics, Silesian Technical University, 16 Akademicka St., 44–100 Gliwice, Poland

Abstract. This article presents the use of fluid flow approximation to model interactions between a set of TCP, UDP and XCP flows in the environment of IP routers using AQM (Active Queue Management) algorithms to control traffic congestion. In contrast to other works, independent UDP and TCP streams are considered and the model allows to start and end data transmissions in TCP, UDP and XCP streams at any time moment. It incorporates several Active Queue Management mechanisms: RED, NLRED, CHOKe.

Key words: fluid flow modeling, Internet, TCP/IP, UDP, XCP, active queue management, non-linear RED.

1. Introduction

Queuing models have been used to evaluate performace of computer networks since their early times, but now – because of an exponential increase of Internet structure, see Fig. 1 [1], new solutions, adapted to large-scale networks are proposed, [2–4]. These models should be able not only to consider large topologies but also should perform transient analysis to see time-dependent behaviour of flows and queues. Only this way we may capture the dynamics of traffic control algorithms and practically evaluate their utility.



Source: Internet Systems Consortium (www.isc.org)

Fig. 1. Survey data collected and published quarterly by Internet Systems Consortium

Basically, apart from discrete time simulation which is very time-consuming in case of intransient-state analysis, we dispose analytical and numerical approaches such as Markov chains models, diffusion approximation and fluid flow approximation. Markov models and simulation models represent a network behaviour on packet level: an event corresponds here to an arrival or departure of a packet. Such models may be detailed, flexible and accurate but the number of events and states to consider is prohibitive, so they are only used in analysis of small configurations and small number of transmitted flows.

Diffusion and fluid flow methods consider continuous flows; events correspond to the changes of flows. In case of diffusion approximation, the flows are defined by their mean and variance, in the fluid flow approach it is even simpler, only time-dependant means of flows matter. In diffusion approximation the changes of queues are given by second order partial equations, in fluid flow approximations they are modelled by first order ordinary differential equations. This simplicity of fluid flow approach makes it easy to consider large topologies: models having millions of nodes (thousands of nodes in case of the diffusion appoximation) are already reported [5]. It also allows to include easily in the model the rules implemented by TCP (Transmission Control Protocol) to react to congestion signalled by packet losses.

The Internet is closely associated with the use of the TCP/IP protocol, its transmission delays and congestions are generally controlled by TCP mechanisms. Transport layer protocols (such as TCP or SCTP, i.e. scaled TCP) change the sending rate depending on network congestion. The active queue management mechanisms, such as AQM/RED (Random Early Detection), are more sophisticated congestion control mechanisms [6]. UDP (User Datagram Protocol) has no congestion control mechanism and XCP (eXplicit Control Protocol) [7] is an router-assisted congestion control mechanism. XCP is a sort of window-based flow control mechanisms. XCP routers inform the senders about the degree of congestion at the bottleneck. With such feedback, an XCP source host can respond to congestion status of the network [8].

Fluid flow modeling techniques were proposed to model the behavior of Internet traffic [9–11], they are especially suitable for TCP traffic modeling [12]. Existing fluid flow network simulators can mainly simulate persistent TCP flows [13]. Although short-living TCP flows, UDP flows, and XCP have been independently modeled using fluid flow modeling techniques [8, 14], fluid network models with heterogenous flows have been very rarely investigated [13, 15].

---

*e-mail: joanna@iitis.pl

AQM mechanisms support the end-to-end congestion control mechanism of TCP. In the last years a lot of AQM algorithms have been proposed and studied [16–18]. Most of these studies were performed in simulation environment. The analytical studies of AQM models were performed in open-loop scenario because of the difficulty in analyzing AQM mathematically [19–21].

In this paper we analyze the transient behavior of the several AQM mechanisms by extending our earlier open-loop models presented in [20–22]. We use the fluid flow approximation method to show the behavior of the AQM mechanisms for various types of connections. We are able to make some comments about RED that were not possible to notice in open-loop scenario.

This paper is organized as follows. Subsection 1.1 briefly describes the works thematically related to this article. Section 2 gives a brief description of the AQM/RED mechanism and introduces two modification of the RED mechanism that are used in this article. Section 3 describes the fluid flow model of TCP/UDP/XCP behavior. Section 4 gives some numerical results. Section 5 concludes this article and discusses future works.

1.1. Related works. The capabilities of simulators did not keep up with the increase of the size of the Internet [23]. Fluid models are reported to be an efficient and accurate tool in performance analysis of large IP networks [9]. The special interest of using fluid flow modeling comes from the fact that discrete event simulation is appropriate only for small networks (10-100 of network nodes and 100–1000 IP flows). And even then is extremely time-consuming in case of transient state analysis technology, refers only to TCP flows over a single link [24]. The article [23] started with the basic fluid model developed in [24] and made considerable improvements and enhancements to it. The idea was to abstract the behavior of IP networks into analytical models and to solve this models numerically to obtain performance metrics that are close to those of the original networks.

Fluid models have been proposed as an alternative to a number of discrete event-driven simulators, which have been actively examined in recent years. In [25] the model proposed in [24] for a non-DiffServ-compliant network is extended to a network supporting assured forwarding per-hop behavior with two drop precedence by means of a generic AQM technique. This approach uses the fluid-flow approximation for a traffic modeling and queue behavior modeling in order to keep the model complexity low for each buffer size or network topology. However, in [25] only a steady-state analysis is performed and the model proposed in it did not consider data-limited TCP flows. The paper [14] proposed the fluid-flow model of a DiffServ network loaded with TCP flows, where RED and WRED techniques are implemented.

There is also a hybrid approach combining fluid flow and packet level model, proposed in [26]. Such models aim to combine the advantages of both (fluid and packet) approaches. The hybrid models are able to handle packets as well as the flows. The hybrid simulation has been also presented in [23]. This simulation has been characterized by the high interaction between packet network segments and fluid network segments. The fluid flow model proposed in [23] has been integrated into NS simulator [27] by constructing fluid link and fluid network objects. The assumption of integration with packet level simulators:

- fluid model can provide delay and loss information for packets passing fluid network segments;
- if traffic from packet segments is negligible to fluid segment, fluid model can be solved independently.

Flow-level network simulators mimic behavior of every flow in a network. It follows that a flow level network simulator can simulate the flows of the large-scale network, where a huge number of packets are transmitted [28]. However, this kind of network simulator is still in a development stage [13]. The existing flow-level simulators [28] can only simulate a network with persistent TCP flows. To evaluate the performance of realistic network, a flow-level network simulator must support heterogeneous flows (flows with different protocols and traffic patterns).

The Internet is heavily dependent on the use of the TCP/IP protocol stack. Managing the inevitable delays and congestion over the networks is generally supported by AQM/RED mechanisms. Many AQM mechanisms have been proposed in many papers. The more and more sophisticated AQM mechanisms are widely employed, but their nonlinear dynamics are usually ignored. Although RED mechanism is conceptually very simple, its interaction with TCP protocol is rather complex [29].

It is still accepted assumption that Internet traffic is dominated by TCP traffic [30]. But in the current network UDP is an indispensable part of the network traffic [31]. It is associated with the popularization of the multimedia applications [32, 33]. The rise of new streaming applications (such as IPTV and new P2P protocols) will increase the use of UDP as a transport protocol. The article [15] modified the model presented in [34] and considered the impact of UDP on the stability of mixed system. In this article the proportion of UDP to whole traffic was small and the authors have made the conclusions that the results for mixed TCP and UDP traffic were similar to those of pure TCP traffic. Not many works regarding fluid flow models take into consideration the UDP traffic. In articles [15] and [33] the authors have made an assumption that each session is assumed to be either a TCP or UDP session. Although the UDP traffic was considered as an element which has an influence on the TCP traffic.

The XCP protocol introduced the new concept of separating utilization control from fairness control [35]. This protocol has high potential for effective congestion control. Both theoretical and simulation studies have shown that XCP controllers are stable, efficient, and fair [7]. There exist some analytical studies on XCP using fluid-flow approximation. In [35] stability of XCP has been analyzed. The authors assumed an identical propagation delay for all XCP streams. By extending the analytic model proposed in [35], the authors of [7] have shown that operation of XCP becomes unstable when the available bandwidth of an XCP router's output link is not fixing. Article [28] has presented the analysis of XCP stability in a network with XCP streams with different propagation delays.

The contribution of our works goes beyond a numerical implementation of the ideas presented in [23, 24]. We also model the behavior of TCP/UDP/XCP mixed traffic. Our fluid flow model is a type of switched system [36]. We extended the fluid-flow model that considered the impact of the UDP traffic on the stability of the bottleneck network. Our UDP model is based on work presented in the article [13] and in contrast to UDP model presented in [15] and [33] it is independent from TCP model. Our XCP model is based on the studies presented in [7]. Going beyond the RED [37] mechanism modeled in [24] and other AQM mechanism modeled in [23], we also incorporate CHOKe mechanism and our NLRED mechanism.

2. RED mechanism and its variations

In the last few years numerous AQM mechanisms have been studied [16–18, 38–45]. The representative AQM mechanism is the Random Early Detection (RED) mechanism [37], which randomly drops an arriving packet to avoid the global synchronization problem. Several researchers conclude that tuning of RED parameters is a very difficult task. This is why so many variants of RED have been proposed in the literature.

In *passive* queue management (with FIFO scheduling), packets coming to a buffer are rejected only if there is no space in the buffer to store them. The RED calculates the average queue size using a low-pass filter with a weighted moving average. The average queue size (avg) is compared to two thresholds, a minimum threshold (Min_{th}) and a maximum threshold (Max_{th}). If $avg < Min_{th}$ all packets are admitted, if $Min_{th} < avg < Max_{th}$ then dropping probability p is growing linearly from 0 to $p_{\max}$:

$$p = p_{\max}\frac{avg - Min_{th}}{Max_{th} - Min_{th}} \tag{1}$$

and if $avg > Max_{th}$ then all packets are dropped. The value of $p_{\max}$ has also a strong influence on the RED performance: if it is too large, the overall throughput is unnecessarily choked and if it's too small, the danger of synchronization arises; [46] recommends $p_{\max} = 0.1$. The problem of the choice of parameters is still discussed, see e.g. [47–49]. In NLRED algorithm (proposed by authors in [50, 51]) the probability of packet dropping function is defined as follows:

$$p(x, a_1, a_2, p_{\max}) = \begin{cases} 0 & \text{for } x < Min_{th} \\ \varphi_0(x) + a_1\varphi_1(x) + a_2\varphi_2(x) & \\ & \text{for } Min_{th} <= x <= Max_{th} \\ 1 & \text{for } x > Max_{th} \end{cases} \tag{2}$$

where a_1 and a_2 are undetermined parameters, x is the current queue size and basic functions are defined:

$$\varphi_0(x) = p_{\max}\frac{x - Min_{th}}{Max_{th} - Min_{th}}, \tag{3}$$

$$\varphi_1(x) = (x - Min_{th})(Max_{th} - x), \tag{4}$$

$$\varphi_2(x) = (x - Min_{th})^2(Max_{th} - x). \tag{5}$$

Set of p-function was presented on Fig. 2.

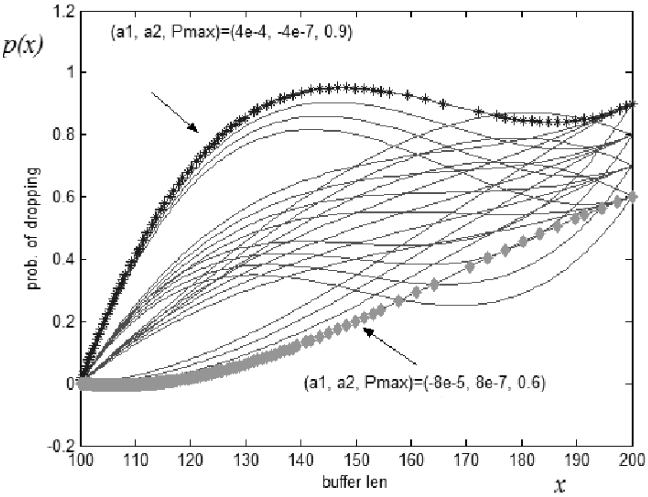


Fig. 2. Set of sample p-function

The CHOKe [52] is a stateless AQM algorithm slightly similar to RED, proposed not only to control TCP packets but also to prevent uncontrollable UDP connections to monopolize the links [34]. It uses incoming packages to punish streams with the highest demand for bandwidth.

Similarly to the RED mechanism, there are two threshold values: Min_{th} and Max_{th}. At the arrival of a new package, the new walking average queue length is calculated. If the average queue length is less than Min_{th}, the packet is placed in the buffer. When the average queue length is greater than Min_{th}, CHOKe pulls randomly one packet from the FIFO buffer ("CHOKe victim") and verifies whether it belongs to the same stream as an incoming packet.

If the both packets belong to the same stream, they are removed (this situation is called "CHOKe hit"). Otherwise, the randomly selected packet is returned to the buffer and the arrived packet is placed into the queue with probability P. This probability is calculated in the same manner as in the case of RED algorithm. The event is called "CHOKe miss".

In next section we present solution techniques that allow one to obtain the transient behavior of described above AQM mechanisms for various types of connections (mixed TCP/UDP/XCP traffic). We are able to make some comments that were not possible to notice in open-loop scenario.

3. The fluid model of TCP/UDP/XCP behavior

This section describes how to model the AQM router supporting TCP/UDP/XCP flows using fluid flow equations. Our dynamic model of TCP behavior is based on the model developed by [24, 34], which uses fluid flow and stochastic differential equation analysis. We use the fluid flow model of XCP derived in [7].

In [24] 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 have also shown how to obtain ordinary differential equations by taking expectations of the 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 model proposed in this paper expands this approach to include UDP and XCP streams. The model is given by the sets of nonlinear differential equations.

The dynamics of the TCP window for the i-th stream is approximated by:

$$\frac{dW_i(t)}{dt} = \frac{1}{R_i(t)} - \frac{W_i(t)W_i(t-R(t))}{2R_i(t-R_i(t))}p(t-R_i(t)), \quad (6)$$

where $W_i(t)$ – expected TCP sending window sizes (packets), $R_i(t) = \frac{q(t)}{C} + T_p$ – round-trip time (sec), $q(t)$ – queue length (packets), C – link capacity (packets/sec), T_p – propagation delay (sec), p – probability of packet drop.

UDP stream is a CBR stream with assumed number of packet being sent per time unit. The sending rate (output from the model $Y_i^{UDP}(t)$) of the i-th UDP stream is approximated by the following equation:

$$Y_i^{UDP}(t) = U. \quad (7)$$

The dynamics of the XCP sending rate for the i-th stream is approximated by:

$$\frac{dY_i^{XCP}(t)}{dt} = -\frac{\alpha}{d}(Y_i^{XCP}(t)(t-d) - C) - \frac{\beta}{d^2}q(t-d), \quad (8)$$

where α and β are two control parameter constants and d is the average round-trip time.

The dynamics of a queue is approximated by:

$$\frac{dq(t)}{dt} = \sum_{i=1}^{n_1} \frac{W_i(t)}{R_i(t)} + \sum_{i=1}^{n_2} \frac{Y_i^{UDP}(t)}{R_i(t)} + \sum_{i=1}^{n_3} Y_i^{XCP}(t) - C, \quad (9)$$

where n_1 – number of TCP streams, n_2 – number of UDP streams, n_3 – number of XCP streams.

The solution of the fluid model can be obtained by solving the set of differential equations defined above. The equations have been solved numerically using time stepped numerical fluid model solver written in Python. Figure 3 depicts the flowchart of our fluid model solver.

The traffic composed of TCP and UDP streams has been already considered in [6, 15]. 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: TCP stream being limited through UDP stream.

In this article, the TCP and UDP streams were treated as separate streams. According to our best knowledge, this approach was previously not considered in the literature. Although in [26] and [53] a separate UDP stream was used, but our work differs at this point that the TCP can start from various initial window sizes. Additionally, we can set up a time after which the stream (TCP, UDP or XCP) will be started. The dynamics of the queue described by the equation 9 has been influenced only by these streams that have started and have not yet sent a predetermined number of packets.

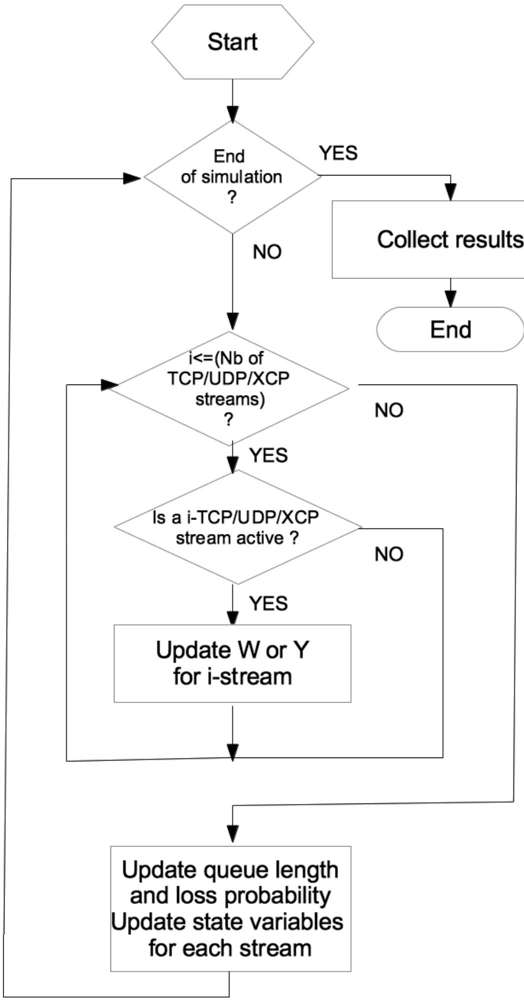


Fig. 3. Flowchart of our fluid flow model solver

In order to illustrate the operation of our TCP/UDP/XCP fluid flow model, we consider some AQM schemes (RED, NLRED and CHOKe) to address the congestion control issue.

4. Experimental results

This article presents the fluid-based analysis of the RED mechanism and its two variants (CHOKe and NLRED) supporting TCP, UDP and XCP streams. We use the numerical scheme described in Sec. 3 for obtaining the transient behavior of queue length, packet losses and TCP window dynamics.

All our computations were made with the use of PyLab (Python numeric computation environment) [54] which is a combination of Python, NumPy, SciPy, Matplotlib, and IPython. The graphs shown below present transient system behavior, the time axis is drawn in seconds.

We assume the following parameters of the AQM buffer: $Min_{th} = 10$, $Max_{th} = 15$, buffer size (measured in packets) $= 20$, weight parameter $\alpha = 0.007$ [37] and the parameters of TCP connection:

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

The XCP control parameters (α and β) were set to their recommended values: 0.4 and 0.226 [35]. We have considered n_3 XCP streams with common round trip delay of $d = 10$ ms [13].

The obtained mean queue lengths for TCP connections ($W_i = 1$, TCP sends 50000 packets) are:

- for RED $(p_{\max} = 0.1) = 9.9931$,
- for RED $(p_{\max} = 0.6) = 8.3359$,
- for NLRED $(p_{\max} = 0.6,\ a_1 = 0.00008,\ a_2 = 0.0000008) = 8.3327$,
- for NLRED $(p_{\max} = 0.9,\ a_1 = 0.0004,\ a_2 = 0.000004) = 7.9111$,
- for CHOKe=7.9866.

For 1 TCP and 1 UDP ($U = 3$) connections (two sources starts immediately, TCP ($W_i = 1$) sends 50000 packets, UDP 16000):

- for RED $(p_{\max} = 0.1) = 10.7132$,
- for NLRED $(p_{\max} = 0.9,\ a_1 = 0.0004,\ a_2 = 0.000004) = 8.4765$,
- for CHOKe=8.9812.

Figure 4 shows the queue behavior in case of one XCP flow and FIFO queue. The number of packets are seen to change periodically between limitting values that depend on the delay and values of parameters α and β.

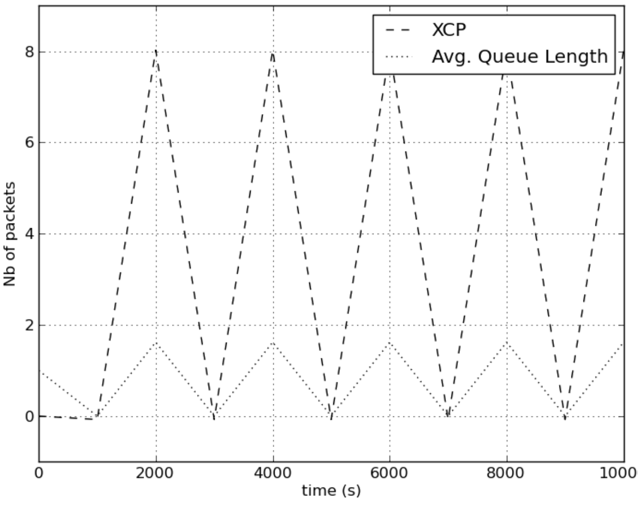


Fig. 4. FIFO queue, 1 XCP flow

Figure 5 shows the queue behavior in case of one TCP flow and FIFO queue. The size of congestion window increases until the buffer becomes full. Packets are dropped, the size of congestion window decreases causing a slow decrease of the queue length – this pattern is repeated periodically. In the last part of the graph it is possible to note that the activity of TCP ends (window size and queue occupancy decline to 0).

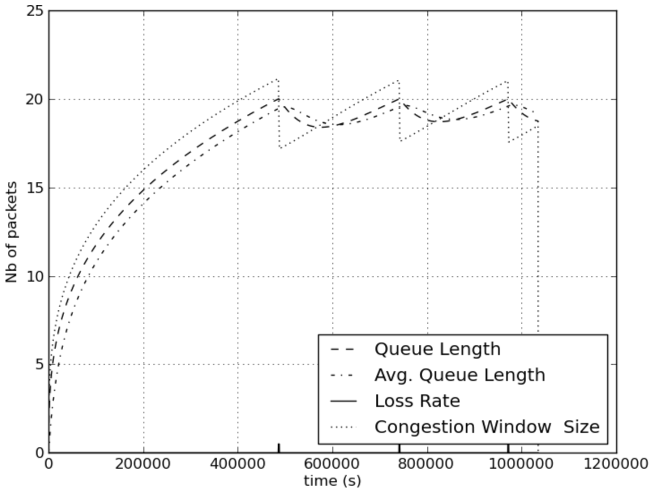


Fig. 5. FIFO queue, 1 TCP flow

Figure 6 presents the analogous situation in case of two TCP flows. The curves are similar to the previous, the only difference is that the congestion window in the case of two TCP flows never achieves such large values as in the case of one TCP flow.

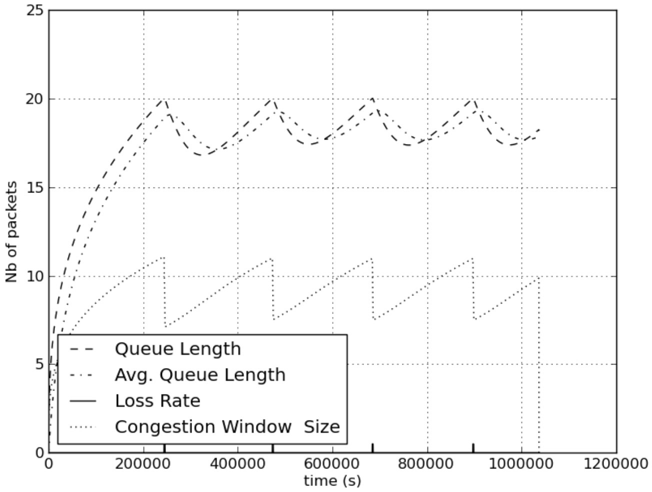


Fig. 6. FIFO queue, 2 identical TCP flows

Figures 7 and 8 present the queue evolution when RED mechanism is implemented. The loss rate is in this cases smaller. Queue length and weighted moving average queue length oscillate between their mean values.

Figure 8 shows the situation when the second TCP stream starts sending packets after a certain period of time. Introducing a second stream causes disturbances in the window evolution. The system needs a time to stabilize.

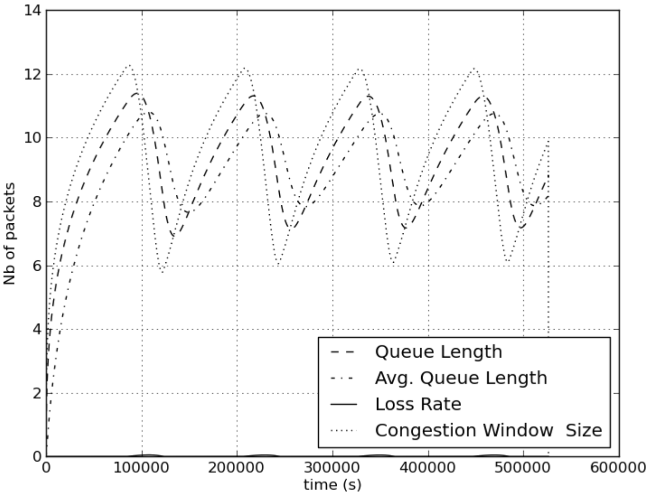


Fig. 7. RED queue, 1 TCP flow

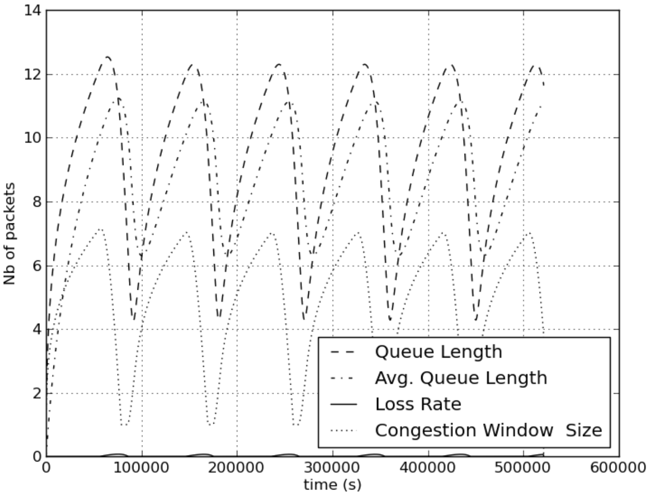


Fig. 8. RED queue, 2 identical TCP flows

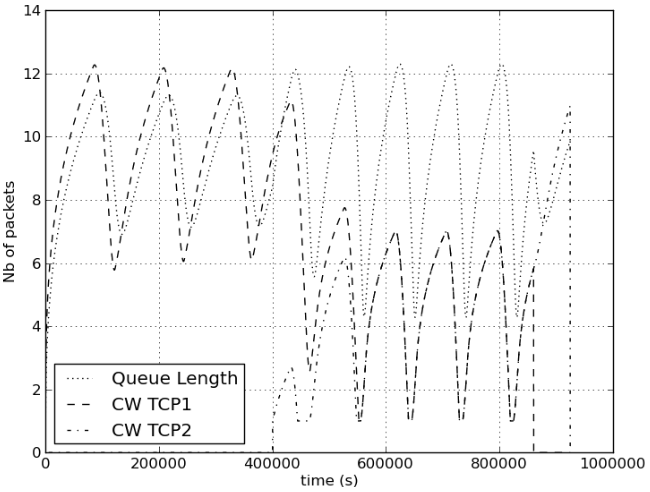


Fig. 9. RED queue, 2 identical TCP flows

Figure 10 shows the situation where fixed intensity UDP stream (3 packets per time unit) disturbs TCP window to evolve. At the moment of UDP transmission interruption, TCP window reaches the size analogous to situation presented in Fig. 7.

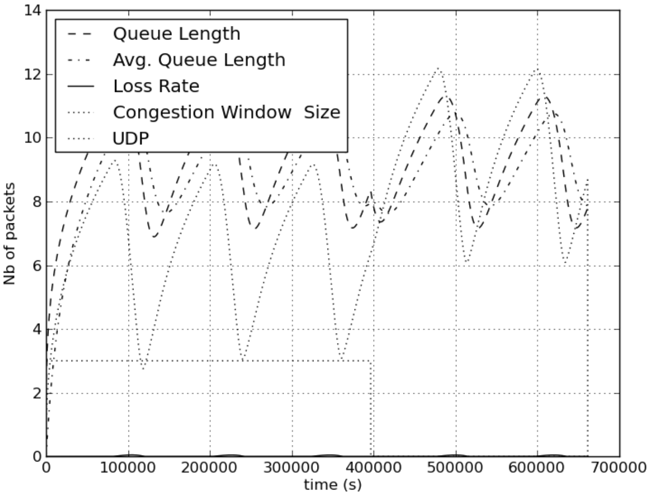


Fig. 10. RED queue, 1 TCP and 1 UDP flow

Figure 11 shows the behavior of TCP stream in a group with several UDP streams.

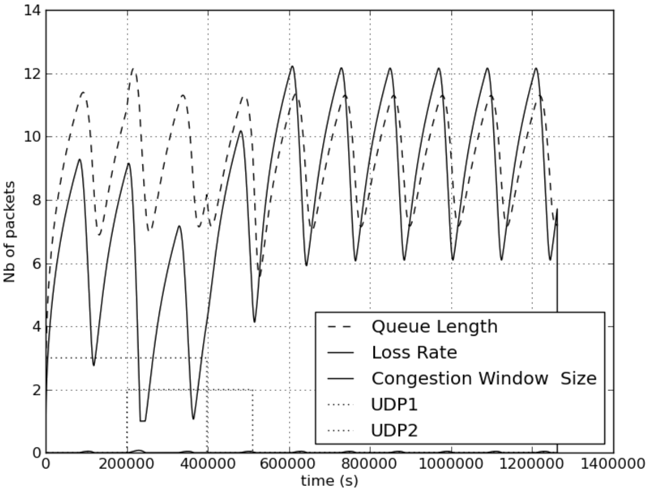


Fig. 11. RED queue, 1 TCP and 2 UDP flows

Figure 12 shows the situation where XCP stream disturbs TCP window to evolve. The problem is due to a lack of co-operation between AQM and XCP.

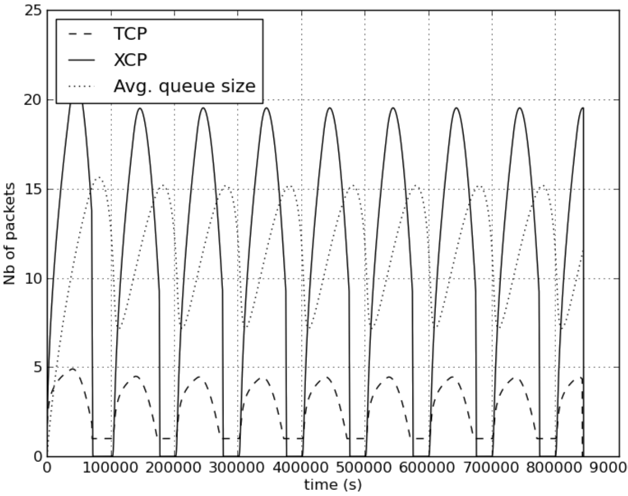


Fig. 12. RED queue, 1 TCP and 1 XCP flow

Figure 13 shows the behavior of the queue with RED and NLRED algorithm. The characteristic of the plots is the same to the RED queue. But the mean values of queue length presented earlier confirm the advantage of NLRED algorithm.

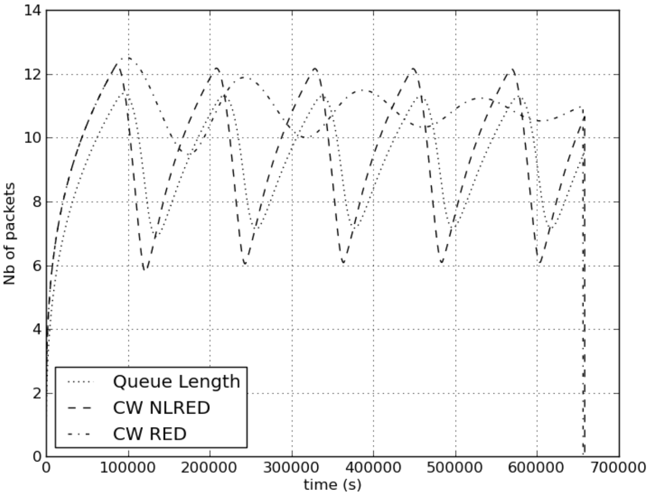


Fig. 13. TCP congestion window evolution for RED and NLRED queue

Figure 14 shows the behavior of the RED/NLRED queues for mixed TCP/UDP traffic. For such kind of traffic decreasing of congestion window causes not as slowly decreasing of queue length as for TCP traffic. This figure shows a very interesting effect. For NLRED queue (after the UDP transmission came to the end), TCP window reaches the stability quicker than while UDP transmission was still in action.

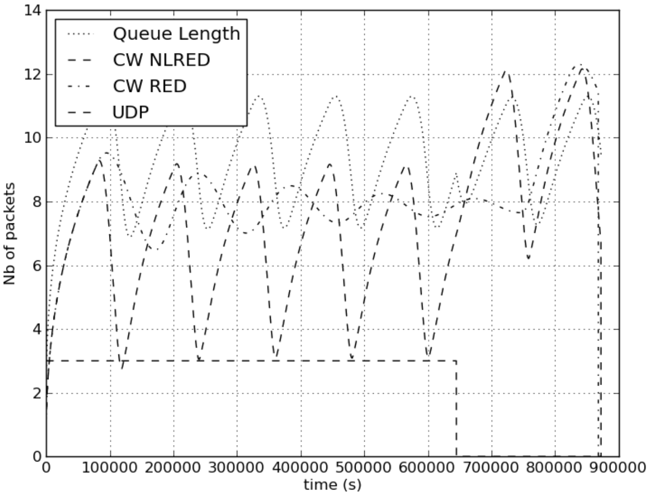


Fig. 14. TCP congestion window evolution for RED and NLRED queue, 1 TCP and 1 UDP stream

Comparing the behavior of the CHOKe algorithm with the RED algorithm (Fig. 15) one can see that the CHOKe algorithm works better in the case of aggressive (stealing most of the bandwidth) streams. When the number of streams increases, the importance of the CHOKe algorithm decreases. The probability of selecting a good victim decreases and the packets are removed by the RED mechanism. Comparing the results one can see that the differences between the obtained average queue length for CHOKe and for RED algorithms decreases when the number of streams increases (Fig. 16).

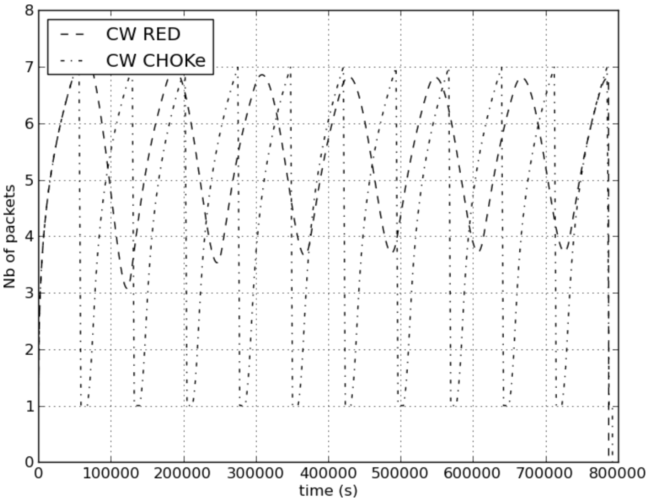


Fig. 15. TCP congestion window evolution for RED and CHOKe queue, 1 TCP stream

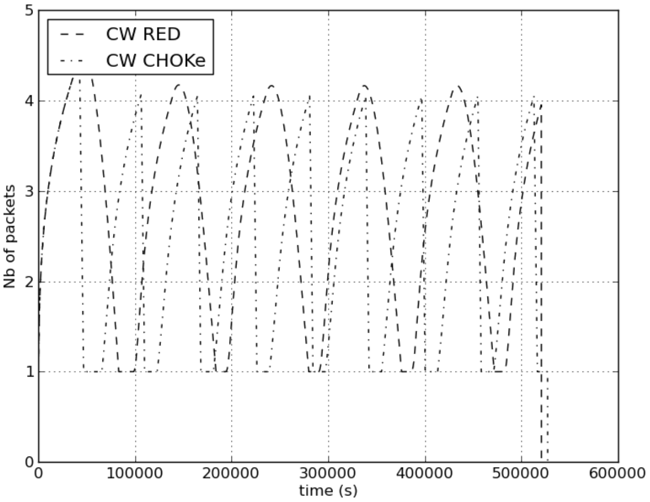


Fig. 16. TCP congestion window evolution for RED and CHOKe queue, 4 TCP streams

5. Conclusions

In this article we present a new approach for using the fluid flow approximation to model the behavior of AQM mechanisms. Compared to previous works [6, 55, 56] we have changed the way to model the UDP sources. Previously it was assumed that each UDP stream is associated with a TCP stream, affecting the dynamics of the TCP window. Here, UDP streams are treated as separate, autonomous ones. Comparing the obtained results with the results of authors' earlier works one can conclude that the results of analogues experiments (the same number of TCP and UDP streams) are similar, which confirms the accuracy of the new approach. On the other hand, the new approach is more flexible and allows gaining access into the dynamics of the TCP window in completely new situations (interaction between heterogenous streams).

The analysis presented in this article takes into account the time-limited streams. The most works in this area assume synchronization of all TCP sources. The model presented in this article allows to start TCP/UDP/XCP transmissions at any point of time. It also allows to specify the number of packets that terminates the TCP/UDP/XCP transmission. Therefore the above approach makes the observation of the dynamics of transmission possible at times when other sources start or end their transmissions.

The paper presents also the impact of AQM mechanism on the dynamics of the TCP window: we considered RED, NLRED and CHOKe algorithms. The article confirms the advantage of proposed NLRED algorithm over standard RED, previously studied in open loop scenario [21]. Our future works will focus on the integration of our model with packet level simulation, this approach was proved to be very efficient [10, 18]. In traditional packet-level discrete event simulators the time consumption increases rapidly with the number of TCP flows being simulated and the use of mixed models can increase the scalability of modeling. This article confirms the advantage of CHOKe algorithm over standard RED for aggressive streams. Using the CHOKe algorithm is insignificant in the case of a large number of streams with the similar intensity.

Acknowledgements. This research was partially financed by the Polish Ministry of Science and Higher Education under the project No. N N516479640.

REFERENCES

[1] Internet Systems Consortium homepage, http://www.isc.org (2013).

[2] D.M. Nicol, M. Liljenstam, and J. Liu, "Advanced concepts in large-scale network simulation", *Winter Simulation Conf.* 1, 153–166 (2005).

[3] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving datacenter performance and robustness with multipath TCP", *ACM SIGCOMM Computer Communication Rev.* 41 (4), 266–277 (2011).

[4] A. Domański, J. Domańska, and T. Czachórski, "The impact of self-similarity on traffic shaping in wireless LAN", *Lecture Notes in Computer Science* 5174, 156–168 (2008).

[5] T. Czachórski, M. Nycz, and T. Nycz, "Modelling transient states in queueing models of computer networks: a few practical issues", *Communications in Computer and Communication Science* 279, 58–72 (2014).

[6] A. Domański, J. Domańska, and T. Czachórski, "Comparison of AQM control systems with the use of fluid flow approximation", *Communications in Computer and Information Science* 291, 82–90 (2012).

[7] Y. Zhang and M. Ahmed, "A control theoretic analysis of XCP", *IEEE Infocom* 4, 2831–2835 (2005).

[8] Y. Sakumoto, H. Ohsaki, and M. Imase, "On XCP stability in heterogenous network", *IEEE Symp. on Computers and Communications* 1, 531–537 (2007).

[9] Y. Gu, Y. Liu, and D. Towsley, "On integrating fluid model with packet simulation", *IEEE INFOCOM* 4, 2856–2866 (2004).

[10] S. Kumar, S.-J. Park, and S. Sitharama Iyengar, "A loss-event driven scalable fluid simulation method for high-speed networks", *Computer Networks: Int. J. Computer and Telecommunications Networking* 54 (1), 112–132 (2010).

[11] A. Gkogkas, R. Roverso, and S. Haridi, "Accurate and efficient simulation of bandwidth dynamics for peer-to-peer overlay networks", *VALUETOOLS '11: 5th Int. ICST Conf. Performance Evaluation Methodologies and Tools* 1, 352–361 (2011).

[12] P.A. Shah, S. Yasmin, S. Asghar, A. Qayyum, and H.B. Hasbullah, "A fluid flow model for SCTP traffic over the internet", *Int. Conf. on Emerging Technologies (ICET)* 1, 1–6 (2012).

[13] T. Nishioka, Y. Sakumoto, H. Ohsaki, and M. Imase "Design and implementation of flow-level simulator for a network with heterogeneous flows", *Int. Symp. on Applications and the Internet* 1, 78–84 (2009).

[14] M. Barbera, A. Lombardo, G. Schembra, and C.A. Trecarichi, "Fluid flow analysis of TCP flows in a DiffServ environment", *Eur. Trans. on Telecommunications* 17 (5), 505–524 (2006).

[15] L. Wang, Z. Li, Y.-P. Chen, and K. Xue, "Fluid-based stability analysis of mixed TCP and UDP traffic under RED", *10th IEEE Int. Conf. on Engineering of Complex Computer Systems* 1, 341–348 (2005).

[16] A. Heider, "Improved congestion control for packet switched data networks and the internet", *PhD Thesis*, Electrical and Electronic Engineering, University of Canterbury, Christchurch, 2004.

[17] A. Heider and A. Nakao, "On path switching in overlay network", *Proc. Australasian Telecom. Networks and Applications Conf. (ATNAC)* 1, 355–360 (2008).

[18] P. Padhy and R. Sundaram, "Analysis and design of improved pi-pd controller for tcp aqm routers", *Proc. Int. Conf. on Power, Control and Embedded Systems (ICPCES)* 1, 1–5 (2010).

[19] C. Liu and R. Jain, "Improving explicit congestion notification with the mark-front strategy", *Computer Networks* 35 (2–3), 185–201 (2000).

[20] J. Domańska, A. Domański, and T. Czachórski, "The drop-from-front strategy in AQM", *Lecture Notes in Computer Science* 4712, 61–72 (2007).

[21] J. Domańska, D.R. Augustyn, and A. Domański, "The choice of optimal 3-rd order polynomial packet dropping function for NLRED in the presence of self-similar traffic", *Bull. Pol. Ac.: Tech.* 60 (4), 779–786 (2012).

[22] J. Domańska and A. Domański, "Adaptive RED in AQM", *Communications in Computer and Information Science* 39, 174–183 (2009).

[23] Y. Liu, F.L. Presti, V. Misra, D. Towsley, and Y. Gu, "Scalable fluid models and simulations for large-scale IP networks", *ACM Trans. on Modeling and Computer Simulation* 14 (3), 305–324 (2004).

[24] V. Misra, W.-B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED", *ACM SIGCOMM* 1, 151–160 (2000).

[25] Y. Chait, C. Hollot, V. Misra , D. Towsley, and H. Zhang, "Providing throughput differentiation for TCP flows using adaptive two color marking and multi-level AQM", *IEEE INFOCOM* 1, 23–27 (2002).

[26] C. Kiddle, R. Simmonds, C. Williamson, and B. Unger, "Hybrid packet/fluid flow net- work simulation", *Parallel and Distributed Simulation* 1, 143–152 (2003).

[27] http://www.nsnam.org (2014).

[28] Y. Sakumoto, R. Asai, H. Ohsaki, and M. Imase, "Design and implementation of flow-level simulator for performance evaluation of large scale networks.", *IEEE Int. Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems* 1, 166–172 (2007).

[29] C.V. Hollot, V. Misra, and D. Towsley, "Analysis and design of controllers for AQM routers supporting TCP flows", *IEEE Trans. on Automatic Control* 47 (6), 945–959 (2002).

[30] W. John and S. Tafvelin, "Analysis of internet backbone traffic and header anomalies observed", *ACM SIGCOMM Conf. on Internet Measurement* 1, 111–116 (2007).

[31] J. Domańska, A. Domański, S. Nowak, and T. Czachórski, "A contribution to the fair scheduling for the TCP and UDP Streams", *Communications in Computer and Information Science* 79, 207–216 (2010).

[32] W. John, S. Tafvelin, and T. Olovsson, "Passive internet measurement: overview and guidelines based on experiences", *Computer Communications* 33 (5), 533–550 (2010).

[33] Pengxuan Mao, Yang Xiao, Shaohai Hu, and Kiseon Kim, "Stable parameter settings for PI router mixing TCP and UDP traffic", *IEEE 10th Int. Conf. on Signal Processing (ICSP)* 1, 2547–2550 (2010).

[34] C.V. Hollot, V. Misra, and D. Towsley, "A control theoretic analysis of RED", *IEEE INFOCOM* 3, 1510–1519 (2001).

[35] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product network", *SIGCOMM* 1, 89–102 (2002).

[36] J. Klamka, A. Czornik, and M. Niezabitowski, "Stability and controllability of switched systems", *Bull. Pol. Ac.: Tech.* 61 (3), 547–555 (2013).

[37] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance", *IEEE ACM Trans. on Networking* 1 (4), 397–413 (1993).

[38] H.-J. Ho and W.-M. Lin, "AURED – Autonomous Random Early Detection for TCP congestion control", *3rd Int. Conf. on Systems and Networks Communications* 1, 79–84 (2008).

[39] L. Enhai, L. Yan, and P. Ruimin, "An improved random early detection algorithm based on flow prediction", *Second Int. Conf. on Intelligent Networks and Intelligent Systems* 1, 425–428 (2009).

[40] M.A. Qadeer, V. Sharma, A. Agarwal, and S.S. Husain, "Differentiated services with multiple random early detection algorithm using ns2 simulator", *2nd IEEE Int. Conf. on Computer Science and Information Technology* 1, 144–148 (2009).

[41] S. Bhatnagar and R. Patro, "A proof of convergence of the B-RED and P-RED algorithms for random early detection", *IEEE Communications Letters* 13 (10), 809–811 (2009).

[42] B. Zheng and M. Atiquzzaman "DSRED: an active queue management scheme for new generation network", *IEEE Conf. on Local Computer Networks* 1, 242–251 (2000).

[43] S. Athuraliya, V.H. Li, S.H. Low, and Q. Yin, "REM: active queue management", *IEEE Network* 15 (30), 48–53 (2001).

[44] K. Zhou, K.L. Yeung, and V. Li, "Nonlinear RED: a simple yet efficient active queue management scheme", *Computer Networks: Int. J. Computer and Telecommunications Networking* 50, 3784–3794 (2006).

[45] H. Abdel-jaber, M. Mahafzah, F. Thabtah, and M. Woodward, "Fuzzy logic controller of Random Early Detection based on average queue length and packet loss rate", *Int. Symp. Performance Evaluation of Computer and Telecommunication Systems* 1, 428–432 (2008).

[46] J. Domańska, A. Domański, and T. Czachórski, "Implementation of modified AQM mechanisms in IP routers", *J. Communications Software and Systems* 4 (1), CD-ROM (2008).

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

[48] M. May, C. Diot, B. Lyles, and J. Bolot, *Influence of Active Queue Management Parameters on Aggregate Traffic Performance, Research Report*, Institut de Recherche en Informatique et en Automatique, Yvelines, 2000.

[49] J. Domańska, A. Domański, and D.R. Augustyn, "The impact of the modified weighted moving average on the performance of the RED mechanism", *Communications in Computer and Information Science* 160, 37–44 (2011).

[50] D.R. Augustyn, A. Domański, and J. Domańska, "Active queue management with non linear packet dropping function", *6th Int. Conf. on Performance Modelling and Evaluation of Heterogeneous Networks* 1, 137–146 (2010).

[51] D.R. Augustyn, A. Domański, and J. Domańska, "A choice of optimal packet dropping function for active queue management", *Communications in Computer and Information Science* 79, 199–206 (2010).

[52] R. Pan, B. Prabhakar, and K. Psounis, "CHOKe, a stateless AQM scheme for approximating fair bandwidth allocation", *IEEE INFOCOM* 1, 942–952 (2000).

[53] T.K. Yung, J. Martin, M. Takai, and R. Bagrodia, "Integration of fluid-based analytical model with packet-level simulation for analysis of computer networks", *SPIE* 1, 130–143 (2001).

[54] www.scipy.org (2014).

[55] J. Domańska, A. Domański, and T. Czachórski, "Fluid flow analysis of RED algorithm with modified weighted moving average", *Communications in Computer and Information Science* 356, 50–58 (2013).

[56] A. Domański, J. Domańska, and T. Czachórski, "Comparison of CHOKe and gCHOKe active queues management algorithms with the use of fluid flow approximation", *Communications in Computer and Information Science* 370, 363–371 (2013).