BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 62, No. 4, 2014
DOI: 10.2478/bpasts-2014-0076

Active queue management with discrete sliding modes in TCP networks

P. IGNACIUK* and M. KARBOWAŃCZYK

Institute of Information Technology, Lodz University of Technology, 215 Wólczańska St., 90-924 Łódź, Poland

Abstract. In this paper, a new active queue management (AQM) algorithm for data traffic control in TCP/IP networks is developed. The algorithm design is based on the principles of discrete sliding-mode control. Unlike majority of earlier studies, the design procedure considers the effects of both non-negligible delay in transferring data and feedback information and unpredictable capacity variations. The switching function is selected to incorporate a delay compensation mechanism, which ensures efficient network operation even for large bandwidth-delay product connections. The proposed algorithm, implemented as a packet marking scheme, is tested in discrete event ns-2 simulator. The results show that the algorithm provides fast convergence to steady state after sudden, unanticipated capacity changes. By generating smaller overshoots, the proposed algorithm also allows for reducing buffer space requirements to avoid packet loss as compared to the benchmark AQM solutions.

Key words: sliding-mode control, congestion control, active queue management (AQM), discrete-time control systems.

1. Introduction

The recent traffic measurements, given e.g. in Cisco reports and forecasts [1], indicate that the amount of information exchanged over the Internet continues to grow. Since data in the Internet is passed over diverse types of networks, efficient regulatory mechanisms for the systems with variable transfer capabilities need to be provided. In particular, the data flow control under dynamically changing networking conditions, disturbances, and modeling uncertainties remains an open and challenging research issue. In this paper, the problem of TCP/IP (Transmission Control Protocol / Internet Protocol) congestion control is addressed from a control system perspective with explicit consideration of uncertain link capacity (bandwidth) variations.

Looking from the historical perspective, following the Jacobson and Braden algorithm [2], a number of different proposals for TCP congestion control were presented. In general, the proposed solutions can be divided into two main categories. The first category concentrates on the TCP congestion window management at the network end-points. The reported algorithms appropriately modify the window increment / decrement functions of the TCP sources, e.g. SACK [3], TCP NewReno [4], or use – different from the packet loss – indicators of congestion, e.g. round-trip time (RTT) measurements in TCP Vegas [5] and other delay-based control schemes [6]. The second category encompasses active queue management (AQM) solutions, which provide network assisted congestion control. The key idea behind AQM follows from the observation that, by monitoring the packet queue length, a network node can detect more promptly and accurately the incipient congestion than the remote sources. Then, preemptive measures can be taken to throttle the incoming rate before the actual loss of packets is experienced. For example, when the packet queue length in the output link buffer reaches a certain level, then an AQM node can selectively discard the incoming packets to prevent further input rate increase. Selective packet drop serves as an implicit feedback mechanism. Instead of discarding the packets, the node can also signal the incipient congestion by marking appropriate bits in the packet headers using explicit congestion notification (ECN) [7]. The receiver incorporates the feedback information read from the marked bits in the acknowledgments destined for the source. Upon the reception of a marked acknowledgment, the source reduces its window size thus increasing the chance of avoiding packet loss at a bottlenecked AQM node. A working example of an AQM scheme is random early detection (RED) designed by Floyd and Jacobson [8]. Due to the vulnerability of single-bit congestion notification, various researchers also proposed the use of two-bit [9], or multi-bit fields [10, 11] for achieving better accuracy and faster convergence rates in data flow control. However, unlike the single-bit ECN, the multi-bit congestion notification requires appropriate changes to be introduced into the existing TCP rules, which hinders wider acceptance of those proposals.

In this work, a new packet marking scheme for single-bit ECN congestion control is designed following strict control-theoretic approach. The design involves the concepts of discrete sliding-mode (DSM) control [12–14], which is well known to provide efficient performance in various engineering areas. However, despite its significant potential as a robust control method, so far the use of sliding-mode control (SMC) to regulate data traffic in communication networks has been quite limited. One of the principal obstacles in adopting SMC

*e-mail: przemyslaw.ignaciuk@p.lodz.pl

in the considered field is information transfer delay, inherent in communication networks. The effects of non-negligible delay should be accounted for in a way that ensures stable sliding motion without compromising performance in variable networking conditions [15–17].

The majority of SMC solutions for data flow control reported so far in the literature are intended for the networks with implicit feedback, working according to the TCP-AQM rules. A few AQM design examples were given in continuous-time domain, e.g. [18, 19], yet without explicit consideration of delay in the feedback and data paths. The effects of delay are covered in [20–23]. In [20], the time delay of input signal is taken into account in the design of an AQM control algorithm, but stability is analyzed only with regard to matched uncertainties. For a similar model with input delay considered, the effects of mismatched uncertainties have been studied in [21]. On the other hand, both the input and state delay have been taken into account in [22], and the maximum value of delay allowable for preserving the system stability has been estimated. A linear matrix inequality approach has been employed in the design of the observer-based AQM controller for a system with uncertainties, input delay, and saturated control [23]. The authors show that the observer-based controller [23] achieves faster response and less oscillatory transient behavior in comparison with the algorithm described in [20]. All those works, however, neglect the sampling effects. A discrete-time SMC approach to AQM controller design has been presented in [24], but the result is derived without explicit consideration of input or state delay. Outside the TCP-AQM paradigm the SMC design examples can be found also for DiffServ [25, 26] and general-type networks [27, 28].

As opposed to the previous SMC approaches to data flow control in TCP-AQM networks, in this paper both the effects of finite sampling rate and non-negligible delay are given explicit consideration in the controller design. The linearized TCP model is discretized and projected onto n-dimensional state space. The parameters of switching function are chosen for dead-beat response to ensure fast convergence to equilibrium. The proposed algorithm is implemented as a packet marking scheme with the marking probability modified according to the current network state and past control decisions. ns-2 simulations demonstrate that the developed algorithm ensures fast reaction to unanticipated capacity changes with reduced buffer size requirements as compared to the classical AQM solutions.

The paper is organized in the following way. Firstly, in Sec. 2, the model of TCP dynamics is presented in a convenient for controller design state-space form. Afterwards, in Sec. 3, the design procedure is conducted with the emphasis placed on the switching function selection. In that section, a nonlinear control law is proposed and the choice of controller parameters is discussed for different traffic scenarios. Section 4 reports on the controller performance in comparison to the benchmark AQM algorithms. Finally, Sec. 5 comprises the conclusions.

2. System model

One of the most popular models of TCP dynamics was derived through stochastic fluid-flow approximation of packet traffic by Misra and co-workers in [29]. In this paper, for the purpose of sliding-mode controller design, the continuous-time, nonlinear model [29] is linearized, discretized, and presented in an augmented state space. The state variables are selected to incorporate the signal history within *RTT*.

2.1. Fluid-flow nonlinear TCP model. According to [29], the mathematical description of essential TCP dynamics (with timeout effects neglected) can be described by the pair of coupled, nonlinear differential equations:

$$\dot{W}(t) = \frac{1}{R(t)} - \beta W(t)\frac{W[t-R(t)]}{R[t-R(t)]}p[t-R(t)], \quad (1)$$

$$\dot{q}(t) = \begin{cases} N(t)\dfrac{W(t)}{R(t)} - C(t), & q(t) > 0, \\ \max\left\{0, N(t)\dfrac{W(t)}{R(t)} - C(t)\right\}, & q(t) = 0, \end{cases} \quad (2)$$

where

- $W(t)$ is expected window size (ensemble averaging), $W(t) \in [0,\ W_{\max}]$, where $W_{\max} > 0$ denotes the maximum window size,
- $q(t)$ is expected packet queue length in the output link buffer (ensemble averaging), $q(t) \in [0,\ B_{\text{size}}]$, where $B_{\text{size}} > 0$ is the buffer capacity,
- $C(t)$ is time-varying capacity available for the controlled flows (in packets/s), $C(t) \in [0,\ C_{\max}]$, where $C_{\max}$ is the maximum capacity allocated to the controlled (elastic) traffic,
- $R(t) = T_p + q(t)/C(t)$ is the flow *RTT* that comprises propagation delay T_p, typically assumed constant (which is particularly well justified when the route is fixed during the transmission), and time-varying queuing delay $q(t)/C(t)$,
- $p(t)$ is probability of packet mark (or drop),
- $N(t)$ is load factor (the number of TCP connections contributing to the queue build-up in the output link buffer),
- β is window decrease parameter, usually taken equal to 1/2, although a more accurate value is sometimes assumed equal to ln(2) [30], or 2/3 [31].

The packet marking probability $p(t)$ is the control input and the packet queue length $q(t)$ is selected as the system output. The available bandwidth $C(t)$ constitutes the part of the overall link capacity $C_{\max}$ which is not consumed by the inelastic or high-priority traffic. In the wireless environment, function $C(t)$ also captures the link capacity fluctuations (occurring for instance due to the fading and shadowing phenomena). Note that in the wireless networks the channel bit rate (and thus the actual bandwidth available for data transfer) varies with time according to the signal strength and bit error rate. Unlike majority of earlier works, in this paper, capacity variations are explicitly taken into account in the control algorithm design. Since capacity variations are not known *a priori*, from the control system perspective $C(t)$ constitutes an exogenous uncertain signal – a disturbance.

2.2. Linearization and discretization. Depending on the choice of variables different approximations of system (1)–(2) can be obtained. In order to ensure efficient data transfer performance for connections with various, possibly disparate delays, the delay information is retained in the system description. The capacity available to serve the traffic may vary with time in an unpredictable way. The effects of finite sampling rate, which is unavoidable in the practical network realization, is also taken into account explicitly. Consequently, a more accurate system representation for sliding-mode controller design as compared to the earlier settings is obtained.

Let W and q be chosen as the state variables. System (1)–(2) is driven by two inputs: p – the controller command and C – the unknown perturbation. The operating point is specified by the quadruple $(W_0,\ q_0,\ p_0,\ C_0)$ such that

$$\dot{W} = 0 \quad \text{and} \quad \dot{q} = 0 \tag{3}$$

for $(W,\ q,\ p,\ C) = (W_0,\ q_0,\ p_0,\ C_0)$. Thus, by equating the left-hand sides of (1) and (2) to zero, the following set of relations is obtained at the operating point

$$\begin{gathered} \beta p_0 W_0^2 = 1, \quad N W_0 = C_0 R_0, \\ \text{and} \quad R_0 = T_p + q_0/C_0, \end{gathered} \tag{4}$$

where R_0 is the equilibrium *RTT*. For given load N and propagation delay T_p, the set of feasible operating points is defined as

$$\begin{gathered} \Omega_{(N,T_p)} = \{(W_0, q_0, p_0, C_0) : W_0 \in (0, W_{\max}), \\ q_0 \in (0, B_{\text{size}}), p_0 \in (0,1), C_0 \in (0, C_{\max})\}. \end{gathered} \tag{5}$$

Introducing the deviations

$$\begin{gathered} \delta W(t) = W(t) - W_0, \quad \delta q(t) = q(t) - q_0, \\ \delta p(t) = p(t) - p_0, \quad \delta C(t) = C(t) - C_0, \end{gathered} \tag{6}$$

the small-signal approximation of system dynamics (see the Appendix) can be written in a state-space form as

$$\begin{gathered} \dot{\mathbf{x}}(t) = \mathbf{A_c}\mathbf{x}(t) + \mathbf{b_c}\delta p(t - R_0) + \mathbf{d_c}\delta C(t), \\ y(t) = \mathbf{q_c}^T \mathbf{x}(t), \end{gathered} \tag{7}$$

where $\mathbf{x}(t) = [\delta q(t) \delta W(t)]^T$ is the deviation state vector, $\mathbf{A}_c$ is its 2×2 state matrix, $\mathbf{b}_c$, $\mathbf{d}_c$, and $\mathbf{q}_c$ are 2×1 vectors:

$$\begin{gathered} \mathbf{A_c} = \begin{bmatrix} -\dfrac{1}{R_0} & \dfrac{N}{R_0} \\ 0 & -\dfrac{2N}{R_0^2 C_0} \end{bmatrix}, \quad \mathbf{b_c} = \begin{bmatrix} 0 \\ -\dfrac{\beta R_0 C_0^2}{N^2} \end{bmatrix}, \\ \mathbf{d_c} = \begin{bmatrix} -\dfrac{T_p}{R_0} \\ 0 \end{bmatrix}, \qquad \mathbf{q_c} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \end{gathered} \tag{8}$$

Assuming that the system is sampled with period T and R_0 constitutes an integer multiple of T, i.e. $R_0 = n_0 T$, where $n_0 > 0$, the zero-order hold discrete-time form of (7),

$$\begin{gathered} \mathbf{x}[(k+1)T] = \mathbf{A_d}\mathbf{x}(kT) + \mathbf{b_d}\delta p[(k-n_0)T] + \mathbf{d_d}\delta C(kT), \\ y(kT) = \mathbf{q_d}^T \mathbf{x}(kT), \end{gathered} \tag{9}$$

for $k = 0, 1, 2, \ldots$. Further on, the independent variable kT will be written shortly as k. Using the inverse Laplace transform, the discrete-time state matrix $\mathbf{A}_d = e^{\mathbf{A_c}T} = \mathcal{L}^{-1}\{s\mathbf{I} - \mathbf{A}_c\}|_{t=T}$ for $C_0 R_0 \neq 2N$,

$$\begin{aligned} \mathbf{A_d} &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \\ &= \begin{bmatrix} e^{-\dfrac{T}{R_0}} & \dfrac{C_0 N R_0}{C_0 R_0 - 2N}\left(e^{-\dfrac{2NT}{R_0^2 C_0}} - e^{-\dfrac{T}{R_0}}\right) \\ 0 & e^{-\dfrac{2NT}{R_0^2 C_0}} \end{bmatrix}. \end{aligned} \tag{10}$$

On the other hand, the discrete-time input and disturbance vectors

$$\mathbf{b_d} = \left(\int_0^T e^{\mathbf{A_c}\tau} d\tau\right) \mathbf{b_c}$$

and

$$\mathbf{d_d} = \left(\int_0^T e^{\mathbf{A_c}\tau} d\tau\right) \mathbf{d_c}$$

are calculated as

$$\begin{aligned} \mathbf{b_d} &= \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \\ &= \beta \begin{bmatrix} \dfrac{C_0^4 R_0^4}{2N^2(C_0 R_0 - 2N)} e^{-\frac{2NT}{R_0^2 C_0}} \\ -\dfrac{C_0^3 R_0^3}{N(C_0 R_0 - 2N)} e^{-\frac{T}{R_0}} - \dfrac{C_0^3 R_0^3}{2N^2} \\ \dfrac{C_0^3 R_0^3}{2N^3} e^{-\frac{2NT}{R_0^2 C_0}} - \dfrac{C_0^3 R_0^3}{2N^3} \end{bmatrix}, \end{aligned} \tag{11}$$

and

$$\mathbf{d_d} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} T_p\left(e^{-\frac{T}{R_0}} - 1\right) \\ 0 \end{bmatrix}. \tag{12}$$

The output vector $\mathbf{q}_d = \mathbf{q}_c$.

In the special case $C_0 R_0 = 2N$, one obtains

$$\begin{aligned} \mathbf{A_d} &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} e^{-\dfrac{T}{R_0}} & \dfrac{NT}{R_0} e^{-\frac{T}{R_0}} \\ 0 & e^{-\frac{T}{R_0}} \end{bmatrix}, \\ \mathbf{b_d} &= \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = 4\beta N \begin{bmatrix} \left(1 + \dfrac{T}{R_0}\right) e^{-\frac{T}{R_0}} - 1 \\ e^{-\frac{T}{R_0}} - 1 \end{bmatrix}, \end{aligned} \tag{13}$$

$\mathbf{d}_d$ given by (12), and $\mathbf{q}_d = \mathbf{q}_c$.

2.3. Delay-free model in augmented state space. In order to circumvent the problems related to controller design in the presence of non-negligible delay, the system dynamics can be projected onto extended state space with extra state variables representing delayed signal history, as shown previously in the context of integrating [15] and first-order processes [32].

Let $\mathbf{v}(k) = [v_1(k),\ v_2(k),\ ...,\ v_n(k)]^T$ be a new state vector with

- $v_1(k) = y(k) = \delta q(k)$ representing the difference between the packet queue length and its equilibrium value q_0,
- $v_2(k) = \delta W(k)$ the difference between the current window size and W_0,
- and the remaining state variables

$$v_i(k) = \delta p(k - n + i - 1) \tag{14}$$

for $i = 3, ..., n$ representing the controller command history.

Then, denoting the controller command by $u(k) = \delta p(k)$, i.e. the deviation of packet marking probability from p_0, and the deviation of available capacity from C_0 by $d(k) = \delta C(k)$ discrete-time system (9) is represented in the extended state space as

$$\begin{aligned} \mathbf{v}(k+1) &= \mathbf{A}\mathbf{v}(k) + \mathbf{b}u(k) + \mathbf{d}d(k), \\ y(k) &= \mathbf{q}^T\mathbf{v}(k). \end{aligned} \tag{15}$$

In (15), $\mathbf{A}$ is $n \times n$ state matrix, $\mathbf{b}$, $\mathbf{d}$, and $\mathbf{q}$ are $n \times 1$ input, disturbance, and output vectors, respectively,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & b_1 & 0 & \dots & 0 \\ 0 & a_{22} & b_2 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} d_1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \tag{16}$$

and the system order $n = n_0 + 2 = (R_0/T) + 2$ depends on the discretization period and *RTT* of the connections passing through the AQM router. Note that since $R_0 > 0$, the system is at least of order 3. The coefficients a_{11}, a_{12}, a_{22}, b_1, b_2 are taken from (10) and (11) if $C_0R_0 \neq 2N$, and from (13) otherwise. The constant d_1 is determined from (12). The deviation dynamics should be kept in the vicinity of the operating point. Hence, the desired system state is the origin of the deviation space.

3. Sliding-mode control law

The purpose of control action can be formulated as to reach the desired operating point (to drive the queue length to a set-point level), and to maintain the system dynamics near the operating point despite modeling inaccuracies and external disturbances. In the considered application, the queue length drift from equilibrium due to capacity variations should cause neither buffer overflow (and packet drop), nor buffer depletion and decreased usage of the available bandwidth. The principal obstacle in maintaining $q(t)$ confined to $(0,\ B_{\text{size}})$ when the bandwidth diverges from C_0 is the time between taking the control decision and experiencing the modified input rate at the node that adjusted the packet marking probability – round-trip delay R.

3.1. Design of switching function. The controller design is performed using the theory of DSM control – an inherently robust regulation technique [12–14]. A key point in the design of sliding-mode controllers is selection of appropriate switching function (or sliding plane) that determines the desirable dynamical properties of the developed control system. Due to the presence of non-negligible delay, in order to provide stable and fast system response, the switching function should incorporate a delay compensator.

Let the switching function

$$s(k) = -\mathbf{c}^T\mathbf{v}(k) \tag{17}$$

with vector

$$\mathbf{c}^T = [c_1\ c_2\ c_3\ ...\ c_n], \qquad \mathbf{c}^T\mathbf{b} \neq 0, \tag{18}$$

describing the sliding hyperplane $\mathbf{c}^T\mathbf{v}(k) = 0$. Substituting (15) into the equation $\mathbf{c}^T\mathbf{v}(k+1) = 0$ and rearranging, one obtains the equivalent control

$$u_{eq}(k) = -(\mathbf{c}^T\mathbf{b})^{-1}\mathbf{c}^T\mathbf{A}\mathbf{v}(k)$$

and the closed-loop state matrix

$$\mathbf{A}_{cl} = [\mathbf{I}_n - \mathbf{b}(\mathbf{c}^T\mathbf{b})^{-1}\mathbf{c}^T]\mathbf{A},$$

where $\mathbf{I}_n$ denotes $n \times n$ identity matrix.

A good control scheme for data transmission networks should provide fast responsiveness to the changes of networking conditions. Therefore, parameters of the sliding plane are selected for a dead-beat scheme. Dead-beat control requires all the closed-loop poles to be placed at the origin. The characteristic polynomial of $\mathbf{A}_{cl}$ is calculated as

$$\begin{aligned} &z^n + \frac{c_{n-1} - (a_{11} + a_{22})\,c_n}{c_n} z^{n-1} \\ &+ \frac{a_{11}\,(a_{22}c_n - c_{n-1}) - a_{22}c_{n-1} + c_{n-2}}{c_n} z^{n-2} + ... \\ &+ \frac{a_{11}\,(a_{22}c_5 - c_4) - a_{22}c_4 + c_3}{c_n} z^3 \\ &+ \frac{a_{11}\,(a_{22}c_4 - c_3) - a_{22}c_3 + b_1c_1 + b_2c_2}{c_n} z^2 \\ &+ \frac{a_{11}\,(a_{22}c_3 - b_2c_2) + a_{12}b_2c_1 - a_{22}b_1c_1}{c_n} z. \end{aligned} \tag{19}$$

For all the roots of (19) to be placed at the origin the determinant $\det(z\mathbf{I}_n - \mathbf{A}_{cl})$ should equal to z^n, which is satisfied when simultaneously

$$\begin{aligned} 0 &= c_{n-1} - (a_{11} + a_{22})\, c_n, \\ 0 &= a_{11}\,(a_{22}c_n - c_{n-1}) - a_{22}c_{n-1} + c_{n-2}, \\ &\vdots \\ 0 &= a_{11}\,(a_{22}c_5 - c_4) - a_{22}c_4 + c_3, \\ 0 &= a_{11}\,(a_{22}c_4 - c_3) - a_{22}c_3 + b_1c_1 + b_2c_2, \\ 0 &= a_{11}\,(a_{22}c_3 - b_2c_2) + a_{12}b_2c_1 - a_{22}b_1c_1. \end{aligned} \tag{20}$$

Equation set (20) is solved recursively. First, c_{n-1} is determined from the first equation in the set as $c_{n-1} = (a_{11}+a_{22})c_n$. Next, c_{n-1} is substituted into the second equation in (20). In this way, c_{n-2} may be expressed in terms of system parameters and the last component of vector $\mathbf{c}$. Continuing the substitutions, the following relation may be observed

$$c_{n-j} = c_n \sum_{i=0}^{j} a_{11}^{j-i} a_{22}^{i}, \quad \text{for} \quad j = 1, ..., n-3 \tag{21}$$

and

$$\begin{aligned} c_2 &= \left(c_n \sum_{i=0}^{n-2} a_{11}^{n-2-i} a_{22}^{i} - b_1c_1 \right) / b_2, \\ c_1 &= c_n a_{11}^{n-1} / (a_{11}b_1 + a_{12}b_2 - a_{22}b_1). \end{aligned} \tag{22}$$

Substituting (21) and (22) into (17), one obtains

$$\begin{aligned} \mathbf{c}^T\mathbf{v}(k) &= c_1v_1(k) + c_2v_2(k) \\ &+ c_n \sum_{j=3}^{n} \left(\sum_{i=0}^{n-j+1} a_{11}^{n-j+1-i} a_{22}^{i} \right) v_j(k). \end{aligned} \tag{23}$$

For $j = 2, 3, ..., n$ the state variables $v_j(k) = \delta p(k-n+j-1)$. Hence, the sum

$$\begin{aligned} &c_n \sum_{j=3}^{n} \left(\sum_{i=0}^{n-j+1} a_{11}^{n-j+1-i} a_{22}^{i} \right) v_j(k) \\ &= \sum_{j=k-n_0}^{k-1} \left(\sum_{i=0}^{k-j} a_{11}^{k-j-i} a_{22}^{i} \right) \delta p(j), \end{aligned} \tag{24}$$

Without loss of generality c_n may be taken as unity. Since $v_1(k) = \delta q(k)$ and $v_2(k) = \delta W(k)$, then using (24) one may represent switching function (17) as

$$\begin{aligned} s(k) &= -g_1\delta q(k) - g_2\delta W(k) \\ &- \sum_{j=k-n_0}^{k-1} \left(\sum_{i=0}^{k-j} a_{11}^{k-j-i} a_{22}^{i} \right) \delta p(j), \end{aligned} \tag{25}$$

where

$$g_1 = \frac{a_{11}^n}{a_{11}b_1 + a_{12}b_2 - a_{22}b_1}$$

and

$$g_2 = \frac{(a_{12}b_2 - a_{22}b_1)\, a_{11}^{n-1}}{b_2\,(a_{11}b_1 + a_{12}b_2 - a_{22}b_1)} + \frac{a_{22}}{b_2} \sum_{i=0}^{n-2} a_{11}^{n-2-i} a_{22}^{i}. \tag{26}$$

The delay compensating features are synthesized in the last term in (25).

3.2. Control law. For $t \in [kT, (k+1)T)$ the marking probability according to DSM control law is calculated as

$$p(t) = p(k) = p_0 + \gamma \mathrm{sgn}[s(k)], \tag{27}$$

where the sgn function is defined as

$$\mathrm{sgn}(s) = \begin{cases} -1, & \text{if} \quad s \le 0, \\ 1, & \text{if} \quad s > 0. \end{cases} \tag{28}$$

In order to reduce chattering, instead of sgn, a sigmoid function sig $= s/(|s| + \delta)$, $\delta > 0$, can be applied [33]. Both functions are illustrated in Fig. 1.

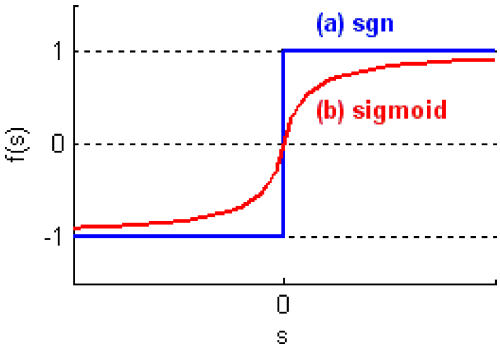

Fig. 1. Sgn (a) and sigmoid (b) functions

The gain γ is adjusted as

$$0 < \gamma \le \min\{p_0, 1 - p_0\} \tag{29}$$

so that feasible marking probability $p \in [0,\ 1]$ is ensured.

Equation (27) with switching function (25) represent a fixed-gain variable-state nonlinear control law. In order to determine the current packet marking rate $p(k)$, the controller uses the measurement of instantaneous queue length $\delta q(k)$, the window size estimate (obtained from the observation of packet incoming rate) $\delta W(k)$, and the marking rate history recorded within the last *RTT*. Since all the controller parameters can be computed offline, good operational efficiency in software implementation is ensured. The structure of the designed controller is illustrated in Fig. 2.

Note that if the sigmoid function with $\delta > 1$ is applied, one obtains a low-gain proportional controller, which is suitable if the available capacity undergoes high-frequency variations, such as in multi-user wireless networks.

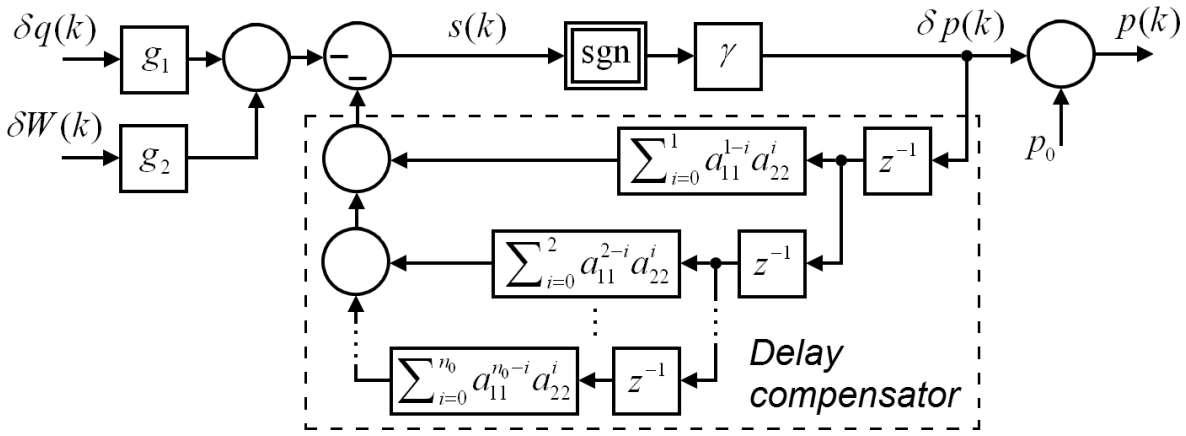

Fig. 2. Proposed control strategy

4. Simulations

The properties of designed AQM strategy (27) are verified in a series of tests conducted in ns-2 discrete event simulator [34]. The network topology used in the experiments is depicted in Fig. 3. In the considered scenario, the data exchange occurs among three nodes: TCP traffic source (**S**), router supporting AQM (**R**), and destination (**D**). Source **S** reflects an aggregate entry point of $N = 50$ TCP NewReno flows. The link connecting nodes **S** and **R** is characterized by the capacity 100 Mbit/s and propagation latency 30 ms, and the link connecting nodes **R** and **D** by the maximum capacity 10 Mbit/s and propagation latency 70 ms. Thus, the round-trip propagation delay $T_p = 200$ ms.

Each flow is fed with non-exhausting FTP data stream. The packet size is set as 1040 bytes (1000-byte payload + 40-byte header) and the target queue length as $q_0 = 100$ packets. The queue limit is adjusted as 2000 packets so that packet loss is avoided and buffer size requirements can be assessed. With the sampling period chosen as $T = 30$ ms, a discrete system of order $n = 12$ is obtained. In the model, the window decrease factor β is assumed equal 0.7.

Performance of the proposed DSM controller is compared with two classical AQM schemes: RED [8] and PI [35]. In the tests, RED $\max_{th}$ is set as 200 packets and PI reference queue length as 100 packets. The remaining parameters of benchmark solutions are left at the default ns-2 values.

Test 1. In the first series of simulations, the actual capacity available for the flows at the bottleneck link R–D varies with time as specified in Table 1. Consequently, as shown in Fig. 4, after each change the available capacity remains constant for several *RTT*s and ultimately returns to the operating point value set as $C_0 = 1000$ packets/s (approx. 8 Mbit/s). The DSM controller gain is chosen according to (29) as $\gamma = \min\{p_0, 1 - p_0\} = p_0 = 0.039$.

The queue length evolution obtained in the test is depicted in Fig. 5 (initial phase) and Fig. 6 (for $t > 10$ s), and marking probability obtained from the DSM algorithm in Fig. 7. One can notice a significant difference between the maximum queue length among the compared algorithms. In the case of RED and PI the initial overshoot results in the maximum of 1000 packets. Moreover, q oscillates with large amplitude before it settles near q_0. In turn, in the case of DSM algorithm the maximum queue length equals 502 packets. Since a lower maximum queue implies smaller buffer capacity to avoid packet drop, the proposed algorithm offers a less memory consuming solution than the benchmark ones. Furthermore, unlike RED, the DSM controller keeps the queue length strictly positive. As a result, the entire available capacity is used to serve the packet traffic and maximum throughput is obtained [36].

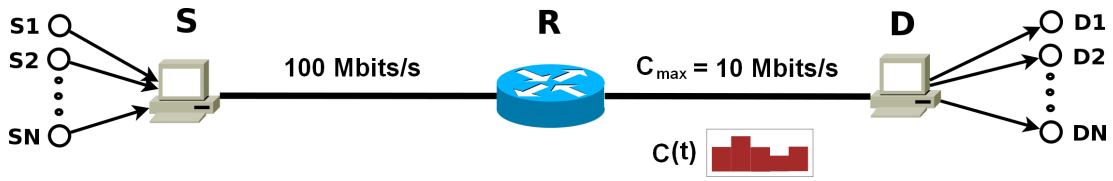

Fig. 3. Network topology used in experiments

Table 1
Available capacity at the bottleneck link (Test 1)

Simulation time t [s]	$C\ (t)$ [packets/s]	$C\ (t)$ [Mbit/s]
0–10	1000	7.93
10–20	1250	9.92
20–30	1000	7.93
30–40	750	5.95
40–50	1000	7.93

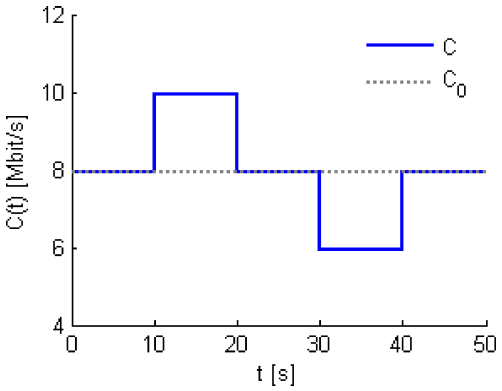

Fig. 4. Available capacity in Test 1

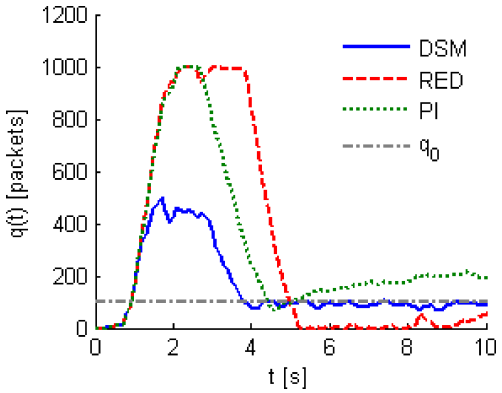

Fig. 5. Queue length interval 0–10 s

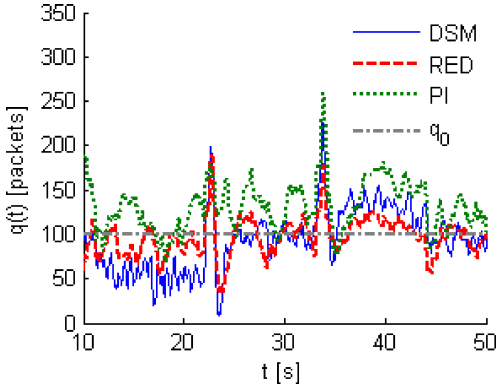

Fig. 6. Queue length, interval 10–50 s

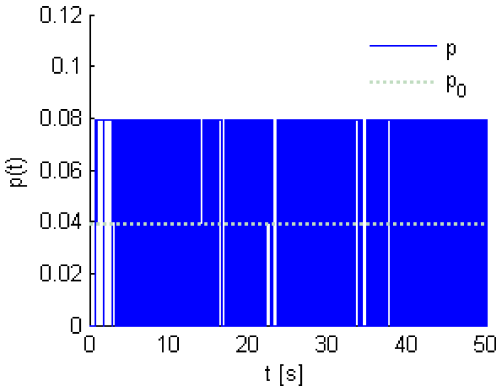

Fig. 7. Marking probability (DSM)

It follows from Fig. 5 that after a link capacity change the DSM controller drives the queue length to the neighborhood of new operating point. The modified queuing delay may cause a drift in the end-to-end latency, which is disadvantageous for multimedia streams. However, it can be noticed that when the capacity returns to C_0 the queue length converges to q_0 and remains in the vicinity of q_0. In order to reduce the queue length sensitivity to persistent capacity changes, feedforward disturbance compensation can be applied [37, 38].

Test 2. Owing to its fast dynamics, the DSM controller promptly reacts to capacity variations yet may lead to increased delay jitter (originating from queue length fluctuations). In the second test, the controller performance is evaluated in the presence of random capacity variations – uniform in the range [800, 1200 packets/s] – illustrated in Fig. 8. In order to mitigate the influence of these high-frequency capacity changes, the DSM controller gain is decreased to 0.01. Also, instead of sgn, a sigmoid function with $\delta = 10$ is applied.

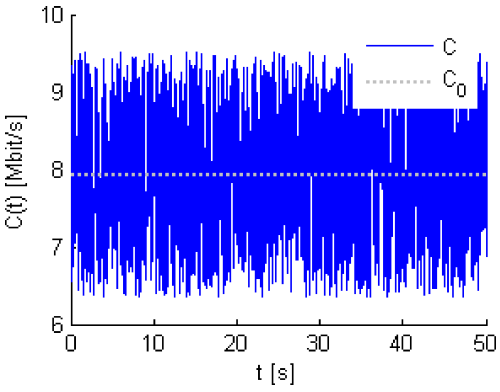

Fig. 8. Available capacity with high frequency components

The resulting queue length is shown in Fig. 9 and marking probability in Fig. 10. It is clear from the graphs that the communication system stability is preserved despite high-frequency capacity fluctuations. Both DSM and RED algorithms keep the queue length in a close vicinity of the operating point. As compared with the results of Test 1 (Fig. 6), the range of q variations has been reduced, which implies smaller delay jitter without increase in average queuing delay. Therefore, the proposed control strategy is flexible enough to

serve both loss and delay sensitive traffic with intuitive tuning limited to the choice of just two parameters – γ and δ.

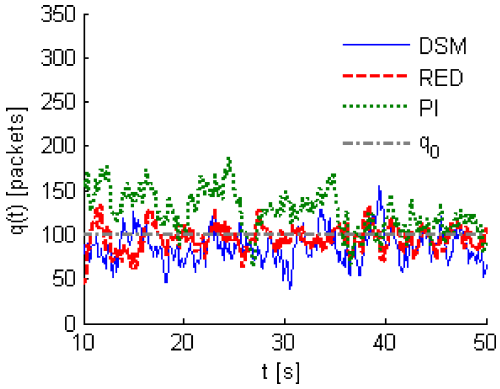

Fig. 9. Queue length

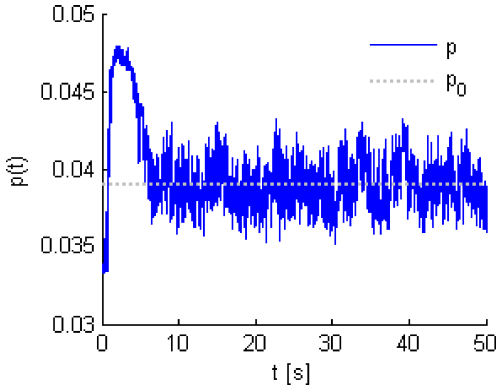

Fig. 10. Marking probability (DSM)

The primary purpose of the considered model is to examine the influence of uncertain, *a priori* unknown capacity changes that reflect the situation in real communication networks. For that model a robust data flow control solution has been established. However, it would be interesting to evaluate performance of the proposed DSM strategy also in the presence of other parameter uncertainty. Therefore, in two further simulations, in addition to random capacity variations, the effects of load (number of flows) N changes and random data loss experienced by the input stream are analyzed.

Test 3. The system model in Subec. 2.2 is constructed under the typical condition that number of flows N is constant during the transmission. However, one cannot actually expect that this condition will be guaranteed in a working TCP/IP network. Therefore, in the third test, the controller performance is evaluated in the presence of load changes – in the range [40, 60 flows] illustrated in Fig. 11. The capacity undergoes random variations depicted in Fig. 8. The algorithm parameters are adjusted in the same way as in Test 2, i.e. assuming that the number of flows is constant and equals $N_0 = 50$.

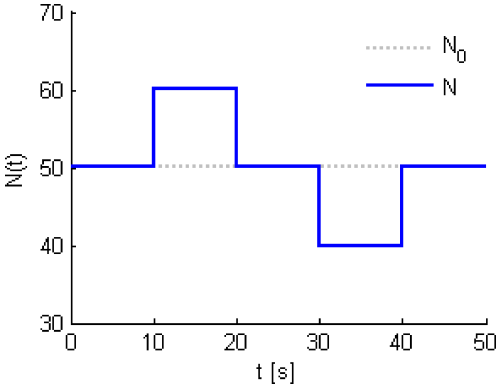

Fig. 11. Variable number of flows in Test 3

The resulting queue length is shown in Fig. 12 and marking probability in Fig. 13. It follows from Fig. 12 that the communication system stability is preserved under variable load conditions. Although the shifts in the number of flows affect the average queue length (similarly as in the case of RED and PI controllers, though with smaller overshoot), it can be noticed that when the number of flows returns to N_0 the queue length converges to q_0, and remains in the vicinity of q_0 afterwards.

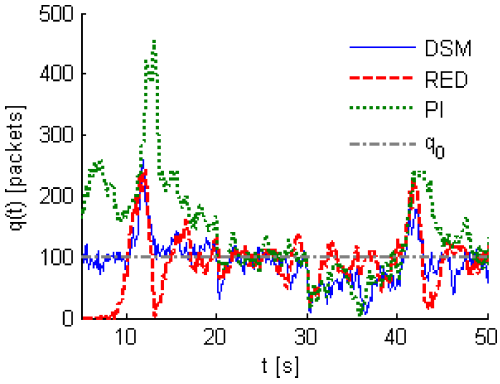

Fig. 12. Queue length under variable load

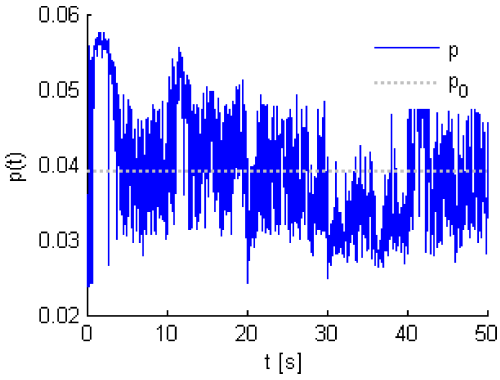

Fig. 13. Marking probability under variable load

Test 4. In the fourth series of simulations, the influence of random packet loss on the performance of proposed control law is examined. The test is conducted with 2% loss rate

on the path from source **S** to steering node **R**. The capacity on the outgoing link is subject to uniform variations in the range [800, 1200 packets/s] illustrated in Fig. 8. To achieve fast reaction to variable networking conditions the control law is tuned in the same way as in Test 1, i.e. the sgn function is applied and the DSM controller gain is chosen as $\gamma = 0.039$.

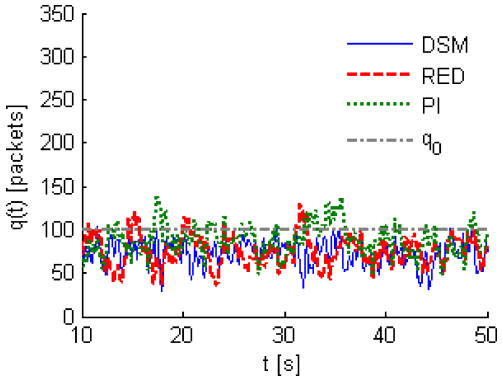

Fig. 14. Queue length in packet loss scenario

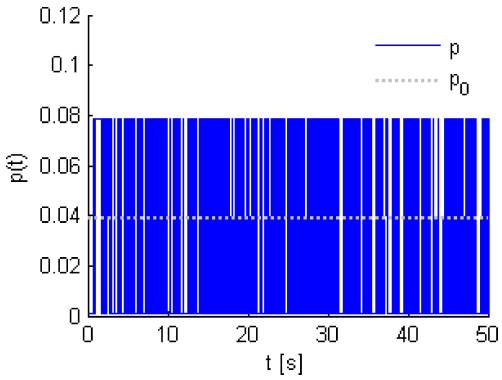

Fig. 15. Marking probability in packet loss scenario

The obtained queue length is shown in Fig. 14 and marking probability in Fig. 15. The average queue length moves below the set-point value $q_0 = 100$ packets. The deficiency in the incoming rate (and the resulting queue length) due to losses does not trigger oscillatory behavior. However, a further increase in loss rate adversely affects the network throughput. The queue length may drop to zero and not all of the available bandwidth will be utilized. Nevertheless, the stability is maintained.

5. Conclusions

In this paper, a new AQM packet marking algorithm for TCP networks was designed using the theory of DSM control. In contrast to the typical approaches found in the literature, the proposed design methodology gives explicit consideration to non-negligible latency in data and feedback information exchange, finite sampling frequency of transfer rate adaptations, and uncertain bandwidth fluctuations. By considering the combined effect of delay and a variable bandwidth in the design process, the proposed algorithm offers faster reaction to the changes of networking conditions (reflected in parameter fluctuations) and smaller range of queue length variations than the classical AQM schemes. As a result, more efficient buffer space management (reduced capacity requirements) can be achieved in high-speed networks.

The control algorithm is expressed in a closed form with constant gain vector precomputable before the actual control process commences. Since the controller parameters can be calculated offline and the algorithm uses only fundamental arithmetic and logic operations, good efficiency in software (or hardware) implementation is guaranteed. The tuning to a particular pattern of capacity variations and link type reduces to the choice of just two parameters and can be automated. As a result, the administrative effort to set up the algorithm at a network node is negligible.

Since the queue length does not exhibit large overshoots in response to parameter variations, e.g. a sudden change in the available bandwidth or network load, smaller buffer suffices to eliminate the risk of packet loss due to overflow. In consequence, the proposed algorithm is well suited to serve loss sensitive traffic, such as batch transfer of financial data. Its possible drawback, originating from the predictor-like structure incorporated to mitigate the effects of delay, is that for a persistent capacity change the queue length settles at new equilibrium. The resulting change in the queuing delay affects the overall latency of transmitted data streams, which may adversely affect multimedia traffic. In order to reduce queue length sensitivity to capacity changes, feed-forward disturbance compensation can be applied in a way similar to [37, 38].

Appendix

It is assumed that R is constant at the operating point and the nested dependence of the time-delay argument $t - R(t)$ on the queue length and available bandwidth can be ignored (which is a reasonable assumption in the majority of practical cases [39]). Therefore, in the linearization procedure $t - R(t)$ is set as $t - R_0$. Taking N and T_p as constants and preserving the information about input delay, the following simplified system dynamics is obtained

$$\dot{W}(t) = \frac{1}{T_p + q(t)/C(t)} - \frac{\beta W^2(t)}{T_p + q(t)/C(t)} p(t - R_0), \tag{30}$$

$$\dot{q}(t) = \frac{NW(t)}{T_p + q(t)/C(t)} - C(t). \tag{31}$$

Let the right-hand sides of (30) and (31) be denoted by

$$f(W, q, p_R, C) = \frac{1}{T_p + q/C} - \frac{\beta W^2}{T_p + q/C} p_R, \tag{32}$$

$$g(W, q, C) = \frac{NW}{T_p + q/C} - C, \tag{33}$$

where $p_R(t) = p(t - R_0)$. Then, after taking the partial derivatives of f and g and considering (4), one has at the operating point

$$\begin{aligned} \frac{\partial f}{\partial C} &= \frac{q}{C^2}\frac{1}{(T_p + q/C)^2} - \frac{q}{C^2}\frac{\beta W^2}{(T_p + q/C)^2} p_R = 0, \\ \frac{\partial f}{\partial W} &= -\frac{2\beta W p_R}{T_p + q/C} = -\frac{2N}{R_0^2 C_0}, \\ \frac{\partial f}{\partial p_R} &= -\frac{\beta W^2}{T_p + q/C} = -\frac{\beta R_0 C_0^2}{N^2}, \\ \frac{\partial f}{\partial q} &= \frac{\beta W^2 p_R - 1}{C\,(T_p + q/C)^2} = 0, \end{aligned} \tag{34}$$

and

$$\begin{aligned} \frac{\partial g}{\partial C} &= \frac{q}{C^2}\frac{NW}{(T_p + q/C)^2} - 1 = -\frac{T_p}{R_0}, \\ \frac{\partial g}{\partial W} &= \frac{N}{T_p + q/C} = \frac{N}{R_0}, \\ \frac{\partial g}{\partial q} &= -\frac{NW}{C\,(T_p + q/C)^2} = -\frac{1}{R_0}. \end{aligned} \tag{35}$$

Thus, the small-signal representation of (30) and (31),

$$\delta\dot{W}(t) = -\frac{2N}{R_0^2 C_0}\delta W(t) - \frac{\beta R_0 C_0^2}{N^2}\delta p(t - R_0), \tag{36}$$

$$\delta\dot{q}(t) = -\frac{1}{R_0}\delta q(t) + \frac{N}{R_0}\delta W(t) - \frac{T_p}{R_0}\delta C(t). \tag{37}$$

Acknowledgements. This work has been performed within a framework of the project "Design and validation of control algorithms in networked dynamical systems" financed by the National Science Centre of Poland – the decision number DEC-2012/05/D/ST6/03030. P. Ignaciuk is a scholarship holder of the Ministry of Science and Higher Education for Outstanding Young Researchers.

REFERENCES

[1] Cisco Public Information,*Cisco Visual Networking Index: Forecast and Methodology 2011–2016* 1, CD-ROM (2012).

[2] V. Jacobson and R. Braden, *TCP Extensions for Long-Delay Paths*, RFC 1072, 1988.

[3] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanov, *TCP Selective Acknowledgment Options*, RFC 2018, 1996.

[4] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, *The NewReno Modification to TCP's Fast Recovery Algorithm*, RFC 6582, 2012.

[5] L. S. Brakmo and L. L. Peterson, "TCP Vegas: end-to-end congestion avoidance on a global Internet", *IEEE J. Sel. Area. Comm.* 13 (8), 1465–1480 (1995).

[6] M. Rodríguez-Pérez, S. Herrería-Alonso, M. Fernández-Veiga, and C. López-García, "Common problems in delay-based congestion control algorithms: a gallery of solutions", *Eur. T. Telecommun.* 22 (4), 168–178 (2011).

[7] S. Floyd, "TCP and explicit congestion notification", *ACM SIGCOMM Comput. Comm. Rev.* 24 (5), 10–23 (1994).

[8] S. Floyd and V. Jacobson, "Random Early Detection gateways for congestion avoidance", *IEEE ACM T. Network* 1 (4), 397–413 (1993).

[9] Y. Xia, L. Subramanian, I. Stoica, and S. Kalyanaraman, "One more bit is enough", *Proc. ACM SIGCOMM* 1, 37–48 (2005).

[10] D. Katabi, M. Handley, and Ch.E. Rohrs, "Congestion control for high bandwidth-delay product networks", *Proc. ACM SIGCOMM* 1, 89–102 (2002).

[11] H. Wu, F. Ren, D. Muc, and X. Gong, "An efficient and fair explicit congestion control protocol for high bandwidth-delay product networks", *Comput. Commun.* 32 (7–10), 1138–1147 (2009).

[12] Č. Milosavljević, "General conditions for the existence of a quasisliding mode on the switching hyperplane in discrete variable structure systems", *Automat. Rem. Contr.* 46 (3), 307–314 (1985).

[13] W. Gao, Y. Wang, and A. Homaifa, "Discrete-time variable structure control systems", *IEEE T. Ind. Electron.* 42 (2), 117–122 (1995).

[14] B. Bandyopadhyay and S. Janardhanan, *Discrete-Time Sliding Mode Control. A Multirate Output Feedback Approach*, Springer-Verlag, Berlin, 2006.

[15] P. Ignaciuk and A. Bartoszewicz, "Linear quadratic optimal discrete time sliding mode controller for connection oriented communication networks", *IEEE T. Ind. Electron.* 55 (11), 4013–4021 (2008).

[16] P. Ignaciuk and A. Bartoszewicz, "Linear quadratic optimal sliding mode flow control for connection-oriented communication networks", *Int. J. Robust Nonlin. Contr.* 19 (4), 442–461 (2009).

[17] X. Yu, B. Wang, and X. Li, "Computer-controlled variable structure systems: the state-of-the-art", *IEEE T. Ind. Inform.* 8 (2), 197–205 (2012).

[18] F. Ren, C. Lin, and X. Yin, "Design a congestion controller based on sliding mode variable structure control", *Comput. Commun.* 28 (9), 1050–1061 (2005).

[19] P. Yan, Y. Gao, and H. Özbay, "A variable structure control approach to active queue management for TCP with ECN", *IEEE T. Contr. Syst. T.* 13 (2), 203–215 (2005).

[20] F. Yin, G.M. Dimirovski, and Y. Jing, "Robust stabilization of input delay for Internet congestion control", *Proc. Amer. Contr. Conf.* 5576–5580 (2006).

[21] H. Wang, Y. Jing, Y. Zhou, Z. Chen, and X. Liu, "Sliding mode control for uncertain time-delay TCP/AQM network systems", *Proc. IFAC World Congr.* 12013-12018 (2008).

[22] Y. Jing, L. He, G. M. Dimirovski, and H. Zhu, "Robust stabilization of state and input delay for Active Queue Management algorithm", *Proc. Amer. Contr. Conf.* 3083–3087 (2007).

[23] Y. Zhou, H. Wang, Y. Jing, and X. Liu, "Observer based robust controller design for Active Queue Management", *Proc. IFAC World Congr.* 12007–12012 (2008).

[24] M. Yan, T. D. Kolemisevska-Gugulovska, Y. Jing, and G. M. Dimirovski, "Robust discrete-time sliding-mode control algorithm for TCP networks congestion control", *Proc. Int. Conf. Telecommun. Modern Sat. Cable & Broadcast. Serv.* 1, 393–396 (2007).

[25] X. Zheng, N. Zhang, G.M. Dimirovski, and Y. Jing "Adaptive sliding mode congestion control for DiffServ network" *Proc. IFAC World Congr.* 12983–12987 (2008).

[26] N. Zhang, M. Yang, Y. Jing, and S. Zhang, "Congestion control for DiffServ network using second-order sliding mode control", *IEEE T. Ind. Electron.* 56 (9), 3330–3336 (2009).

[27] P. Ignaciuk and A. Bartoszewicz, "Discrete-time sliding-mode congestion control in multisource communication networks with time-varying delay", *T. Contr. Syst. T.* 19 (4), 852–867 (2011).

[28] P. Baburaj and B. Bandyopadhyay, "Discrete-time integral sliding-mode flow control for connection-oriented communication networks", *Proc. Int. Conf. Contr. Automat. Rob. Vision* 1, CD-ROM (2012).

[29] 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", *Proc. ACM SIGCOMM* 151-160 (2000).

[30] S. S. Kunniyur and R. Srikant, "End-to-end congestion control schemes: utility functions, random losses and ECN marks", *IEEE ACM T. Network.* 11 (5), 689–702 (2003).

[31] S. S. Kunniyur and R. Srikant, "An Adaptive Virtual Queue (AVQ) algorithm for active queue management", *IEEE ACM T. Network.* 12 (2), 286–299 (2004).

[32] P. Ignaciuk and A. Bartoszewicz, "Dead-beat and reaching-law-based sliding-mode control of perishable inventory systems", *Bull. Pol. Ac.: Tech.* 59 (1), 39–49 (2011).

[33] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, *Sliding Mode Control and Observation*, Springer, New York, 2014.

[34] *The Network Simulator* – ns-2. http://www.isi.edu/nsnam/ns/

[35] C. Hollot, V. Misra, D. Towsley, and W.-B. Gong, "On designing improved controllers for AQM routers supporting TCP flows", *Proc. INFOCOMM* 1, 1726–1734 (2001).

[36] M. Karbowańczyk and J. Żuk, "Quantity-based flow control strategy for connection-oriented communication networks", *Bull. Pol. Ac.: Tech.* 60 (2), 293–300 (2012).

[37] A. Bartoszewicz, T. Molik, and P. Ignaciuk, "Discrete time congestion controllers for multi-source connection-oriented communication networks", *Int. J. Contr.* 82 (7), 1237–1252 (2009).

[38] P. Ignaciuk and A. Bartoszewicz, "Flow control in connection-oriented networks – a time-varying sampling period system case study", *Kybernetika* 44 (3), 336–359 (2008).

[39] S. H. Low, F. Paganini, and J. C. Doyle, "Internet congestion control", *IEEE Contr. Syst. Mag.* 22 (1), 28–43 (2002).