Robust Congestion Controller for a Single Virtual Circuit in Connection-Oriented Communication Networks

Piotr Leśniewski and Andrzej Bartoszewicz

Abstract In this contribution, we consider the problem of data flow control for a single virtual connection in communication networks. The connection is described by the maximum link capacity, the non-negligible propagation delay and an unknown, time-varying data loss rate. We propose a discrete-time sliding mode controller, which generates non-negative and upper bounded transmission rates. In addition, it ensures that the queue length in the bottleneck link buffer is always limited. Moreover, with a sufficiently large memory buffer in the bottleneck node, it guarantees full utilization of the available bandwidth. The controller uses a dead-beat sliding hyperplane in order to ensure fast response to unknown changes of the link capacity and to an unpredictable data loss rate. However, if the straightforward dead-beat paradigm was used, unacceptably large transmission rates would be generated. Therefore, we use the reaching law approach in this chapter to decrease excessive magnitudes of the control signal at the start of the control process.

1 Introduction

In connection-oriented communication networks, data units are almost never sent directly from their source to their destination. Instead, they must pass through a series of intermediate nodes. When one of those nodes cannot pass on all of the received data, due to the limited bandwidth of its outgoing link, a part of the incoming data is stored in the memory buffer of this node, awaiting for later transmission. Such an event is called a congestion. In order to minimize queueing delays and to maximize throughput, congestion controllers must be applied [4–6, 15, 17, 19, 24, 26]. The difficulty in the design of congestion control algorithms is caused by rapid changes of the available bandwidth, unpredictable packet losses, and long propagation delays.

P. Leśniewski (✉) · A. Bartoszewicz
Institute of Automatic Control, Technical University of Łódź,
18/22 Bohdana Stefanowskiego St., 90-924 Łódź, Poland
e-mail: piotr.lesniewski2@gmail.com

A. Bartoszewicz
e-mail: andrzej.bartoszewicz@p.lodz.pl

A. Rauh and L. Senkel (eds.), *Variable-Structure Approaches*,
Mathematical Engineering, DOI 10.1007/978-3-319-31539-3_2

When the congestion of a specific link is observed, an information must be sent to all sources that are transmitting data using this link. However, the delivery of this information is not instantaneous, but occurs after the feedback propagation delays. Then data sources adjust their transmission rates in order to reduce the congestion. The adjusted rates begin to affect the congested link after the propagation delay of the sent data. Thus, in modern communication networks that are characterized by time delays and large bandwidth the need for efficient data flow control algorithms cannot be neglected.

Sliding mode control is a widely recognized methodology [8, 10, 25, 27, 29], suitable for a large group of nonlinear, time-varying and uncertain systems. The main advantages of sliding mode control are its high computational performance and robustness [9]. This technique was first designed for continuous-time systems. However, as an overwhelming majority of control algorithms are nowadays implemented in digital hardware, discrete-time sliding mode control [1–3, 7, 11, 14, 18, 20, 23, 28, 31] is an interesting and up-to-date research field. The main concept of sliding mode control is to force the representative point (state) of the system from its initial position towards the sliding hyperplane. The period when the state approaches the hyperplane is called the reaching phase. The controller should then enforce a sliding mode, during which the representative point moves on ("slides" along) the hyperplane or in its vicinity.

The first step in designing a sliding mode control is selecting the parameters of the hyperplane so as to obtain the desired performance of the closed-loop system. This can be done in various ways such as dead-beat design, quadratic optimization or the pole placement method. Then, a controller that guarantees the convergence to the hyperplane and the stability of the sliding mode is designed. There are two approaches to solving this problem. The first one begins with proposing a control law, and then proving, that the properties mentioned above are guaranteed when it is applied. However, in this contribution we will use the other approach, which is based on the reaching law. Using this approach, one first defines the desired evolution of the sliding variable. Then, the sliding mode control that enforces this evolution is derived. This methodology was presented for continuous-time systems in [12], and then extended to discrete-time systems in [13]. The approach presented in [13] has then become quite popular among sliding mode control researchers [14, 16, 21, 22, 30].

In this work, we design a discrete-time reaching law-based sliding mode congestion controller for connection-oriented networks. During the design process, we take into account not only unpredictable bandwidth changes and inevitable propagation delays, but also time-varying transmission losses.

The remainder of this work is organized as follows. In Sect. 2 we present the model of the virtual circuit. Then, we design the reaching law-based sliding mode control in Sect. 3. In Sect. 4 we demonstrate and prove analytically the important properties of the presented controller. Section 5 contains the results of computer simulations, that verify the performance of the controller. Finally, Sect. 6 comprises the conclusions of this work.

2 Network Model

Let us analyze a virtual circuit of a connection-oriented network. The circuit consists of a data source, some intermediate nodes and a destination. Figure 1 depicts the scheme of the model. There is a single bottleneck in the considered network, and a congestion controller is placed at the bottleneck node. This controller generates a signal (denoted by u), that determines the transmission rate of the source. This signal arrives at the source after the backward delay T_B, and upon receiving it, the source transmits the specified amount of data. The data are passed from node to node, until, after the forward delay T_F they reach the bottleneck node. Inevitably, during transmission, some data packets are lost, and so only αu data arrive at the congested node, where

$$0 < \alpha_{\min} \leq \alpha \leq \alpha_{\max} \leq 1. \tag{1}$$

The round trip time (RTT), which is the delay between generating the control signal and the requested data arrival at the bottleneck node, can be expressed as the sum of the backward and forward propagation delays

$$RTT = T_B + T_F. \tag{2}$$

We denote the discretization period by T, the bottleneck queue length at time kT by $y(kT)$, and its demand value by y_d. There are no data in the buffer before the start of the control process, i.e. $y(kT < 0) = 0$. Furthermore, it is assumed that the round trip time is a multiple of the discretization period, i.e. $RTT = mT$, where m is a positive integer. Since the first data will arrive at the queue after RTT, then $y(kT \leq RTT) = 0$.

The available bandwidth of the bottleneck link is modelled as an a priori unknown non-negative function of time $d(kT)$. Only the maximum value of this function denoted by $d_{\max}$ is known in advance. As sometimes the amount of data stored in the bottleneck node can be insufficient to fully utilize the bandwidth, an additional function $h(kT)$ representing the amount of data actually leaving the buffer at time kT is introduced. Therefore,

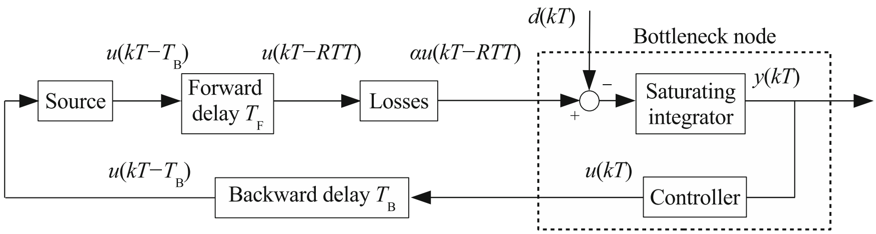

Fig. 1 Network model

$$0 \leq h(kT) \leq d(kT) \leq d_{\max} \tag{3}$$

for any $k \geq 0$.

We can represent the queue length as the difference between incoming and outgoing amounts of data, i.e.

$$y(kT) = \alpha \sum_{j=0}^{k-1} u(jT - RTT) - \sum_{j=0}^{k-1} h(jT) = \alpha \sum_{j=0}^{k-m-1} u(jT) - \sum_{j=0}^{k-1} h(jT). \tag{4}$$

The system can also be expressed using the standard state-space notation

$$\begin{aligned} \mathbf{x}[(k+1)T] &= \mathbf{A}\mathbf{x}(kT) + \boldsymbol{\delta}\mathbf{A}\mathbf{x}(kT) + \mathbf{b}u(kT) + \mathbf{o}h(kT) \\ y(kT) &= \mathbf{r}^T \mathbf{x}(kT), \end{aligned} \tag{5}$$

where $\mathbf{x}(kT) = [x_1(kT) \;\; x_2(kT) \ldots x_n(kT)]^T$ is the state vector, $y(kT) = x_1(kT)$ represents the queue length, and the remaining state variables are the delayed values of the control signal, i.e.

$$x_i(kT) = u[(k - n + i - 1)T] \tag{6}$$

for $i = 2, \ldots, n$, where $n = m + 1$. In (5) $\mathbf{A}$ is a $n \times n$ state matrix and $\boldsymbol{\delta}\mathbf{A}$ is a $n \times n$ model uncertainty matrix

$$\mathbf{A} = \begin{bmatrix} 1 & \alpha_{\max} & 0 & & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & & 0 \end{bmatrix}, \quad \boldsymbol{\delta}\mathbf{A} = \begin{bmatrix} 0 & \delta\alpha & 0 & & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & & 0 \end{bmatrix}, \tag{7}$$

where $\delta\alpha \in [\alpha_{\min} - \alpha_{\max}, 0]$, and $\mathbf{b}$, $\mathbf{o}$, and $\mathbf{r}$ are the following $n \times 1$ vectors

$$\mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{o} = \begin{bmatrix} -1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{r} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{8}$$

The desired state of the system is denoted by $\mathbf{x}_{\mathrm{d}} = [y_{\mathrm{d}} \;\; 0 \;\; \cdots \;\; 0]^T$, where $y_{\mathrm{d}} =$ const. The model may represent a single virtual circuit in Asynchronous Transfer Mode (ATM) or MultiProtocol Label Switching networks.

3 Non-Switching Reaching Law-Based SM Controller

We now derive a non-switching reaching law-based sliding mode controller, and apply it to the system described in the previous section. We start by choosing the sliding variable as

$$s(kT) = \mathbf{c}^T \mathbf{e}(kT), \tag{9}$$

where

$$\mathbf{e}(kT) = \mathbf{x}_\mathrm{d} - \mathbf{x}(kT) \tag{10}$$

is the closed-loop control error. With this choice of the switching variable s the equation $s(kT) = 0$ determines the sliding hyperplane. The vector $\mathbf{c}$ is chosen so that $\mathbf{c}^T\mathbf{b} \neq 0$ and the closed-loop system exhibits the desired performance. We select the vector $\mathbf{c}$ so that the closed-loop system exhibits dead-beat dynamics to obtain finite-time error convergence to zero. We begin by calculating the value of the sliding variable in the next time instant. As we are interested in the poles of the closed-loop system, we omit the disturbance and the modelling uncertainty terms. Using Eqs. (5) and (9) we arrive at

$$s[(k+1)T] = \mathbf{c}^T \left[\mathbf{x}_\mathrm{d} - \mathbf{A}\mathbf{x}(kT) - \mathbf{b}u(kT)\right]. \tag{11}$$

We find that the control signal that satisfies $s[(k+1)T] = 0$ is given by

$$u(kT) = \left(\mathbf{c}^T\mathbf{b}\right)^{-1} \mathbf{c}^T \left[\mathbf{x}_\mathrm{d} - \mathbf{A}\mathbf{x}(kT)\right]. \tag{12}$$

Let us point out, that control signal (12) is calculated only to find the form of vector $\mathbf{c}$ that ensures dead-beat dynamics in sliding mode. This signal is not used in the final reaching law-based sliding mode controller. By substituting the control signal (12) into (5) we obtain the closed-loop system matrix as

$$\mathbf{A}_\mathrm{c} = \left[\mathbf{I}_\mathrm{n} - \mathbf{b}(\mathbf{c}^T\mathbf{b})^{-1}\mathbf{c}^T\right]\mathbf{A}. \tag{13}$$

The matrix $\mathbf{A}_\mathrm{c}$ has the following characteristic polynomial

$$\det\left(z\mathbf{I}_\mathrm{n} - \mathbf{A}_\mathrm{c}\right) = z^n + \frac{c_{n-1} - c_n}{c_n} z^{n-1} + \cdots + \frac{c_2 - c_3}{c_n} z^2 + \frac{\alpha_{\max} c_1 - c_2}{c_n} z. \tag{14}$$

A discrete-time system is asymptotically stable if and only if all of its eigenvalues are located inside the unit circle. Moreover, to obtain dead-beat performance all of the eigenvalues must be located in the origin of the z-plane. Therefore, the characteristic polynomial must have the following form

$$\det\left(z\mathbf{I}_{\mathrm{n}} - \mathbf{A}_{\mathrm{c}}\right) = kz^{n}, \tag{15}$$

where k is an arbitrary constant. We find that (14) simplifies to (15) when the following vector $\mathbf{c}$ is chosen

$$\mathbf{c} = \begin{bmatrix} \frac{1}{\alpha_{\max}} & 1 & 1 & \cdots & 1 \end{bmatrix}^{T}. \tag{16}$$

The aim of the controller is to decrease the absolute value of the switching variable until it reaches a band around $s(kT) = 0$, which is further called the quasi-sliding band. After entering this band, the switching variable should never leave it again. Contrary to some previous works [13], in the chosen definition of the quasi-sliding mode, crossing the hyperplane is allowed but not necessary. As it will be demonstrated later, this modification allows us to eliminate the undesirable phenomenon of chattering.

After determining the appropriate sliding hyperplane coefficients we now consider the following reaching law, which describes the desired evolution of the sliding variable

$$s[(k+1)T] = \{1 - q[s(kT)]\}s(kT) - \tilde{F}(kT) - \tilde{S}(kT) + F_1, \tag{17}$$

where

$$\tilde{S}(kT) = \mathbf{c}^{T}\boldsymbol{\delta}\mathbf{A}\mathbf{x}(kT) = \frac{\delta\alpha x_2(kT)}{\alpha_{\max}} \tag{18}$$

is the impact of the model uncertainty (the unknown and time-varying transmission losses). Function

$$\tilde{F}(kT) = \mathbf{c}^{T}\mathbf{o}h(kT) = -\frac{h(kT)}{\alpha_{\max}} \tag{19}$$

represents the influence of the disturbance (in our case the outgoing flow of data) on the sliding variable. The term

$$F_1 = -\frac{d_{\max}}{2\alpha_{\max}} \tag{20}$$

is employed to compensate for the mean value of $\tilde{F}(kT)$. The term $S_1 = \frac{S_U + S_L}{2}$ (where S_U and S_L are the upper and lower bounds of $\tilde{S}(kT)$), used in some earlier works [13] to compensate for the mean value of the model uncertainty, could lead to generating negative control signals. As in the considered system this is not feasible, we discard this term. The variable convergence rate $q[s(kT)]$ is given by

$$q[s(kT)] = \frac{s_0}{s_0 + |s(kT)|}, \tag{21}$$

where

$$s_0 > \frac{d_{\text{max}}(2\alpha_{\text{max}} - \alpha_{\text{min}})}{2\alpha_{\text{max}}\alpha_{\text{min}}} \tag{22}$$

is a design parameter, that must satisfy the above inequality in order to ensure the convergence to the vicinity of the sliding hyperplane. This is another modification of the reaching law developed by Gao et al. [13] as in their work the parameter q was constant. As we can observe from (21), the parameter $q[s(kT)]$ decreases with the increase of $|s(kT)|$. Therefore, the value of the control signal needed to ensure the desired sliding variable evolution (17) will not change so dramatically for different absolute values of the sliding variable. This allows us to find a better compromise between acceptable control signal values when $|s(kT)|$ is large and fast convergence in the vicinity of the sliding hyperplane.

We now calculate the control signal that makes the sliding variable actually evolve according to (17). We will start by using (5) to rewrite (9) as

$$s[(k+1)T] = \mathbf{c}^T\mathbf{x}_{\text{d}} - \mathbf{c}^T\left[\mathbf{A}\mathbf{x}(kT) + \boldsymbol{\delta}\mathbf{A}\mathbf{x}(kT) + \mathbf{b}u(kT) + \mathbf{o}h(kT)\right]. \tag{23}$$

By comparing the right-hand sides of (17) and (23) we obtain

$$u(kT) = (\mathbf{c}^T\mathbf{b})^{-1}\left\{q[s(kT)]s(kT) + \frac{d_{\text{max}}}{2\alpha_{\text{max}}} - \mathbf{c}^T(\mathbf{A} - \mathbf{I}_{\text{n}})\mathbf{x}(kT)\right\}. \tag{24}$$

We notice, that by choosing $\mathbf{c}$ according to (16), we have obtained

$$\mathbf{c}^T(\mathbf{A} - \mathbf{I}_{\text{n}}) = [0 \quad \ldots \quad 0]. \tag{25}$$

Using (8), (16), (21) and (25) we can rewrite (24) as

$$u(kT) = \frac{s_0 s(kT)}{s_0 + |s(kT)|} + \frac{d_{\text{max}}}{2\alpha_{\text{max}}}. \tag{26}$$

In this way, we have completed the design of the reaching law-based sliding mode congestion controller. It will be shown in the next section that control strategy (26) as opposed to (12) ensures upper bounded transmission data rates, which is a highly desirable property in the application considered in this chapter.

4 Properties of the System

In this section we demonstrate that the application of the proposed controller guarantees several important properties of the considered system.

Theorem 1 *The sliding variable satisfies*

$$s(kT) \geq -\frac{d_{\max} s_0}{2\alpha_{\max} s_0 - d_{\max}} \tag{27}$$

for all $k \geq 0$.

Proof The sliding variable in the initial time instant is equal to $y_{\mathrm{d}}/\alpha_{\max}$, and thus satisfies (27). Therefore, in order to prove the theorem, it is sufficient to demonstrate that if (27) holds for every $k \leq l$, then it also holds for $k = l + 1$.

We observe from (26) that the control signal value always increases with the increase of $s(kT)$. Therefore, substituting the right-hand side of (27) into (26), we observe, that $u(kT) \geq 0$ for all $k \leq l$. This in turn implies that $\tilde{S}(kT) \leq 0$ for all $k \leq l$. Moreover, we observe from (19) that $\tilde{F}(kT) \leq 0$ for all $k \geq 0$.

Substituting the right-hand side of (27) and the worst-case scenario values $\tilde{F}(kT) = 0$, $\tilde{S}(kT) = 0$ into (17), we notice that indeed if (27) holds for all $k \leq l$ then it must also hold for $k = l + 1$. Using this observation and the principle of mathematical induction we conclude, that (27) will hold for all $k \geq 0$.

Since (27) holds for all $k \geq 0$ then, following the reasoning from Theorem 1, the control signal is always non-negative. In the next theorem we show that this control signal is also upper bounded by an a priori known value. As the amount of data sent by the source during a single discretization period is limited by the capacity of the outgoing link (and obviously cannot be negative) both of these properties are crucial for applying the flow controller in a real network.

Theorem 2 *The control signal satisfies the inequality*

$$u(kT) \leq \frac{y_d s_0}{s_0 \alpha_{\max} + y_d} + \frac{d_{\max}}{2\alpha_{\max}} \tag{28}$$

for any $k \geq 0$.

Proof The outgoing amount of data $h(kT)$ is defined in such a way that $x_1(kT) \geq 0$ for any $k \geq 0$. Furthermore, since the control signal is always non-negative, then we notice from (6) that $x_i(kT) \geq 0$ for $i = 2, \ldots, n$ and all $k \geq 0$. Using these observations together with (9) and (16), we see that the sliding variable cannot exceed its initial value, i.e.

$$s(kT) \leq \frac{y_{\mathrm{d}}}{\alpha_{\max}} \tag{29}$$

for all $k \geq 0$. As already stated the control signal always increases with the increase of $s(kT)$. Therefore, by substituting the right-hand side of inequality (29) into (26) we obtain the greatest possible value of $u(kT)$ and find that (28) indeed holds.

Theorem 3 *Once the inequality*

$$s(kT) \leq \frac{s_0\left[(\alpha_{\max}-\alpha_{\min})\left(\frac{y_d s_0}{s_0\alpha_{\max}+y_d}+\frac{d_{\max}}{2\alpha_{\max}}\right)+\frac{d_{\max}}{2}\right]}{\alpha_{\max}s_0-\left[(\alpha_{\max}-\alpha_{\min})\left(\frac{y_d s_0}{s_0\alpha_{\max}+y_d}+\frac{d_{\max}}{2\alpha_{\max}}\right)+\frac{d_{\max}}{2}\right]} \tag{30}$$

is satisfied, it remains true for the remainder of the control process.

Proof As $x_2(kT)$ is a delayed value of the control signal, using (28) with (18) we get

$$\tilde{S}(kT) \geq \frac{\alpha_{\min}-\alpha_{\max}}{\alpha_{\max}}\left(\frac{y_{\mathrm{d}} s_0}{s_0\alpha_{\max}+y_{\mathrm{d}}}+\frac{d_{\max}}{2\alpha_{\max}}\right) \tag{31}$$

for all $k \geq 0$. Furthermore, we notice from (19) that

$$\tilde{F}(kT) \geq -\frac{d_{\max}}{\alpha_{\max}} \tag{32}$$

also for all $k \geq 0$.

Substituting these lower bounds of $\tilde{S}(kT)$ and $\tilde{F}(kT)$ and the right-hand side of inequality (30) into (26), we conclude that indeed once (30) holds, it will remain true for the remainder of the control process.

Remark 1 Taking into account the results of Theorems 1 and 3, we observe that the sliding variable will converge to the quasi-sliding mode band defined by

$$\begin{aligned}\frac{-d_{\max}s_0}{2\alpha_{\max}s_0-d_{\max}} &\leq s(kT)\\ &\leq \frac{s_0\left[(\alpha_{\max}-\alpha_{\min})\left(\frac{y_{\mathrm{d}} s_0}{s_0\alpha_{\max}+y_{\mathrm{d}}}+\frac{d_{\max}}{2\alpha_{\max}}\right)+\frac{d_{\max}}{2}\right]}{\alpha_{\max}s_0-\left[(\alpha_{\max}-\alpha_{\min})\left(\frac{y_{\mathrm{d}} s_0}{s_0\alpha_{\max}+y_{\mathrm{d}}}+\frac{d_{\max}}{2\alpha_{\max}}\right)+\frac{d_{\max}}{2}\right]},\end{aligned} \tag{33}$$

and after entering it, it will never leave it again.

Buffer overflows occur when the incoming data and the current amount of stored data exceed the buffer capacity of the congested node. As this event leads to data losses and causes the need for retransmission, it is highly undesirable in communication networks. In the next theorem we determine the size, which the queue length at the bottleneck node will never exceed. If the buffer capacity is equal to or greater than this value, then there is no risk of buffer overflows.

Theorem 4 *The queue length never exceeds the following value*

$$y(kT) \leq y_d + \frac{d_{\max}s_0}{2s_0-d_{\max}/\alpha_{\max}}. \tag{34}$$

Proof Using (9) we can rewrite (27) as

$$y(kT) \leq y_{\mathrm{d}} + \frac{d_{\max} s_0}{2s_0 - d_{\max}/\alpha_{\max}} - \alpha_{\max} \sum_{i=2}^{n} x_i(kT). \tag{35}$$

All of the state variables, except for the first one are the delayed values of the control signal. As already demonstrated, the control signal is always non-negative, therefore (35) implies (34).

In order for the flow control algorithm to be efficient, it should result in the greatest throughput that is possible in the network. In Theorem 5, we calculate the smallest value of the demand queue length that guarantees, that the queue length remains always strictly positive, after the first data reach it. This property corresponds to 100 % utilization of the available bandwidth.

Theorem 5 *If the demand queue length satisfies*

$$y_d > \frac{\alpha_{\max} s_0 d_{\max} (2\alpha_{\max} - \alpha_{\min})}{2\alpha_{\max}\alpha_{\min} s_0 - d_{\max}(2\alpha_{\max} - \alpha_{\min})} + \frac{\alpha_{\max} d_{\max} (n-1)}{\alpha_{\min}}, \tag{36}$$

then the queue length will be strictly positive for any $k \geq n$.

Proof To keep the notation clear, we introduce

$$\theta = \frac{2\alpha_{\max} - \alpha_{\min}}{2\alpha_{\max}\alpha_{\min}}. \tag{37}$$

In the proof we will consider two possible ranges of the sliding variable separately

Case 1 If

$$s[(k-n+1)T] > \frac{s_0 d_{\max}\theta}{s_0 - d_{\max}\theta}, \tag{38}$$

then the queue length for any $k \geq n-1$ can be calculated, using (4) as

$$\begin{aligned} y[(k+1)T] &= y(kT) + \alpha u[(k-n+1)T] - h(kT) \\ &\geq y(kT) + \alpha_{\min}\left[\frac{s_0 s[(k-n+1)T]}{s_0 + |s[(k-n+1)T]|} + \frac{d_{\max}}{2\alpha_{\max}}\right] - d_{\max}. \end{aligned} \tag{39}$$

The expression in square brackets in (39) always increases with the increase of $s[(k-n+1)T]$. Therefore, substituting (38) we get

$$\begin{aligned} y[(k+1)T] &> (s_0 - d_{\max}\theta)\frac{d_{\max}\theta}{s_0 - d_{\max}\theta} + \frac{d_{\max}}{2\alpha_{\max}} \\ &\quad + y(kT) - d_{\max} = y(kT) + d_{\max} - d_{\max} \geq 0. \end{aligned} \tag{40}$$

Case 2 On the other hand, if

$$s[(k-n+1)T] \leq \frac{s_0 d_{\max}\theta}{s_0 - d_{\max}\theta}, \tag{41}$$

then the queue length for $k \geq n-1$ can be expressed as

$$\begin{aligned} y[(k+1)T] &= y[(k-n+1)T] + \alpha \sum_{i=2}^{n} x_i[(k-n+1)T] - \sum_{i=k-n}^{k} h(iT) \\ &\geq y[(k-n+1)T] + \alpha_{\min} \sum_{i=2}^{n} x_i[(k-n+1)T] - (n-1)d_{\max}. \end{aligned} \tag{42}$$

Furthermore, we obtain from (9) and (16) that

$$s(kT) = \frac{y_{\mathrm{d}}}{\alpha_{\max}} - \frac{y(kT)}{\alpha_{\max}} - \sum_{i=2}^{n} x_i(kT). \tag{43}$$

Multiplying both sides of (43) by $\alpha_{\min}$ and rearranging the terms one obtains

$$\frac{\alpha_{\min} y(kT)}{\alpha_{\max}} + \alpha_{\min} \sum_{i=2}^{n} x_i(kT) = \frac{\alpha_{\min} y_{\mathrm{d}}}{\alpha_{\max}} - \alpha_{\min} s(kT). \tag{44}$$

As $\alpha_{\min}/\alpha_{\max} \leq 1$, and the queue length cannot be negative, we can observe, that

$$y(kT) + \alpha_{\min} \sum_{i=2}^{n} x_i(kT) \geq \frac{\alpha_{\min} y_{\mathrm{d}}}{\alpha_{\max}} - \alpha_{\min} s(kT). \tag{45}$$

Substituting (45) and (41) into (42), we get

$$y[(k+1)T] \geq \frac{\alpha_{\min} y_{\mathrm{d}}}{\alpha_{\max}} - \alpha_{\min} \frac{s_0 d_{\max}\theta}{s_0 - d_{\max}\theta}. \tag{46}$$

We find, that if (36) holds, then the right-hand side of the above inequality is always strictly positive. This conclusion, together with the result of Case 1 finalizes the proof.

5 Simulation Results

To verify the properties of the proposed control strategy, computer simulations were performed. The round trip time RTT =12 ms and the discretization period $T = 1$ ms. Therefore, $m = 12$ and $n = 13$. The packet loss ratio is bounded by $\alpha_{\min} = 0.82$ and

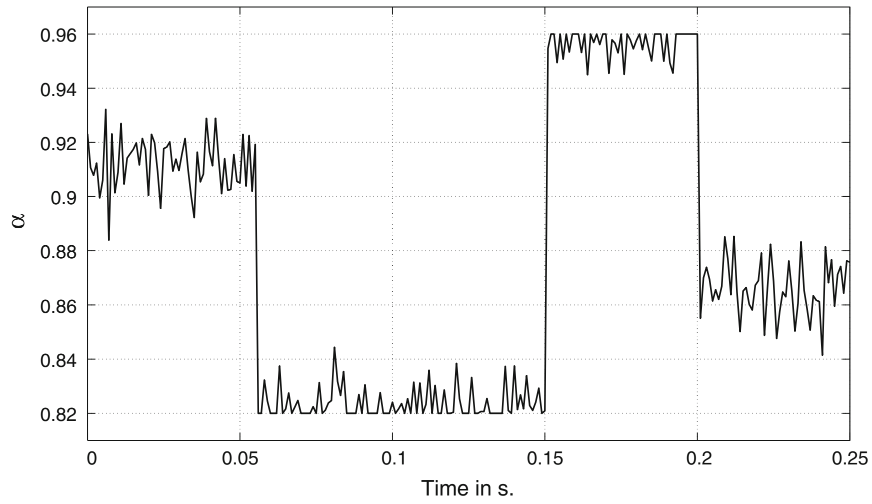

Fig. 2 Data loss rate

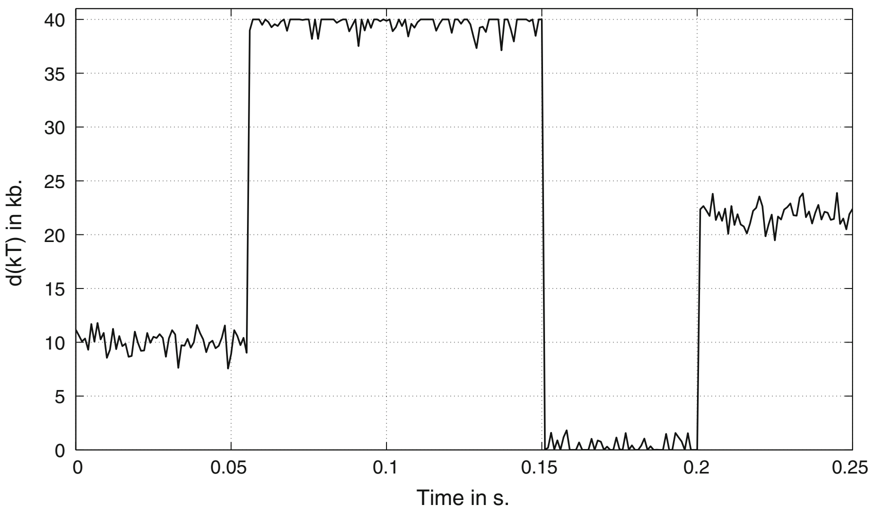

Fig. 3 Available bandwidth

$\alpha_{\max} = 0.96$, and the actual transient of the loss ratio is depicted in Fig. 2. The parameter $d_{\max} = 40$ kb and the available bandwidth chosen for the simulation is shown in Fig. 3. Both of the unpredictable functions exhibit rapid changes between small and large values which correspond to the most difficult conditions that could arise in the system. Note that we have taken into account both of the worst-case scenario combinations, namely

- $d(kT)$ is close to its maximum value, while α is close to minimum between 0.05 and 0.15 s
- $d(kT)$ is close to its minimum value, while α is close to maximum between 0.15 and 0.2 s.

Let us point out that irrespective of the chosen control algorithm, the upper bound of the transmission rate of the data source must satisfy

$$u_{\max} \geq \frac{d_{\max}}{\alpha_{\min}} \tag{47}$$

in order to fully utilize the available bandwidth of the bottleneck node. This can be observed as follows: imagine a situation, in which the available bandwidth is equal to $d_{\max}$, and the transmission loss parameter to $\alpha_{\min}$. Generating a constant control signal, that would be smaller then the right-hand side of (47) would result in less data arriving at the buffer, then leaving it in every discretization period. In this way, the buffer would become empty after some time, and the available bandwidth would not be fully utilized.

The choice of controller parameters y_d and s_0 will now be considered. The generated control signal will never exceed the right-hand side of inequality (28). As this signal is upper bounded by $u_{\max}$ we obtain the following condition on the controller parameters

$$\frac{y_d s_0}{s_0\alpha_{\max} + y_d} \leq u_{\max} - \frac{d_{\max}}{2\alpha_{\max}}. \tag{48}$$

The expression $s_0\alpha_{\max} + y_d$ is always positive, therefore it follows from (48) that

$$y_d\left(s_0 - u_{\max} + \frac{d_{\max}}{2\alpha_{\max}}\right) \leq \left(u_{\max} - \frac{d_{\max}}{2\alpha_{\max}}\right) s_0\alpha_{\max}. \tag{49}$$

Depending on the choice of s_0, the term $s_0 - u_{\max} + \frac{d_{\max}}{2\alpha_{\max}}$ can be positive, negative or equal to zero. We will now analyze these three cases separately.

Case 1 If

$$s_0 > u_{\max} + \frac{d_{\max}}{2\alpha_{\max}}, \tag{50}$$

then it follows from (49) that parameter y_d must satisfy

$$y_d \leq \frac{\left(u_{\max} - \frac{d_{\max}}{2\alpha_{\max}}\right) s_0\alpha_{\max}}{s_0 - u_{\max} + \frac{d_{\max}}{2\alpha_{\max}}} \tag{51}$$

so that the control signal will never exceed $u_{\max}$.

Case 2 If

$$s_0 < u_{\max} + \frac{d_{\max}}{2\alpha_{\max}}, \tag{52}$$

then parameter y_{d} must satisfy

$$y_{\mathrm{d}} \geq \frac{\left(u_{\max} - \frac{d_{\max}}{2\alpha_{\max}}\right) s_0 \alpha_{\max}}{s_0 - u_{\max} + \frac{d_{\max}}{2\alpha_{\max}}} \tag{53}$$

so that the maximum value of the control signal will not exceed $u_{\max}$. It is easy to observe using (47) and (52), that the right-hand side of inequality (53) is strictly negative. Therefore, any positive value of the demand queue length y_{d} satisfies this inequality.

Case 3 We now consider

$$s_0 = u_{\max} + \frac{d_{\max}}{2\alpha_{\max}}. \tag{54}$$

Using (47) we notice that the right-hand side of (49) is strictly positive. Therefore, for s_0 given by (54) any value of y_d ensures that the control signal will not exceed $u_{\max}$.

To sum up the three cases: for all $s_0 \leq u_{\max} + \frac{d_{\max}}{2\alpha_{\max}}$ any positive value of y_d ensures that the control signal will not exceed $u_{\max}$. On the other hand, if $s_0 > u_{\max} + \frac{d_{\max}}{2\alpha_{\max}}$, then the demand queue length y_{d} must be selected according to (51).

It is assumed that the source can send a maximum of 54 kb of data in a single discretization period. Moreover, inequality (36) must be satisfied to achieve full bandwidth utilization. The choice of parameters s_0 and y_{d} is depicted in Figs. 4 and 5. The dashed black line in Fig. 4 corresponds to $s_0 = u_{\max} + \frac{d_{\max}}{2\alpha_{\max}}$. In both figures the solid black line reflects the smallest y_{d} that satisfies condition (36), while the dotted black line represents the greatest value of y_{d} that for a given s_0 ensures, that the control signal will not exceed $u_{\max} = 54$ kb. Therefore, to achieve maximum throughput and at the same time not exceed the data source capabilities, one should select a combination of parameters that is below the dotted black line and above the solid black one. The set of admissible parameter combinations is marked in grey. The last question is which particular point from this set should be chosen. It is clear from inequality (34) that the maximum bottleneck buffer queue length decreases with the increase of s_0 and decrease of y_{d}. Therefore, to reduce memory requirements, one should select the combination of parameters that is close to the intersection of the solid and dotted lines. Motivated by this reasoning, we have selected $s_0 = 34.72$ kb and $y_{\mathrm{d}} = 710$ kb.

The results of the simulation are shown in Figs. 6, 7 and 8. Figure 6 shows the control signal. As predicted by Theorems 1 and 2, it never exceeds 54 kb and is always non-negative. Furthermore, when comparing Figs. 3 and 6 we observe, that although the control signal tracks the value of the available bandwidth, the rapid changes have

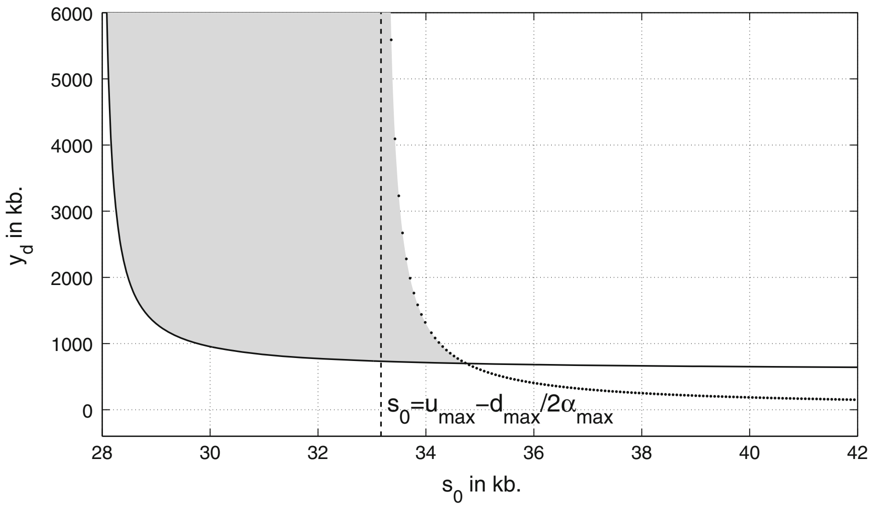

Fig. 4 The admissible set of parameter combinations for the first simulation scenario

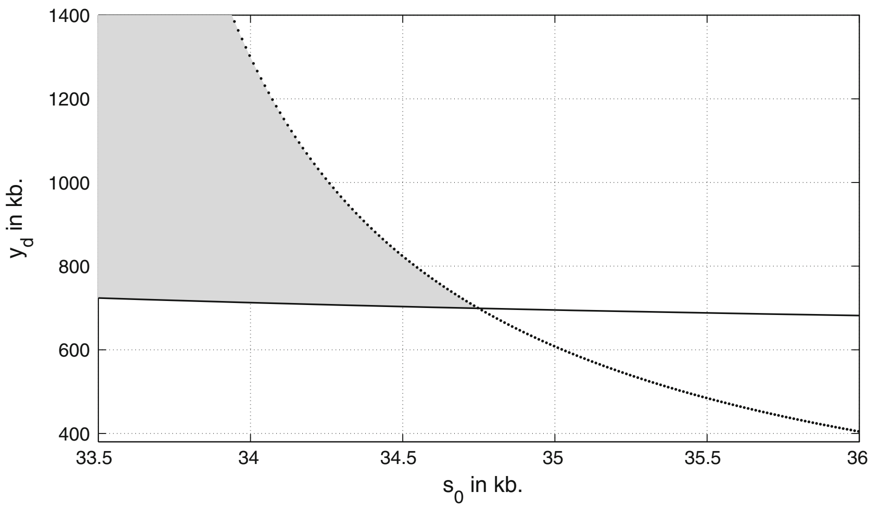

Fig. 5 The enlarged neighbourhood of the selected parameter values

been smoothed out, which is advantageous for the general transmission efficiency in the network. The bottleneck queue length is shown in Fig. 7. It never exceeds the value of 760 kb predicted by Theorem 4 and as shown in Theorem 5, after the first data reach it, it never drops to zero. Therefore, the risk of buffer overflow is eliminated and full utilization of the available bandwidth is guaranteed. Figure 8 depicts the evolution of the sliding variable. As predicted by Theorem 1 it is always greater

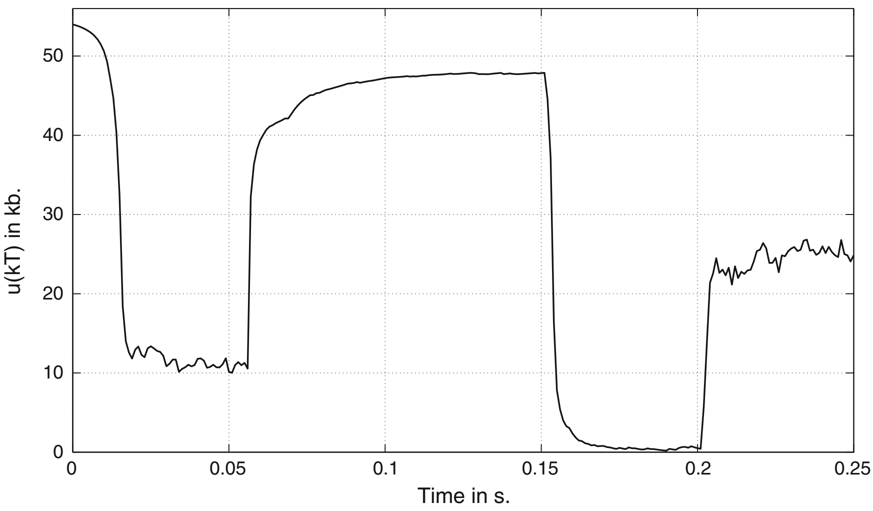

Fig. 6 Control signal

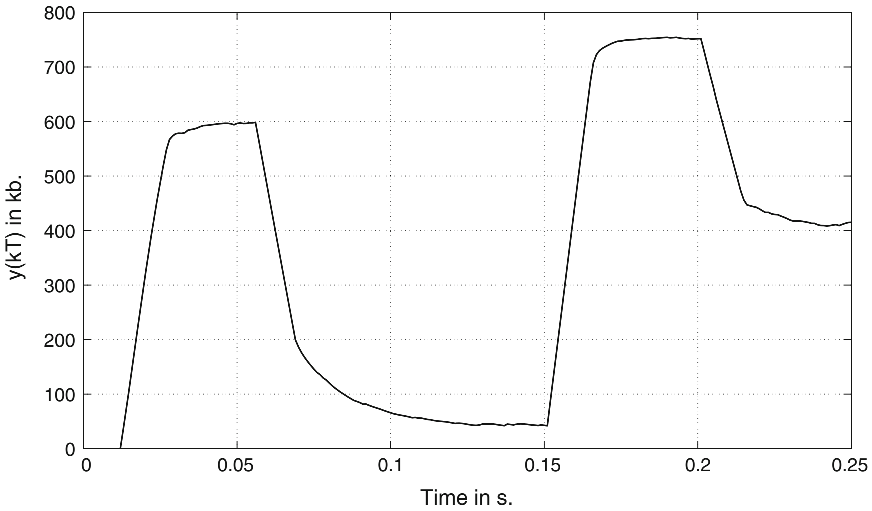

Fig. 7 Bottleneck node queue length

than $-52.08\,\text{kb}$, and as demonstrated in Theorem 3 once it drops below 370 kb it never exceeds this value again.

In the second simulation scenario the available memory capacity at the bottleneck node is insufficient to enable full bandwidth utilization. The buffer capacity is $y_{\max} = 620\,\text{kb}$. Therefore, in order to prevent buffer overflows, we have to select such values

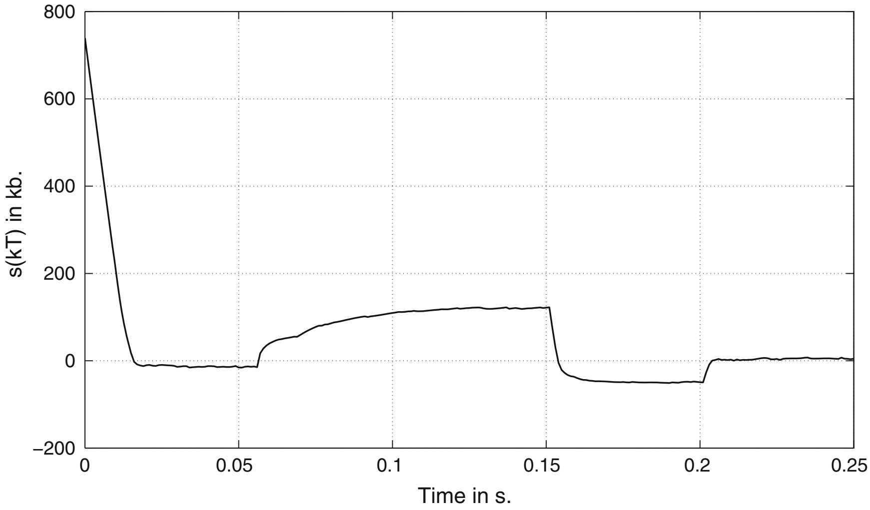

Fig. 8 Sliding variable evolution

of y_d and s_0 that the right-hand side of inequality (34) does not exceed y_{max}. This corresponds to satisfying the following condition

$$y_d \leq y_{max} - \frac{d_{max} s_0}{2s_0 - d_{max}/\alpha_{max}} \tag{55}$$

The choice of parameters for this simulation scenario is presented in Fig. 9. As in the previous case, the dotted black line represents the greatest value of y_d that prevents from exceeding the admissible control signal range, and the solid black line corresponds to the smallest values of y_d that satisfy condition (36). Moreover, the dashed black line corresponds to points in which the right-hand side of (55) is equal to the left-hand side of this inequality. Therefore, one should select parameter values that are below this line to eliminate the risk of buffer overflows. As we can observe from Fig. 9 it is impossible to select a combination of parameters which satisfies all the three requirements at the same time, namely there are no points that are at the same time below both the dotted black line and the dashed black line and above the solid black line. Therefore, in order to prevent buffer overflows, we select the intersection of the dashed line with the dotted black line. In this way we fully utilize the available control signal range without exceeding it. The risk of buffer overflows is also eliminated. However, as this intersection point lies below the solid black line, we do not obtain full utilization of the available bandwidth. This will be apparent in the simulation results.

The results of simulations are shown in Figs. 10, 11, 12 and 13. Figure 10 depicts the control signal, which again is always non-negative and never exceeds the available value of 54 kb. When compared to the previous scenario, we observe, that the changes

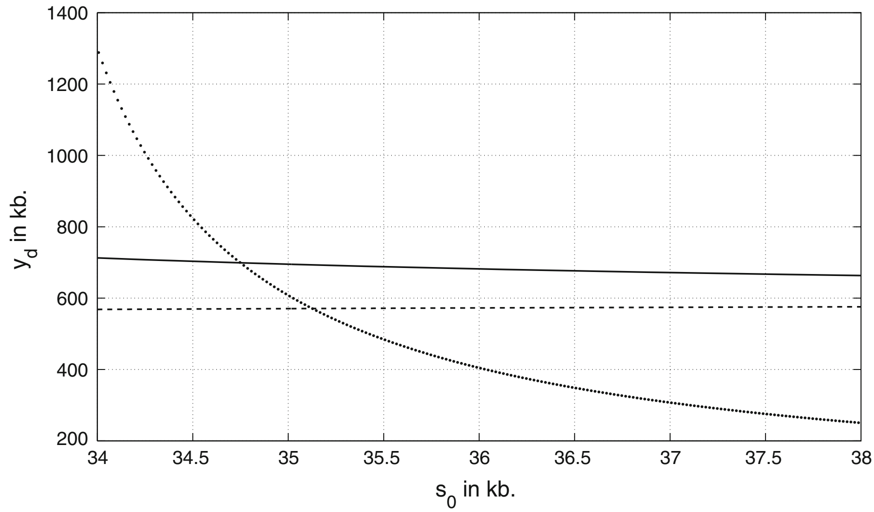

Fig. 9 The choice of parameters for the reduced bottleneck node memory capacity

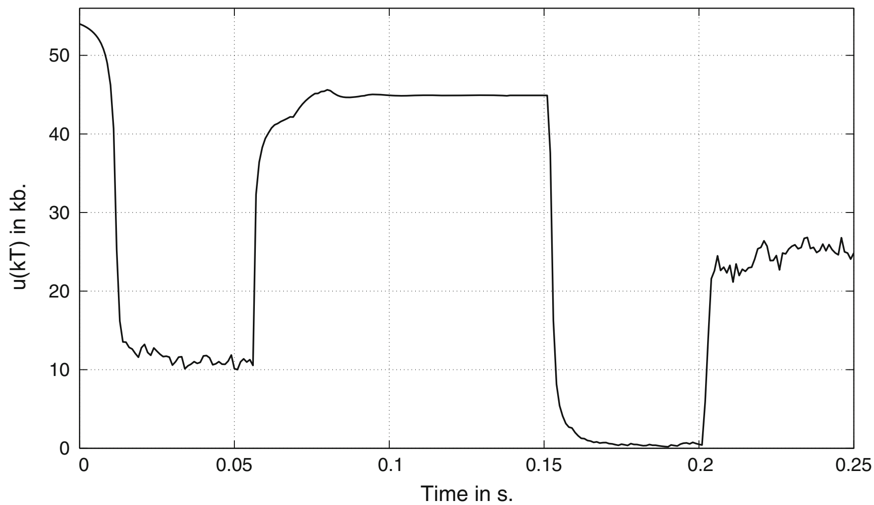

Fig. 10 Control signal for the reduced bottleneck node memory capacity

of the control signal are slightly more rapid. This is due to the fact, that the smaller available memory capacity can be used to smooth the changes in bandwidth to a lesser extent. The bottleneck node queue length is shown in Fig. 11. It never exceeds the value of 620 kb predicted by Theorem 4, but in contrast to the previous case, because of the insufficient memory capacity, at some times the queue is empty. When this occurs, the available bandwidth is not fully utilized, and we depict the unused part of the bandwidth in Fig. 12. The unused bandwidth at the start of the control process

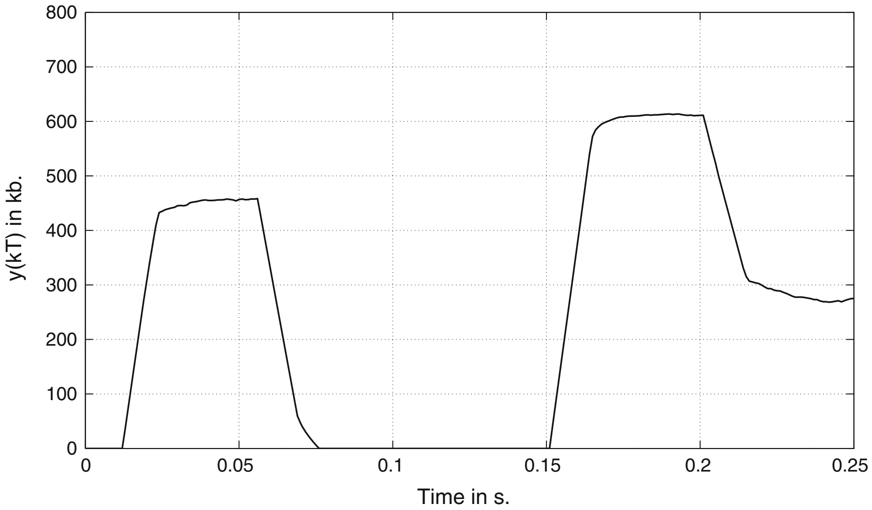

Fig. 11 Bottleneck node queue length for the reduced bottleneck node memory capacity

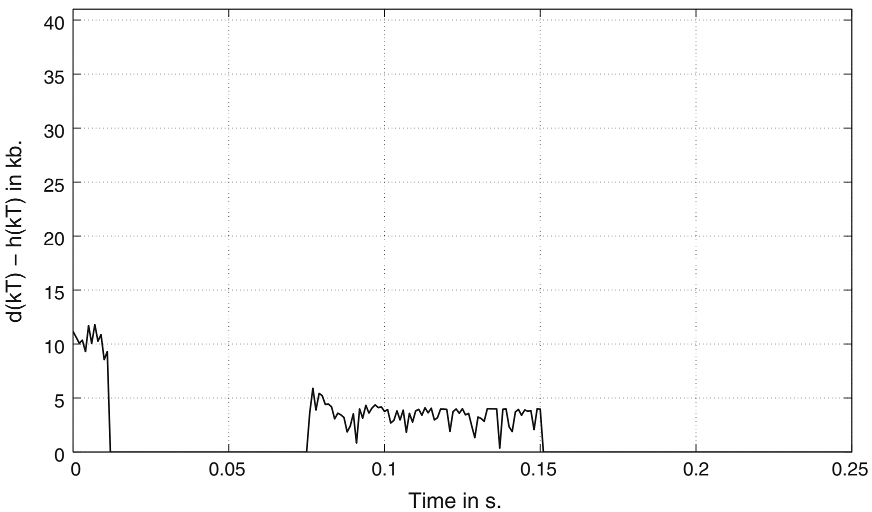

Fig. 12 Unused bandwidth for the reduced bottleneck node memory capacity

occurs before the first data arrive at the bottleneck node, and therefore cannot be eliminated by any control strategy. As we can notice from Fig. 12 the bandwidth after the first round trip time passes is still utilized fairly well, despite the limited buffer size. Figure 13 shows the evolution of the sliding variable. Again, as predicted in Sect. 4, it never drops below -51.14 kb and once it gets smaller than 328 kb it never exceeds this value again.

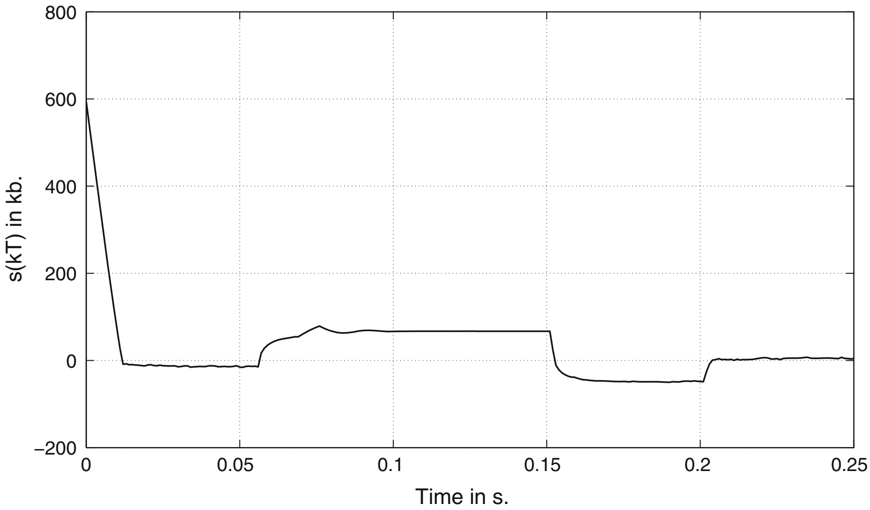

Fig. 13 Sliding variable evolution for the reduced bottleneck node memory capacity

6 Conclusions

In this chapter we have presented a robust sliding mode control strategy for congestion control of a single virtual circuit in connection-oriented communication networks. We have applied the reaching law methodology and the dead-beat control technique in designing the control law. The presented controller enforces a chattering free quasi-sliding mode. Furthermore, it eliminates the risk of data losses, and can ensure full utilization of the bottleneck link bandwidth, in spite of rapid, unpredictable changes of the transmission loss rate and the available bandwidth. In our future work we will extend the presented approach to the case of multisource data transmission networks with varying time delays.

Acknowledgments This work has been performed in the framework of a project "Optimal sliding mode control of time delay systems" financed by the National Science Centre of Poland decision number DEC 2011/01/B/ST7/02582. Kind support provided by the Foundation for Polish Science under Mistrz grant is also acknowledged.

References

1. Abidi K, Xu JX, Yu X (2007) On the discrete-time integral sliding-mode control. IEEE Trans Autom Control 52(4):709–715
2. Bandyopadhyay B, Fulwani D (2009) High-performance tracking controller for discrete plant using nonlinear sliding surface. IEEE Trans Ind Electron 56(9):3628–3637

3. Bartolini G, Ferrara A, Utkin V (1995) Adaptive sliding mode control in discrete-time systems. Automatica 31:769–773
4. Bartoszewicz A (2006) Nonlinear flow control strategies for connection oriented communication networks. Proc IEE Part D: Control Theory Appl 153(1):21–28
5. Bartoszewicz A, Leśniewski P (2014) An optimal sliding mode congestion controller for connection-oriented communiction networks with lossy links. Int J Appl Math Comput Sci 24(1):87–97
6. Bartoszewicz A, Leśniewski P (2014) Reaching law-based sliding mode congestion control for communication networks. IET Control Theory Appl 8(17):1914–1920
7. Corradini ML, Orlando G (1998) Variable structure control of discretized continuous-time systems. IEEE Trans Autom Control 43(9):1329–1334
8. DeCarlo RS, Żak S, Mathews G (1988) Variable structure control of nonlinear multivariable systems: a tutorial. Proc of the IEEE 76:212–232
9. Draženović B (1969) The invariance conditions in variable structure systems. Automatica 5:287–295
10. Edwards C, Spurgeon S (1998) Sliding mode control: theory and applications. Taylor & Francis, London
11. Galias Z, Yu X (2008) Analysis of zero-order holder discretization of two-dimensional sliding mode control systems. IEEE Trans Circuits Syst II 55(12):1269–1273
12. Gao W, Hung J (1993) Variable structure control of nonlinear systems: a new approach. IEEE Trans Ind Electron 40(1):45–55
13. Gao W, Wang Y, Homaifa A (1995) Discrete-time variable structure control systems. IEEE Trans Ind Electron 42(2):117–122
14. Golo G, Milosavljević C (2000) Robust discrete-time chattering free sliding mode control. Syst Control Lett 41(1):19–28
15. Jagannathan S, Talluri J (2002) Predictive congestion control of ATM networks: multiple sources/single buffer scenario. Automatica 38(5):815–820
16. Janardhanan S, Bandyopadhyay B (2007) Multirate output feedback based robust quasi-sliding mode control of discrete-time systems. IEEE Trans Autom Control 52(3):499–503
17. Jing Y, Yu N, Kong Z, Dimirovski G (2008) Active queue management algorithm based on fuzzy sliding model controller. In: Proceedings of the 17th IFAC world congress, pp 6148–6153
18. Kurode S, Bandyopadhyay B, Gandhi P (2011) Discrete sliding mode control for a class of underactuated systems. In: Proceedings of 37th annual conference on IEEE industrial electronics society, pp 3936–3941
19. Laberteaux KP, Rohrs CE, Antsaklis PJ (2002) A practical controller for explicit rate congestion control. IEEE Trans Autom Control 47(6):960–978
20. Lai NO, Edwards C, Spurgeon S (2007) On output tracking using dynamic output feedback discrete-time sliding-mode controllers. IEEE Trans Autom Control 52(10):1975–1981
21. Mehta A, Bandyopadhyay B (2010) The design and implementation of output feedback based frequency shaped sliding mode controller for the smart structure. In: Proceedings of IEEE international symposium on industrial electronics, pp 353–358
22. Mija S, Susy T (2010) Reaching law based sliding mode control for discrete MIMO systems. In: Proceedings of IEEE international conference on control, automation, robotics and vision, pp 1291–1296
23. Milosavljević C (1985) General conditions for the existence of a quasisliding mode on the switching hyperplane in discrete variable structure systems. Autom Remote Control 46(3):307–314
24. Quet PF, Ataslar B, Iftar A, Ozbay H, Kalyanaraman S, Kang T (2002) Rate-based flow controllers for communication networks in the presence of uncertain time-varying multiple time-delays. Automatica 38(6):917–928
25. Shtessel Y, Edwards C, Fridman L, Levant A (2014) Sliding mode control and observation. Springer, New York
26. Tanenbaum AS, Wetherall DJ (2010) Computer networks. Prentice Hall, Boston

27. Utkin V (1977) Variable structure systems with sliding modes. IEEE Trans Autom Control 22(2):212–222
28. Utkin V, Drakunov SV (1989) On discrete-time sliding mode control. In: Proceedings of IFAC conference on nonlinear control, pp 484–489
29. Utkin V, Guldner J, Shi J (1999) Sliding mode control in electromechanical systems. Taylor & Francis, London
30. Xia Y, Liu GP, Shi P, Chen J, Rees D, Liang J (2007) Sliding mode control of uncertain linear discrete time systems with input delay. IET Control Theory Appl 1(4):1169–1175
31. Yu X, Wang B, Li X (2012) Computer-controlled variable structure systems: the state of the art. IEEE Trans Ind Inf 8(2):197–205

http://www.springer.com/978-3-319-31537-9

Variable-Structure Approaches
Analysis, Simulation, Robust Control and Estimation of Uncertain Dynamic Processes
Rauh, A.; Senkel, L. (Eds.)
2016, X, 361 p. 139 illus., 23 illus. in color., Hardcover
ISBN: 978-3-319-31537-9