Computer Standards & Interfaces xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Access delay in LonTalk MAC protocol

Marek Miśkowicz

AGH University of Science and Technology, Department of Electronics, al. Mickiewicza 30, 30-059 Kraków, Poland

ARTICLE INFO

Available online xxxx

Keywords:
Carrier Sense Multiple Access (CSMA) protocols
Markov models
Performance evaluation
Local Operating Networks (LonWorks)
ANSI/CEA-709.1

ABSTRACT

The study deals with the analysis of latency introduced by the media access control algorithm of LonTalk protocol registered as ANSI/CEA-709.1 standard and used in LonWorks control networking technology. The LonTalk protocol provides multicast communication which is a distinctive feature among control network protocols. The predictive *p*-persistent CSMA protocol built in the MAC sublayer uses the memoryless backoff, additive increase/additive decrease contention window adjustments, provides collision avoidance and optional collision detection.

The behavior of the LonTalk MAC protocol, unlike of the other CSMA schemes, is forced not only by the traffic rate but also by the structure of the workload transmitted through the channel. Therefore, the predictive *p*-persistent CSMA performance depends on the load scenario defined as the specification of the input traffic generated to the network. In the study, a unified method of load scenario definition integrating various addressing and message service types, is used. The contribution of the paper is the adaptation of the analytical approach based on Markov chains to the evaluation of mean access delay of LonTalk protocol for any load scenario.

1. Introduction

A time delay experienced by a message sent between application tasks residing on different nodes is a critical issue in real-time control networks. Latency characteristics determine the application profile of control networking. Industrial communication systems must meet Quality of Service (QoS) requirements to obtain satisfactory performance. To achieve this goal, a deep understanding of QoS issues is needed.

A core component of the application-to-application delay is a latency in accessing the channel introduced by the Media Access Control (MAC) protocol of the sending node. The present study deals with the analysis of latency introduced by the channel access algorithm of LonTalk/ANSI/CEA-709.1 protocol used in LonWorks control networking technology [1,2]. LonTalk supports the *best-effort service delivery* where the network does not provide any guarantees that data is really delivered. A best-effort network operates according to the principle "best possible effort taken" so data delivery time depends on the current traffic load. In the MAC protocols designed according to best-effort strategy, like in Carrier Sense Multiple Access (CSMA) family, the access delay depends not only on MAC algorithm operation but also on the traffic rate produced by the other nodes sharing the channel.

Local Operating Networks (LON, LonWorks) is one of the leading technologies in control networking addressed due to its architectural flexibility to a wide range of applications. In particular, LonWorks platform has become a classic solution in building automation, and home networking [3,4]. On the MAC layer of LonTalk, the *predictive p-persistent CSMA* protocol is used. Like all CSMA-based protocols, the predictive *p*-CSMA belongs to random access schemes based on contention. Rather than reserve the bandwidth, nodes compete for a shared channel resulting in probabilistic coordination. When a node wins a contention, transmits at the full channel bandwidth. Neither a priori coordination among the nodes, nor clocks synchronization is required.

The evaluation of the random MAC performance is a complex task since an analytical model has to follow stochastic protocol behavior. This problem becomes more difficult in adaptive random schemes, like the predictive *p*-persistent CSMA, where the protocol adapts to varying load conditions.

The most known representative of contemporary CSMA schemes is undoubtedly the group of protocols for wireless local area networks (IEEE 802.11) [6]. A huge amount of research has been carried out on the performance analysis of IEEE 802.11 in the past decade. However, the results obtained for IEEE 802.11 cannot be adapted to LonTalk/ANSI/CEA-709.1 due to differences in operation of both protocols [1,2,6]. First of all, the IEEE 802.11, like the other CSMA schemes, does not provide multicast communication, and the acknowledgements are sent in a dedicated time interval preceded by the Short Inter Frame Space (SIFS). Instead, LonTalk supports multicast addressing of messages as an efficient solution of improving utilization of the network bandwidth since a single multicast message is received by all the nodes specified in the message address field. Furthermore, in LonTalk the acknowledgement packets compete for the channel together with messages according to the contention algorithm. The next difference is that the IEEE 802.11 belongs to protocols with the *non-memoryless backoff*, i.e., the contending nodes freeze their backoff timers when the transmission is detected in the channel, and resume the backoff states when the current transmission is

E-mail address: miskow@agh.edu.pl.

0920-5489/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2008.03.025

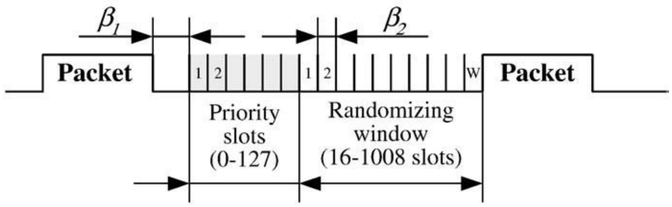

Fig. 1. LonTalk/ANSI/CEA-709.1 packet cycle.

completed. Instead, the MAC scheme in LonTalk uses the *memoryless backoff*, i.e., the contenders draw backoff times in every packet cycle anew and cancel them when the transmission is detected in the channel. Finally, the IEEE 802.11 adopts the collisions avoidance, rather than collision detection, and LonTalk provides both the (mandatory) collision avoidance and the (optional) collision detection in wired systems. The collision avoidance in LonTalk consists in the additive increase/additive decrease contention window adjustments, and IEEE 802.11 exploits the truncated exponential backoff.

However, the most distinctive feature of the predictive CSMA among the other CSMA schemes is that the predictive p-persistent CSMA behavior is forced not only by the traffic rate but also by the *structure of the workload* transmitted through the channel. Therefore, the predictive p-persistent CSMA performance depends on the *load scenario* defined as the specification of the input traffic generated to the network. A definition of the load scenario integrates the description of the message service (acknowledged, or unacknowledged one), and addressing (unicast, or multicast) of each input traffic component.

The contribution of the paper is the adaptation of the analytical approach based on Markov chains to the evaluation of mean access delay of LonTalk MAC protocol for any load scenario. Several papers with performance analysis of the predictive p-persistent CSMA scheme including the simulation analyses (e.g. [7]), and the analytical approaches [5,8,10–12] have been published. We take advantage in particular of the analytical derivation of the mean access delay for the slotted-CSMA with memoryless backoff presented in [10].

The paper is structured as follows. In Section 2, the LonTalk MAC protocol specification and the network model are introduced. The framework of the analytical approach to the evaluation of the mean access delay based on Markov chains is summarized in Section 3. The composition of the transition matrix for any load scenario is explained with some examples in Section 4. Furthermore, the numerical results according to the analytical procedure for exemplified load scenarios are reported at the end of Section 4. Finally, the conclusions are drawn.

2. Protocol specification and network model

The LonTalk packet cycle consists of two phases (Fig. 1). The first phase is optional and dedicated to priority messages. During the second phase nodes randomize their access to the medium. Since the goal of our analysis is the competitive scheme of LonTalk, we assume there are no priority slots in a packet cycle.

2.1. Competitive LonTalk MAC scheme

The predictive p-persistent CSMA belongs to slotted-CSMA protocols. The algorithm operates in the following way. A node attempting to transmit monitors the state of the channel. If the channel is busy, the node continues sensing. When the node detects no transmission during the *minimum interpacket space* of β_1 period, delays a random backoff.

If the channel is still idle when the random delay expires, the node transmits. Otherwise, the node receives incoming packet and competes for the channel access again. If more than one node chooses the same slot number, and where that slot has the lowest number selected by any node with a packet to send, then a *collision* happens. All the packets involved in a collision are corrupted.

The backoff time is expressed as a pseudorandom number of *contention slots* of β_2 duration drawn from the uniform distribution between 0 and W, where W is the *size of the contention window*. The predictive p-CSMA is an adaptive version of p-CSMA because a window size is dynamically adjusted to the current channel load. If the channel is idle, the contention window consists of 16 time slots. When the channel load increases, the number of slots grows by factor BL, called the *estimated backlog*. The backlog BL can range from 1 to 63 and the size of the window varies from 16 to 1008 slots since

$$W = \mathrm{BL} \cdot W_{\mathrm{base}}, \tag{1}$$

where W_{base} is the size of the *basic contention window* (16 slots). Thus, the level of persistence of p-CSMA equals $1/(16\mathrm{BL})$, is variable, and has either the lower ($1/16=0.0625$), or the upper bound ($1/1008=0.0009$).

In the predictive p-CSMA, the optional collision detection can be introduced. The aim of collision detection is that the sender does not have to wait for time-out before attempting to resend the messages.

As follows from the protocol specification, the access to the shared channel is organized in packet cycles. Each *packet cycle* is an attempt of a packet transmission undertaken by node(s) that has data ready for sending. A packet cycle begins with an interpacket gap and a random number of contention slots followed by a packet transmission. The result of each transmission attempt is a *successful transmission* of a packet or a *collision*.

2.2. Backlog counting algorithm

The backlog estimation is based on the calculation of the number of packets expected in a competition for the channel in the next packet cycle. The current state of the backlog counter BL varies from one to the next packet cycle and relies on the accumulation of consecutive backlog increments and decrements [1,2]. Backlog counting built in the node firmware relies on the following principles:

– the backlog BL is incremented after a successful transmission of a packet by a number of ($Delta\_BL-1$) encoded in the header of this packet,
– the backlog is decremented by one in idle packet cycle,
– the backlog counter is incremented optionally by one in case of collision if the nodes are equipped with the collision detection.

A number encoded in the 6-bit long data field *Delta_BL* of a packet header represents the number of acknowledgements that will be generated by receiver(s) as a result of packet reception (Fig. 2). This number equals one for unicast messages. Similarly, for multicast messages the number encoded in the *Delta_BL* is greater than one but does not exceed $M=63$ so the maximum size of a group of receiving nodes addressed by a single message equals 63. In the predictive p-CSMA, acknowledgement packets are not privileged in the channel access, and compete for the channel jointly with messages.

Each node calculates the channel backlog autonomously based on the backlog counter implemented in LonWorks node firmware. To keep the *consistency* of backlog states, all the nodes in the network should modify their backlog counters in the same way. The consistency is kept if each node is able to detect collisions even if it is not a sender of collided packets.

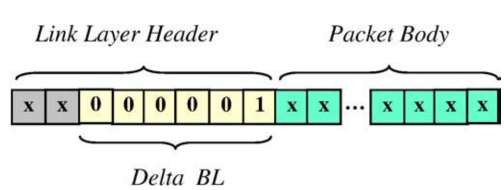

Fig. 2. *Delta_BL* is the 6-bit long data field in the 8-bit Link Layer header. *Delta_BL*=1 set in the figure corresponds to the unicast acknowledged message.

Please cite this article as: M. Miśkowicz, Access delay in LonTalk MAC protocol, Computer Standards & Interfaces (2008), doi:10.1016/j.csi.2008.03.025

2.3. Message service types

As mentioned above, in LonWorks networks the messages can be sent to a *single* receiver (*unicast*), or a *group* of receiving nodes (*multicast*). Multicast transactions save communication bandwidth since a single multicast message can substitute a set of unicast messages addressed to each recipient individually.

LonTalk provides the following message services: acknowledged service, unacknowledged service, unacknowledged repeated service, request/response service.

The *acknowledged service* is a default type of the message service in CSMA-based networks. As a response to acknowledged message reception, each receiver generates a *positive acknowledgement* to the sender. If acknowledgement(s) are not all received, the sender times out and retries the transaction. From the communication point of view the *request/response service* provides the same traffic contribution as the acknowledged service. The only difference is that a response unlike an acknowledgement can carry application data.

In the *unacknowledged repeated service*, acknowledgements are not applicable. The acknowledged repeated transaction is completed after sending all message repetitions. The particular case of the unacknowledged repeated service is the simple *unacknowledged service* where repetitions of the message are not sent. From the perspective of traffic modeling, the unacknowledged repeated service may be considered as the multiplication of the simple unacknowledged service.

2.4. Network model

To recognize the predictive CSMA performance, we introduce some simplifications to the real network model. In particular, we assume that:

1. The network is in the *saturation status* where each node has always a packet to send. If an acknowledged message has been received by recipient(s), this node(s) generates an acknowledgement packet and places it in the output queue before messages waiting for a transmission.
2. Each node is a *source of messages* unless it receives an acknowledged message. Then, it generates an acknowledgement packet and switches its status to the *source of acknowledgements* (i.e. schedules acknowledgement packet as the next packet for a transmission). As stated, the predictive *p*-CSMA behavior is forced not only by the traffic rate but also by the structure of the traffic transmitted in the channel. A key assumption we make is that the destination address(es) of transmitted messages are uniformly distributed such that each message is sent to the node that currently possesses a status of a source of messages. The protocol analysis deals with the steady state of the network when the mean size of the contention window reaches asymptotically a constant value. The proportion between the number of sources of messages and the number of sources of acknowledgements determines the transition probabilities between backlog stages which are evaluated in Section 4.4.
3. The network consists of a single segment that does not contain store-and-forward routers.
4. Backlog states are consistent in all the nodes in the network.
5. The number of concurrent *outgoing transactions* being in progress is unlimited (i.e., each node tries to send a new packet even if acknowledgement(s) of previously sent packets have not been yet successfully received).
6. The processing speed of the node is infinite, and in particular, the communication channel is not too fast for the node CPU.
7. The collision is detected at the end of packet transmission and the preamble preceding packet transmission is assumed to be of zero length. As a result of this assumption, the whole packets are in fact transmitted either in the successful, or in the unsuccessful packet cycles.

3. Network-induced delay and access delay analysis

3.1. Application-to-application delay

Network-induced delay is inevitable in networked control systems where the devices communicate via the shared channel. From the point of view of providing the real-time requirements, the application-to-application delay is a critical performance measure in distributed real-time systems. By the definition, the *application-to-application delay* is the time delay experienced by a message sent between application tasks that usually reside on different nodes in the networked real-time system. The application-to-application delay depends on the implementation of the system and can be decomposed as follows [13,14]:

- scheduling delay of the application task that produces a message at the sending node,
- processing and queueing delay in the upper layers of the sending node,
- queueing delay in the MAC layer of the sending node including the network access delay,
- message transmission delay,
- propagation delay,
- queueing delay in the MAC layer of the receiving node,
- processing and queueing delay in the upper layers of the receiving node,
- scheduling delay of the application task that consumes a message at the receiving node.

Several components of the application-to-application delay are not determined by the network but by the node CPUs. The processing and queueing delay in the upper layer protocols depend mainly on the system software and on the processor/memory speeds at the sending/receiving nodes. The queueing delay in the MAC layer of the receiving node depends on how quickly the processor can respond to a message arrival signalled by the MAC layer [14].

Scheduling delay depends on the scheduling algorithm operation, the number of application tasks managed by a node application software and their complexity. If the scheduling is based on the best-effort strategy, which is used in event-driven schedulers, the scheduling delay depends also on the rate of event occurrences that trigger execution of application tasks.

The message transmission delay is determined by the bandwidth of the channel, and the message propagation delay is given by the signal propagation speed. Both of these delays can be regarded as fixed. However, the queueing delay in the MAC layer of the sending node depends on the MAC protocol that is used. In the MAC protocols designed according to best-effort strategy (e.g. in CSMA family), the queueing delay is additionally influenced by the traffic rate produced by the other nodes sharing the channel. Therefore, the queueing delay, and the access delay as its most important fraction, are the functions of the channel load. If the channel is heavily loaded, the access delay can be a dominant component of the application-to-application delay.

In the context of LonWorks technology, the analysis of scheduling delay is reported in [13], and the access delay introduced by the MAC layer of the sending node is a subject of the present study.

3.2. Access delay definition

As the maximum access delay in CSMA is generally unbounded, the *mean access delay* is chosen to evaluate the latency introduced by the LonTalk MAC layer.

The *mean access delay* is defined as an average time from the instant the node starts trying to send a packet until the beginning of its successful transmission [15]. The access delay is a fraction of a queueing delay. The latter covers all the time a packet spends in the outgoing buffer, and the former defines only the time when the sender tries to transmit a packet.

The channel access delay in LonTalk MAC random protocol consists of the following components (Fig. 3):

- deferring transmission when the channel is busy as detected by carrier sense hardware,
- delaying transmission by the fixed interval called the minimum interpacket gap (β_1) following any transmission in the channel to ensure that all the nodes can sense an idle channel,
- deferring transmission for the random delay (from 0 to 1007 β_2 contention slots) to reduce the probability of packet collision during the contention,
- deferring transmission before any of the transmission attempts if a packet is involved in collision(s).

3.3. Analytical approach to mean access delay analysis

We evaluate the mean access delay for the slotted-CSMA on the basis of the estimation of an average time interval between consecutive successful channel access attempts undertaken by a given node that always has packets to send (Fig. 3). The mean access delay t_{mean} scaled in bits for the slotted-CSMA with varying contention window and memoryless backoff is given by [10]:

$$t_{\mathrm{mean}} = \left(\frac{1}{p_{\mathrm{succ}}} - 1\right) n\tau_{\mathrm{coll}} + n\tau_{\mathrm{succ}} - \mathrm{PktLength} \tag{2}$$

where $p_{succ}=p_{succ}(n)$ is the probability of a successful transmission, $\tau_{succ}=\tau_{succ}(n)$, $\tau_{coll}=\tau_{coll}(n)$ denote the mean lengths of successful and unsuccessful packet cycles, respectively, PktLength is the packet length, and n is the number of contending nodes.

The mean lengths of the appropriate packet cycles, $\tau_{succ}(n)$, $\tau_{coll}(n)$ are simply expressed by:

$$\tau_{\mathrm{succ}}(n) = \beta_1 + [d_{\mathrm{succ}}(n) - 1]\beta_2 + \mathrm{PktLength} \tag{3}$$

$$\tau_{\mathrm{coll}}(n) = \beta_1 + [d_{\mathrm{coll}}(n) - 1]\beta_2 + \mathrm{PktLength} \tag{4}$$

where $d_{succ}(n)$ denotes the mean slot number, at which a node winning the competition starts the transmission, $d_{coll}(n)$ is the mean slot number when a collision occurs, β_1 is the duration of the minimum interpacket gap, and β_2 is the contention slot width. All the parameters τ_{succ}, τ_{coll}, β_1, β_2, PktLength in the formulas (11) and (12) are specified in bits.

Summing up, in order to estimate the mean access delay t_{mean}, the following measures have to be calculated (see formulas (2), (3) and (4)):

- the probability of a successful transmission $p_{succ}(n)$, or the probability of collision since $p_{coll}(n)=1-p_{succ}(n)$,
- the mean slot numbers when the successful transmission starts $d_{succ}(n)$, and the collision occurs $d_{coll}(n)$.

Since the contention window varies in the predictive p-persistent CSMA during the network operation, the analytical approach has to involve Markov chains to estimate the distribution of the window size in the network steady state [10]. The window size distribution is determined by the *stationary distribution of the backlog* $\pi=[\pi_k]$, $k=1,\ldots,\mathrm{BL}_{max}$ according to the formula (1). In order to compute $\pi=[\pi_k]$, the *transition matrix* with a set of transition probabilities of the Markov chain has to be defined. As will be further discussed in Section 4.4, the transition probabilities depend on the load scenario.

If the backlog stationary distribution $\pi=[\pi_k]$ is computed, the mean slot numbers $d_{succ}(n)$, $d_{coll}(n)$ when respectively the successful/unsuccessful transmission starts for the predictive p-persistent CSMA are given by [10]:

$$d_{\mathrm{succ}}(n) = \sum_{k=1}^{\mathrm{BL}_{\max}} \pi_k \left[\frac{\sum_{s=1}^{16k} (16k - s)^{n-1} s}{\sum_{s=1}^{16k} (16k - s)^{n-1}}\right] \tag{5}$$

$$d_{\mathrm{coll}}(n) = \sum_{k=1}^{\mathrm{BL}_{\max}} \pi_k \left(\frac{1}{(16k)^{n-1}} \sum_{s=1}^{16k} s^{n-1}\right) \tag{6}$$

and the probability of a successful transmission is expressed by:

$$p_{\mathrm{succ}}(n) = n \sum_{k=1}^{\mathrm{BL}_{\max}} \pi_k \left[\sum_{s=1}^{16k} \frac{1}{16k}\left(\frac{16k - s}{16k}\right)^{n-1}\right] \tag{7}$$

Under some constraints, the formula (2) can be further simplified [10]. More specifically, if the number of contending nodes is large, both $d_{succ}(n)$ and $d_{coll}(n)$ approach asymptotically one so $\tau_{succ} \cong \tau_{coll} \cong \beta_1 + \mathrm{PktLength}$. Furthermore, if the interpacket space β_1 is negligible compared with the packet length PktLength ($\beta_1 << \mathrm{PktLength}$), then $\tau_{succ} \cong \tau_{coll} \cong \mathrm{PktLength}$. Consequently, after substituting the corresponding simplification to Eq. (2), the mean access delay for high number of contenders can be approximated by the simple formula:

$$t_{\mathrm{mean}} \cong \frac{\mathrm{PktLength}}{p_{\mathrm{succ}}} n \tag{8}$$

3.4. Stationary distribution of backlog counter

It is well known that the stationary distribution of a Markov chain is an eigenvector of the transition matrix **P**, associated with the eigenvalue equal to one. The vector $\pi=[\pi_k]$ includes the long-term probabilities π_k that the channel backlog will be at the stage k in the steady state, that is:

$$\pi_k = \lim_{l \to \infty} \Pr\ \{\mathrm{BL}(l) = k\}$$

The probability π_k is the relative frequency that a channel enters the backlog stage k in the steady state. The numerical methods of the stationary distribution computation are discussed in [9]. We compute the stationary distribution directly as the appropriate eigenvector of the transition matrix **P**. Thus, to compute the steady-state vector π of a Markov chain, the following linear system has to be solved:

$$[\mathbf{G}|\boldsymbol{e}]^T \boldsymbol{\pi} = \boldsymbol{b} \tag{9}$$

where, $\mathbf{P}=[p_{k,\ k+m}]$ is a transition matrix $\mathrm{BL}_{max}\times\mathrm{BL}_{max}$; the elements $p_{k,\ k+m}$ of the matrix **P** will be defined in Section 4.4,

$\mathbf{G}=\mathbf{P}-\mathbf{I}$, where **I** is an identity matrix $\mathrm{BL}_{max}\times\mathrm{BL}_{max}$,

$\boldsymbol{e}=[e_k]$ is a vector, where $e_k=1$; $k=1,\ldots,\mathrm{BL}_{max}$,

$[\mathbf{G}|\boldsymbol{e}]$ is a matrix $\mathrm{BL}_{max}\times(\mathrm{BL}_{max}+1)$, where the last column of this matrix is the vector $\boldsymbol{e}$,

$\boldsymbol{b}=[b_k]$ is a vector, where $b_k=0$, $b_{k+1}=1$; $k=1,\ldots,\mathrm{BL}_{max}$.

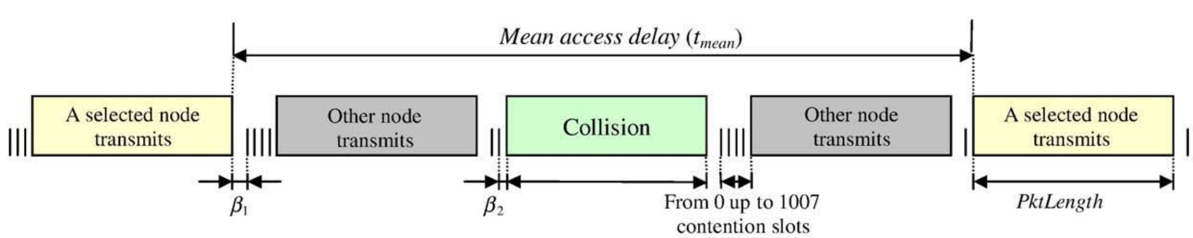

Fig. 3. A packet access delay definition.

4. Composition of transition matrix

Now we will describe how to define the transition matrix for a given load scenario.

4.1. Analytic model of backlog

Let BL(l) be a discrete-time stochastic process representing the *backlog stage* at the lth packet cycle in the network consisting of n nodes, where BL(l)=1,…,BL_{max}. As was mentioned in Section 2.2, BL_{max} is set to 63 in the existing implementation of the predictive p-persistent CSMA [1]. We assume that the process BL(l) is a global measure of the current channel load. BL(l) is a Markov chain with transition probabilities $p_{k,\,k+m}$; $k=1,\ldots,BL_{max}$; $m=1,\ldots,BL_{max}-1$.

As stated in Section 2.2, the current state of the backlog counter is tuned by the traffic transmitted through the channel and by collision occurrences. In any load scenario, we can distinguish the following types of packet cycles:

(1) an unsuccessful transmission of any packet due to a collision, which causes the channel backlog BL to be increased by one in the next packet cycle: $BL(l+1)=BL(l)+1$,
(2) a successful transmission of the acknowledged message addressed to a number of m recipients that results in the channel backlog BL increase by $m-1$, $m\geq 1$ in the next packet cycle: $BL(l+1)=BL(l)+m-1$,
(3) a successful transmission of the acknowledgement packet or the unacknowledged message, which causes the channel backlog BL to be decreased by one in the next packet cycle: $BL(l+1)=BL(l)-1$.

Modeling the impact of backlog constraints, two extra conditions for the backlog minimum $BL(l)=BL_{min}=1$ and the backlog maximum $BL(l)=BL_{max}$ have to be included:
(4) if the backlog has reached the last stage $BL(l)=BL_{max}$, remains at it even after an unsuccessful transmission of any packet or a successful transmission of multicast (m) message where $m\geq 2$,
(6) if the backlog has entered the first stage $BL(l)=BL_{min}=1$, remains at it even after successful transmission of an acknowledgement or the unacknowledged message.

The transition probabilities depend on that how many messages of particular type are produced by the application tasks in particular nodes. In order to model the input traffic we introduce a unified method of load scenario definition integrating various addressing and message service types.

4.2. Unified approach to input traffic specification

Let us call the traffic of original messages as the *input traffic*, or the *primary traffic*. The acknowledgements packets, and the messages retransmitted due to collisions constitute the *derivative traffic*. Acknowledgement packets generated in case of a successful reception of acknowledged messages form the *control traffic overhead* since they do not carry application data. The *total traffic* in the channel is a *superposition* of the traffic of messages and acknowledgements.

The specification of a *load scenario* describes the structure of the input traffic generated to the network (i.e. a percentage of particular message types). Each original message is characterized as regards to its contribution to derivative traffic that will be generated in case of its successful reception. Therefore, a specification of a message type has to include the message service (acknowledged or unacknowledged one), and the addressing (unicast, or multicast).

Let the load scenario be specified by a set of numbers $0\leq\alpha_i\leq 1$, $i=0,\ldots,M$ representing percentages of multicast (i) messages in the input traffic. Thus, there are $M+1$ input traffic components and:

$$\sum_{i=0}^{M}\alpha_i = 1 \tag{10}$$

In the existing implementation of LonTalk protocol, the maximum number of recipients (M) of a single multicast message is 63 as stated in Section 2.2. By the convention, we denote as multicast (0) the unacknowledged message since its successful transmission causes no acknowledgement to be generated.

Let us transform the numbers α_i to the set of relative coefficients γ_i, $i=0,\ldots,M$ denoted as the *scaling factors*. The transformation relies on the normalization of a set of α_i to their minimum. Assume that the multicast (j) messages, $j\in\{0,\ldots,M\}$, are the smallest non-zero component in the input traffic. Then, the scaling factors are simply defined as:

$$\gamma_i \overset{df}{=} \frac{\alpha_i}{\alpha_j};\, i=0,\ldots,M. \tag{11}$$

where by the assumption $\alpha_j = \min\limits_{i=0,1,\ldots,M}\{\alpha_i;\, \alpha_i\neq 0\}$.

By the defition: $\gamma_i\geq 1$, $i\neq j$, and $\gamma_i=1$, $i=j$. Note that the case $\gamma_i=1$ for $i\neq j$ represents the situation when more than one component of original messages provides the same and the smallest contribution to the input traffic. For example, the traffic load that consists of 50% of unacknowledged messages, 25% of unicast, and 25% of multicast (2) messages is represented by $\alpha_0=0.5$, $\alpha_1=0.25$, $\alpha_2=0.25$, or alternatively by the set of scalling factors: $\gamma_0=2$, $\gamma_1=1$, $\gamma_2=1$. The scaling factors $\gamma_0,\ldots,\gamma_M$ provide a complete specification of the traffic in the channel because the index of γ_i defines a size of a backlog change in case of a successful transmission of the multicast (i) message, and the value of γ_i determines the probability of a successful transmission of that message.

The traffic description by the set of α_i, $0\leq\alpha_i\leq 1$ coefficients is equivalent to the specification by scalling factors γ_i, $\gamma_i\geq 1$.

Please cite this article as: M. Miśkowicz, Access delay in LonTalk MAC protocol, Computer Standards & Interfaces (2008), doi:10.1016/j.csi.2008.03.025

4.3. Traffic superposition

Since a successfully transmitted acknowledged multicast (i) message generates a number of i acknowledgements, the total traffic in the channel is a superposition of the traffic of messages and traffic of acknowledgements. Applying the assumption about distribution of message destination addresses, the total traffic might be described as follows:

$$\alpha_j{}'\left(\gamma_0+\gamma_1+\gamma_1+\gamma_2+2\gamma_2+\ldots+\gamma_M+M\gamma_M\right)=1 \tag{12}$$

where $\alpha_j{}'$ represents the percentage of multicast (j) messages in the total traffic. Solving the Eq. (12) for $\alpha_{j'}$ we obtain:

$$\alpha_j{}'=\frac{1}{\sum_{i=0}^{M}\left[(i+1)\gamma_i\right]} \tag{13}$$

Next, the percentage $\alpha_m{}'$ of the multicast (m) messages in the total traffic:

$$\alpha_m{}'=\gamma_m\alpha_j{}',\quad m=0,\ldots,M \tag{14}$$

and finally after setting Eq. (13) into Eq. (14):

$$\alpha_m{}'=\frac{\gamma_m}{\sum_{i=0}^{M}\left[(i+1)\gamma_i\right]} \tag{15}$$

4.4. Transition probabilities

Assume the channel backlog BL equals k at a certain packet cycle. Denote by $p_{k,\,k+m-1}$ the probability of the channel backlog increase by $m-1$ from the state k to $k+m-1$ in the next packet cycle. As follows from the predictive p-persistent CSMA protocol specification, for $m=1$ or $3\le m\le M$, $p_{k,\,k+m-1}$ equals the probability of a successful transmission of the multicast (m) message since it causes the backlog increase by $m-1$. According to Eq. (15), the probability of a successful transmission of the multicast (m) message simply amounts $\alpha_m'(1-p_k)$ where p_k is the probability of collision provided that BL=k (Eqs. (16a) and (16c)). The probability $p_{k,\,k+1}$ of incrementing the backlog by one is the sum of a successful transmission of the multicast(2) message equal to $\alpha_2{}'(1-p_k)$, and the probability of a collision p_k (Eq. (16b)). Finally, the probability $p_{k,\,k-1}$ of decrementing the backlog by one is the sum of the probability of a successful transmission of the multicast (0) (unacknowledged) message $\alpha_0{}'(1-p_k)$ and the probability of a successful transmission of the acknowledgement packet. The latter equals $(1-p_k)\sum_{i=1}^{M} i\alpha_i'$ since each of the input traffic components $\alpha_i{}'$, $i=0,\ldots,M$ contributes as $i\alpha_i{}'$ to the traffic of acknowledgements (Eq. (16d)). The other transitions of backlog states are prohibited. Summarizing, the transition probabilities are given by:

$$p_{k,k+m-1}=\alpha_m'(1-p_k),\quad M\geq m\geq 3 \tag{16a}$$

$$p_{k,k+1}=\alpha_2'(1-p_k)+p_k \tag{16b}$$

$$p_{k,k}=\alpha_1'(1-p_k) \tag{16c}$$

$$p_{k,k-1}=\alpha_0'(1-p_k)+(1-p_k)\sum_{i=1}^{M} i\alpha_i{}'=(1-p_k)\sum_{i=0}^{M} i\alpha_i{}' \tag{16d}$$

Setting the expression (14) to a set of the formulas (16a)–(16d), the probabilities of switching the backlog states are given by the following set of equations:

$$\begin{aligned}
p_{k,k}&=\frac{\gamma_1(1-p_k)}{\sum_{i=0}^{M}\left[(i+1)\gamma_i\right]}\\
p_{k,k+1}&=\frac{\gamma_2(1-p_k)}{\sum_{i=0}^{M}\left[(i+1)\gamma_i\right]}+p_k\\
p_{k,k+m-1}&=\frac{\gamma_m(1-p_k)}{\sum_{i=0}^{M}\left[(i+1)\gamma_i\right]},\ 3\leq m\leq M\\
p_{k,k-1}&=(1-p_k)\left[1-\frac{\sum_{i=0}^{M}\gamma_i}{\sum_{i=0}^{M}\left[(i+1)\gamma_i\right]}\right]
\end{aligned} \tag{17}$$

The probability p_k of collision provided that a number of nodes equals n and BL=k is given by:

$$p_k=1-n\sum_{s=1}^{16k}\frac{1}{16k}\left(\frac{16k-s}{16k}\right)^{n-1} \tag{18}$$

Note that the elements of a transition matrix **P** are the function of the number of nodes n because $p_k=p_k(n)$ according to the formula (18).

Please cite this article as: M. Miśkowicz, Access delay in LonTalk MAC protocol, Computer Standards & Interfaces (2008), doi:10.1016/j.csi.2008.03.025

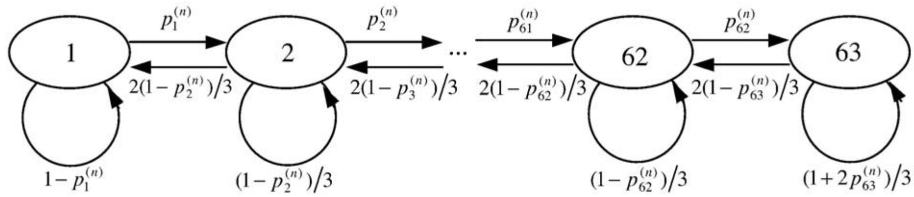

Fig. 4. State transition diagram of the Markov chain for load scenario defined by $\alpha_0=0.5$, $\alpha_1=0.5$. Symbol $p_k^{(n)}$ is used in order to emphasize that the probability of collision is a function of the number of nodes.

4.5. Examples of load scenarios

Let us consider an example of a load scenario, where the half of the input traffic is formed by the unacknowledged messages (multicast (0)), and the other half by the acknowledged unicast messages (multicast (1)). This load scenario is represented by the coefficients $\alpha_0=0.5$, $\alpha_1=0.5$ or equivalently by the scaling factors $\gamma_0=1$, $\gamma_1=1$, $\gamma_i=0$; $i=2,3,\ldots,M$ according to the input traffic specification method reported in Section 4.2. The transition matrix $\mathbf{P}_1$ for $\gamma_0=1$, $\gamma_1=1$ is presented below, and the diagram of the corresponding Markov chain is shown in Fig. 4:

$$\mathbf{P}_1=\begin{bmatrix}
1-p_1 & p_1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0\\
2(1-p_2)/3 & (1-p_2)/3 & p_2 & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots\\
0 & 2(1-p_3)/3 & (1-p_3)/3 & p_3 & 0 & \ddots & \ddots & \ddots & \ddots & \vdots\\
\vdots & 0 & 2(1-p_4)/3 & (1-p_4)/3 & p_4 & 0 & \ddots & \ddots & \ddots & \vdots\\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots\\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 2(1-p_{61})/3 & (1-p_{61})/3 & p_{61} & 0\\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & 2(1-p_{62})/3 & (1-p_{62})/3 & p_{62}\\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 2(1-p_{63})/3 & (1+2p_{63})/3
\end{bmatrix}$$

The next example of a load scenario corresponds to the multicast addressing. The situation when all the messages generated by the application tasks in each node are multicast (3) (i.e., $\alpha_3=1$, and $\gamma_3=1$, $\gamma_i=0$; $i=0,1,2,4,5,\ldots,M$) is exemplified by the transmition matrix $\mathbf{P}_2$, and the corresponding diagram of the Markov chain (Fig. 5):

$$\mathbf{P}_2=\begin{bmatrix}
3(1-p_1)/4 & p_1 & (1-p_1)/4 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0\\
3(1-p_2)/4 & 0 & p_2 & (1-p_2)/4 & 0 & 0 & \ddots & \ddots & \ddots & \vdots\\
0 & 3(1-p_3)/4 & 0 & p_3 & (1-p_3)/4 & 0 & 0 & \ddots & \ddots & \vdots\\
\vdots & 0 & 3(1-p_4)/4 & 0 & p_4 & (1-p_4)/4 & 0 & 0 & \ddots & \vdots\\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0\\
\vdots & \ddots & \ddots & \ddots & 0 & 3(1-p_{60})/4 & 0 & p_{60} & (1-p_{60})/4 & 0\\
\vdots & \ddots & \ddots & \ddots & \ddots & 0 & 3(1-p_{61})/4 & 0 & p_{61} & (1-p_{61})/4\\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & 3(1-p_{62})/4 & 0 & (1+3p_{62})/4\\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 3(1-p_{63})/4 & (1+3p_{63})/4
\end{bmatrix}$$

Finally, we show the example of the load scenario where the input traffic consists of three equal components as follows: unacknowledged messages, acknowledged unicast messages, and multicast (4) messages. This load scenario is represented by the coefficients $\alpha_0=1/3$, $\alpha_1=1/3$, $\alpha_4=1/3$, or equivalently by the scaling factors $\gamma_0=1$, $\gamma_1=1$, $\gamma_4=1$, $\gamma_i=0$; $i=2,3,5,6,\ldots,M$. The corresponding transition matrix $\mathbf{P}_3$ has the following form:

$$\mathbf{P}_3=\begin{bmatrix}
7(1-p_1)/8 & p_1 & 0 & (1-p_1)/8 & 0 & \cdots & \cdots & \cdots & \cdots & 0\\
3(1-p_2)/4 & (1-p_2)/8 & p_2 & 0 & (1-p_2)/8 & 0 & \ddots & \ddots & \ddots & \vdots\\
0 & 3(1-p_3)/4 & (1-p_3)/8 & p_3 & 0 & (1-p_3)/8 & 0 & \ddots & \ddots & \vdots\\
\vdots & 0 & 3(1-p_4)/4 & (1-p_4)/8 & p_4 & 0 & (1-p_4)/8 & 0 & \ddots & \vdots\\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0\\
\vdots & \ddots & \ddots & \ddots & 0 & 3(1-p_{60})/4 & (1-p_{60})/8 & p_{60} & 0 & (1-p_{60})/8\\
\vdots & \ddots & \ddots & \ddots & \ddots & 0 & 3(1-p_{61})/4 & (1-p_{61})/8 & p_{61} & (1-p_{61})/8\\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & 3(1-p_{62})/4 & (1-p_{62})/8 & (1+7p_{62})/8\\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 3(1-p_{63})/4 & (1+3p_{63})/4
\end{bmatrix}$$

4.6. Numerical results

In Fig. 6a and b two families of stationary distributions $\pi=[\pi_k]$, $k=1,\ldots,63$ of the backlog (for selected numbers of contending nodes) corresponding to the load scenarios represented by the transition matrix $\mathbf{P}_1$ and $\mathbf{P}_2$ are shown. As follows from Fig. 6, the backlog stationary distribution is symmetric for $\mathbf{P}_1$ and right-skewed for $\mathbf{P}_2$ (i.e., with a longer tail for higher backlog, and with the mass of the distribution concentrated for lower backlog states). The skewness of $\pi=[\pi_k]$ is caused by the multicast messages. Note that the Makov chain defined by $\mathbf{P}_1$ is a *random walk* since $p_{k,\,k+m}=0$ for $|m|>1$, i.e., only transitions between consecutive backlog stages are possible. The mean backlog and the standard deviation of distributions increase with growing number of contending nodes, n. This is intuitively obvious since the mean size of contention window must be greater if the channel load represented by n is heavier.

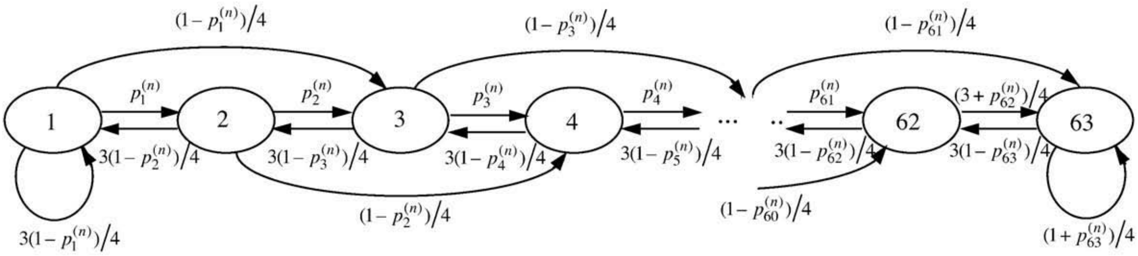

Fig. 5. State transition diagram of the Markov chain for multicast (3) load scenario.

In Fig. 7, the plots of mean access delay versus the number of contending nodes for various load scenarios are presented. The numerical results of the mean access delay have been computed according to the analytical approach represented by the formulas (2)–(7), (9) and (17). Each point in the plot has been found as a solution of the linear system given by Eq. (9) for a specified load scenario and particular number of nodes. The results for the following load scenarios are reported:

- $\alpha_0=1$ denoted as *UNACK*,
- $\alpha_0=0.5$, $\alpha_1=0.5$ denoted as *50% UNACK/50% unicast*,
- $\alpha_1=1$ as *unicast*,
- $\alpha_3=1$ as *multicast*(3),
- $\alpha_0=1/3$, $\alpha_1=1/3$, $\alpha_4=1/3$ as *general*,
- 0.0625-persistent CSMA.

The predicitve p-CSMA is reduced to the 0.0625-persistent CSMA protocol for the load scenario if no collision detection is provided, and no multicast addressing is used. Then, the predictive part of the algorithm is deactivated, and the contention window is constant and consists of 16 slots [10]. Moreover, the 0.0625-persistent CSMA approximates well the predictive p-persistent CSMA performance for light channel load regardless of the load scenario as is shown also in Fig. 7 as regards the mean access delay.

The latency in accessing the channel increases quasi-linearly with growing number of contending devices. This is due to the fact that the probability of collision $p_{\text{coll}}=1-p_{\text{succ}}$ is bounded for any load scenario for the predictive p-persistent CSMA. Thus, the mean access delay is almost

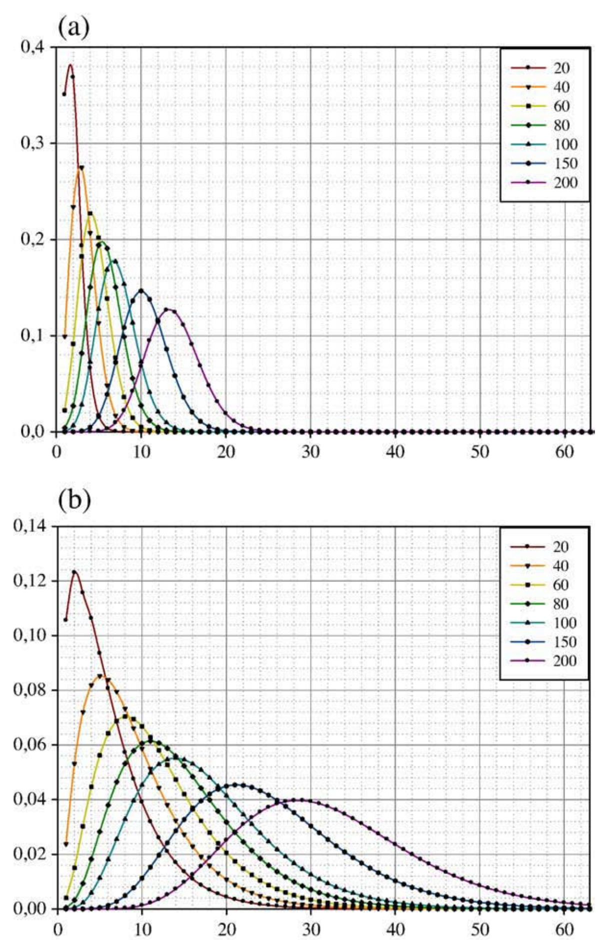

Fig. 6. Stationary distributions $\pi=[\pi_k]$, $k=1,\ldots,63$ for various number of contending nodes (from 20 to 200) and the two load scenarios defined by $\alpha_0=0.5$, $\alpha_1=0.5$ (a), and $\alpha_3=1$ (b).

Please cite this article as: M. Miśkowicz, Access delay in LonTalk MAC protocol, Computer Standards & Interfaces (2008), doi:10.1016/j.csi.2008.03.025

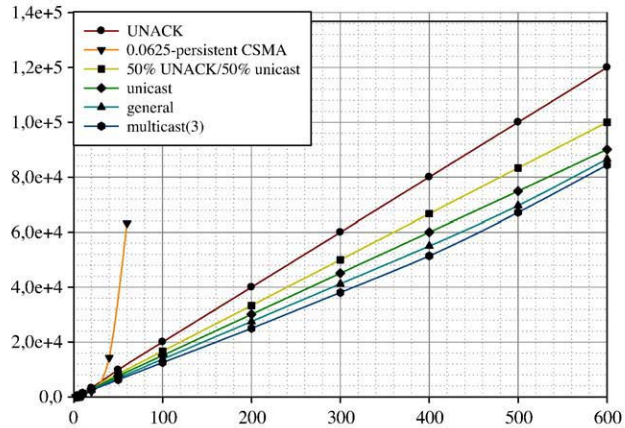

Fig. 7. Mean access delay vs. number of contending nodes for various load scenarios denoted as follows: *UNACK* (α_0=1), *50% UNACK/50% unicast* (α_0=0.5, α_1=0.5), *unicast* (α_1=1), *general* (α_0=1/3, α_1=1/3, α_4=1/3), *multicast* (3) (α_3=1) and for 0.0625-persistent CSMA.

linear function of the number of contenders according to the formula (8) which approximates t_{mean} for high contention. In particular, adding a new active node to the existing network causes to increase the mean access delay of about 1.3PktLength for multicast (3) scenario to 2PktLength for the load scenario denoted by UNACK. Since the probability of successful transmission approach one for multicast transactions in large groups, the rate of mean access increase should be close to PktLength per node.

However, note that the better characteristics of the mean access delay obtained for load scenarios with multicast transactions are achieved at the cost of minimization of the fraction of bandwidth devoted to a transmission of application data. This is a fundamental tradeoff of the predictive *p*-persistent CSMA operation similar to the tradeoff related to the throughput maximization reported in [11].

Summing up, the rate of the mean access delay increase ranges from PktLength to 2PktLength per active node for the range of high network load. The former corresponds to the multicast transactions in large groups, and the latter to unacknowledged message service, respectively. Thus, the rate of the mean access delay increase becomes lower if most packets sent are acknowledgements.

5. Conclusions

The paper deals with the evaluation of the mean access delay versus the channel load for the LonTalk media access control protocol according to the analytical approach based on Markov chains. It is proved that the LonTalk mean access delay increases almost linearly with the growing number of transmitting nodes if the number of contenders is high. The rate of the mean access delay increase ranges from PktLength to 2PktLength per active node for the range of high network load. The former corresponds to the multicast transactions in large groups, and the latter to unacknowledged message service, respectively. Thus, the rate of the mean access delay increase becomes lower if most packets sent through the network are acknowledgements. This is one of fundamental tradeoffs of the LonTalk MAC operation.

References

[1] LonTalk Protocol Specification, Echelon Corporation, 1995 Version 3.0.
[2] Network communication protocol including a reliable multicasting technique, United States Patent 5297143, 1994.
[3] M. Miśkowicz, R. Golański, LON technology in wireless sensor networking applications, Sensors 6 (1) (2006) 30–48.
[4] D. Dietrich, D. Loy, H.-J. Schweinzer, Open Control Networks: LonWorks/EIA 709 Technology, Kluwer Academic Publishers, 2001.
[5] P. Buchholz, J. Plonnigs, Analytical analysis of access-schemes of the CSMA type, Proceedings of IEEE International Workshop on Factory Communication Systems WFCS'2004, 2004, pp. 127–136.
[6] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Computer Society, 1997 IEEE standard 802.11.
[7] X. Chen, G.-S. Hong, A simulation study of the predictive p-persistent CSMA protocol, Proceedings of 35th Annual Simulation Symposium, 2002, pp. 345–351.
[8] X. Chen, G.-S. Hong, Real-time performance analysis of a fieldbus-based network, Proc of Information, Decision and Control Conference. Adelaide, 2002, pp. 1–113.
[9] W.J. Stewart, Introduction to the Numerical Solution of Markov Chains, Princeton Univ. Press, Princeton, 1999.
[10] M. Miśkowicz, Analysis of mean access delay in variable-window CSMA, Sensors 7 (12) (2007) 3535–3559.
[11] M. Miśkowicz, Limits of performance of adaptive variable-window CSMA, Proceedings of IFAC International Conference on Fieldbuses & Networks in Industrial & Embedded Systems FeT'2007, 2007, pp. 239–244.
[12] M. Miśkowicz, A generalized analytic approach to the evaluation of predictive p-CSMA/CD saturation performance, Proceedings of IEEE International Workshop on Factory Communication Systems WFCS'2006, 2006, pp. 342–352.
[13] M. Miśkowicz, Latency characteristics of event-driven task scheduler embedded in Neuron Chip, International Journal of Computer Science and Network Security 7 (12) (2007) 132–139.
[14] N. Malcolm, W. Zhao, Hard real-time communication in multiple-access networks, Journal of Real-Time Systems 8 (1) (1995) 35–77.
[15] J. Yeh, Simulation of local computer networks — a case study, Computer Networks 3 (1979) 401–417.

Marek Miśkowicz reveived his M.Sc. and Ph.D. degrees in Electronic Engineering respectively in 1995 and 2004 from AGH University of Science and Technology (AGH-UST), Krakow, Poland. Currently, he is an Assistant Professor in the Department of Electronics, AGH-UST. His main research interests focus on performance modeling of networked sensor and control systems, in particular, media access control and event-based sampling schemes. He has published more than 30 papers in international journals and conference proceedings.

Please cite this article as: M. Miśkowicz, Access delay in LonTalk MAC protocol, Computer Standards & Interfaces (2008), doi:10.1016/j.csi.2008.03.025