A Generalized Analytic Approach to the Evaluation of Predictive 
$p$-CSMA/CD Saturation Performance

Marek Miškowicz
AGH University of Science and Technology
Department of Electronics
Al. Mickiewicza 30, 30-059 Kraków
miskow@agh.edu.pl

Abstract

The paper addresses the issue of the worst-case performance evaluation of the predictive $p$-CSMA scheme used as the MAC protocol in LonWorks/EIA-709 networked control systems. The performance of the predictive $p$-CSMA significantly depends on the specification of the traffic scenario since it determines the efficiency of load prediction built in the protocol. The analytical approach based on Markov chains is applied. Transition probabilities for a general-case load scenario are derived and validated by the simulation. The saturation throughput and collision rate are further estimated. Finally, the simple procedure based on the equilibrium analysis to find the sustained throughput and collision rate for a general-case load scenario, is formulated.

1. Introduction

The key throughput/delay characteristics and the application profile of LonWorks/EIA-709 networked sensor/control systems are defined by the predictive CSMA scheme used in the Media Access Control sublayer of LonTalk/EIA-709.1 protocol.

The predictive $p$-CSMA is the adaptive version of a classical $p$-CSMA developed by Kleinrock and Tobagi [12]. Using the $p$-persistent CSMA, a node transmits with a fixed probability $p$ if the channel is idle, and defers a transmission with a probability $(1-p)$, when the channel is busy [12]. As opposed to a classical $p$-CSMA, in the predictive $p$-persistent CSMA, the probability $p$ is variable and dynamically adjusted to the expected traffic load using a backoff scheme. The goal of persistence tuning is to keep the high protocol performance even if the channel is loaded with a variable and bursty traffic.

Classical performance analyses of MAC protocols typically assume an infinity number of nodes, generating a traffic with Poisson arrivals. Unlike in data networks, MAC protocols for sensor/control networks have to be able to support variable and correlated traffic since a generation of messages is stimulated by dynamics of physical processes rather than by independent human activities. The traffic in sensor/control networking might be bursty. This is particularly the case when the application architecture is event-triggered, which LonWorks technology exemplifies. In systems with the event-driven input perception, data is transmitted in response to external events. Moreover, the event-triggered architecture is prone to event showers, i.e. burst of highly correlated events, often released by a single physical event that cause congestion of the system [11]. Such an effect occurs for example when a fire is detected in a building and a number of redundant temperature and smoke sensors begin reporting the event. Therefore, the fundamental assumption of stochastically distributed traffic and the traditional model of Poisson arrivals are not valid for many sensor/control applications.

A typical LON application assumes the channel is lightly loaded in a normal type of operation. Under light traffic, the predictive CSMA reduces to a pure 0.0625-persistent CSMA regardless of a structure of the traffic in a channel. In general, the network congestion does not appear in a normal system operation. However, it is highly desirable to recognize the system performance in overload since such a situation can occur.

In order to estimate how the protocol copes with overload situations, we choose the network saturation conditions that define the worst-case protocol performance. The saturation throughput is the most representative measure describing the robustness of the protocol to overload.

The saturated status is defined as a situation when all the nodes in a network segment have packets to send. This paper is a continuation of previous author’s work [6] where the saturation performance of the predictive $p$-CSMA for a channel without collision detection is analyzed in [6].

The present study deals with the analysis of the predictive $p$-persistent CSMA with collision detection for a general-case load scenario, that is, for any proportion between acknowledged/unacknowledged message service, and unicast/multicast transactions.
The presented analysis is based on Markov chains. In particular, it is shown how the performance of the predictive $p$-CSMA depends on the load scenario.

Several papers deal with the performance analysis of predictive $p$-persistent CSMA protocol [3,4,6-10]. The most distinctive study presenting the analytic approach to the predictive $p$-CSMA performance evaluation basing on queuing theory has been developed by Buchholz and Plönnigs and presented in [4]. This reference includes also a general Markov model for predictive CSMA.

However, the efficiency of protocol predictability has not been yet displayed quantitatively. For example, it has not been analytically proved that the sustained throughput of predictive $p$-CSMA is determined by a structure of the traffic being transmitted and is independent of the number of sending nodes. This latter remark is sometimes mentioned by LonWorks technology users. This paper addresses also this issue among other things.

The paper is structured as follow. Section 2 presents the predictive $p$-CSMA protocol specification and the backlog counting algorithm. In Section 3, the framework of the analytical approach, and definitions of performance measures, are specified. Section 4 shows examples of how to apply the analysis for particular load scenarios. The simulative verification of analytic results is also reported. In Section 5, the generalization of the analytic approach for various load scenarios, is formulated. Finally, basing on the equilibrium analysis, the sustained probability of a successful transmission, and the worst-case throughput for a generalized load scenario, is derived.

2. Protocol specification

The LonTalk/EIA 709.1 packet cycle consists of two phases (Fig. 1). The first phase is optional and dedicated for priority messages. During the second phase nodes randomize their access to the medium. Since the goal of our analysis is the predictive $p$-CSMA, we assume there are no priority slots in a packet cycle.

2.1. LonTalk/EIA 709.1 packet cycle

The predictive $p$-CSMA belongs to slotted-CSMA algorithms where the time axis is split into segments, called slots, whose duration is equal to $\beta_1$ (Fig. 1).

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 $\beta_1$ period, delays a random number of time slots of $\beta_2$ duration. 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 choose the same slot, and where that slot is the earliest one from slots selected by all the contending nodes, then a collision happens. All the packets involved in a collision are corrupted. The time by which the competing node defers the transmission is expressed as a pseudorandom number of time slots $\beta_2$ drawn from the uniform distribution between 0 and $W$, where $W$ is the size of the randomizing (competition) window.

The predictive $p$-CSMA is an adaptive version of $p$-CSMA, where a size of the randomizing window is dynamically adjusted to the current channel load. If the channel is idle, the randomizing window consists of 16 time slots. When the channel load increases, the number of slots grows by factor $BL$, called the estimated channel backlog. The backlog $BL$ can range from 1 to 63 and the size of the window varies from 16 to 1008 time slots, since the following relationship is met:

$$W = BL \cdot W_{base},$$

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

2.2. Backlog concept

The backlog estimation is based on the calculation of the number of packets expected in a competition for the channel during the next packet cycle. The current value 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’s firmware, relies on the following principles [2]: (i) successive backlog increments are based on the information included in the header of each packet that is sent or successfully received by a particular node; this information is encoded in the 6-bit long field $Delta _{BL}$; (ii) successive backlog decrements by one at the end of packet cycles. Both backlog modifications are independent of each other and occur in every cycle.

Optionally, the backlog counter might be also incremented by one in case of collision if the nodes are equipped with the collision detection [1,2]. In order to detect collisions, dedicated hardware in the transceiver is needed. When a collision is detected by the transceiver, LonWorks node firmware is signalled using a particular pin in the communication port [13].

$Delta _{BL}$ represents the number of acknowledgements that will be generated by receiver(s).
as a result of packet reception. For unicast messages \( Delta_{BL} = 1 \). Similarly, for multicast messages \( 1 < Delta_{BL} \leq 63 \) since the maximum size of a group of receiving nodes equals 63.

The backlog decrements are based on time-out technique, i.e., when Packet Cycle Timer expires [1]. Hence, the backlog is decreased by one either during busy, or in case of idle packet cycles (i.e., if no node transmits). However, in order to ensure the effective backlog increase by one, the collision has to be detected before the Packet Cycle Timer expiration and should disable the periodical backlog decrement.

On a basis of the backlog counting algorithm we can conclude that after a successful transmission of a message, the backlog \( BL \) is incremented by \( (Delta_{BL} - 1) \). It is a resultant of the increment by \( Delta_{BL} \) due to the number of expected acknowledgements, and the decrement by one at the end of a packet cycle. In particular, as a result of a successful transmission of an acknowledged unicast message, the channel backlog does not change, since \( Delta_{BL} = 1 \). Similarly, after a successful transmission of an acknowledgement or an unacknowledged message, the channel backlog \( BL \) is decremented by one, since \( Delta_{BL} = 0 \).

Note some remarks: (i) the backlog concept takes into account the derivative traffic only i.e., acknowledgements and retransmissions of packets corrupted due to collisions, (ii) the backlog \( BL \) cannot change more than one in consecutive packet cycles if no multicast transactions are realized, (iii) by the assumption, idle cycles do not occur in saturation conditions.

2.3. Backlog counting consistency

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 nodes in the network should modify their backlog counters in the same way.

However, backlog states can differ between the nodes. First, the packets with invalid CRC introduce the backlog inconsistency since all recipients reject it, and only the packet sender modifies the backlog counter according to its \( Delta_{BL} \). Thus, the noise introduced by Physical Layer can break backlog consistency since recipient(s) cannot read \( Delta_{BL} \) field of corrupted packets (note that recipients mean all the nodes in a network segment where a packet is broadcasted, not only nodes specified in the destination address field of a packet).

Second, most LON transceivers available on the market enable only senders to detect possible packet collisions and increment their backlog counters. Remaining nodes in the network segment do not increase their backlog counters since that nodes are not able to detect collisions. Consequently, the backlog counter can lose its global character and becomes a local parameter.

The inconsistency in backlog counting is the undesirable effect since causes unfair access to the channel. It is not a problem if a channel stays under light traffic load, because idle cycles can recover the backlog consistency if the minimum window size is reached for some time. However, if the channel is heavy loaded, the senders detecting a collision have a smaller chance of winning a competition for the channel in the next packet cycles. Then, unfairness in channel access is introduced in the long run. Thus, "incomplete" collision detection is an example of a certain imperfection in the protocol implementation, breaking its intended behavior, displayed in [2].

We assume that either Physical Layer, or Link Layer of the protocol do not introduce the backlog inconsistency, i.e., either the transmitting node, or the receiver(s) modify their backlog counters in the same way. It is achieved if the channel can be assumed to be noise-free and all the transceivers are able to detect collisions even if they are not senders of colliding packets. Then, backlog can be considered as a global channel parameter.

3. Framework of the analytic approach

By the definition, the saturated status of a network is reached if each node has a packet to send. We assume that the network consists of \( n \), a fixed number of nodes, and each node after the completion of a successful transmission immediately has a new packet available for sending. If an acknowledged message has been received by its recipient, a receiving node generates an acknowledgement and places it in the output queue before messages waiting for a transmission.

We assume also that the backlog consistency between the nodes is kept, and that both:
- the number of concurrent outgoing transactions being in progress (i.e., each node tries to send a new packet even if acknowledgement(s) of previously sent packets have not been yet successfully received), and
- the number of concurrent incoming transactions (defined by the length a queue of acknowledgements waiting for a transmission), are unlimited.

3.1. Analytical model of the channel backlog

Let \( BL^{(n)}(l) \) be a stochastic process representing the backlog stage at the \( l \)th packet cycle in a network consisting of \( n \) nodes, where \( BL^{(n)}(l) = 1,...,63 \).

\( BL^{(n)}(l) \) is a Markov chain with the transition probabilities \( p^{(n)}_{i,j} \), \( i,j = 1,...,63 \). Suppose that the backlog enters the stage \( k \) at the \( l \)th packet cycle, that is, \( BL^{(n)}(l) = k \). Let \( Pr[BL^{(n)}(l+1) = k + s \mid BL^{(n)}(l) = k] \) be the transition probability that backlog enters the stage

345
(k + s) in the ((l+1)th packet cycle from the stage k in the lth cycle. Denote this transition probability in short:

\[ Pr^{(n)}(BL^{(n)}(l + 1) = k + s | BL^{(n)}(l) = k) = p_{k,k+s}^{(n)} \] (2)

The first goal of our analysis is to find a stationary distribution \( \pi = [\pi_k], k = 1, \ldots, 63 \) of backlog. Next, the saturation channel backlog \( BL^{(n)} \) defined as the expected backlog in the long-term prospect:

\[ BL^{(n)} = E[BL^{(n)}(l \rightarrow \infty)] = \sum_{k=1}^{BL_{\text{max}}} k\pi_k \] (3)

where \( E[\cdot] \) is the expectation operator, is calculated. The saturation window is defined as the mean size of the randomizing window, see formula (1):

\[ \bar{W}^{(n)} = 16BL^{(n)} \] (4)

Finally, the probabilities of successful/ unsuccessful transmission, throughput, and collision rate are found.

### 3.2. Backlog saturation distribution

In order to find the saturation backlog, the steady-state vector, or stationary distribution, \( \pi = [\pi_k], k = 1, \ldots, 63 \) of a Markov chain has to be calculated. The stationary distribution \( \pi \) is an eigenvector of the transition matrix \( P^{(n)} \), associated with the eigenvalue 1. Each entry in the transition matrix \( P^{(n)} \) is defined by the transition probability \( p_{i,j}^{(n)}, i, j = 1, \ldots, 63 \). The vector \( \pi = [\pi_k] \) includes the long-term probabilities \( \pi_k \) that the channel backlog will be at the stage \( k \) in the steady state, that is:

\[ \pi_k = \lim_{l \to \infty} Pr\{BL^{(n)}(l) = k\} \] (5)

The direct method of calculation of the equilibrium probabilities of backlog stages using a Markov chain is applied, that is, in order to compute the steady-state vector \( \pi \) of a Markov chain, the following linear equation has to be solved:

\[ [G | e]^{T} \pi = b \]

where \( P^{(n)} = [p_{i,j}^{(n)}] \) is a transition matrix 63 x 63; \( p_{i,j}^{(n)} \) are given by (5), \( G = P^{(n)} - I \), where \( I \) is an identity matrix 63 x 63, \( e = [e_i] \) is a vector, where \( e_i = 1 \) \( i = 1, \ldots, 63 \), \( [G | e] \) is 63 x 64 matrix, where the last column of this matrix is the vector \( e \), \( b = [b_i] \) is a vector, where \( b_i = 0, b_{i+1} = 1; i = 1, \ldots, 63 \).

The key problem in Markov-based approach is to calculate the transition probabilities. These probabilities depend on a definition of a load scenario.

### 3.3. Channel throughput and collision rate

We follow the throughput and collision rate definitions formulated in [5]. The throughput is defined as the average fraction of time used for successful transmissions of packets in a channel [5]. To be precise, this is the Network Layer throughput definition. Note that such a definition takes into account successful transmissions of all the packets (i.e. not only messages, but also acknowledgements and retransmissions of corrupted packets). We assume that collisions are detected at the end of packet transmissions [1,13].

The throughput defined as the percentage of a channel bit rate can be calculated as follows:

\[ \text{Throughput}(n) = \frac{\text{PktLength}}{t_{\text{coll}}(n) p_{\text{coll}}^{(n)} / p_{\text{succ}}^{(n)} + \tau_{\text{succ}}(n)} \]

where \( \tau_{\text{succ}}(n) \) and \( \tau_{\text{coll}}(n) \) are the mean lengths of the successful/ unsuccessful packet cycle in network saturation, respectively, given by the formulas:

\[ \tau_{\text{succ}}(n) = \beta_1 + [d_{\text{succ}}(n) - 1]\beta_2 + \text{PktLength} \]

\[ \tau_{\text{coll}}(n) = \beta_1 + [d_{\text{coll}}(n) - 1]\beta_2 + \text{PktLength} \]

\( d_{\text{succ}}(n) \) denotes the mean slot number, at which a node winning a competition starts the transmission [4]:

\[ d_{\text{succ}}(n) = \sum_{k=1}^{BL_{\text{max}}} \pi_k \left[ \sum_{i=1}^{16} (16k-s)^{n-1} \sum_{s=1}^{16} (16k-s)^{n-1} \right] \]

\( d_{\text{coll}}(n) \) is the mean slot number, at which a collision occurs [4]:

\[ d_{\text{coll}}(n) = \sum_{k=1}^{BL_{\text{max}}} \pi_k \left[ \frac{1}{(16k)^{n-1}} \sum_{s=1}^{16} (16k-s)^{n-1} \right] \]

Throughout the present study, it is assumed: \( \beta_1 = 4 \) [bits], \( \beta_2 = 2 \) [bits], \( \text{PktLength} = 96 \) [bits], the preamble is assumed to be zero bits long (see [13] for a preamble length selection).

The collision rate is defined as a fraction of time during that transmitted packets are involved in collisions [5]. The collision rate might be calculated as follows:

\[ \text{CollisionRate}(n) = \frac{\text{PktLength}}{\tau_{\text{coll}}(n) + \tau_{\text{succ}}(n) p_{\text{succ}}^{(n)} / p_{\text{coll}}^{(n)}} \]

### 3.4. Probability of successful transmission/collision

Basing on the distribution of the contention window \( \pi = [\pi_k] \), the saturation probability of a successful transmission \( p_{\text{success}}^{(n)} \) might be calculated as the appropriate expectation:

\[ p_{\text{success}}^{(n)} = n \sum_k \pi_k \left[ \sum_{i=1}^{16} \frac{1}{16k} \left( 16k-s \right)^{n-1} \right] \]

The expression in the square brackets in (12) is the probability that a certain node among a number of \( n \) contending nodes wins a competition for the channel if a window contains of 16k slots. Hence, this probability is expressed as the sum of the following probabilities calculated for each one from 1, ..., 16k:

- probability that a winner selects a certain slot \( s, s = 1, \ldots, 16k \), which equals to \( 1/(16k) \),
- probability that all the other \( (n-1) \) nodes draw one from \( (16k-s) \) later slots, which equals to \( (16k-s)/(16k)^{n-1} \).
Next, the saturation probability of collision $p_{\text{coll}}^{(n)}$ might be simply found as:

$$p_{\text{coll}}^{(n)} = 1 - p_{\text{suc}}^{(n)}.$$  \hfill (13)

Both $p_{\text{suc}}^{(n)}$ and $p_{\text{coll}}^{(n)}$ are the relative frequencies that the successful/unsuccessful transmission occurs. Unlike for $p_{\text{suc}}^{(n)}$ and $p_{\text{coll}}^{(n)}$, the sum of throughput and collision rate is not necessarily equal to one.

3.5. Distribution of destination addresses

In order to calculate the probability of a successful transmission of a message/acknowledgement, we have to evaluate, how many nodes in the steady-state saturation conditions try to send the messages and the acknowledgements, respectively. Unlike some CSMA-based protocols (e.g. 802.11), in the predictive $p$-CSMA either the acknowledgements, or the messages compete for a channel in the same way.

Let us define a node having a message waiting for a transmission as a source of messages, and a node that competes for a channel to send an acknowledgement as a source of acknowledgements. Since we assume 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), the sum of the number of message sources and the number of sources of acknowledgements in saturation status equals the number of the nodes in the network.

After a successful reception of a message, every recipient generates an acknowledgement and (i) switches its status to the source of acknowledgements if previously it was the source of messages (i.e. schedules an acknowledgement packet as the next packet for a transmission), (ii) queues a new acknowledgement if it was the source of acknowledgements.

The probability of a successful transmission of a message/acknowledgement is proportional to the number of sources of messages/acknowledgements in the steady state of the saturated network.

A key assumption we make is that the destination address(es) of transmitted messages are distributed rather than concentrated on particular nodes. To be precise, it means that every message is addressed to recipient(s) that are currently sources of messages. Such an addressing scheme causes that the number of acknowledgement sources reaches its maximum, that is, every acknowledgement packet successfully transmitted causes a new acknowledgement source to arise. If a message recipient currently possesses a status of a source of acknowledgements, then a new acknowledgement is queued, and the number of acknowledgement sources in the network does not increase.

The uniform distribution of destination addresses seems to be the most natural assumption in network modeling, and quite realistic if a network size is large.

4. Analysis of various load scenarios

As it was stated, the performance of the predictive $p$-CSMA significantly depends on the traffic scenario, that is, the proportions between the acknowledged/unacknowledged message service, and unicast/multicast transactions. It is because the specification of a traffic scenario determines the efficiency of a load prediction.

As follows from Section 2.2, the traffic of messages is not included in the backlog estimation. If original messages present a significant percentage of the total traffic in the channel (i.e. many transactions are unacknowledged), the protocol prediction becomes less powerful. For example, for the acknowledged service and unicast transactions, in average 50% of the total load is predicted. If multicast messages are transmitted, more than 50% of the total load is predicted by the backlog measure. Finally, if all the messages are unacknowledged, then the prediction is deactivated.

We will investigate the protocol behavior for some particular load scenarios as follows: (i) unacknowledged service, (ii) multicast transactions, (iii) an example of a general case scenario.

4.1. Unacknowledged message service

According to the backlog counting algorithm (see Section 2.2), if the unacknowledged message service (UNACK) is used (regardless of the number of message recipients), the channel backlog is: (i) incremented by one, when the collision occurs, (ii) decremented by one, when the transmission of an UNACK message is successful. The transition probabilities for UNACK service are given by the following formulas:

$$p_{k,k+1}^{(n)} = p_{k}^{(n)}, \quad k = 1, \ldots, 62$$

$$p_{k,k}^{(n)} = \begin{cases} 1 - p_{k}^{(n)} & k = BL_{\text{min}} = 1 \\ p_{k}^{(n)} & k = BL_{\text{max}} = 63 \end{cases}$$

$$p_{k,k-1}^{(n)} = 1 - p_{k}^{(n)}, \quad k = 2, \ldots, 63$$

$$p_{k,0}^{(n)} = 0, \quad \text{otherwise}$$  \hfill (14)

The transition matrix $P^{(n)}$ for UNACK service is:

$$P^{(n)} = \begin{bmatrix}
1 - p_{1}^{(n)} & 0 & \cdots & \cdots & \cdots & 0 \\
1 - p_{2}^{(n)} & p_{1}^{(n)} & 0 & \cdots & \cdots & \cdots \\
0 & 1 - p_{2}^{(n)} & p_{1}^{(n)} & \ddots & \ddots & \ddots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & \cdots & \cdots & \cdots & 1 - p_{63}^{(n)} & p_{62}^{(n)} \\
0 & \cdots & \cdots & \cdots & \cdots & 0 & 1 - p_{63}^{(n)} & p_{62}^{(n)}
\end{bmatrix}$$

and the corresponding state transition diagram of the Markov chain is shown in Fig. 2.
4.2. Multicast transactions

Multicast transactions is an efficient solution for savings of the network bandwidth since a single multicast message is received by all group members.

Assume all transactions in a network containing a number of $n$ nodes are multicasted to $m$ receivers. As a result of successful message receptions, each receiving node generates a single acknowledgement. Thus, assuming the uniform distribution of message destination addresses (see Section 3.5) in the steady-state network saturation status, the total traffic transmitted in the channel consists of $1/(m+1)$ of messages and $m/(m+1)$ of acknowledgements. Thus, the probability of a successful transmission of an acknowledgement is $m$ times as much as the probability of a successful transmission of a message.

According to the backlog counting algorithm, the transition probabilities for ACK/multicast($m$) scenario:

$$p_{k+1}^{(m)} = p_k^{(m)}$$

$$p_{k+1}^{(m)} = (1 - p_k^{(m)})/m + 1, \quad k = 1, \ldots, 64 - m$$

$$p_{k+1}^{(m)} = (1 - p_k^{(m)})m/m + 1, \quad k = 2, \ldots, 63$$

$$p_k^{(m)} = 0, \quad \text{otherwise}$$

The state transition diagram of the Markov chain for ACK/multicast(3) is shown in Fig. 3. The ACK/unicast scenario is defined by $m = 1$ and a transition matrix is:

$$P = \begin{bmatrix}
1 - p_1^{(0)} & p_1^{(0)} & 0 & \cdots & 0 \\
1 - p_2^{(0)} & 1 - p_2^{(0)} & p_1^{(0)} & \cdots & 0 \\
0 & 1 - p_2^{(0)} & 1 - p_2^{(0)} & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & \cdots & \cdots & 1 - p_2^{(0)} \\
0 & \cdots & \cdots & \cdots & 0 \\
1 - p_2^{(0)} & \cdots & \cdots & \cdots & 0 \\
0 & \cdots & \cdots & \cdots & 1 - p_2^{(0)} \\
0 & \cdots & \cdots & \cdots & 0
\end{bmatrix}$$

4.3. General case load scenario

Finally, let us consider an example of a general case load scenario, where any combination of the message services and transaction types specified in any proportions, is possible. We will show the construction of the transition matrix on a basis of a concrete example. Suppose that the total traffic of messages generated to the network by a number of $n$ nodes consists of 3 equal parts as follows:

- UNACK messages,
- ACK/unicast messages,
- ACK/multicast(4) messages.

Message types are not assigned to particular nodes so each node can transmit various kind of messages. A primary traffic of acknowledged messages triggers off the derivative traffic of acknowledgements. The total traffic in the channel is a superposition of original messages and acknowledgements. The traffic of acknowledgements is 5 times as much as the traffic of each message stream (as a sum of acknowledgements generated by multicast(4) messages and unicast ones). Summing up, taking into account all the contributions, the total traffic transmitted in a channel consists of 4 types of packets:

- UNACK messages (1/8 of packets),
- ACK/unicast messages (1/8 of packets),
- ACK/multicast(4) messages (1/8 of packets),
- acknowledgements (5/8 of packets).

According to the backlog counting algorithm, the transition probabilities for the example of a general case scenario, specified above, are defined as follows:

$$p_{k+1}^{(n)} = p_k^{(n)}$$

$$p_{k+1}^{(n)} = (1 - p_k^{(n)})/8, \quad k = 1, \ldots, 62$$

$$p_{k+1}^{(n)} = (1 - p_k^{(n)})/8, \quad k = 1, \ldots, 63$$

$$p_{k+1}^{(n)} = (1 - p_k^{(n)})/8, \quad k = 1, \ldots, 60$$
\[ p_{k,1}^{(n)} = \frac{5(1 - p_{k}^{(n)})}{8} \quad k = 2, \ldots, 63 \]
\[ p_{k,3}^{(n)} = 0, \quad \text{otherwise.} \quad (16) \]

### 4.4. Saturation performance for various scenarios

The saturation channel backlog \( BL_{k}^{(n)} \) against the number of nodes for various load scenarios is presented in Fig. 4. The range of network sizes, interesting from the practical point of view, varies from dozens to even a few hundreds of nodes, since a network segment in LON networks might contain more than one subnet (a single subnet can have the maximum number of 127 nodes [1]).

The analysis of results shows that the saturation channel backlog is, as expected, a non-decreasing function of the network size. At the lower range the saturation backlog increases linearly as the number of nodes grows. All the graphs presented in Fig. 4 have the same shape, but different slopes depending on the load scenario as follows:
- 0.06 (i.e. \( 0.06 \times 16 = 1 \) time slot of \( \beta_2 \) duration) per node for unicast/ACK scenario,
- 0.04 (i.e. 0.6 slots per node) for UNACK service,
- 0.11 (i.e. 1.76 slots per node) for multicast(2).

It means in particular that adding a new node to the existing network causes to increase the mean size of a competition window in saturation conditions from 0.6 time slot of \( \beta_2 \) duration for unicast/ACK scenario to 1.76 time slot for multicast(2). For networks containing more than about 700 nodes (for UNACK messages) or even 1200 (for multicast(2)), the influence of the upper bound of the channel backlog \( BL_{max}=63 \) prevents a further extension of the competition window, and the predictive CSMA reduces to 0.0009-persistent CSMA. Summing up, the prediction is effective for the network sizes up to at least a few hundreds of nodes.

The saturation throughput/collision rate versus the number of nodes are presented in Fig. 5.

For small networks (up to 10 nodes), the saturation throughput equals about 80% for any ACK service. For larger networks, the throughput deteriorates, but establishes next at the constant level for a network containing more than 100 nodes. We call both measures the sustained throughput and the sustained collision rate. Both measures constitute the worst-case protocol performance if the prediction built in the protocol is effective, (i.e. current backlog is not limited by \( BL_{max} \)). The sustained throughput depends on the load scenario, and equals from about 48% for UNACK service, and 63% for unicast to about 72% for multicast(2) (Fig. 4).

### 4.5. Simulative verification of analytical results

In order to verify the analytic approach we have run the simulations for the network saturation status. The simulation model implemented in LabView corresponds to the analytic model. In particular, the channel is assumed to be noise-free, the channel backlog is considered as a global channel measure, and the numbers of both outgoing and incoming transactions managed by a node, are unlimited.

The simulator tries to distribute destination addresses of messages as much as possible. The addresses of receiving nodes are randomly selected from the nodes that are currently the sources of messages.

However, due to random behavior of CSMA-based protocols there is no guarantee that no recipient possesses a status of the acknowledgement source at some time. For example, if the number of the message sources is less than \( m \) at the time when a multicast message to \( m \) recipients is transmitted, then at least one message recipient must be a source of acknowledgements. The statistics about queuing of acknowledgement packets are reported by the simulation.
program, but such situations are really rare and do not influence the results considerably.

The simulation starts when the channel is idle and all the nodes are sources of messages. Next, the transient zone appears, when the nodes permanently try to access the channel and the mean channel backlog grows. Since the simulation model belongs to non-terminating systems and the steady state theoretically is never reached, we detect it with a finite accuracy. The detection relies on the search of the constant value of the mean backlog, rather than the constant value of the current backlog. Therefore, we used the moving averages defined over a window of observations (i.e. a certain number of packet cycles, increasing with the number of nodes). Moving averages filter the higher frequency components in the mean backlog, arisen from the random behavior of the CSMA algorithm on the one hand, and remove also the influence of the transient zone on the estimation of saturation backlog on the other. Simulation outputs are the histogram of backlog states, saturation backlog, relative frequencies of successful and unsuccessful transmissions, throughput, collision rate.

Figures 6 and 7 shows the percentages of sources of messages/acknowledgements in the steady state of saturated network for multicast (2), and an example of a general case scenario specified in Section 4.3, respectively. In the latter case, the number of message sources is defined as a sum of UNACK messages, and multicast(4) ones. Proportions between appropriate packet sources determines the transition probabilities between backlog stages. As follows from (15), the number of acknowledgement sources should be two times as much as the number of message sources for multicast(2), which is definitely displayed in Fig. 6. Similarly, as follows from (16), the relative number of acknowledgement sources should equal 62.5%, and message sources 37.5%, respectively. Both values might be read from Fig. 7 also.

Fig. 8 shows the relative frequencies of collisions and successful transmissions of messages and acknowledgements for a general case scenario. All these measures are the experimental equivalents of appropriate probabilities defined in Section 3.4. Fig. 8 shows that the successful transmissions of acknowledgements are about 5/3 times more frequent than transmissions of messages. This result is consistent with transition probabilities defined in (16). The sustained frequency of collisions read from Fig. 8 equals about 27% (compare with the last row in Table 1, where the appropriate probability is equal to 3/11).
5. Generalization of the approach for various load scenarios

Let us generalize the obtained results for various load scenarios. The range of traffic-dependent backlog modifications is limited by the length of 6-bit long Delta_BL field. Thus, from the point of view of the traffic prediction, a number of 64 packet types might be distinguished as follows: (i) UNACK messages and acknowledgements (Delta_BL = 0); (ii) ACK/multicast(i) messages (Delta_BL = i, i = 1,...,63). In particular, the multicast(1) represents a unicast/ACK message.

By the definition, the primary traffic consists of original messages only. Acknowledgements are generated in case of successful reception of ACK messages. As it was stated, the total traffic in the channel is a superposition of the traffic of messages and acknowledgements.

5.1. Primary traffic specification

Let us assume that the multicast(j) messages (j = 0,...,63) present the smallest non-zero component in the primary traffic of original messages. Next, denote the scaling factors $\gamma_{i}, i = 0,...,63$ defining all the other components of the primary traffic with a normalization to the smallest one. Note that by the definition: $\gamma_i \geq 1, i \neq j$, and $\gamma_j = 1, i = j$. Scailing factors $\gamma_0, ..., \gamma_{63}$ define completely a structure of the primary traffic.

5.2. Traffic superposition

Since a successfully transmitted acknowledged multicast(i) message generates a number of i acknowledgements, the total traffic in the channel:

$$\alpha_i (\gamma_0 + \gamma_1 + \gamma_2 + 2\gamma_2 + ... + \gamma_{63} + 63 \gamma_{63}) = 1$$

where $\alpha_i$ represents a percentage of multicast(i) messages in the total traffic. Futhermore:

$$\alpha_i = \frac{1}{\sum_{i=0}^{63} [(i+1)\gamma_i]}$$

(18)

The formula (18) defines how to calculate $\alpha_i$ on a basis of scaling factors $\gamma_i, i = 0,...,63$. Next:

$$\alpha_k = \gamma_k \alpha_j = \frac{\gamma_k}{\sum_{i=0}^{63} [(i+1)\gamma_i]}, \forall k = 0, ..., 63$$

(19)

5.3. Transition probabilities for a general scenario

The appropriate transition probabilities are calculated basing on (19) as follows:

$$p_{k,k} = \alpha_i \gamma_0 = \frac{\gamma_k}{\sum_{i=0}^{63} [(i+1)\gamma_i]}$$

$$p_{k,k+1} = 1 + (\alpha_2 - 1) \frac{\gamma_2}{\sum_{i=0}^{63} [(i+1)\gamma_i]}$$

$$p_{k,k+m} = \frac{\gamma_m}{\sum_{i=0}^{63} [(i+1)\gamma_i]}$$

(20)

$$p_{k,k+1} = 1 + (\alpha_2 - 1) \frac{\gamma_2}{\sum_{i=0}^{63} [(i+1)\gamma_i]} + 1 - p_{suc}$$

The formula (23), derived above, specifies the worst-case probability of backlog in the channel state at such a value that the expected change of the backlog equals zero, i.e. the probability of backlog increase is equal to the probability of backlog decrease:

$$p_{k,k+1} = p_{k,k+2} + 3p_{k,k+3} + ... + 62p_{k,k+62}$$

(21)

or equivalently

$$\sum_{k=0}^{62} p_{k,k+1} = 0$$

(22)

Setting the appropriate transition probabilities (20) to (22), we obtain the formula for sustained probability of a successful transmission:

$$p_{suc} = \frac{\sum_{i=0}^{63} [(i+1)\gamma_i]}{\sum_{i=0}^{63} [(i+2)\gamma_i]}$$

(23)

The formula (23), derived above, specifies the worst-case probability of a successful transmission for the predictive p-CSMA protocol loaded with any traffic scenario if the protocol behavior is not limited by the maximum channel backlog. The equation (23) is a simple and closed-form formula obtained without a calculation of a Markov chain stationary distribution but only using the equilibrium analysis.

The formula (23) presents a new and important result for predictive p-CSMA performance evaluation. The sustained probability of a successful transmission does not depend on the network size, but only on the structure.
of the traffic transmitted in the channel, represented by scaling factors \( \gamma_0, \ldots, \gamma_{63} \). Although, the effect of keeping the high throughput of the predictive CSMA has been known [3,7,9], there has been yet no explicit quantitative evaluation of sustained probability of successful/unsuccessful transmission.

5.5. Sustained probability \( P_{\text{succ}} \) versus throughput

Since the saturation probability reaches its sustained value if a network contains more than 50 nodes, then the following approximation is valid:

\[
d_{\text{succ}}(n) \approx d_{\text{coll}}(n) \approx 2 \quad \text{for} \ n > 50 \text{ nodes} \quad (24)
\]

Next, assuming typical settings (i.e. \( \beta_1 = 4 \) [bits] and \( \beta_2 = 2 \) [bits]), we have:

\[
\tau_{\text{succ}} \equiv \tau_{\text{coll}} = \beta_1 + \beta_2 + \text{PktLength} \approx \text{PktLength} \quad (25)
\]

with a few percent accuracy for typical packet lengths (e.g. 12 bytes). Finally, for \( n > 50 \) nodes:

\[
\text{Throughput}(n) \approx P_{\text{succ}}, \quad (26)
\]

\[
\text{CollisionRate}(n) \approx P_{\text{coll}} \quad (27)
\]

Therefore, the sustained probability of successful and unsuccessful transmission, \( P_{\text{succ}} \), \( P_{\text{coll}} \) might be used as the approximations of the saturation throughput/collision rate. Table 1 summarizes the sustained throughput and collision rate for selected load scenarios. A comparison with sustained throughput, obtained using a calculation of stationary distribution (see Section 4.4), shows that both results are consistent, and the approximations (26) and (27) are valid with a few percent accuracy.

**Table 1** The sustained probability of successful and unsuccessful transmission as approximations of the worst-case throughput and collision rate for particular load scenarios.

<table>
<thead>
<tr>
<th>Load scenario</th>
<th>( P_{\text{succ}} )</th>
<th>( P_{\text{coll}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNACK: ( \gamma_0 = 1 )</td>
<td>( P_{\text{succ}} = \gamma_0 )</td>
<td>( P_{\text{coll}} = \gamma_0 )</td>
</tr>
<tr>
<td>50% UNACK, 50% ACK/unicast: ( \gamma_0 = 0.5 ), ( \gamma_1 = 0.5 )</td>
<td>( P_{\text{succ}} = \gamma_0 )</td>
<td>( P_{\text{coll}} = \gamma_0 )</td>
</tr>
<tr>
<td>ACK/unicast: ( \gamma_1 = 1 )</td>
<td>( P_{\text{succ}} = \gamma_1 )</td>
<td>( P_{\text{coll}} = \gamma_1 )</td>
</tr>
<tr>
<td>50% unicast, 50% multicast(2): ( \gamma_0 = 0.5 ), ( \gamma_2 = 0.5 )</td>
<td>( P_{\text{succ}} = \gamma_2 )</td>
<td>( P_{\text{coll}} = \gamma_2 )</td>
</tr>
<tr>
<td>ACK/multicast(2): ( \gamma_2 = 1 )</td>
<td>( P_{\text{succ}} = \gamma_2 )</td>
<td>( P_{\text{coll}} = \gamma_2 )</td>
</tr>
<tr>
<td>General case scenario (Sec. 4.3) ( \gamma_0 = \gamma_1 ), ( \gamma_1 = \gamma_2 ), ( \gamma_4 = \gamma_5 )</td>
<td>( P_{\text{succ}} = \gamma_1 )</td>
<td>( P_{\text{coll}} = \gamma_1 )</td>
</tr>
</tbody>
</table>

Note that the sustained probability of collision for a general case scenario (Sec. 4.3) found in the last row of Table 1 is consistent with the relative frequency of collisions estimated by simulation (see Fig. 8).

Strictly speaking, the formula (23) is valid for \( BL_{\text{max}} \to \infty \). Then, the sustained probability of a successful transmission ranges from \( \gamma_0 \) for UNACK message service to \( 64\% \) for multicast(63) transactions. Since \( BL_{\text{max}} = 63 \) in the existing protocol implementation [1], the upper bound of the throughput is limited by the performance of 0.0009-persistent-CSMA protocol. Due to the upper limit of the contention window size, the maximum throughput of predictive CSMA is not higher than about 80% for any case load scenario.

6. Conclusions

The analytic approach to estimate the worst-case performance evaluation of the predictive p-CSMA for any case of load scenario, has been presented. Some new results have been obtained. First, the calculation of transition probabilities are exemplified and validated by a simulation for a given specification of a general-case load scenario. Next, the generalization of the analytic approach is formulated.

Finally, the simple and closed-form formula based on the equilibrium analysis to find the sustained throughput and collision rate, is formulated.

**References**