Channel Backlog Estimation in LonWorks®

authors:

Zdeněk Hanzálek, Josef Čapek

Trnka Laboratory of Automatic Control

Department of Control Engineering

Karlovo nám. 13, Czech Technical University in Prague

121 35, Prague 2, Czech Republic

hanzalek@rtime.felk.cvut.cz

http://novell.felk.cvut.cz/~hanzalek.rtime/

phone: ++420 2 24357434

fax: ++420 2 24357610

1. Introduction

The fieldbuses are used primarily as communication systems for exchange of information between automation systems and distributed field devices. Many fieldbus systems today are open standard systems. The users are no longer tied to individual vendors and they have an opportunity of choosing a suitable product from a wide variety of products. Chosen fieldbus type and its correct functionality play the key role in design of the distributed control system. That is why various studies have been published on this topic [1], [2], and [7].
One of the most important fieldbus standards in building automation is LonWorks. It was developed by Echelon® and in addition to many others, LonWorks uses wide variety of Physical layers including power lines. LonTalk, which is communication protocol of LonWorks, uses non-deterministic media access method (MAC) called predictive p-persistent CSMA. Despite of non-deterministic MAC method LonWorks is used in soft real-time applications with moderate traffic. An advantage of the protocol is that it keeps the collision ratio independent of the channel utilisation and it uses technique for partial predication of the channel backlog.

Many people have analysed MAC layers of communication protocols in last three decades. Orientation of research in seventies was given namely by L. Kleinrock, who, together with Tobagi, has presented queuing models of Aloha and CSMA methods under infinitely large population assumption [6, part I] and who solved several issues like hidden terminal problem [6], part II. Tobagi and Hunt in [13], presented throughput-delay analysis of multiple CSMA methods and they analysed the effect of bimodal distributions of parameters. In eighties, many people analysed CSMA/CD (the MAC method used in the Ethernet). Coyle and Liu analysed the stability of communication protocols, they also defined a stability measure, and presented analysis assuming a finite number of hosts. Takaki and Kleinrock [17] presented an analytical formula for throughput of persistent CSMA/CD as a function of offered traffic. Many people have also analysed Ethernet: Metcalfe and Boggs presented the first throughput analysis of the Ethernet. Almes and Lazowska, [15], studied response times of the Ethernet as a function of offered traffic. Tobagi and Gonsalves in 1986 [16], simulated the Ethernet studied the influence of several parameters to the Ethernet's performance. They also studied the influence of different distances among particular nodes. Boggs at al [18] in 1988 studied some implementation problems regarded with Ethernet. Mole studied Binary Exponential Backoff Congestion control mechanism of the Ethernet and he proposed a new one, which solves some drawbacks of BEB [11].

A communication protocol is a set of rules usually used for information exchange among nodes in computer networks. They have to be exact and error-free in order to the communication is efficient and the network resources are well utilized. The communication protocols tend to be developed using rigorous methods and they are checked and proved by standardization organizations like ANSI or IEEE. Formal methods and formal description techniques are used in development of communication protocols, their verification and testing of their conformance against standards. Among most often used formal description techniques belong LOTOS, Estelle, and SDL. Protocol verification is the proving of protocol design represented by formal specification. Rather tan on verification issues this text focuses on performance analysis of communication networks, which is quantifying of the network behaviour in terms of some meaningful numerical measures. For us, the key to define a general performance measure is to consider networks as general open queuing systems with inputs, called the offered load and outputs, which we call the throughput. *Offered traffic*, [3] is an average number of jobs offered to the queuing system over a time interval. Traffic offered to channel consists not only of new packets but also of previously collided packets (we have a non-retrial queuing system) and we say that the offered load is a fraction of network bandwidth that the nodes would use if they had complete access to the channel (when there were no collisions on the channel). We define the *throughput* as the fraction of the nominal network bandwidth that is used for carrying successfully transmitted data and the throughput is also defined as an average number of successfully transmitted packets per packet transmission time [3].

This article is organised in the following way: first MAC of LonTalk protocol is briefly explained, then token player for timed Petri Nets is presented and finally a Petri Net model of LonTalk MAC sub-layer is given in the three steps. It is demonstrated why collision ratio rise up in the model with propagation delay and how this problem can be resolved by the channel backlog estimation.

2. MAC sublayer of the LonTalk Communication Protocol

The LonTalk *MAC (Media Access Control)* sublayer uses a protocol called *Predictive p-persistent CSMA (Carrier Sense, Multiple Access)*, which is a collision detection technique that randomises channel access using knowledge of the expected channel load [4]. Optional features of this media access protocol, like optional priority slots or optional collision resolution, are not assumed in the following text.

2.1. Predictive p-persistent CSMA

Like CSMA, Predictive p-persistent CSMA senses the medium before transmitting. A node attempting to transmit monitors the state of the channel (see Figure1), and when it detects no transmission during β_1 period, it asserts the channel is idle. Then it generates the random delay Δ_T before transmission. If the channel is idle when the delay Δ_T expires, the node transmits; otherwise, Link layer receives an incoming packet and the algorithm repeats.
Predictive p-persistent CSMA generates the delay Δ_T as an integer number of discrete time slots of width β_2. The delay Δ_T is generated from the randomising window (0..*W_Base*), which changes with respect to actual and predicated traffic on the channel. In other words *W_Base* is defined as product of *BL* (an integer estimate of the current channel backlog) and a basic window size. If there is no traffic on the channel or if the traffic is very low, then *BL* is equal to 1. With growing utilisation of the channel the *BL* grows and the randomising window (0..*W_Base*) enlarges. β_1 and β_2 are time constants given by Physical layer parameters and respect propagation delay defined by the media length, detection and turn-round delay within MAC sublayer. In Figure 1 Δ_T*_mean* is given as *W_Base/2* because variable Δ_T is uniformly distributed.
The MAC algorithm predictability is based on backlog estimation. Each node maintains an estimate of the current channel backlog *BL*, which is incremented as a result of the packet transmission/reception and decremented periodically every packet cycle. Each packet of MAC sublayer contains a variable representing prediction of the traffic arising as a result of processing this packet (the variable represents the number of

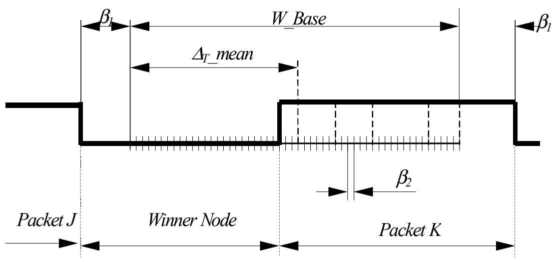

Figure 1 – MAC protocol timing

messages that the packet shall generate upon reception). By adjusting the size of the randomising window as a function of the predicated load, the algorithm keeps the collision ratio constant and independent of the load.

2.2. Interface to Link Layer and interface to Physical Layer

MAC sublayer is responsible for the access to the media. Communication among MAC sublayer, Link layer and Physical layer is shown in Figure 2. A frame reception is handled entirely by Link Layer, which notifies MAC sublayer about each correctly received packet via the *Frame_OK* primitive. Link layer uses primitive *M_Data_Request* to pass an outbound *Link Protocol Data Unit (LPDU)* to MAC sublayer. Next using *P_Data_Request* the packet is sent to Physical layer for an immediate transmission. Physical layer returns one of the three admissible results after the transmission:

- *success*, indicating the packet transmission
- *request_denied*, indicating an activity was detected on the line prior to the start of the transmission
- *collision*, indicating a collision was detected during the transmission.

Ch_Activity is an indication of the channel activity provided by Physical layer, and *P_Data_Indication* is information about an incoming packet.

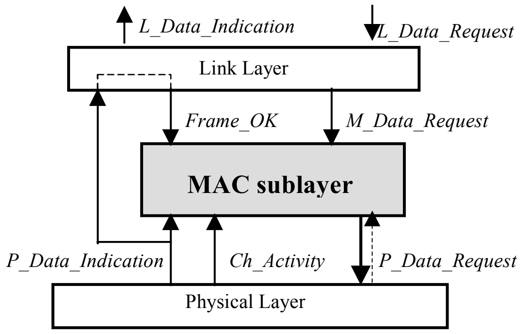

Figure 2 – MAC sublayer

3. Petri Net token player

A simulation of the above mentioned communication protocol with Petri Net requires a model, which relates time to the transitions. The time related to the transitions is either deterministic time (*timed transitions*) or stochastic time with uniform probability distribution function (*stochastic transitions*). Not all events that occur in the protocol correspond to the end of time-consuming activities. Such behaviour can be described with transitions that consume no time called *immediate transitions*.

The model adopted in this text considers that a transition does not reserve the tokens in its input places. If required the reservation can be conveniently modeled using immediate transitions to separate conflict from transition timing specification.

At each transition firing, the counters of all timed transitions, which are disabled, are restarted whereas the counters of all the timed transitions, which are enabled, hold their present value (called *enabling memory* model). Single server semantics is assumed. This means that enabling sets of tokens are processed serially and one can imagine that a self-loop place marked with one token is associated to each transition.

With respect to definitions in [5], and [10] the Petri Net used in this text should be exactly called *Extended Generalised Deterministic and Stochastic Petri Net* as it allows assigning uniform probability distribution function to the transitions.

The functionality of the model under consideration is fully specified by the interpretation of the token player given by Algorithm 1.

Algorithm 1:

Input data:
marked Petri Net <*P,T,Post,Pre,M*> with *type* (immediate/timed/stochastic) and *time* of each transition
length_of_simulation – scalar integer value
Output data:
firing_sequence

%phase1 – initialisation
set *counter* of all transitions to initial value
%main simulation cycle
while $current\_time \leq length\_of\_simulation$
    %phase 2 - find enabled and firable transitions
    for all transitions T_j
        if $\forall P_i \in {}^{\circ}T_j \;/ M(i) \geq Pre(i,j)$ **then** T_j is enabled **else** set *counter* of T_j to initial value
        if T_j is enabled and *counter* of T_j is equal to 0 **then** T_j is firable
    %phase 3 – either change marking or change time
    if there is firable transition
        then fire one randomly chosen firable transition
            add fired transition into *firing_sequence*
            set *counter* of fired transition to initial value
        else decrement *counters* of all enabled transitions and increment *current_time*

With respect to Algorithm 1 the following drawbacks should be mentioned:

- In case of an effective conflict, no token reservation is assumed. It means that the first fireable transition (enabled & counter=0) wins. When "reservation" behaviour is needed, immediate transition should precede a timed or stochastic transition (e.g. T_{27} precedes T_{28} in Figure8).
- In case of the actual conflict (two and more fireable transitions in conflict), representing the system non-determinism, the winning transition is chosen in random manner. It means that one possible firing sequence is chosen from several ones.
- The initial value of the transition counter is either 0 (in case of an immediate transition), or positive non-zero integer value (in case of a timed transition) or a random integer number from interval <1,upper_margin> (in case of a stochastic transition with uniform distribution).

4. Predictive p-persistent CSMA model

A complete Petri Net model of one node MAC sublayer using Predictive p-persistent CSMA is shown in Figure 11. Immediate transitions are not labelled (e.g. T_1, T_2, T_3), timed transitions are labelled by "t" with corresponding time (e.g. 5 ticks in case of T_4) and stochastic transitions are labelled by "s" with the corresponding upper margin of the uniform distribution interval. Figure 11 consists of the following components:

- Idle Channel Detection Model is situated at the top (P_3 - P_5, T_2 - T_4).
- Left side of the figure represents the Backlog Estimator Model (P_{17} - P_{27}, $T_{18} – T_{36}$).
- Physical Layer Model is at the right side ($P_8 – P_{12}$, $T_8 – T_{11}$).
- Medium Access Model is at the bottom of the figure (P_{13} - P_{16}, $T_{12} – T_{17}$).

This model is relatively complex and influences of various components on the system performance (namely network throughput) are not clear when the simulation is carried out. That is why the model will be constructed in three steps (Figure 3, Figure 7 and Figure 11) and simulation results will be evaluated after each of them.

4.1. No propagation delay, no backlog estimation

The Petri Net shown in Figure 3 models MAC sublayer of one node. Propagation delays in Physical layer and backlog estimation are not considered in this model.
Transition T_1 is supposed to be connected to Link layer (this fact is not represented in Figure 3 as this one-node model is further integrated with other nodes and the traffic generation). So T_1 fires when the node finished sending a previous packet (there is a token in P_1) and there is a new packet to be sent (model receives *M_Data_Request* from Link layer). A token in place P_2 represents a situation, when MAC sublayer has data for transmission, but the channel was not idle in last β_1 period. The Idle Channel Detection part (conservative component $\{P_3, P_4, P_5\}$) monitors the channel activity (conservative component $\{P_{11}, P_{12}\}$ where P_{11} represents an active channel and P_{12} represents a passive channel). During any channel activity place P_3 is marked. If there is no activity on the bus, the token "moves" from place P_3 to P_4. As soon as there is some activity on the channel, the token returns from place P_4 to place P_3. If the token resides in place P_4 continuously during β_1 period (corresponding to the timing of T_4), transition T_4 fires, and the token moves into place P_5. Marked place P_5 means that the channel is in the idle state and consequently the transition T_6 can be fired. If traffic appears on the channel when the token resides in place P_5, then it returns immediately to place P_3, and the action repeats. The transition T_7 fires after waiting for a random time period, and the token moves into place P_7 (*P_Data_Req* primitive in Figure 2) - the model checks again the state of the channel (please notice there is no actual conflict between T_{12} and T_8 because of the conservative component $\{P_{11}, P_{12}\}$). If there is some activity (*request_denied*), the token is returned to P_2 through transition T_{12} and the MAC algorithm repeats. If there is no activity on the channel a packet cycle starts by firing transition T_8 (*success*). For the simulation purposes it is supposed that each packet cycle consists of one request frame of the fixed length (transition T_9) and one response frame of the fixed length (transition T_{10}). In other words, each request frame is supposed to be acknowledged by one response frame from one receiver node. Firing the transition T_{11} represents closing the packed cycle and the system returns to the initial state (token in P_1).

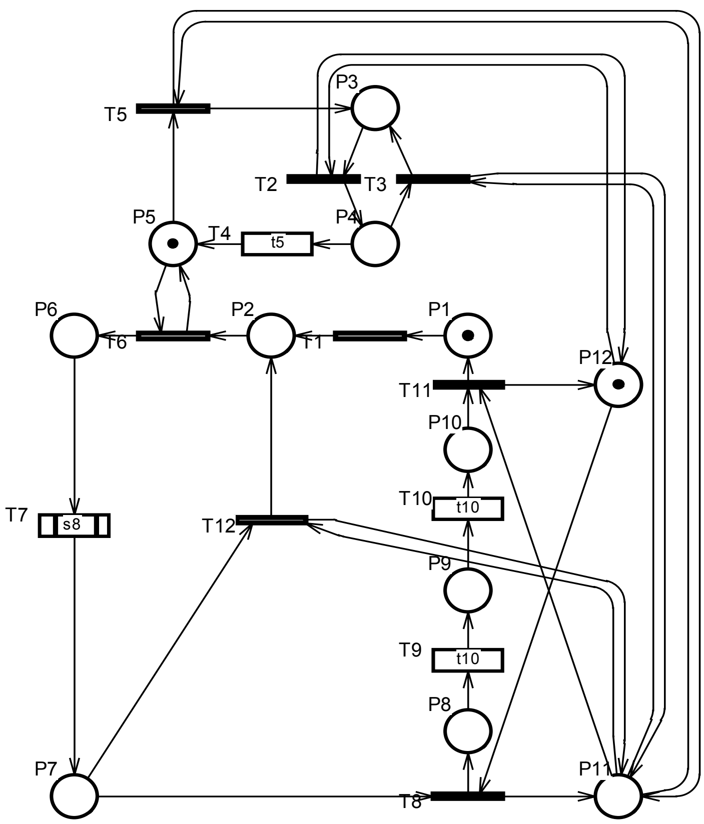

Figure 3 – Node structure (no propagation delay, no backlog)

Figure 4 shows a configuration of the network consisting of five nodes (*MAC node 1* to *MAC node 5* and corresponding *M_Data_Request* messages) sharing the bus represented by the conservative component $\{P_{11}, P_{12}\}$. Marked place P_{12} represents passive channel and, on the contrary, marked place P_{11} represents active channel.

Media access is here fully determined by the shared place P_{12}. The node firing its transition T_8 gets the bus and other nodes cannot interrupt the transmission, as place P_{12} is empty during the packet cycle. So consequently no collision can occur and the system saturates with increasing offered traffic, defined as the traffic from upper layers corresponding to the number of firings of T_0 per length of simulation (when packets are of constant length τ or when τ is the mean packet length, the offered traffic could be 'normalised' to τ).

The Petri Net token player explained in the previous paragraph was used to simulate the network abbreviated in Figure 4. Each simulation was run separately for given offered traffic (number of firings of the stochastic transition corresponding to its upper margin X). Analysis of the resulting firing sequence showed the number of firings of the transitions T_8 and T_{12} corresponding to the successful access or to the denial.

The simulation results are shown in Figure 5 and 6. If the traffic is low, then there are very few denied requests and the network throughput is more or less equal to the offered traffic (all requests are satisfied). When the offered traffic increases, the network throughput reaches its saturation point and there are more denied requests.

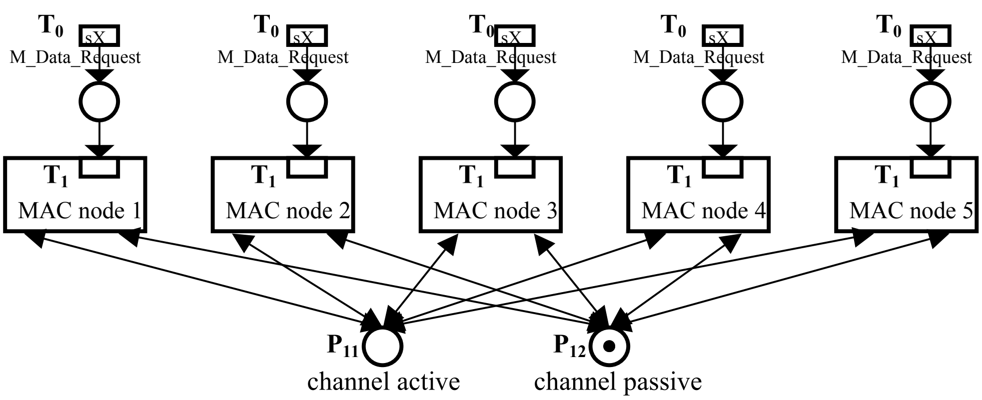

Figure 4 – Interconnection of five nodes, (no propagation delay, no backlog)

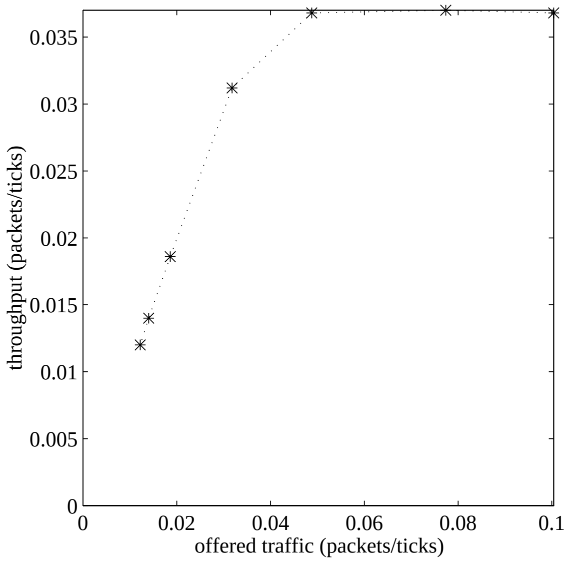

Figure 5 – Network throughput (no propagation delay, no backlog)

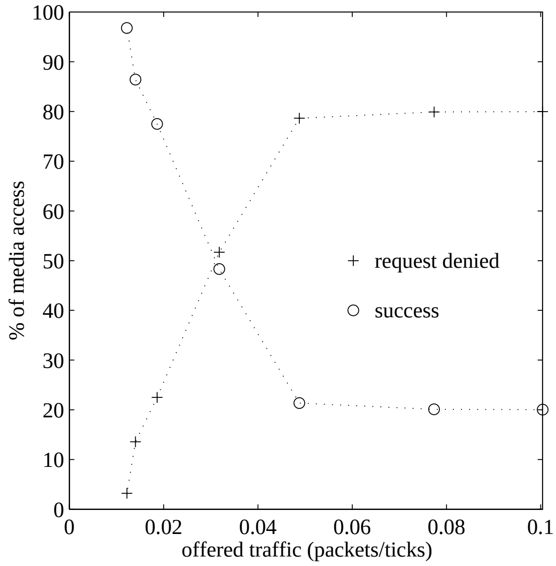

Figure 6 - Ratio of successful attempts to access the media (no propagation delay, no backlog)

4.2. Propagation delay, no backlog estimation

Contrary to the previous paragraph, if two physical nodes A and B perform media access at the same moment, then both of them can detect free state of the channel and consequently both of them can start to send their data. This results in a collision. Petri Net given in the previous paragraph does not model this situation as P_{12} is shared by several nodes and the actual conflict between T_8 in the node A and T_8 in the node B is resolved by token player in random way (this resolution of the non-determinism is acceptable since the network analysis is based on the statistical processing of the firing sequence).

Furthermore, the model presented in the previous paragraph does not take into consideration the propagation delay depending on the propagation characteristics of the channel (media length, number of repeaters). In fact, node A is notified about the activity of node B after the propagation delay and vice versa. The collision then occurs not only when both nodes access the media exactly at the same moment but also in the case when the time difference (between the moment when node A starts to send data and node B starts to send data) is less than propagation delay between node A and node B. Consequently, when collision occurs, neither node A nor node B succeeds, because data have been corrupted.

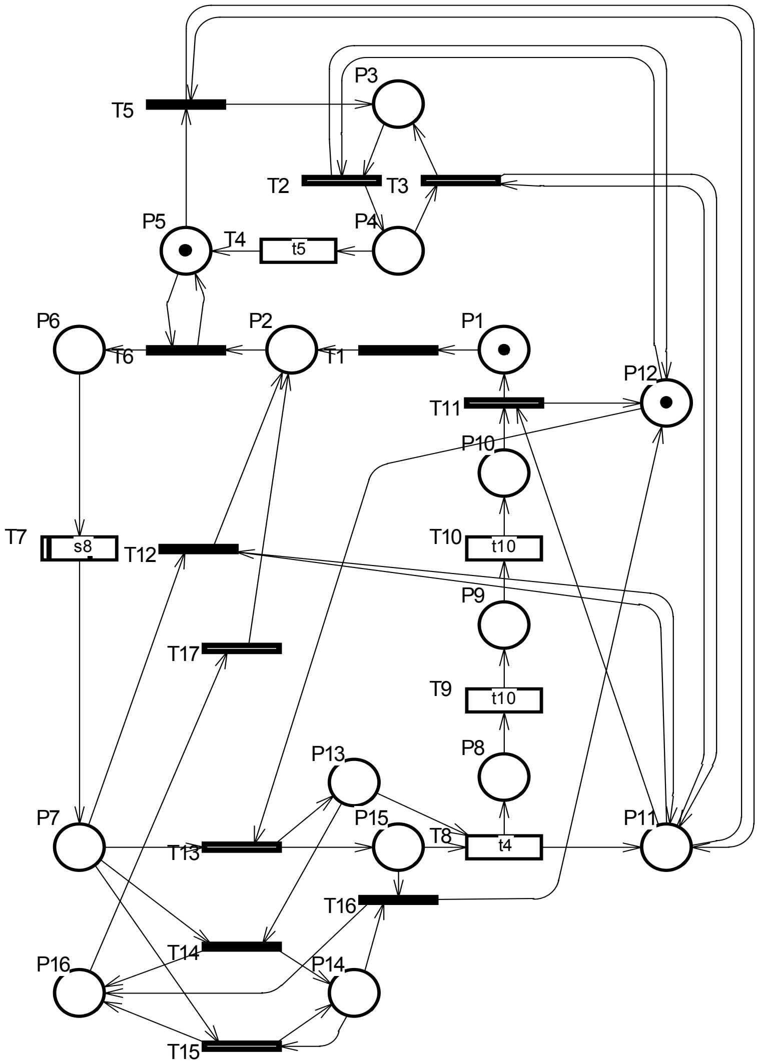

Figure 7 – Node structure (propagation delay, no backlog)

The Petri Net model of one node MAC sublayer with propagation delay is shown in Figure 7 and the complete abbreviated network model is shown in Figure 8. The transition T_8 represents the propagation delay. Shared

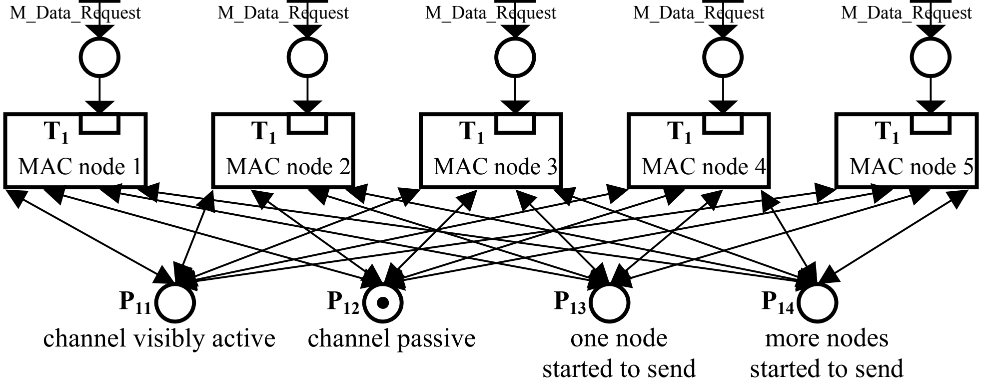

Figure 8 – Interconnection of five nodes, (propagation delay, no backlog)

place P_{13} represents the case when one node started to send data and shared place P_{14} represents the case when two or more nodes started to send data. If a node tries to accesses the media (token in P_7) and the channel is not visibly active (no token in P_{11} – transition T_{12} is not enabled) then three possible situations can occur:

1. If the node is the first one starting to send data, then T_{13} fires. If this situation does not change during timing of T_8 then there was no collision during the time corresponding to the propagation delay and T_8 is fired (*success*). The channel becomes visibly active. If this situation changes during the timing of T_8 (the token was moved from shared place P_{13} to shared place P_{14} by another node trying to access the media) then T_{16} is fired, the channel becomes passive (P_{12}) and a collision is detected (T_{17}).
2. If the node is the second one sending data, then T_{14} fires and a collision is immediately detected (T_{17}).
3. If the node is the third (fourth, and more) one sending data, then T_{15} fires and a collision is also immediately detected (T_{17}).

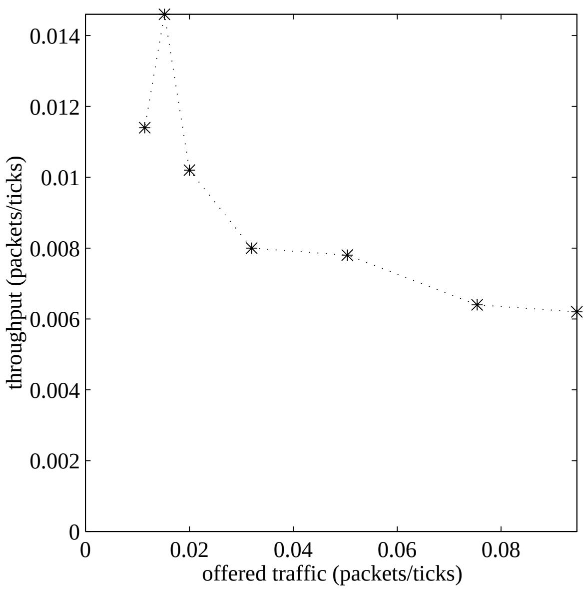

Figure 9 - Network throughput (propagation delay, no backlog)

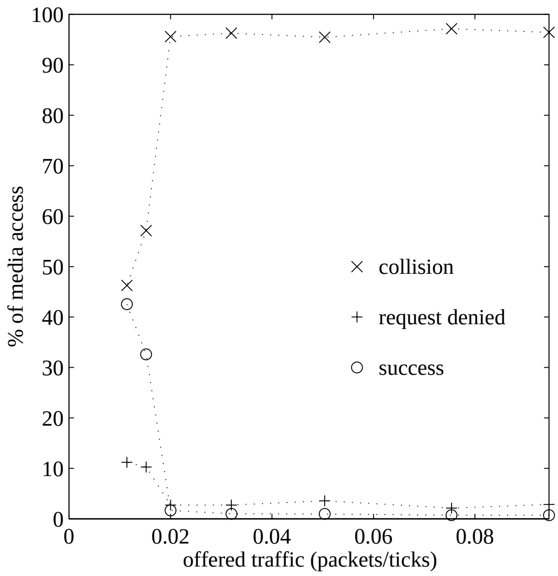

Figure 10 – Ratio of MAC results (propagation delay, no backlog)

With respect to the model mentioned above the following should be noticed:

- All shared places indicating channel status form one conservative component $\{P_{11}, P_{12}, P_{13}, P_{14}\}$ containing one token. The first node, which starts sending, detects the collision too. Only this node deposits token from P_{14} to P_{12}.

- The collision is detected in the interval (0, timing of T_8) with probability distribution, which can differ from the real one depending on the physical distance between various nodes on the network. The timing of T_8 should be interpreted as maximum propagation delay between two nodes.

Figure 10 illustrates the resulting collisions having significant influence on the network throughput (see Figure 9). When the offered traffic is very low, then the system behaves in a similar way as the one without propagation delay (compare Figure 5 and Figure 9). As far as the offered traffic increases, there are more collisions and the network throughput drops down rapidly. This phenomenon is sensitive on the propagation delay (corresponding to the timing of T_8) and CSMA random delay (corresponding to timing of T_7). In order to demonstrate a positive influence of the backlog estimation in the next paragraph, the timing of T_7 and T_8 have been chosen in such a way that excessively many collisions occur even in the network consisting of five nodes.

4.3. Propagation delay, backlog estimation

The Petri Net model of one node with Predictive p-persistent CSMA is shown in Figure 11.

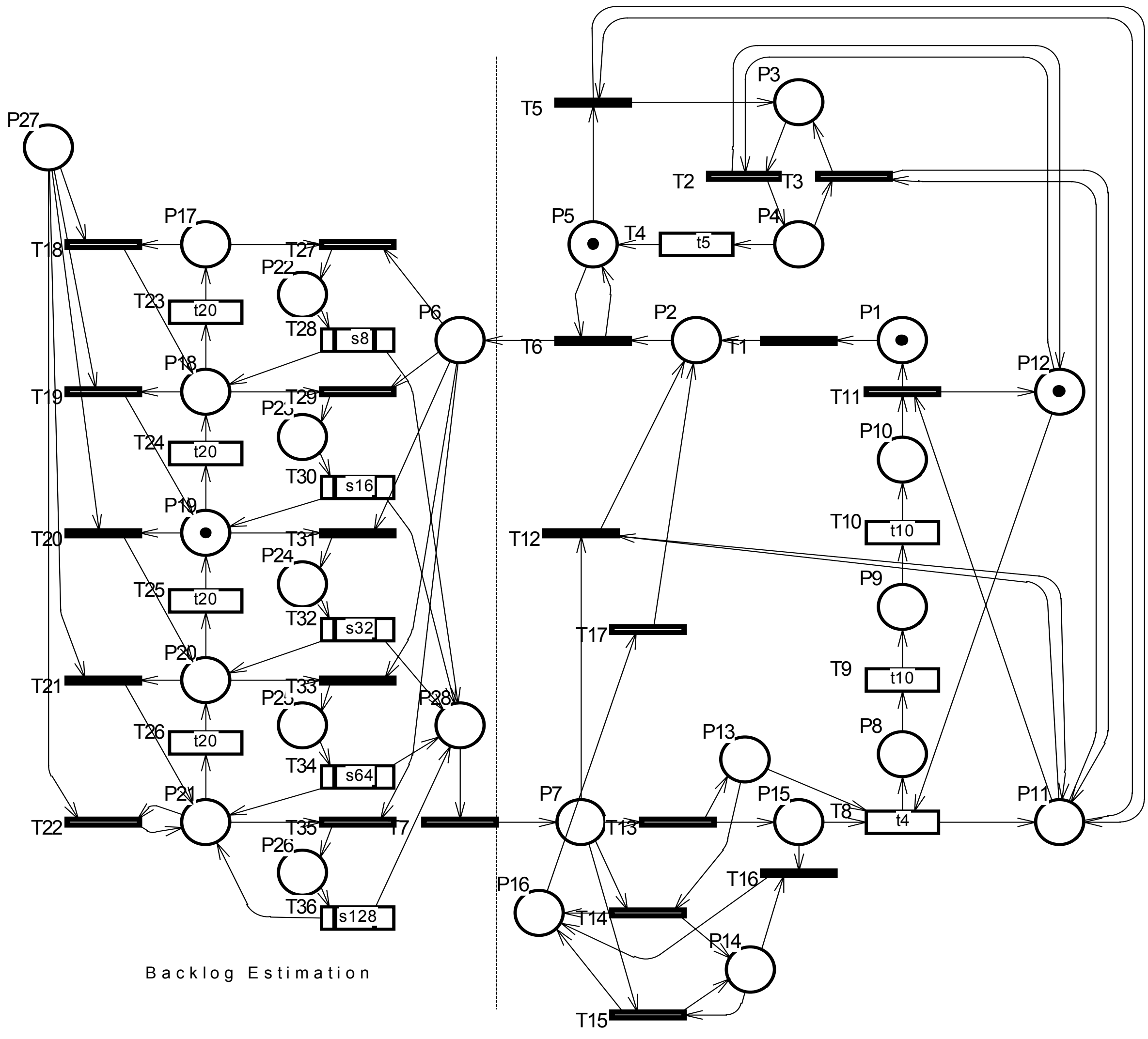

Figure 11 – Node structure (propagation delay, backlog)

The channel backlog (*BL*) corresponds to the mean value of the predicated channel load. All nodes increment the estimated backlog by one when they receive a request frame (owing to the broadcast capability of the bus topology, a particular node is aware of this frame even if it is not the destination node) or when they send a request frame. The estimated channel backlog is decremented by one at the end of each packet cycle (request and response frame). The model assumes that all frames have the same length (10 ticks) and that all packets are acknowledged (T_9 – request, T_{10} - response).
Because of the lack of space only 5-level backlog estimator is modelled in Figure 11. Places P_{17}, P_{18}, P_{19}, P_{20}, and P_{21} represent the state of the backlog estimator and correspond to the current value of channel backlog *BL*. Place P_{17} represents BL=1, $P_{18} \sim BL=2$, and so on. Message *Frame_OK* (marked interconnection place P_{27} – see also Figure 12), indicates a successfully received request frame being a broadcast message from Physical layer (from the transition T_9 in sending node). Consequently transitions T_{18}, T_{19}, T_{20}, T_{21} increment the current value of *BL*. Transition T_{23}, T_{24}, T_{25}, and T_{26} are *timed transitions* with deterministic firing time corresponding to the packet cycle. Via T_{28}, T_{30}, T_{32}, T_{34}, *or* T_{36} Predictive p-persistent CSMA generates a random time delay from a random timing interval, which is dependent on the estimated channel backlog. Please notice, that just one of the transitions T_{28}, T_{30}, T_{32}, T_{34}, T_{36} is fired at a given time.

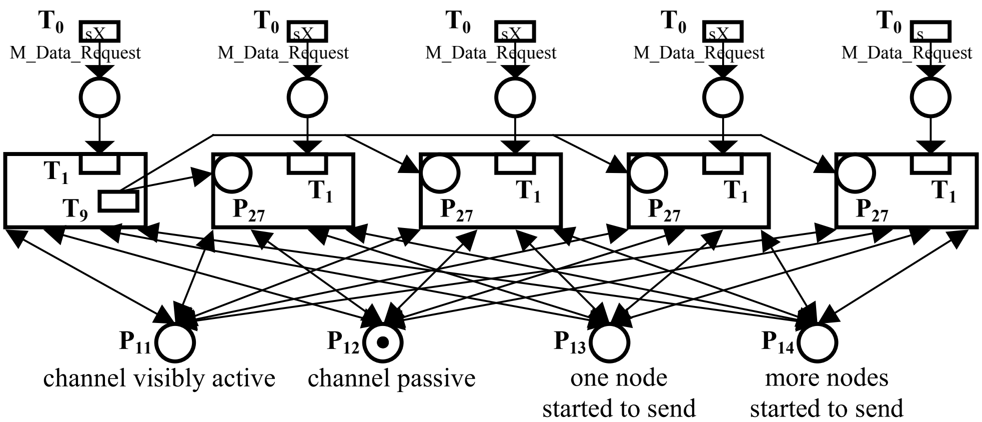

Figure 12 – Interconnection of five nodes, (propagation delay, backlog estimation)

Simulated network consisting of five nodes is shown in Figure 12. Broadcast communication of a request frame is realised by arcs going form T_9 in each particular node to P_{27} in all other nodes (for simplicity reasons Figure 12 shows just broadcasting from the first node to the other nodes).
The characteristics in Figure 14 show that collision ratio is kept constant even in case of a big traffic load and the network throughput in Figure 13 is very close to the one without the propagation delay (see Figure 5). This is achieved due to the approximation of the channel backlog (higher traffic ⇒ higher backlog ⇒ longer average random delay ⇒ less collisions) without charging low traffic by long random delay.

PHYSICAL MEANING OF THE TRANSITIONS IN FIGURE 11:

T_1 data from Link Layer entering MAC,
T_2 no traffic on the channel
T_3 , T_5 traffic on the channel appeared,
T_4 delay β_1
T_6 entry point for waiting process
T_7 entry point for media access
T_8 success
T_9 request frame
T_{10} response frame
T_{11} end of packet cycle
T_{12} request denied
T_{13} first node starting to send data
T_{14} second node starting to send data
T_{15} more than second node starting to send data
T_{16} first node starting to send data detects collision
T_{17} collision
T_{18} T_{19} T_{20} T_{21} BL increment - reception of request frame
T_{23} T_{24} T_{25} T_{26} BL decrement - approximation of end of packet cycle
T_{27} to T_{34} BL increment – ready to send request frame

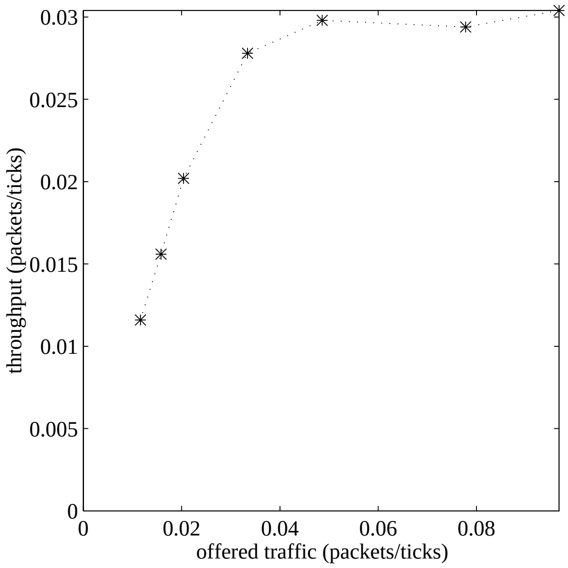

Figure 13- Network throughput (propagation delay, backlog)

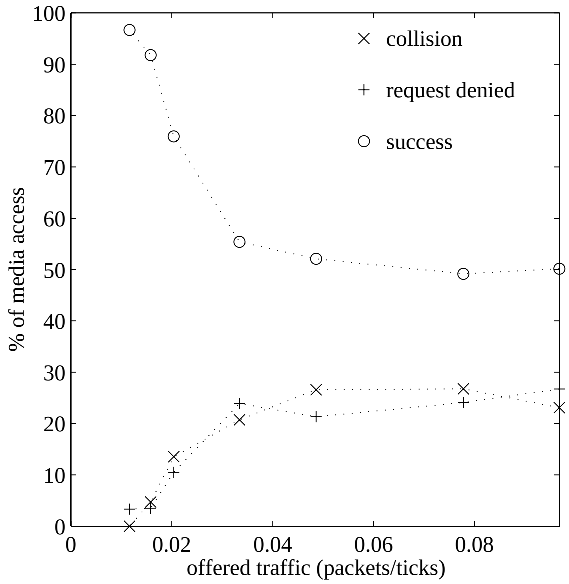

Figure 14 – Ratio of MAC results (propagation delay, backlog)

5. Conclusion

Acceptable throughput characteristic of LonWorks is based on the fact that LonTalk protocol dynamically adjusts the number of randomising time slots by predicting channel backlog. By actively managing the collision rate, the LonTalk protocol provides superior layers with acceptable communication performance even for low data rates, heavy loads and large networks.

The Petri Net model of the LonTalk MAC sublayer presented in this article enables to observe various phenomena of *Predictive p-persistent CSMA*. The model shows major tendencies even if backlog estimator by five values is just an approximation of the real backlog estimator used by LonTalk protocol and even if each request frame was supposed to be acknowledged by just one response frame.
Simulated results (token player, network interconnection and analysis implemented in Matlab) are not in contradiction with the ones given by formulas approximating CSMA throughput in [3]. Major tendencies of real results achieved on test bed of 36 nodes [9] comply very well to our simulation results. Advantage of the modelling approach given in this text in comparison to the real tests is that one can observe all needed

events/variables/characteristics and carry our the experiments in order to clarify various tendencies separated from other influences. Furthermore the modelling and simulation enables to treat more complex cases (e.g. backlog estimation, various probability distribution functions) than fully analytical approach based e.g. on the queuing theory.

References

[1] Blum, I., Juanole, G.: Formal Specification and Analysis of a Control System Based on Computer Networks, WFCS'97 – IEEE workshop on Factory Communication Systems, October 1-3, 1997, Grafismar Barcelona

[2] Hilmer, H., Kochs, H.D., Dittmar, E.: A Fault Tolerant Communication Architecture for Real-time Control Systems, WFCS'97 – IEEE workshop on Factory Communication Systems, October 1-3, 1997, Grafismar Barcelona

[3] Kleinrock, L. and Tobagi, F. A., "Packet Switching in radio Channels: Part I, II ", IEEE Trans. Commun. Corn-23, Dec. 1975, pp. 1400-1433.

[4] ECHELON® CORPORATION, Document No.19550, ref 078-0125-01A, „LonTalk® Protocol Specification" –Version 3.0. Palo Alto, 1994.

[5] Tadao Murata: "Petri Nets: Properties, Analysis and Applications" Proceedings of the IEEE, vol. 77, No. 4, April 1989.

[6] L. Gallon, G. Juanole, I. Blum: "Modelling ad Analysis of the ARINC specification 629 CP MAC layer protocol" Proceedings of the IEEE, wfcs' 97, October 1997.

[7] Tovar E., Vasques F., Pre-run-time schedulability analysis of P-Net fieldbus networks, IECON '98, Aachen, September 1998.

[8] M.K.Molloy, "Performance analysis using stochastic Petri nets", IEEE Computer Society, Transactions on Computers, vol. C-31, no. 9. pp. 913-917. 1982.

[9] Enhanced Media Access Control with LonTalk® Protocol, LonWorks® engineering bulletin, January 1995, Part Number 005-0001-01 Rev.C

[10] Marsan E., Balbo G., Conte G., Donatelli S., Franceshinis, "Modelling with Generalized Stochastic Petri Nets", John Wiley, 1995.

[11] Molle M. "A New Binary Logarithmic Arbitration Method for Ethernet" Computer Systems Research Institute, University of Toronto, Toronto Canada M5S 1A1. Technical Report CSRI-298, April 1994.

[12] Goodman J. Greenberg A, Madras N, March P, "Stability of Exponential Backoff" Journal of the ACM, Vol. 35, No. 3, July 1988, pp/ 579-602.

[13] Fouad A. Tobagi and V. Bruce Hunt. Performance Analysis of Carrier Sense Multiple Access with Collision Detection. Computer Networks 4(5):245-259, October/November, 1980.

[14] K. Jensen, How to find Invariants for Coloured Petri Nets, Computer Sc. Dep. Aarhus University Aarhus, Danemark, DIAMI PB 120, May 1980.

[15] G. Almes and E. Lazowska "The Behavious of Ethernet-Like Computer Communication Networks" Proc. Of the 7the Symp. On Operating Systems Principles, pages 66-81. ACM SIGCOMM, Asilomar, California, December, 1979.

[16] T. Gonsalves and F. Tobagi, "On the Performance Effects of Station Locations and Access Protocol Parameters in Ethernet Networks" IEEE TransComm. COM-36(4).

[17] H. Takaki and L. Kleinrock "Throughput Analysis for Petristent CSMA Systems" IEEE TransComm. COM-33(7).

[18] D. Boggs, J. Mogul, Ch. Kent, "Measured Capacity of an Ethernet: Myths and reality" WRL Research Report 88/4, Hamilton Av. Palo Alto, Cal. 9430, USA.