MODELING OF LONTALK ALGORITHM IN CONTROL NETWORKS

Percy Javier Igei Kaneshiro
University of São Paulo, Escola Politécnica, São Paulo, SP, BRAZIL
percy.kaneshiro@poli.usp.br

José Isidro Garcia Melo
University of São Paulo, Escola Politécnica, São Paulo, SP, BRAZIL
University of Valle, Faculty of Engineering, Cali, CA, COLOMBIA
joisigar@poli.usp.br

Paulo E. Miyagi
University of São Paulo, Escola Politécnica, São Paulo, SP, BRAZIL
pemiyagi@usp.br

Carlos E. Cugnasca
University of São Paulo, Escola Politécnica, São Paulo, SP, BRAZIL
carlos.cugnasca@poli.usp.br

Abstract. *The industrial organization of productive processes have presented a tendency for dispersion and distribution of manufacturing plants based on an increase of resources and functionality of information and mobility technology. In this sense, Local Operating Networks (LON, LonWorks) is the name of one of the leading technologies in sensor/control networking, addressed to a wide range of applications with topology free technology. In spite of its popularity, only a few design tools are available to simulate LonWorks architectures and predict their performance. In this paper, a model of the collision resolution algorithm of LonWorks is described. The approach developed in this work is based on the characterization of LonWorks networks as Discrete Event Dynamic Systems (DEDS), since dynamic behavior is defined through the discrete events and discrete states. The proposed procedure employs techniques that are derived from interpreted Petri net, which has been used as an efficient tool for modeling, analysis and control of DEDS. In this context, the media access control sublayer (MAC) is modeled in different levels of abstraction: a conceptual model which is obtained using the PFS (Production Flow Schema) technique, and a functional model by using MFG (Mark Flow Graph). The MFG abstraction level describes details in a functional form preserving the description activities of upper levels. This procedure allows the structured development of models, facilitating the modeling process of the algorithm specification. The result presented in this paper for a single network segment can be integrated into a global networks modeling, since the proposed procedure decomposes the model systematically according to a hierarchal approach into different modules. In this way, it is straightforward to integrate single segments networks models into complex networks.*

Keywords*: LonWorks, Petri net, discrete event dynamic system, distributed production, fieldbus.*

1. Introduction

In the last few years, markets are becoming global and independent of geographic barriers. Many manufacturing industries have been established in a distributed and dispersed way, exploring the potential of mobility and information technology (IT), which provides the means for a larger transnational cooperation (Junqueira, et al.,2006). Considering this, systems need to be supported by formal concepts and modeling techniques, because the high complexity of component-based systems often compromises its consistency. The high complexity is caused mainly by the non-deterministic and concurrent behavior and interaction of components. This complexity also leads to strong dependencies between a component and its environment. In this context, this work aims at a modeling procedure of a communication protocol that guarantees its predictable behavior and permits distributed industrial implementation. This paper is an introductory work related to modeling a distributed application in industrial environments with topology free technology contrary to the central architecture common used in industrial applications. These centralized architectures, such as PC-Host and PLC control, are typically specified with deterministic industrial protocol works, such as campus area network (CAN) and PROFIBUS (Cavalieri, 1996; Marschall, 1996).

Nowadays, building automation and industrial systems integrates computing, communication, and control into different levels of operations and information processes. These distributed control systems are called control network, and have sensors, actuators, and controllers interconnected by communication networks. Local Operating Networks (LonWorks), designed by Echelon Corporation, to be used in building and industrial automation applied to distributed control, sensor, and actuator networks. Characterized by decentralized organization and asynchronous data transfer, LonWorks can support large-scale fieldbus networks using multiple media with peer-to-peer or master-slave architectures. Due to architectural flexibility, LonWorks smart devices are intended not only to acquire data from the environment in a sensor network, but also to interact with the object being sensed in a feedback loop as a control network.

LonWorks has become a classic solution in building automation, and home networking including all key building automation subsystems: heating, ventilating and air conditioning, lighting, security, fire detection, access control, energy monitoring (Miskowicz, Golanski, 2006). Among others, LonWorks platforms are also used in semiconductor manufacturing, pulp and paper equipment, material handling, textile machinery, petrochemical, food and beverage, automotive manufacturing and wastewater treatment.

The ANSI/EIA 709.1 control network standard, also known as Lontalk protocol is the communication protocol of LonWorks and provides services to all seven layers of the ISO/OSI reference model (ECHELON, 1995).

This protocol uses non-deterministic media access method (MAC) called predictive p-persistent CSMA. Despite the non-deterministic MAC method, Lontalk is used in soft real time applications with moderate traffic. An advantage of this protocol is that it keeps the collision ratio independent of the channel utilization and it uses technique for partial predication of the channel backlog (Bauer, Rossler, 2002).

The predictive p-persistent CSMA algorithm is a collision detection technique that randomizes channel access using knowledge of the expected channel load. In this way, collision rate is decreased effectively under heavy network load while sufficient throughput and bandwidth utilization under light load can still be guaranteed. Previous studies confirm the advantage of this algorithm (ECHELON, 1995).

Some works approach the modeling of predictive p-persistent CSMA. Miskowics (2003) studied how to optimize the predictive p-persistent CSMA in terms of the soft real-time requirements, based on a channel-centric simulation model. Wang et. al. (2005) studied the protocol with a node-centric simulation model and compared it with a real network and a Marcov chain-based analytical model.

The purpose of this paper is to propose a novel approach to model this protocol that guarantees its predictable behavior and permits distributed industrial implementation. The approach developed in this work is based on the characterization of LonWorks protocol as Discrete Event Dynamic Systems (DEDS), since the dynamic behavior is defined through the discrete events and discrete states.

On the other hand, Petri net is a convenient tool to model parallel algorithms and communication protocols. (Hanzalek, Svadova, 2001). Petri net is a graphic and mathematical tool for modeling, analyzing, and designing DEDS. Among its main advantages, there is easy graphic interpretation, identifying states and actions in a clear way, and the possibility of representing the system dynamics and structure in many levels of detail (Murata 1989; Peterson 1981). Petri net can also model synchronism, asynchronism, concurrence, causality, conflict, and resource sharing.

The proposed procedure uses techniques that are derived of interpreted Petri net, which has been proven to be as an efficient tool for modeling, analysis and control of DEDS (Reizig 1985; Murata 1989; Peterson 1981). In this context, the media access control sublayer (MAC) is modeled in different levels of abstraction: a conceptual model, which is obtained using the PFS (Production Flow Schema) technique level, and a functional model by using MFG (Mark Flow Graph). The MFG abstraction level describes details in a functional form, in which the description activities of previous levels are preserved.

This paper is organized in the following way: Section 2 The predictive p-persistent CSMA protocol is briefly explained. In Section 3, the technique derived from Petri net is presented. In Section 4, a methodology for modeling the predictive p-persistent CSMA protocol is introduced. Finally, section 5 presents the main conclusions.

2. Predictive p-persistent CSMA protocol

All network protocols use a MAC algorithm to allow devices to determine when they can safely send a packet of data. MAC algorithms are designed to either eliminate or minimize collisions. A collision occurs when two or more devices attempt to send data at the same time. Lontalk protocol uses the predictive p-persistent CSMA protocol that has excellent performance characteristics even during periods of network overload (ECHELON, 1995). This protocol uses a collision detection technique that randomizes the channel access using knowledge of the expected channel load.

Like CSMA, predictive p-persistent CSMA senses the medium before transmitting. The most important difference with CSMA algorithms is its ability to estimate channel traffic load and adjust the probability of access to channel. It is a variant of p-persistent CSMA with the difference that the parameter p, the probability with which a packet is transmitted when the channel is idle, varies according to the traffic condition while it is constant in p-persistent CSMA.

When the traffic is high, p is small; when the traffic decreases, p will increase automatically. In this way, collision rate is decreased effectively under heavy network load while sufficient throughput and bandwidth utilization under light load can still be guaranteed (Xiaoming, Geok-Soon, 2002, Miskowicz, 2003). The behavior of this algorithm can be defined by a set of discrete states that are changed due to the occurrence of instantaneous discrete events. Examples of events are: channel detection (channel can be idle or busy), packet transmission (success or collision), etc. This type of system can be classified as a Discrete Event Dynamic System (DESD) (Reizig 1985; Murata 1989; Peterson 1981). There are various methods for modeling and analyzing DEDS, such as Petri net, Markov chain, Queue Theory, Mini-max Algebra and State machines but among them, Petri net has proved to be effective and very useful because of it graphic and formal description, among other reasons (Reizig 1985; Murata 1989; Peterson 1981). According to Hanzalek and Svadova (2001), Petri net is a convenient tool to model parallel algorithms and communication protocols.

3. The Petri net approach

Petri nets offer a mathematically defined technique for the specification, design, analysis, verification and performance evaluation of concurrent distributed systems. They offer not only precise semantics and a theoretical foundation, but also a graphical form that facilitates the understanding of both information and control flow within the same formalism. As an intuitively appealing graphical form of presentation, Petri net can also model synchronism, asynchronism, concurrency, causality, conflict, and resource sharing. Among the Petri net interpretations for DEDS, the PFS/MFG technique (Miyagi, 1996) is adopted. In the PFS/MFG, an interpreted Petri net-based model of the system is built by using a top-down approach. Firstly, a conceptual model is obtained using the PFS (Production Flow Schema) (Miyagi; Arata, 1997). Then, the PFS model is refined into a functional model using MFG (Mark Flow Graph) (Hasegawa et al., 1987).

The choice of PFS/MFG among the many Petri nets classes around was guided by their ability to build compact, structured, and declarative system models, without compromising the mathematical power of Petri net theory.

The aim of PFS is to identify the activities performed on a flow of discrete items (information or material) at a high level of abstraction. The PFS model has no dynamic.

On the other hand, MFG explicitly shows the behavior of the *activity* and the interaction with external components. The MFG models the system dynamics by modifying the number and distribution of marks in the graph. The MFG consists of the following six elements: ***box***, ***transition***, ***directed arc***, ***mark***, ***gate arc***, and ***output signal arc***.

(1) A ***box*** represents a condition for an action (an event).
(2) A ***transition*** represents an event of the system.
(3) A ***directed arc*** connects a ***box*** and a ***transition***, and its direction shows the input/output relation between them.
(4) A ***mark*** is placed in a ***box*** to indicate that the condition corresponded to the ***box*** is holding.
(5) A ***gate arc*** connects a ***transition*** with a signal source and, depending on the signal, it permits or inhibits the occurrence of the event that corresponds to the connected ***transition***. ***Gate arcs*** are classified into enabling or inhibitor, and internal or external.
(6) An ***output signal arc*** transmits a flag from a ***box*** to an external element of the graph. The value of signal is "1" when there is a *mark* in the ***box***, otherwise it is "0".

The detailed definition of PFS, MFG and its extensions can be found in (Miyagi, 1996; Kaneshiro, et al., 2005).

4. The predictive p-persistent CSMA *algorithm* procedure

The procedure proposed for predictive p-persistent CSMA algorithm modeling is composed of 3 steps, illustrated in Fig. 1 and detailed as follows:

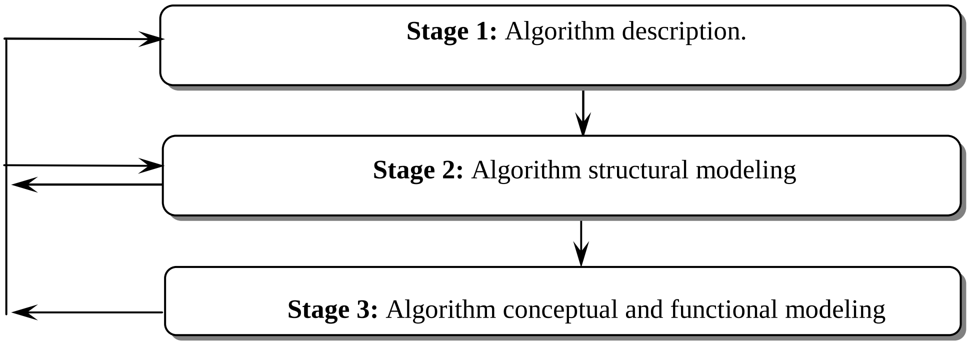

Figure 1.Prcodedure for predictive p-persistent CSMA algorithm modeling

Stage 1: Algorithm description

This is the initial stage in algorithm modeling. Here, a research is carried out with the purpose of reviewing the predictive p-persistent CSMA algorithm and its traffic load prediction mechanism according to the ANSI/EIA 709.1.

This protocol operates in the following way: a ready node, that has a packet to send, monitors the channel before transmission. When it detects no transmission during beta1 period, it asserts the channel is idle. Then it generates the random delay time (RDT). If the channel is still idle when the RDT expires, the node transmits. Otherwise, the node receives the incoming packet and competes for the channel access again. Beta1 is the time constant given by physical layer parameters and respect propagation delay defined by the media length, detection and turn-round delay within MAC sublayer. Beta2 is the basic number of randomizing slots, and is equal to a constant value of 16.

Fig.2 shows the concepts and parameters of this protocol. A ready node monitors the status of the channel and determines the channel to be idle if it senses no transmission during beta1 period. Nodes without a ready packet during this period will remain in synchronization for the duration of the following random delay time and thus, if the packet is ready after the end of beta1 period, it can be transmitted in a valid slot according to the ready nodes. Then, the ready node generates a random delay time from 0 to (16*BL-1) slot time. If the channel is idle when this delay expires, the node transmits; otherwise, the node repeats the MAC algorithm (Wang, et al., 2005).

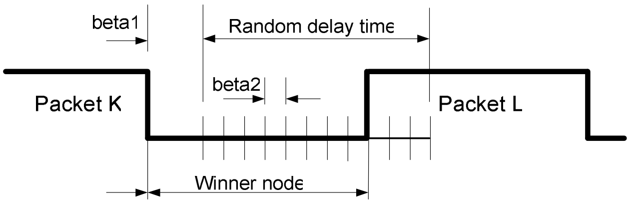

Figure 2. Predictive p-persistent CSMA concepts and parameters

BL is an integer maintained by each node. It is initially set to its minimum value, which is equal to 1, and its maximum value can be 63.

Backlog BL is increased:

- By the backlog increment after sending or receiving a packet with non-zero backlog increment;
- By one whenever a collision is detected by a transmitting node.

Backlog BL is decreased by one:

- If a packet with a backlog increment of “0” is transmitted or received;

If a packet cycle time expires without channel activity

Stage 2: Algorithm structural modeling

This stage consists in developing the predictive p-persistent CSMA structural model. At this stage, the interfaces among MAC sublayer, link layer and physical layer are represented. In Fig. 3, a frame reception is handled entirely by Link Layer, which notifies MAC sublayer about each correct 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 to one of the three admissible results after the transmission (Hanzalek, Capek, 2001):

- **success:** when the packet transmission is finished;
- **request_denied:** when an activity was detected on the channel before the start of the transmission;
- **collision:** when one or more nodes sense the channel idle but in fact another node is beginning to transmit.

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

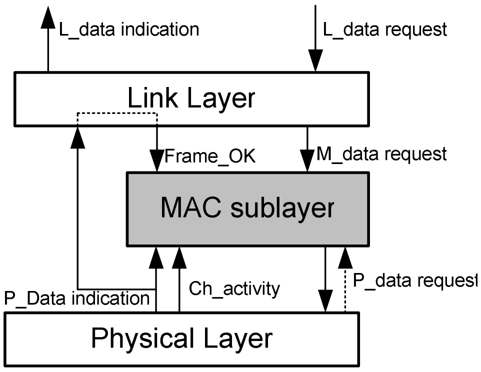

Figure 3. MAC sublayer interfaces (Hanzalek, Capek, 2001)

Fig. 4.a shows a typical network architecture consisting of n nodes (MAC node 1 to MAC node n and corresponding M_data request messages) sharing the physical layer. In this model, the network architecture is implemented by the interactions between different nodes processes and channel activity (physical layer). There are mainly two interactions between nodes and the channel.

One interaction happens when a node releases the channel. At this time, the physical layer should inform to the sensor activity channel of other nodes, that the channel is idle. The other interaction occurs when a node catches the channel. This node checks whether or not a collision appears and then, the channel informs other nodes. Fig 4.b shows the tree components (sensor activity channel, backlog estimation and node substructure) that compose a MAC sublayer node model. The interaction between components and the channel is described in the next stage.

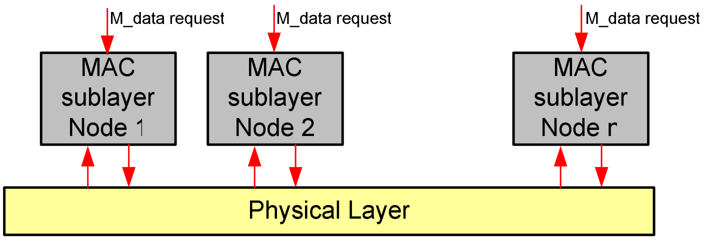

Figure 4.a Network architecture

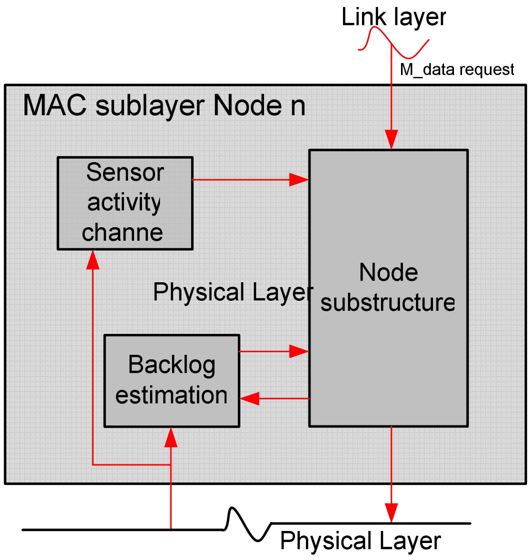

Figure 4.b MAC sublayer node structure

Stage 3: Algorithm conceptual and functional modeling

At this stage, the modeling of the predictive p-persistent CSMA algorithm is performed systematically according to a hierarchal approach. Initially, the algorithm conceptual modeling is developed by gradually refining the models until its functional model is obtained. The conceptual model is particularly important to deal with interpretations of the information obtained in the previous stages. This model describes the MAC algorithm and allows a deep understanding of different parts. Afterwards, the modeling of the interrelation and functions of its components results in the appropriated functional specification. At this stage, the PFS/MFG methodology is adopted. (Miyagi, et al. 2002). The functional model contains the description of the algorithm dynamic behavior.

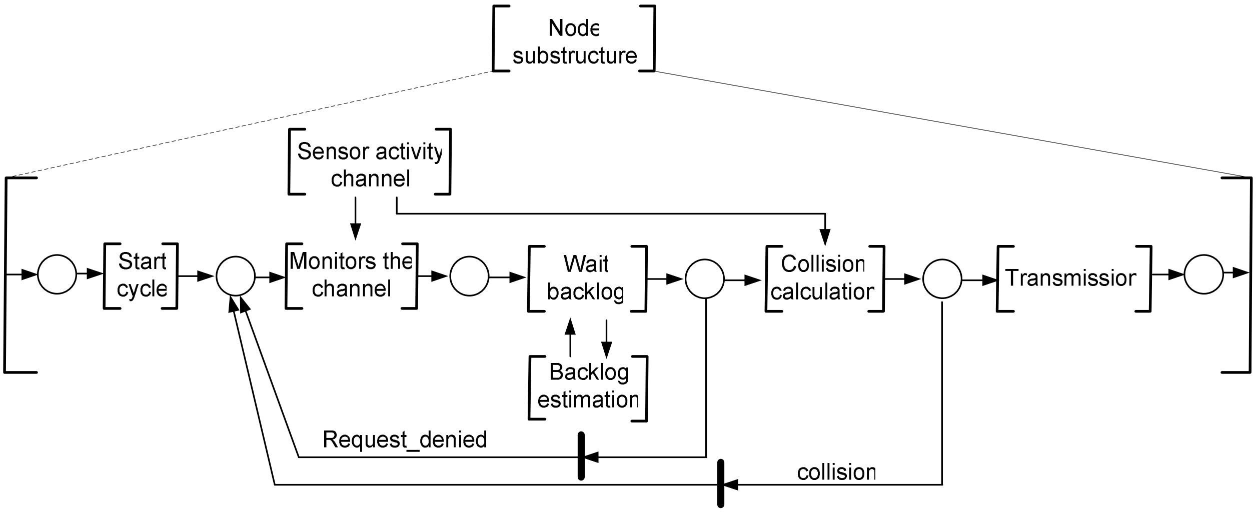

Figure 5. PFS model of MAC sublayer node substructure

Fig. 5 presents the PFS model of one node MAC sublayer using predictive p-persistent CSMA algorithm. In this model, the activity "start cycle" can be initiated when the node finished sending a previous packet and there is a new packet to be sent (node receives M_data_request from Link Layer). The activity "monitors the channel" receives information about the channel activity from the "sensor activity channel" (see Fig. 6). If the channel is busy, the node continues monitoring, but when it detects no transmission during beta1 period, the next activity “wait backlog” is performed. In this activity, the node delays a number of time slots of beta2 duration.

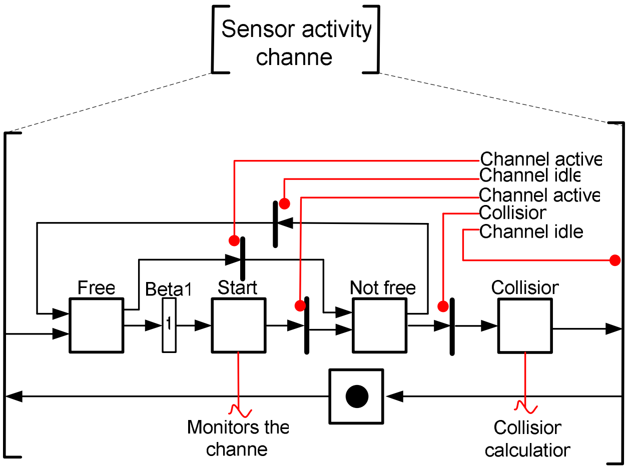

Figure 6. MFG model of the activity “Sensor activity channel”

Fig. 6 describes the dynamic behavior of “sensor activity channel” developed in MFG. In this *activity*, the channel stage is constantly monitored, where the *box* “free” represents the channel is idle, the *box* “not free” represents the channel is active and the *box* “collision” represents a collision of two o more nodes in the channel. If the token resides in *box* “free” continuously during beta1 period, transition beta1 fires, and the token moves into *box* “start”. Otherwise, if traffic appears on the channel when the token resides in *box* “free” and beta1 period is not finished, then the *mark* moves from *box* “free” to *box* “not free”. Marked *box* “start” means that the channel is in the idle estate and sends a signal to perform the *activity* “monitors the channel”. If two or more physical nodes perform media access at the same time, then both of them can detect the free state of the channel and consequently can start to send their data. This results in collision, represented by a *mark* in *box* “collision”. Finally, if there is no *activity* on the channel, the token returns to the *box* “free”.

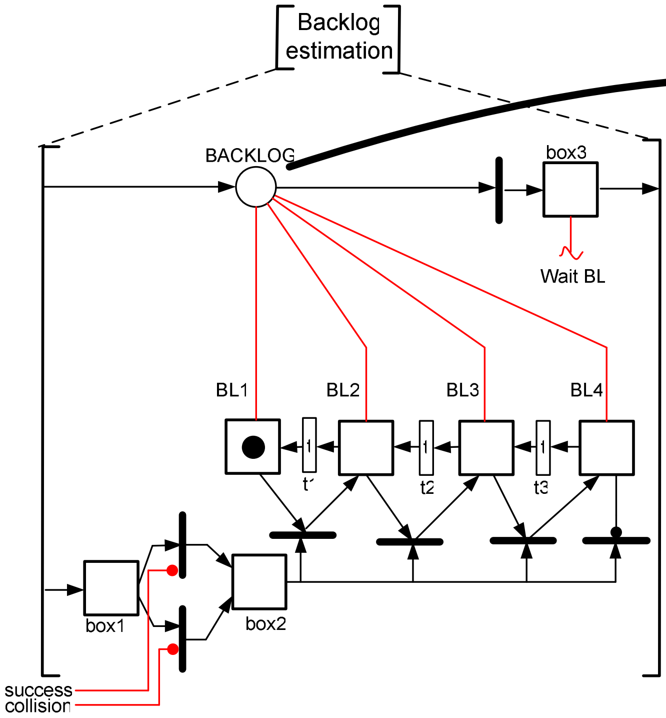

Figure 7.a MFG of the *activity* “Backlog Estimation”

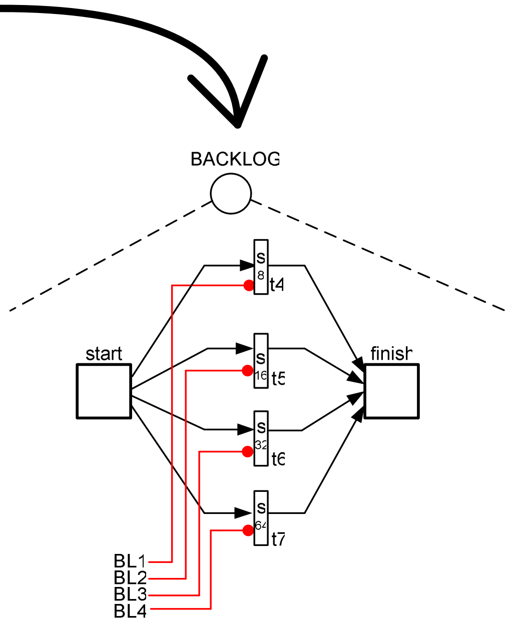

Figure 7.b MFG of the *inter-activity* “Backlog”

The activity “backlog estimation” (see Fig. 7) calculates a random delay according to the channel activity. If the channel is still idle when the random delay expires, the *activity* “collision calculation” is performed. Otherwise, the node receives the request_denied packet and competes for the channel access again. When the *activity* "collision calculation" is finished, the node will perform the *activity* "transmission" if collision is not detected. Otherwise, the node receives the collision packet and competes for the channel access. A collision may occur in a vulnerable period, in which one node senses the channel idle but in fact another node is beginning to transmit. When the transmission ends, the node returns to the sleep state.

The mean value of the predicated channel load is the channel backlog (BL). Due to the limited space available, in Fig. 7.a only 4-level backlog estimator is modeled. *Boxes* BL1, BL2, BL3 and BL4 represent the state of the backlog estimator and correspond to the current value of channel backlog. Timed transitions t1, t2, t3 e t4 are timed transitions (labeled by "t") with deterministic firing time corresponding to the packet cycle. Via t4, t5, t6 or t7, predictive p-persistent CSMA generates a stochastic time delay (labeled "s") from a random timing interval, which is dependent on the estimated channel backlog. According to Fig 7.b, just one of the transition t4, t5, t6 or t7 is fired at a given time.

Once the models are built, simulation can be performed to provide information about the algorithm behaviour, allowing, for example, the detection of deadlock occurrence, possible non-desired states, etc. Then, the models can be analyzed in order to verify if they satisfy the specifications of the algorithm in such a way that its dynamic behavior is equivalent to the expected behavior. The verification can be conducted by simulation technique based on the Petri net properties (Peterson 1981; Murata 1989).

5. Conclusions

This paper is an introductory work that aims for the modeling of a distributed application in industrial environments. This application is topology free technology contrary to central architecture commonly used in industrial applications. The procedure presented, for the modeling of a collision resolution algorithm of LonWorks, permits a predictable behavior at communications level which makes a proper industrial application implementation, well suited for control applications. This procedure allows the structured development of models, facilitating the modeling process of the algorithm specification. A node-centric approach to model this protocol was developed using interpreted Petri net. In the development of the model, the PFS/MFG technique was proposed to facilitate the design. The result shows that this method has the following features:

- It is a systematic approach for the construction of models that can be used for verification and validation of the algorithm.
- It describes the algorithm from conceptual to detailed level according to the hierarchical structure of the systems activities, considering aspects such as modularity and flexibility.

Nowadays, LonWork networks tend to be more complex and usually consist of several segments connected via routers. Thus, the modeling of a single networks segment is only a small part of the overall modeling. However, the procedure proposed in this paper for a single network segment can be integrated into a global networks modeling, since the procedure proposed decomposes the model systematically according to a hierarchal approach into different modules. It is therefore straightforward to integrate single segments networks models into complex networks.

6. Acknowledgements

This work received financial support from the Brazilian agencies CAPES, CNPq and FAPESP. The authors would particularly like to thank Kyatera/TIDIA Program under which the work is developed.

7. References:

Bauer, A., Rossler, P., 2002, "A Hard Real-Time Arbitration Protocol based on EIA-709", Proceeding of the 14th Euromicro Conference on Real-Time Systems, Issue 2002, pp. 127-133.

Xiaoming C., Geok-Soon H., 2002, "A Simulation Study of the Predictive p-Persistent CSMA Protocol", 35th Annual Simulation Symposium, pp. 0345.

ECHELON, 1995, "Determinism in industrial computer control network applications", Echelon White Paper, January.

Hanzalek, Z., Capek, J., 2003, "Channel Backlog Estimation in LonWorks". In: Mahalik, N.P.(Ed.): Fieldbus Technology - Industrial Network Standards for Real-Time Distributed Control ISBN 3-540-40183-0, Springer.

Hanzalek, Z., Svadova, M, 2001, "Matlab Toolbox for PetriNets", 22nd International Conference, ICATPN 2001 Newcastle upon Tyne, UK.

Hasegawa, K.; et al., 1987, "Simulation of discrete production systems based on Mark Flow Graph", Systems Science, Poland, vol. 13, n.1-2, pp.22.

Hasegawa, K.; et al., 1999, "On resource arcs for Petri net modeling of complex shared resource systems", Journal of Intelligent and Robotics Systems, Kluwer Ac. Publ., Amsterdan, vol.26, n.3, pp.423-437.

Kaneshiro, P. I.; Miyagi, P. E.; Villani, E., 2005,"Modeling of Fire Protection Systems in Intelligent Buildings", Proceedings of COBEM XVIII International Congress of Mechanical Engineering, Ouro Preto, ABCM.

Miyagi, P.E., 1996, "Controle Programável - Fundamentos do Controle de Sistemas a Eventos Discretos", Editora Edigard Blücher, São Paulo, SP.

Miyagi, P.E., Arata, W. M., 1997, "Analysis of manufacturing systems – application on PFS model based analysis of manufacturing system for performance assessment", Revista Brasileira de Ciências Mecânicas, ABCM, Campinas, SP, vol. 19, n.1, pp. 58-71.

Junqueira F, Miyagi P.E. A new method for the hierarchical modeling of productive systems. In: 7th IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services, Ontario, Canada , 2006.

Miyagi, P.E.; et al., 2002, "Petri net approach for modelling system integration in intelligent buildings". Journal of The Brazilian Society of Mechanical Sciences, ABCM, Rio de Janeiro, RJ, vol. 23, n.4, pp.341-350.

Miskowics, M., 2003, "Analysis of the LonTalk/EIA-709.1 channel performance under soft real-time". Proceedings of the IEEE ICIT 2003 International Conference on Industrial Technology, Maribor, Slovenia. pp. 705–708.

Miskowics, M., Golanski, R., 2006, "LON Technology in Wireless Sensor Networking Applications". Journal Sensors, vol.6, pp.30-48.

Murata, T., 1989, "Petri nets properties, analysis and applications", Proceedings of the IEEE, vol.77, n.4, pp.541-580.

Peterson, J.L., 1981, "Petri Net Theory and The Modeling of Systems", Prentice Hall, Englewood Cliffs.

Reisig, W., 1985, "Petri Nets an Introduction", Spring Verlag, New York.

Wang, et al., 2005, "Collision Resolution Simulation for Distributed Control Architectures using LonWorks". Proceedings of the 2005 IEEE International Conference on Automation Science and Engineering, Canada. pp.319-326.

Zhu, J.; Denton, R., 1988, "Timed Petri Nets and their Application to Communication Protocol Specification". Proceedings of the 21st IEEE Military Communications Conference: MILCOM 88, pages 195-199, San Diego, USA, pp.195-199.

Cavalieri, S.; et al.,1996, "CAN assessment in time-critical cyclic applications through Petrinet model". Industrial Electronics, Control, and Instrumentation. Proceedings of the IEEE IECON 22nd International Conference vol. 2, n.5, pp.922 – 927.

Marschall, G., 1996, "Petri net simulation of a Fieldbus communication application" Proceedings of the 1996 IEEE Emerging Technologies and Factory Automation., 1996 IEEE Conference on vol. 1, n.18, pp. 63 – 69.

8. Responsibility notice

The authors are the only responsible for the printed material included in this paper.