ISSN 1392-1215 ELEKTRONIKA IR ELEKTROTECHNIKA. 2005. Nr. 4(60)

T 180 TELEKOMUNIKACIJŲ INŽINERIJA

Formalization and Simulation Telecommunication Protocols and Systems using PLA Method

H. Pranevicius, I. Praneviciene, R. Benkunskis
Business Informatics Department, Applied Mathematic Department, Kaunas University of Technology,
Studentu str. 50, LT-5136 Kaunas, Lithuania, phone: +370 37 300376, fax: +370 37 451654, e-mail: hepran@if.ktu.lt

Introduction

The stage of formal specification is one of the most important during the design of software for computer network protocols. Such formal specification is normally used for analysis and implementation purposes. In the stage of analysis it is necessary to resolve two tasks: analysis of logical correctness and evolution of system functioning timing parameters.

Different mathematical schemes are used for creating formal descriptions of systems, such as: different automate models, Petri–nets, data flow and state transition diagrams, temporal logic technique, abstract communicating methods and others.

When a formalization method is chosen, it is desirable that both above mentioned analysis tasks could be resolved on the bases of a single formal description. The aggregate approach has such property and it has been successfully used both for correctness analysis and for simulation of computer network protocols (Pranevicius, et al., 2002). The specification language ESTELLE/Ag and the specifications analysis tool PRANAS-2 have been created on the base of the aggregate method (Ag) (Pranevicius, et al., 1994). There are some differences between ESTELLE/Ag and the ESTELLE standard ISO: the piece-linear aggregate model is used in ESTELLE/Ag. The use of such a model instead of a finite-state automate, which is the formal background of the standard ESTELLE, enables to create models both for validation and simulation. This is possible due to the special structure of the piece-linear aggregate. Apart from the discrete components describing the state of the modules, there are also continuous components to control event-sequences in the module. These continuous components are called operations. By means of operators, sequences of actions are described, the intermediate results of which are invisible on the outside. If such an operation sequence is being performed at a given instance of time the corresponding operation is called “active”. Thus, an individual module involves two types of events: the arrival of an input signal and the completion of an active operation. The specification analysis system PRANAS-2 consists of the following software tools: a specification editor, a validation subsystem and a simulation subsystem. The editor provides the capability to create a specification in ESTELLE/Ag. The validation subsystem permits to construct a validation model for the program generating the reachability graph. After completing the construction of the reachability graph, it is possible to verify the following specification characteristics: completeness, deadlock freeness, boundedness, absence of static deadlock, absence of dynamic deadlock, termination.

The PLA formalization principles and the methods of analyzing the correctness of specifications based on this method are presented in this paper. The PLA specification of the CSMD/CD protocol and the obtained simulation results are presented.

General Principles of the Aggregate Approach

In the application of the aggregate approach for system specification, the system is represented as a set of interacting piece–linear aggregates (PLA). The PLA is taken as an object defined by a set of states Z, input signals X, and output signals Y. The aggregate functioning is considered in a set of time moments $t \in T$. The state $z \in Z$, the input signals $x \in X$, and the output signals $y \in Y$ are considered to be time functions. Apart from these sets, transition H and output G operators must be known as well.

The state $z \in Z$ of the piece-linear aggregate is the same as the state of a piece-linear Markov process, i.e. $z(t)=(\upsilon(t),\ z_{\upsilon}(t))$, where $\upsilon(t)$ is a discrete state component taking values on a countable set of values; and $z_{\upsilon}(t)$ is a continuous component comprising of $z_{\upsilon 1}(t), z_{\upsilon 2}(t), \ldots, z_{\upsilon k}(t)$ co-ordinates.

When there are no inputs, the state of the aggregate changes in the following manner:

$$\upsilon(t)=\text{const}\,,\ \frac{dz_{\upsilon}(t)}{dt}=-\alpha_{\upsilon}\,, \tag{1}$$

where $\alpha_{\upsilon}=(\alpha_{\upsilon 1}, \alpha_{\upsilon 2}, \ldots, \alpha_{\upsilon k})$ is a constant vector.

The state of the aggregate can change in two cases only: when an input signal arrives at the aggregate or when a continuous component acquires a definite value. The theoretical basis of piece-linear aggregates is their representation as piece-linear Markov processes.

Reachable States Approach for Aggregate Model Validation

An essence of the reachable states method is a use of the global state which is considered as a joint state of a system after aggregate system composition (Pranevicius, 1991). A graph of the reachable states is created as oriented one: its nodes stand for global states of the system, its arcs indicate the possible transitions from one state to another. Initial and final states must be specified in working out the graph. The resulting states graph is used for the analysis of defined properties of a system, as some of them are closely related with the graph structure. The given validation method allows to investigate general properties of a system such as boundedness, absence of redundancy in specification, completeness, absence of static deadlocks, absence of dynamic deadlocks, termination.

Invariant Approach for Aggregate Model Validation

A system invariant (**I**) is the assertion, which describes the correct system functioning and it must remain true in spite of the events taking place and system transition from one state to another.

The essence of the method is as follows: assertions are formulated in relation to the co-ordinates of the aggregate model so as to express the requirements for the system functioning.

On the base of a conceptual model of an analyzed system we can describe the system functioning by the event sequence, which may be represented by the graph **G**(V), where V is a set of vertices and $\mathbf{A} = \{a_{ij}\}$ is an adjacency matrix. In this case $V = \{e_1, e_2, \ldots e_n\}$, where e_i is i-th event, n is a number of events. $(e_i e_j) \neq (e_j e_i)$, i.e. the graph is oriented.

The set of states, which the system may enter after the event e_i, is called as the i-th set of possible states (SS_i – symbolic state).

$$SS_i = \{z \in Z | (\exists z')((z' \in Z) \wedge EP_i(z') \wedge (z = H_i(z', P)))\}, \quad (2)$$

where Z is a set of all possible system states, $EP_i(z')$ is an enabling predicate of the event e_i in the state z', P is a set of probabilistic parameters of the system and H_i is a transition operator determining a new system state when the event e_i occurs.

The system considered being in the symbolic state SS_i only if it is in the state z and $z \in SS_i$. Relying this SS_i definition, every event e_i is related to the symbolic state SS_i, therefore replacing the set of vertices V in the graph **G**(V) by $V' = \{SS_1, SS_2, \ldots, SS_n\}$ while the adjacency matrix **A** remains unchanged. We obtain the graph of symbolic states **G**(V') which describes the system operation by determining the possible set of states and transitions from one symbolic state to another.

Conceptual Model of CSMD/CD Protocol

The local network consists of a joint connection bus. Each node that has a station receives information frame flows, which intensity are $\lambda_1, \lambda_2, \ldots, \lambda_N$. The frame that comes into the node enters into the station transition channel. If this channel is occupied the frame is being included into the queue. The time of transmitting the frame successfully is being calculated since the station began attempting to occupy the bus for transmitting till the frame was successfully transmitted. The CSMA/CD protocol network model scheme is presented in Figure 1.

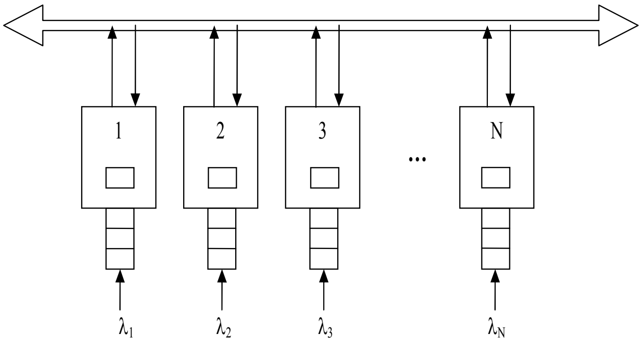

Fig. 1. The system model scheme

The CSMA/CD protocol (IEEE, 1997) allows the station itself to check if the connection bus is occupied by another station, however it does not allow interrupting the transmission process. In order to avoid collisions every sender has to control the signal in the bus and to ascertain that no other station is transmitting at that time. If the signal CS (Carrier Sense) is identified, the station delays the transmission of its frame till the transmission is over and only then it attempts to transmit the frame again.

For the collision to be identified all the stations during the transmission trace the signals in the bus; if the signals under transmission and under tracing differ then the collision is fixed. The state of the collision is expressed in a special bit sequence called the *Jam* sequence. When the transmitting station determines the collision it has to stop the transmission immediately.

After the collision is determined the station has to wait till the bus is free and then to start the transmission again. However a collision may occur anew. In order to avoid multiplex collisions the Ethernet demands every station to wait an appropriate slot of time before a new transmission after the collision. The pause time after the n-th collision is equal to $t_p = Slot\,time * T$, here T is a random integer, which is distributed equally in the range [0, 2^n]. The range value increases to 10 trials (max 16) and after the 10-th it becomes [0, 1024] and unchanging. Technically such a duplication of the waiting time is known as Binary Exponential Backoff. In essence such an increase of the waiting time according to the exponential law means that the Ethernet can recover fast after the collisions as the stations wait longer after each unsuccessful attempt to transmit. In the worst case when two or more stations choose the same waiting time the exponential increase ensures the competition for the bus to be terminated after few collisions.

After detecting a collision each station that was transmitting the frame and encountered a collision attempts to transmit its frame again after an appropriate delay. The station may attempt to transmit the frame for 16 times and the transmission is rejected after the 16-th unsuccessful attempt.

Aggregate Model of the CSMD/CD Protocol

The CSMA/CD protocol network model aggregate scheme consists of one aggregate. Consequently the sets of input and output signals will be empty and all the events in the system will be internal. Seven events can occur in the system, the transition operators define the state the system can enter after each event and the conditions under which it can happen.

1. The set of input signals:

   $X = \varnothing$.

2. The set of output signals:

   $Y = \varnothing$.

3. The set of external events:

   $E' = \varnothing$.

4. The set of internal events:

$$E'' = \{e''_{11} \ldots e''_{1N}, e''_{21} \ldots e''_{2N}, e''_{31} \ldots e''_{3N},$$
$$e''_{41} \ldots e''_{4N}, e''_{51} \ldots e''_{5N}, e''_{61} \ldots e''_{6N}, e''_{71} \ldots e''_{7N}\},$$

here e''_{1i} – a formed frame in the *i*-th station;

e''_{2i} – the *Timer*1 in the *i*-th station has terminated;

e''_{3i} – the *Timer*2 in the *i*-th station has terminated;

e''_{4i} – the *Timer*3 in the *i*- th station has terminated;

e''_{5i} – the *Jam* sequence in the *i*- th station has terminated;

e''_{6i} – the frame transition in the *i*- th station has terminated;

e''_{7i} – the frame of the *i*-th station has transited through the whole bus.

5. Controlling sequences:

$$\{e''_{1i}\} \rightarrow \{\xi_{1i}^{j}\}_{\mathrm{j=1}}^{\infty},$$

here ξ_{1i}^{j} – the time interval, after which the *j*-th frame in the *i*-th station will be formed; it will be distributed according to the exponential law with the parameter λ_i .

6. The discret state:

$$v(t) = \{Timer1_1(t) \ldots Timer1_N(t),$$
$$Timer2_1(t) \ldots Timer2_N(t),$$
$$Timer3_1(t) \ldots Timer3_N(t),\ \ Q_1(t) \ldots Q_N(t),$$
$$Perd_1(t) \ldots Perd_N(t),$$
$$n_1(t) \ldots n_N(t), CS(t), K(t),\ \ Jam_1(t) \ldots Jam_N(t),$$
$$Kolizija(t), \mathrm{Mag\_pab(t)}\},$$

here $Timer1_i(t)$: 0,1 – the *i*-th station collision check timer state;

$Timer2_i(t)$: 0,1 –the possibility waiting to transmit timer state in the *i*-th station;

$Timer3_i(t)$: 0,1 – the *i*-th station pause timer state;

$Q1_i(t)$ – the frame number in the queue of the *i*-th station;

$Perd_i(t)$: 0,1 – the *i*-th station transmission channel state;

$n_i(t)$: 0..16 – the number of the *i*-th station's attempts to transmit the frame*;*

$CS(t)$: 0,1 – the bus state;

$K(t)$: 0, N – the number of the states that transmit;

$Jam_i(t)$: 0,1 – the transmission feature of the *Jam* sequence in the *i*-th station ;

$Kolizija(t)$: 0,1 – the collision;

$Mag\_pab(t)$: 0,1 – the frame that has transited through the whole bus.

7. The continuous state:

$$z_v(t) = \{w(e''_{11}, t) \ldots w(e''_{1N}, t),\ \ w(e''_{21}t) \ldots w(e''_{2N}, t),$$
$$w(e''_{31}t) \ldots w(e''_{3N}, t),\ \ w(e''_{41}t) \ldots w(e''_{4N}, t),$$
$$w(e''_{51}t) \ldots w(e''_{5N}, t),\ \ w(e''_{61}t) \ldots w(e''_{6N}, t),$$
$$w(e''_{71}t) \ldots w(e''_{7N}, t),$$

here $w(e''_{1i}t)$ – the moment when a frame in the *i*-th station will be formed;

$w(e''_{2i}t)$ –*Timer*1 end moment of the *i*-th station;

$w(e''_{3i}t)$ –*Timer*2 end moment of the *i*-th station;

$w(e''_{4i}t)$ – *Timer*3 end moment of the *i*-th station;

$w(e''_{5i}t)$ – *Jam* sequence end moment of the *i*-th station;

$w(e''_{6i}t)$ – the frame transmission end moment of the *i*-th station;

$w(e''_{7i}t)$ – the moment of the *i*-th station's frame transition through the whole bus.

8. The state of the system:

$$z(t) = \{Timer1_1(t) \ldots Timer1_N(t),$$
$$Timer2_1(t) \ldots Timer2_N(t),$$
$$Timer3_1(t) \ldots Timer3_N(t),\ \ Q_1(t) \ldots Q_N(t),$$
$$Perd_1(t) \ldots Perd_N(t),$$
$$n_1(t) \ldots n_N(t), CS(t), K(t),\ \ Jam_1(t) \ldots Jam_N(t),$$
$$Kolizija(t), \mathrm{Mag\_pab(t)}\},\ \ w(e''_{11}t) \ldots w(e''_{1N}, t),$$
$$w(e''_{21}t) \ldots w(e''_{2N}, t),$$
$$w(e''_{31}t) \ldots w(e''_{3N}, t),\ \ w(e''_{41}t) \ldots w(e''_{4N}, t),$$
$$w(e''_{51}t) \ldots w(e''_{5N}, t),$$
$$w(e''_{61}t) \ldots w(e''_{6N}, t),\ \ w(e''_{71}t) \ldots w(e''_{7N}, t)\}.$$

9. The initial state:

$$Timer1_1(t) \ldots Timer1_N(t) = 0;$$
$$Timer2_1(t) \ldots Timer2_N(t) = 0;$$
$$Timer3_1(t) \ldots Timer3_N(t) = 0;\ \ Q_1(t) \ldots Q_N(t) = 0$$
$$Perd_1(t) \ldots Perd_N(t) = 0;\ \ n_1(t) \ldots n_N(t) = 0;$$
$$CS(t) = 0,$$
$$K(t) = 0,\ \ Jam_1(t) \ldots Jam_N(t) = 0;\ \ Kolizija(t) = 0;$$

$$Mag\_pab(t) = 0;\ w(e''_{11}t)\ldots w(e''_{1N},t) = t + \xi^{j}_{1i};$$
$$w(e''_{21}t)\ldots w(e''_{2N},t) = \infty;\ \ w(e''_{31}t)\ldots w(e''_{3N},t) = \infty;$$
$$w(e''_{41}t)\ldots w(e''_{4N},t) = \infty;\ \ w(e''_{51}t)\ldots w(e''_{5N},t) = \infty;$$
$$w(e''_{61}t)\ldots w(e''_{6N},t) = \infty;\ \ w(e''_{71}t)\ldots w(e''_{7N},t) = \infty\ .$$

10. Transition operators:

$H(e''_{1i})$: [a formed frame in the *i*-th station];

$$Timer1_i(t+0) = \begin{cases} 1, \text{ if } Timer3_i(\mathrm{t}) = 0 \wedge Perd_i(\mathrm{t}) = 0 \wedge \\ (CS(t) = 0 \vee (CS(t) = 1 \wedge \\ Mag\_pab(t) = 0 \wedge Kolizija(t) = \\ = 0 \wedge \xi \le p)); \\ 0, \text{ otherwise}; \end{cases}$$

[the *i*-th station‘s *Timer1* is released if the pause timer of this station is terminated, the transmission channel is empty and the station detects a free bus that can be completely unoccupied or the frame transmitted by another station has not yet reached this station; here ζ is a random value equally distributed in the range [0,1]];

$$Timer2_i(t+0) = \begin{cases} 1, \text{ if } Timer3_i(\mathrm{t}) = 0 \wedge Perd_i(\mathrm{t}) = 0 \wedge \\ (CS(t) = 1 \wedge Mag\_pab(t) = 1 \vee \\ Kolizija(t) = 1 \wedge \xi > p); \\ 0, \text{ otherwise}; \end{cases}$$

[the *i*-th station‘s *Timer*2 is released in the pause timer of this station is terminated, the transmission channel is empty and the station detects a free bus];

$$CS(t+0) = \begin{cases} 1, CS(t) = 1 \vee (Timer3_i(t) = 0 \wedge \\ Perd_i(t) = 0 \wedge CS(t) = 0); \\ 0, \text{ otherwise}; \end{cases}$$

[the bus is being occupied if the pause timer is terminated, is released if the pause timer of this station is terminated, the transmission channel is empty and the station detects a free bus or the bus has already been occupied];

$$Q_i(t+0) = \begin{cases} Q_i(t) + 1, \text{if } Perd_i(t) = 1; \\ Q_i(t), \quad \text{otherwise}; \end{cases}$$

[the frame is placed into the queue if the transmission channel is occupied];

$$K(t+0) = \begin{cases} K(t)+1, \text{ if } Timer3_i(\mathrm{t}) = 0 \wedge Perd_i(\mathrm{t}) = 0 \wedge \\ (CS(t) = 0 \vee (CS(t) = \\ = 1 \wedge Mag\_pab(t) = \\ = 0 \wedge Kolizija(t) = 0 \wedge \xi \le p)); \\ K(t), \quad \text{otherwise}; \end{cases}$$

[the number of the stations that transmit is being increased if the pause timer is terminated, the transmission channel is empty and the station detects a free bus]

$$Perdi(t+0) = 1\,;$$

[the transmission channel of the *i*-th station becomes occupied in any case];

$$w(e''_{1i}, t+0) = t + \xi^{j}_{1i}\,;$$

[the time moment when another frame in the *i*-th station will be formed is set];

$$w(e''_{2i}, t+0) = \begin{cases} t + slot\_time, \text{ if } Timer3_i(\mathrm{t}) = 0 \wedge \\ Perd_i(\mathrm{t}) = 0 \wedge \\ (CS(t) = 0 \vee (CS(t) = \\ = 1 \wedge Mag\_pab(t) = 0 \wedge \\ Kolizija(t) = 0 \wedge \xi \le p)); \\ \infty, \quad \text{otherwise}; \end{cases}$$

[the *i*-th station’s *Timer*1 end moment is set if the pause timer of this station is terminated, the transition channel is empty and the station detects a free bus];

$$w(e''_{3i}, t+0) = \begin{cases} t + slot\_time, \text{ if } Timer3_i(\mathrm{t}) = 0 \wedge Perd_i(\mathrm{t}) = 0 \wedge \\ (CS(t) = 1 \wedge Mag\_pab(t) = 1 \vee \\ Kolizija(t) = 1 \vee \xi \le p); \\ \infty, \quad \text{otherwise}; \end{cases}$$

[the *i*-th station’s *Timer*2 end moment is set if the pause timer of this station is terminated, the transition channel is empty and the station detects a free bus];

$$w(e''_{6i}, t+0) = \begin{cases} t + t_{kad}, \text{ if } Timer3_i(\mathrm{t}) = 0 \wedge Perd_i(\mathrm{t}) = \\ = 0 \wedge (CS(t) = 0 \vee \\ (CS(t) = 1 \wedge Mag\_pab(t) = 0 \wedge \\ Kolizija(t) = 0 \wedge \xi \le p)); \\ w(e''_{6i}, \mathrm{t}), \text{ otherwise}; \end{cases}$$

[the *i*-th station’s frame transition end moment is set if the pause timer of this station is terminated, the transition channel is empty and it detects a free bus];

$$w(e''_{7i}, t+0) = \begin{cases} t + t_{kad}, \text{ if } Timer3_i(\mathrm{t}) = 0 \wedge Perd_i(\mathrm{t}) = 0 \wedge \\ (CS(t) = 0 \vee (CS(t) = \\ = 1 \wedge Mag\_pab(t) = 0 \vee \\ Kolizija(t) = 0 \wedge \xi \le p)); \\ \infty, \quad \text{otherwise}; \end{cases}$$

[the whole bus transition time moment of the *i*-th station’s transmitted frame is set if the pause timer of this station is terminated, the transition channel is empty and the stationdetects a free bus];

$H(e''_{2i})$ [the *i*-th station’s *Timer*1 end];

$$Timer1_i(t+0) = 0\,;$$

[the *i*-th station’s *Timer*1 is disconnected];

$$Jam_i(t+0) = \begin{cases} 1, \text{ if } K(t) > 1 \vee Kolizija(t) = 1; \\ 0, \text{ otherwise}; \end{cases}$$

[the *Jam* sequence is released in the *i*-th station if it detects the other stations transmitting or the *Jam* sequence of the other station];

$$n_i(t+0) = \begin{cases} n_i(t) + 1, \text{ if } K(t) > 1 \vee Kolizija(t) = 1; \\ 0, \quad \text{otherwise}; \end{cases}$$

[the number of the *i*-th station’s attempts to transmit the frame is increased if it detects the other stations transmitting or the *Jam* sequence of the other station];

$$CS(t+0)=1;$$

[the bus remains occupied in any case];

$$Perd_i(t+0)=Perd_i(t);$$

[the frame remains in the *i*-th station's transition channel in any case];

$$Kolizija_i(t+0)=\begin{cases}1, & \text{if } K(t)>1\wedge Kolizija(t)=0;\\ 0, & \text{otherwise;}\end{cases}$$

[the collision is fixed in the bus if the station detects the other stations transmitting first];

$$w(e''_{2i},t+0)=\infty;$$

[the *i*-th station's *Timer*1 is stopped];

$$w(e''_{5i},t+0)=\begin{cases}t+2\tau, & \text{if } K(t)>1\vee Kolizija(t)=1;\\ \infty, & \text{otherwise;}\end{cases}$$

[the *i*-th station's *Jam* sequence end moment is set if it detects the other stations transmitting or the *Jam* sequence of the other station];

$$w(e''_{6i},t+0)=\begin{cases}w(e''_{6i}, & \text{if } K(t)=1\wedge Kolizija(t)=0;\\ \infty, & \text{otherwise;}\end{cases}$$

[the *i*-th station's frame transmission end moment remains unchanged if the station is the only one transmitting at the moment and no *Jam* sequence of the other station is detected].

Simulation Results of the CSMA/CD Protocol

The network model under research consists of 6 stations. These stations perform the transmission of ordinary computer data. In order to perform this, frames of a fixed size, 2400 bits with heading are formed. The frame flow is formed according to the Puasonic law. The dependence of the features of this network model on the network load is calculated. The network load is calculated according to the ratio between the required throughput and the possible maximum throughput of the network. The network load is changed by increasing the intensity of the frames under formation in the stations. The station's buffers lengths are restricted to the capacity of 100 frames. The transmission speed in the network is 10 Mb/s. The network parameters are selected according to the IEEE 802.3 standard. The modeling time is 10 s.

Such features are evaluated statistically:

- the average time of data frame transition;
- the average time of data frame presence in the system;
- the average time of data frame waiting in the queue;
- the average length of data frame queue;
- system loss.

Parameters of the protocol.

The duration of the frame = 2400 bit, λ = 0,2 – 0,81 frame/ms, the size of buffer = 100 frames, the transmission speed = 10 Mb/s.

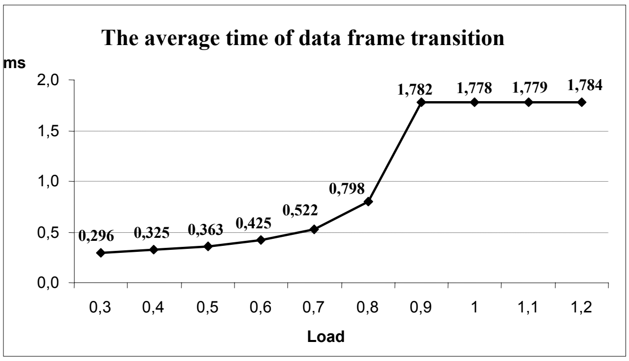

Fig. 2. The dependence of the average timeof data frame transition on load

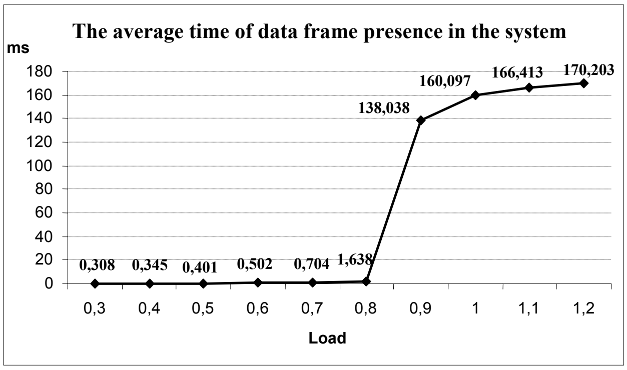

Fig. 3. The dependence of the average time of data frame presence in the system on load

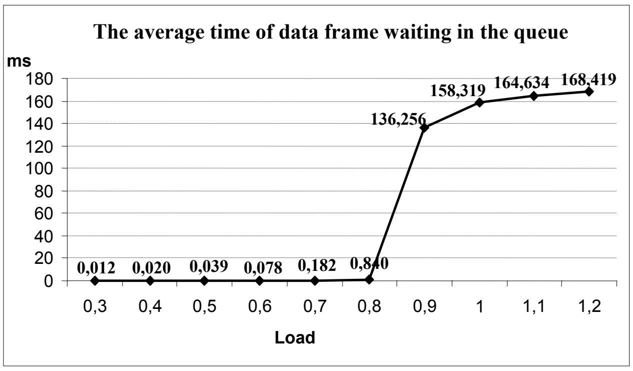

Fig. 4. The dependence of the average time of frame waitin in the queue on load

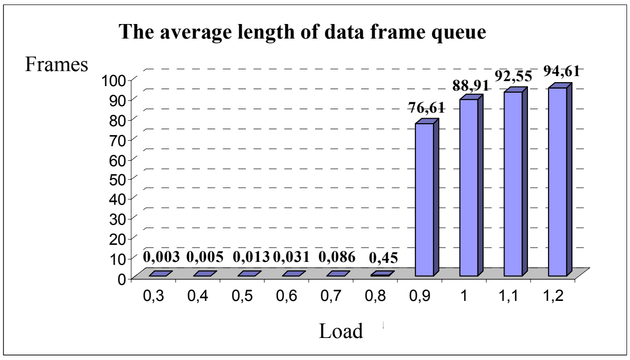

Fig. 5. The dependence of the average length of data frame queue on load

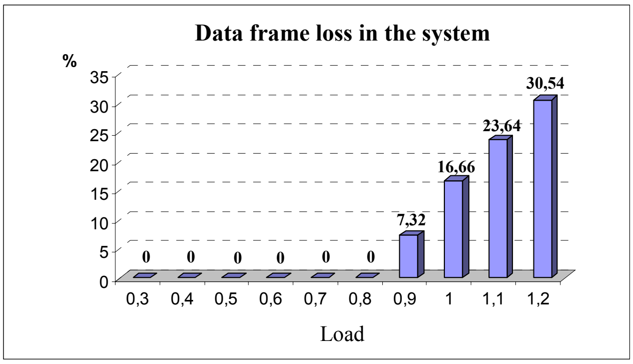

Fig. 6. The dependence of loss in the system on load

Conclusions

We see from the presented figures that when the network load varied to 80% the protocol features increased fractionally. Only when the network load is 90% the loss in the system occur. The simulation results show that this protocol can be used for transmitting the language signals only when the network load is less than 80%. When the load is larger a considerable delay of frames occurs.

References

1. **Pranevičius H., Knight J. A. G. Pranevičienė I**. Formal Specification and Analysis of Distributed Systems Used Aggregate Approach // Information Technology and Control. No.22(1). ISSN 1392-124 X. – Kaunas: Technologija, 2002. – P.31–38.
2. **Pranevicius H., Pilkauskas V., Chmieliauskas A.** Aggregate approach for specification and analysis of computer network protocols. – Kaunas: Technologija, 1994. – 152 p.
3. **Pranevicius H**. Aggregate approach for specification, validation, simulation and implementation of computer network protocols // Lectures notes in Computer Science, 502. – Springer–Verlag, 1991. – P. 433–477.
4. **Institute of Electronical and Electronics Engineers (IEEE).** „Enchaced Media Control Algorithm for IEEE 802.3 CSMA/CD“. Document #80232/D1.8. – New York, 1997 January.

Pateikta spaudai 2005 03 15

H. Pranevičius, I. Pranevičienė, R. Benkunskis. Telekomunikacinių protokolų formalizavimas ir imitacinis modeliavimas PLA metodu // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2005. – Nr. 4(60). – P. 5–10.

Pateikiamas agregatinis formalizavimo metodas, aptariamos galimybės taikyti jį telekomunikaciniams protokolams specifikuotii ir analizuoti. Pateikiamo metodo teorinę bazę sudaro atkarpomis tiesinių agregatų panaudojimas formaliam sistemų specifikavimui. Šio metodo pranašumas tas, kad, panaudojant bendrą formalų aprašymą, galima atlikti sudarytų specifikacijų teisingumo analizę ir imitacinį modeliavimą. Sudaryto formalaus aprašymo teisingumo analizei taikomas pasiekiamų būsenų metodas, kuris leidžia tikrinti tokias sistemos savybes: statines ir dinamines aklavietes, pabaigiamumą, invariantus. Pateikiama protokolo CSMD/SD agregatinė specifikacija panaudota protokolo imitaciniam modeliui sudaryti. Pateikiami imitacinio modeliavimo rezultatai, leidžiantys įvertinti leistiną protokolo apkrovą. Il. 6, bibl. 4 (anglų kalba; santraukos lietuvių, anglų ir rusų k.).

H. Pranevicius, I. Pranevičiene, R. Benkunskis. Formalization and Simulation Telecommunication Protocols and Systems using PLA Method // Electronics and Electrical Engineering. – Kaunas: Technologija, 2005. – No. 4(60). – P. 5–10.

The aggregate approach and its possibilities for specification and analysis of computer network protocols are presented. The theoretical basis of the aggregate approach is a piece-linear aggregate (PLA) for a formal specification of systems. The advantage of this approach is that it permits to create models both for analysis correctness of specifications and simulation. Some methods that can be used for validation and verification of aggregate specifications are presented also. The presented approach is illustrated by formal specification of CSMD/SD protocol and simulation results. Ill. 6, bibl. 4 (in English; summaries in Lithuanian, English, Russian).

Г. Пранявичюс, И. Пранявичене, Р. Бенкунскис. Формализация и имитационное моделирование телекоммуникационных протоколов и систем PLA методом // Электроника и электротехника. – Каунас: Технология, № 4(60). – С. 5–10.

В системе представлен агрегативный метод и его возможности для формальной спецификации и анализа телекоммуникационных систем. Теоретической основой предлагаемого метода является применение кусочно-линейных агрегатов для формализованного описания систем. Достоинством предлагаемого метода является то, что он позволяет используя единую формальную спецификацию выполнять анализ корректности созданной спецификации и создание имитационных моделей. Анализ созданных формальных описаний исследуется методом достижимых состояний, который позволяет использовать такие свойства систем: статические и динамические тупики, завершаемость, инварианты. Представляется агрегативное описание CSMD/SD протокола, которое использовано при создании имитационной модели протокола. Представляемые результаты имитационного моделирования позволяет оценить допустимую нагрузку протокола. Ил. 6, библ. 4 (на английском языке; рефераты на литовском, английском и русском яз.).