CREATION SIMULATION MODELS OF COMPLEX SYSTEMS USING PIECE-LINEAR AGGREGATE FORMALISM

Henrikas Pranevicius

Business Informatics Department, Faculty of Informatics,
Kaunas University of Technology
Studentu 50, LT-51368, Kaunas, LITHUANIA
Tel.:+370-37-300376, fax: +370-37-451654, e-mail: hepran@if.ktu.lt

Abstract

A method for development of simulation models of complex system using piece-linear aggregate (PLA) formalism is presented. PLA formalism depends to class of timed automata. States of PLA are described by two components: discrete and continuous. Coordinates of continuous component define time instructs when occur internal events. External events are related with arrival input signals. Simulated system is presented how a set connected of piece-linear aggregates. PLA's System is created during the main phase of a model development, which later is used for creation of simulation model programmes. Possibilities of PLA model application for simulation of hybrid systems is discussed also. Presented method for the development of simulation models is illustrated by example of simulation model of interval-marker protocol..

Keywords: *Complex systems, simulation, discrete event system specification, piece-linear aggregates, verification, computer network protocols*.

1. INTRODUCTION

Piece-linear aggregates (PLA) formalization approach for creation simulation models was proposed in [1]. Later, Pranevicius [2] developed a method formal description of PLA based on the notion of *controlling sequences*. In [3] has been shown how PLA can be used for formal specification, validation and simulation of computer network protocols. For this task, the language Estelle/Ag (close to ISO standard Estelle) was created using PLA model. The main difference between the two languages is that the basic structural unit of Estelle – module – in Estelle/Ag is defined as a piece-linear aggregate. System PRANAS (PRotocol ANAlysis System) was developed for protocol validation and simulation using formal description of protocols by aggregates.

The method of controlling sequences defines, in fact, operational semantics for the PLA model. This semantics defines how to interpret specifications given in form of aggregates, and is sufficient for simulation and validation tasks [4]. Some correctness analysis using reachability graph of states is also possible, but only in the case of finite number of states. Presented systems simulation approach is comparable with Discrete Event System Specification (DEVS) formalism [5] introduced by B. Zeigler which provides a means of specifying a mathematical object called a system.

2. THE USE OF CONTROLLING SEQUENCES FOR FORMAL DESCRIPTION OF PIECE-LINEAR AGGREGATES

In the application of the aggregate approach for system specification, the system is represented as a set of interacting piece – linear aggregates. 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:

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

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 Markoff processes.

Aggregate functioning is examined on a set of time moments $T = \{t_0, t_1, ..., t_m, ...\}$ at which one or several events take place, resulting in the aggregate state alternation. The set of events *E* which may take place in the aggregate is divided into two non-intersecting subsets $E' = E' \cup E''$. The subset $E' = \{e'_1, e'_2, ..., e'_N\}$ comprises classes of events (or simply events) e'_i, $i = \overline{1, N}$ resulting from the arrival of input signals from the set $X = \{x_1, x_2, ..., x_N\}$. The class of events $e''_i = \{e''_{ij}, j = 1, 2, 3, \ldots\}$, where e''_{ij} is an event from the class of events e''_i taking place the *j*-th time since the moment *t*0. The events from the subset *E′* are called external events. A set of aggregate input signals is unambiguously reflected in the subset *E′* i.e., $X \rightarrow E'$. The events from the subset $E'' = \{e''_1, e''_2, ..., e''_f\}$ are called internal events where $e''_i = \{e''_{ij}, j = 1, 2, 3, \ldots\}$, $i = \overline{1, f}$ are the classes of the aggregate internal events. Here, *f* determines the number of operations taking place in the aggregate. The events in the set *E″* indicate the end of the operations taking place in the aggregate.

For every class of events e''_i from the subset *E″*, control sequences are specified $\{\xi_j^{(i)}\}$, where $\xi_j^{(i)}$ – the duration of the operation, which is, followed by the event e''_{ij} as well as event counters $\{r(e''_i, t_m)\}$, where $r(e''_1, t_m)$, $i = \overline{1, f}$ is the number of events from the class e''_i taken place in the time interval $[t_0, t_m]$.

In order to determine start and end moments of operation, taking place in the aggregate the so-called control sums $\{s(e''_i, t_m)\}$, $\{w(e''_i, t_m)\}$, $i = \overline{1, f}$ are introduced, where $s(e''_i, t_m)$ – the time moment of the start of operation followed by an event from the class e''_i. This time moment is indeterminate if the operation was not started; $w(e''_i, t_m)$ is the time moment of the end of the operation followed by the event from the class e''_i. In case of no priority operations, the control sum $w(e''_i, t_m)$ is determined in the following way:

$$w(e''_i, t_m) = \begin{cases} s'(e''_i, t_m) + \xi_{r(e''_i, t_m)+1}, & \text{if at the moment } t_m \text{ an operation is taking place,} \\ & \text{which is followed by the event } e_i; \\ \infty, & \text{in the opposite case.} \end{cases}$$

The infinity symbol (∞) is used to denote the undefined values of the variables.

The control sum definition presented above is used in simulation. When aggregate models are used for system formalisation and correctness analysis, the control sum may be determined in a simplified way [Pran83]:

$$w(e''_i, t_m) = \begin{cases} < \infty, & \text{if at the moment } t_m \text{ an operation is taking place, followed by the event } e''; \\ \infty, & \text{in the opposite case.} \end{cases}$$

Control sums determine only the possibility conditions for the events after the moment t_m, while the event occurrence moments are not determined.

Let us specify the meaning of the co-ordinates of the aggregate state. The discrete component of the state, $v(t_m) = \{v_1(t_m), v_2(t_m), \ldots, v_p(t_m)\}$, presents the system state, i.e., counters of transmitted and received information packets, readiness for information transmission etc.

$$z_v(t_m) = \{w(e''_1, t_m), w(e''_2, t_m), \ldots, w(e''_f, t_m)\}$$

are control co-ordinates specifying the moment of evolutionary events occurrence.

The control co-ordinate $w(e''_i, t_m)$ corresponds to every each e''_i from the subset of events *E″*, while always $w(e''_i, t_m) \geq t_m$.

The state co-ordinates $z(t_m)$ can change their values only at discrete time moments t_m, $m = 1, 2, \ldots$ of event occurrence, remaining fixed in each interval $[t_m, t_{m+1})$, $m = 0, 1, 2, \ldots$ where t_0 – the initial moment of system functioning.

When the state of the system $z(t_m)$, m = 0, 1, 2, …, is known, the moment t_{m+1} of the following event is determined by a moment of input signal arrival to the aggregate or by the equation:

$$t_{m+1} = \min\{w(e_i'', t_m)\},\ 1 \le i \le f .$$

Class of the next event e_{m+1} is specified by an input signal if it arrives at the time moment t_{m+1} or is determined by the control co-ordinate, which acquire minimal value at the moment t_m, i.e., if $w(e_i'', t_m)$ acquires minimal value, then $e_{m+1} = e_i''$.

The operator *H* states the new aggregate state.

$$z(t_{m+1}) = H[z(t_m), e_i],\ e_i \in E' \cup E'' .$$

The output signals y_i from the set of output signals $Y = \{y_1, y_2, \dots, y_m\}$ can be generated by an aggregate only at occurrence moments of events from the subsets *E*′ and *E*″. The operator *G* determines the content of the output signals:

$$y = G[z(t_m), e_i],\ e_i \in E' \cup E'',\ y \in Y .$$

Further transition and output operators will be denoted $H(e_i)$ and $G(e_i)$.

Let us consider an aggregate system, comprising *K* aggregates, including the environmental one. Let the *k*-th aggregate contains N_K input and M_K output IP. The number of communication channels, transmitting information between the aggregates, let be denoted *L*.

Let us assume the matrix $R = \|r_{ji}\|$ $i = \overline{1, L}$, $j = \overline{1, 2}$ of input IP, the meanings of its elements are the following: r_{1i} is the number of the aggregate receiving input signals from the *i*-th communication channel: $r_{1i} = 1, 2, \dots, K$; r_{2i} is the number of the input IP of the aggregate r_{1i}, incidental to the *i*-th channel, $1 \le r_{2i} \le N_r$.

The inputs of the aggregates are specified by the matrix $H = \|h_{ij}\|$, $i = \overline{1, K}$, $j = \overline{1, \max\{M_K\}}$, $1 \le k \le K$, where h_{ij} is the number of the channel incidental to the *j*-th output IP of the *i*-th aggregate $1 \le h_{ij} \le L$.

The matrices *R* and *H* unambiguously determine the addresses of all output signals of all aggregates in a system. Only one signal can be an output of one aggregate and an input of the another. In view of the fact that an aggregate system is a closed one, i.e., no outside signals arrive and no signals are generated, an occurrence of input signals in the aggregates result from an occurrence of internal events in one of the aggregates of the aggregate system.

Set of events, which are possible in the aggregate system, is a union of subsets of internal events taking place in each aggregate, i.e.:

$$E = \bigcup_{k=1}^{K} E_k'',$$

where E_k'' is a set of events taking place in the *k*-th aggregate.

A set of time moments at which the events from the set *E* take place is denoted as:

$$T = \bigcup_{k=1}^{K} T_k'',$$

where T_k'' is a set of time moments at which internal events of the *k*-th aggregate take place.

The next time moment t_{m+1}, at which the events from the set *E* take place, is determined in the following way:

$$t_{m+1} = \min_k \min_r \left(w_k(e_r'', t_m) \cup w_k(e_r''', t_m)\right),\ 1 \le k \le K,\ e_r'' \in E_k'',\ e_r''' \in E_k'''$$

3. THE SPECIAL CASES OF AGGREGATE MODEL

It will be shown that it is possible to get various cases of other well-known models from the aggregate model.

System of differential equations:

$$v(t_m)=\varnothing\,,\ z_v(t_m)\neq\varnothing\,,\ E'=\varnothing\,,\ E''=\varnothing\,,\ \frac{dz_{v_i}(t)}{dt}=f\left[t,\,z_{v_i}(t_m),\,x(t)\right],\ i=\overline{1,k}\,,\ t\in(0,\infty)\,.$$

The hybrid model:

$$v(t_m)\neq\varnothing\,,\ z_v(t_m)\neq\varnothing\,,\ E'\neq\varnothing\,,\ E''\neq\varnothing\,,\ \frac{dz_v(t)}{dt}=f\left[t,\,z_v(t_m),\,x(t)\right],\ t\in\left[t_m\,,t_{m+1}\right),\ v(t)=\text{const}\,,$$
$$\text{when}\ t\in\left[t_m\,,t_{m+1}\right),\ z(t_{m+1})=H\left(z_v(t_m),e_i\right),\ y=G\left(z_v(t_m),e_i\right),\ e_i\in E'\cup E''\,.$$

4. INTERVAL-MARKER PROTOCOL

4.1. Conceptual model of protocol

The method is characterized by a use of time intervals after the end of transmission of a packet or a special marker. The intervals are related with time moments when a transmission medium becomes free. A station may send information if the transmission medium is free for a certain time duration that depends on a concrete medium occupation procedure, which is executed after the end of sending the packet or the marker. Periodicity of sending is determined by timers used in the procedures of network occupation. If the network is free, the synchronizing markers are sent. The markers are reference points of time moments at which stations of the network may occupy the transmission medium in a case of occurrence of sent packets.

Timers A_i and B_i (i – station number) are used in every station in order to control the transmission medium. Timers A_i and B_i are switched on by assigning $\Delta T_{i,k}$ and ΔT_i values when the transmission medium becomes free after the end of transmitting a packet at the i-th station or a marker at the k-th station. Timers are switched off if the transmitting starts before timer is switched off.. When timer A_i is switched off the i-th station acquires the right to transmit packets for some time or to send certain number of packets. After implementing such a possibility both station timers are switched on. If a station has no packets for transmission, then timer A_i of the i-th station is turned off. When the i-th station timer B_i is switched off the station acquires the right to send packets too. If the station has packets for transmitting, then they are transmitted and the both station timers are switched on. In an opposite case the i-th station transmits a marker and the both station timers are switched off.

Time intervals $\Delta T_{i,k}$ and ΔT_i of the network with a successive numbering of stations and a *bus* (transmission medium) topology are calculated by these formulas:

$$\Delta T_{i,k}=\begin{cases}0,\ \text{jei}\ 1=\left[k+1\right]_{\bmod N};\\ (i-k-1)\Delta\tau,\ \text{jei}\ i>k\,;\\ 2\tau_m(i+1)+\Delta\tau(N-k+i-1),\ \text{jei}\ i\leq k, i\neq\left[k+1\right]_{\bmod N};\end{cases}$$

$$\Delta T_i\geq\left(2\tau_m(i+1)+\Delta\tau(N+i)\right);$$

where $\Delta\tau$ – maximum permissible response time of the station;
N – number of stations in the network;
τ_m – the longest signal spread time through the transmission medium between two most remote network nodes.

4.2. Formal description of the protocol

Analyzed protocol is described by one aggregate, which description is presented below.

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 $E''=\left\{e''_{i,j}, i=\overline{1,9},\ j=\overline{0,N-1}\right\}$;

where: $e''_{i,j}$ – a request to transmit an information packet has arrived to the j-th station; $e''_{2,j}$ – timer A_j *has switched off* at the j-th station; $e''_{3,j}$ – timer B_j has switched off at the j-th station; $e''_{4,j}$ – j-th station has started sending an information packet; $e''_{5,j}$ – j-th station has ended sending an information packet; $e''_{6,j}$ – j-th station has started sending a marker; $e''_{7,j}$ – j-th station has started receiving an information packet; $e''_{8,j}$ – j-th station has ended receiving an information packet; $e''_{9,j}$ – j-th station has started receiving a marker.

5. Controlling sequences: $e''_{1,j} \mapsto \{\eta_j^{(1)}\}_{k=1}^{\infty}, \ldots, e''_{5,j} \mapsto \{\eta_j^{(5)}\}_{k=1}^{\infty}$;

where $\eta_j^{(i)}$ – duration of operation after which $e''_{1,j}$ event occurs, $i = \overline{1,5}$.

6. The discrete component of a state:

$$v(t_m) = \{M(t_m), PRIM_j(t_m), QPI(t_m), QCI(t_m), {}^{\#}Q_j(t_m), MA_j(t_m), ATE_j(t_m), ATEJ_j(t_m), JJ_j(t_m), J1(t_m)\};$$

where $M(t_m)$ – number of information packets arrived to the network;

- $PRIM_j(t_m)$ – number of packets received at the j-th station;
- $QPI(t_m)$ – total amount of transferred useful information;
- $QCI(t_m)$ – total amount of transferred service information;
- ${}^{\#}Q_j(t_m)$ – number of information packets in a queue of the j-th station
- $MA_j(t_m)$ – address of the station-receiver of an information packet sent from the j-th station;
- $ATE_j(t_m)$ – variable defining time moment when an information packet appears at first position in the queue of the j-th station;
- $ATEJ_j(t_m)$ – variable defining $ATE_j(t_m)$ value when an information packet is beginning to send;
- $JJ_j(t_m)$ – variable defining an address of the station-sender of an information packe that was received at the j-th station;
- $J1(t_m)$ – variable defining an address of the station, which sends a marker.

7. The continuous component of a state:

$$z_v(t_m) = \{w(e''_{i,j}, t_m), i = \overline{1,9}, j = \overline{0, N-1}\};$$

where $w(e''_{i,j}, t_m)$ – time moment of $e''_{i,j}, i = \overline{1,9}$ event at the j-th station.

8. Parameters of specification $\tau_{i,j}, V, \tau$:

where $\tau_{i,j}$ – signal spread time from the i-th to the j-th station;

- V – information sending speed through a monochannel;
- τ – duration of marker transmission.

9. The initial state:

$$M(t_0) = 0;\ PRIM_j(t_0) = 0;\ \forall j = \overline{0, N-1};\ QPI(t_0) = 0;\ QCI(t_0) = 0;\ {}^{\#}Q_j(t_0) = 0;$$

$$\forall j = \overline{0, N-1};\ MA_j(t_0) = 0;\ \forall j = \overline{0, N-1};\ ATE_j(t_0) = 0;\ \forall j = \overline{0, N-1};\ ATE_j(t_0) = 0;$$

$$JJ_j(t_0) = 0;\ \forall j = \overline{0, N-1};\ J1_j(t_0) = 0;\ w(e''_{1,j}, t_0) = N \cdot \tau/(j+1),\ \forall j = \overline{0, N-1};$$

$$w(e''_{2,j}, t_0) = \Delta T_j;\ \forall j = \overline{0, N-1};\ w(e''_{i,j}, t_0) = \infty,\ \forall i = \overline{3,9},\ \forall j = \overline{0, N-1}.$$

10. Transition and output operators:

$H(e''_{1,j})$: / a request to transmit an information packet has arrived to the j-th station /

$M(t_{m+1}) = M(t_m) + 1$;

${}^{\#}Q_j(t_{m+1}) = {}^{\#}Q_j(t_m) + 1$;

$ATE_j(t_{m+1}) = t_m$, jei ${}^{\#}Q_j(t_{m+1}) = 1$;

$w(e''_{1,j}, t_{m+1}) = t_m + \eta_j^{(1)}$.

$H(e''_{2,j})$: / timer A_j is switched off at the j-th station /

$w(e''_{2,j}, t_{m+1}) = \infty$;

$$\left.\begin{array}{l} w(e''_{2,j}, t_{m+1}) = t_m + \tau / V + \Delta T_j \\ w(e''_{3,j}, t_{m+1}) = t_m + \tau / V + \Delta T_{j,j} \\ w(e''_{6,j}, t_{m+1}) = t_m \end{array}\right\},$$

if ${}^{\#}Q_j(t_m) = 0$.

$ATEJ_j(t_{m+1}) = ATE_j(t_m)$;

${}^{\#}Q_j(t_{m+1}) = {}^{\#}Q_j(t_m) - 1$;

$ATE_j(t_{m+1}) = t_m$, if ${}^{\#}Q_j(t_{m+1}) \neq 0$;

$MA_j(t_{m+1}) = k : k = RAN(0, N-1), k \neq j$;

if ${}^{\#}Q_j(t_m) \neq 0$.

$w(e''_{3,j}, t_{m+1}) = \infty$;

$w(e''_{5,j}, t_{m+1}) = t_m + \eta_j^{(5)}/V$;

$QPI(t_{m+1}) = QPI(t_m) + \eta_j^{(5)}$;

$w(e''_{4,j}, t_{m+1}) = t_m$.

$H(e''_{3,j})$: / timer B_j is switched off at the j-th station /

$w(e''_{3,j}, t_{m+1}) = \infty$;

${}^{\#}Q_j(t_{m+1}) = {}^{\#}Q_j(t_m) - 1$;

$ATEJ_j(t_{m+1}) = ATE_j(t_m)$;

$ATE_j(t_{m+1}) = t_m$, if ${}^{\#}Q_j(t_{m+1}) \neq 0$;

$MA_j(t_{m+1}) = k : k = RAN(0, N-1), k \neq j$;

if ${}^{\#}Q_j(t_m) \neq 0$.

$w(e''_{2,j}, t_{m+1}) = \infty$;

$w(e''_{4,j}, t_{m+1}) = t_m$;

$w(e''_{5,j}, t_{m+1}) = t_m + \eta_j^{(5)}/V$;

$QPI(t_{m+1}) = QPI(t_m) + \eta_j^{(5)}$.

$H(e''_{4,j})$: / j-th station has started sending an information packet /

$w(e''_{7,j}, t_{m+1}) = t_m + \tau_{j,i}$, $\forall i = \overline{0, N-1}$, and $i \neq j$.

$H(e''_{5,j})$:

$w(e''_{2,j}, t_{m+1}) = t_m + \Delta T_j$;

$w(e''_{3,j}, t_{m+1}) = t_m + \Delta T_{i,j}$;

$w(e''_{8,i}, t_{m+1}) = t_m + \tau_{j,i}$, $\forall i = \overline{0, N-1}$; and $i \neq j$.

$JJ_{MA_j(t_m)}(t_{m+1}) = j$.

$H(e''_{6,j})$:

$J1(t_{m+1}) = j$;

$w(e''_{9,i}, t_{m+1}) = t_m + \tau_{j,i}$, $\forall i = \overline{0, N-1}$; ir $i \neq j$.

$H(e''_{7,j})$:

$w(e''_{2,j}, t_{m+1}) = \infty$;

$w(e''_{3,j}, t_{m+1}) = \infty$.

$H(e''_{8,j})$:

$w(e''_{2,j}, t_{m+1}) = t_m + \Delta T_j$;

$w(e''_{3,j}, t_{m+1}) = t_m + \Delta T_{j, JJ_j(t_m)}$.

$H(e''_{9,j})$:

$w(e''_{2,j}, t_{m+1}) = t_m + \tau / V + \Delta T_j$;

$w(e''_{3,j}, t_{m+1}) = t_m + \tau / V + \Delta T_{j, J1(t_m)}$.

4.3. Results of protocol simulation

Dependencies of an average waiting time of packets on an intensity of the arrived requests are presented in fig. 1-2. Two transfer rates were considered during simulation: 10 Mbps and 50 Mbps. Curves of these dependabilities were obtained by applying three data transfer methods: marker, event-driven interval marker. Another parameters of the model are: length of the information packet – 960 bits, length of the marker – 48 bits. Curves obtained in 1 and 2 figures reflect the network consisting of six stations.

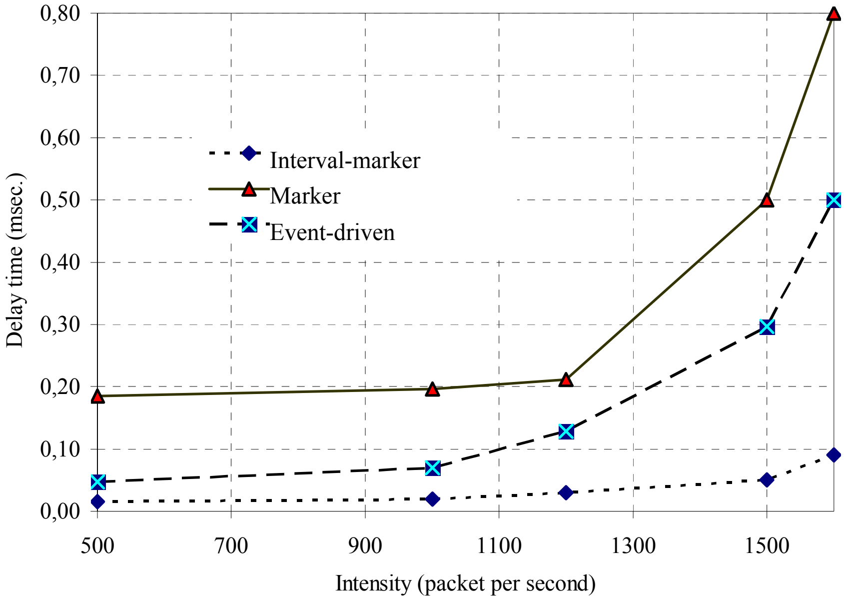

Figure 1. Dependencies of an average waiting time on an intensity of flow of the arriving requests when the bus speed is 10 Mbps

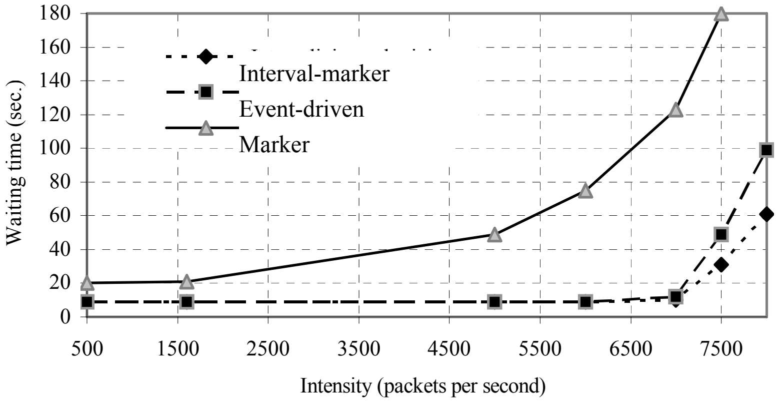

Figure 2. Dependencies of a waiting time on an intensity of the arriving requests when the bus speed is 50 Mbps

5. Conclusions

The paper shows how PLA formalization approach can be used for creation simulation models computer network protocols. Simulation results of interval-marker protocol show that this protocol permits to delay less transferred packets in a whole range of the transmission medium. This is because a control of packet transmission is based on a calculation of time intervals. Results of marker method are worse because the control is based on a continuous marker transmission with a right to send packet to a station that will receive it.

References

[1] Buslenko, N., Kalashnikov, V., Kovalenko, I. (1973). *Lectures on Complex Systems.* Moscow, Soviet. Radio. (In Russian)

[2] Pranevicius, H. (1982). *Models and methods for computer system investigation*. Mokslas, (p. 228). Vilnius. (In Russian)

[3] Pranevicius, H., Pilkauskas, V., Chmieliauskas, A. (1994). *Aggregate approach for specification and analysis of computer network protocols* (p. 228). Kaunas University of Technology, Lithuania, "*Technologija*".

[4] Pranevičius, H. (2005). *Formal specification and analysis: of computer network protocols aggregate approach*. (p.186). Kaunas University of Technology, Lithuania, "*Technologija*". (In Lithuanian)

[5] Zeigler, B.P. (2000). *Theory of Modeling and Simulation* (p.510). Academic Press.