An Evaluation of Network Response Time using a Coloured Petri Net Model of Switched LAN

Dmitry A. Zaitsev

Odessa National Telecommunication Academy,
Kuznechnaya, 1, Odessa, 65029, Ukraine
Web: http://www.geocities.com/zsoftua

Abstract

The enterprise class model of switched LAN in the form of a coloured Petri net is represented. The components of the model are switches, servers and workstations. For the evaluation of network response time a special measuring workstation model is proposed. It counts response times for each request and calculates the average response time. For the simulation of network behaviour and accumulation of statistical information, CPN Tools was applied. Hierarchical nets usage allows the convenient representation of an arbitrary given structure of LAN.

Keywords: LAN; Switch; Response time; Colored Petri net; Evaluation

1. Introduction

The technology of switching [4] is prospective for bandwidth increase in local and global computer networks. But it is hard enough to create an adequate analytical model of a switched network [2]. Petri net models [5] contain facilities for precise description of network architecture and traffic peculiarities and allow the representation of interaction within the client-server systems.

Early represented model [9] has been refined up to enterprise quality. CSMA (Carrier Sense Multiple Access) procedures are implemented. Complete full-duplex mode is simulated with separate input and output frame buffers. The model of switch was arranged for technological convenience with fusion places allowing an easy description of an arbitrary number of ports. Moreover, the general model was supplied with special measuring workstation model that calculates network response time.

Notice that the model is represented with hierarchical coloured [3] timed [7,10] Petri nets. For automated composition of model and accumulation of statistical information during network behaviour simulation, CPN Tools [1] was used.

In the Section 2 we consider the peculiarities of switched Ethernet LAN construction. Model of LAN is described in Section 3, whereas Sections 4, 5, 6 are devoted to sub models of: switch, server, workstation, measuring workstation. Evaluation technique is represented in Section 7 and Section 8 contains the discussion of models parameters.

Results obtained may be used in real-time applications sensitive to delays, as well as at communication equipment, for instance, switches, development.

2. Switched LAN

Recently the Ethernet has become the most widespread LAN. With gigabit technology it started a new stage of popularity. And this is not the limit yet. Hubs are dumb passive equipment aimed only at the connection of devices as wires. The base element of the Local Area Network (LAN) Ethernet (IEEE 802.x) is a switch of frames. Logically a switch is constituted of a set of ports [6]. LAN segment (for example, made up via hub) or terminal equipment such as workstation or server may be attached to each port. The task of a switch is the forwarding of incoming frame to the port that the target device is connected to. The usage of a switch allows for a decrease in quantity of collisions so each frame is transmitted only to the target port and results in an increased bandwidth. Moreover the quality of information protection rises with a reduction of ability to overhear traffic. The scheme of sample switched network is presented in Fig. 1.

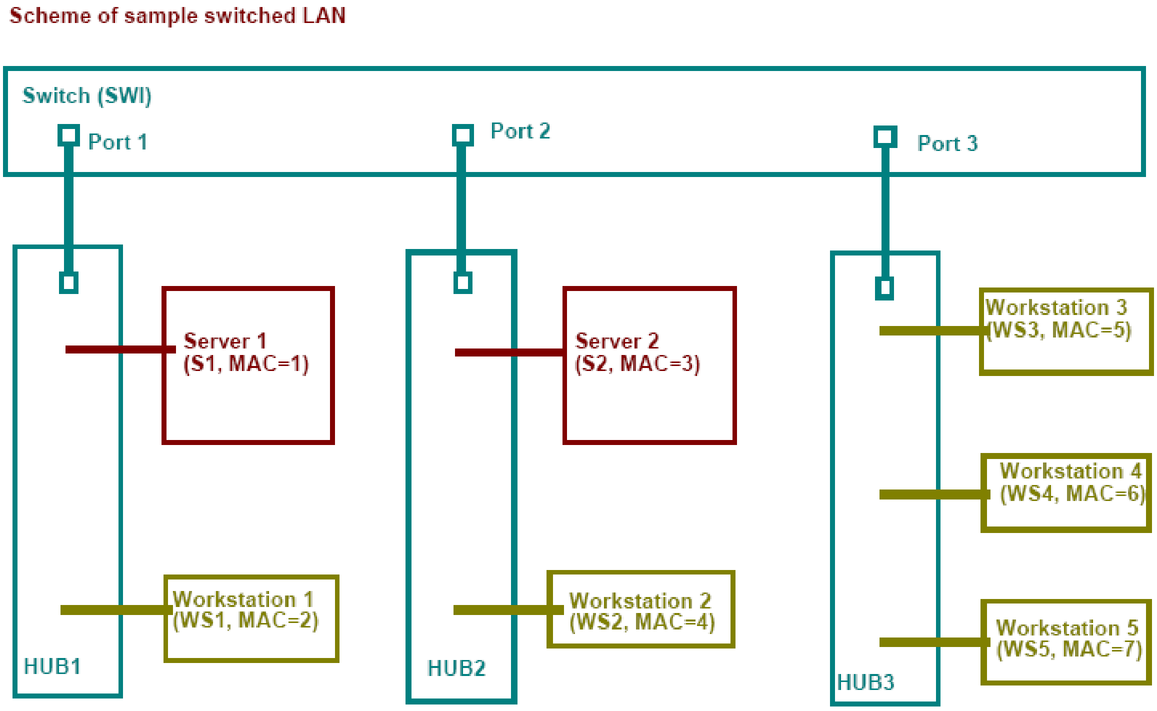

Fig. 1. Scheme of sample switched LAN

As a rule, the Ethernet works in a full-duplex mode now, which allows simultaneous transmission in both directions. To determine the target port number for the incoming frame a static or dynamic switching table is used. This table contains the port number for each known Media Access Control (MAC) address. Only static switching tables will be modelled in the present paper.

3. Model of LAN

A model of sample LAN with topology, shown in Fig. 1, is represented in Fig. 2. Let us describe the model constructed. Notice that the model is represented with coloured Petri net [3] and consists of places, drawn as circles (ellipses), transitions, drawn as bars, and arcs. Dynamic elements of the model, represented by tokens, are situated in places and move as a result of the transitions' firing.

The elements of this model are sub models of: Switch (**SWI**), Server (**S**), Workstation (**WS**) and Measuring Workstation (**MWS**). Workstations **WS1-WS4** are the same type exactly **WS**, whereas workstation **WS5** is the type **MWS**. It implements the measuring of network response time. Servers **S1** and **S2** are the same type exactly **S**. Hubs are a passive equipment and have not an independent model representation. The function of hubs is modelled by common use of the corresponding places **p*in** and **p*out** by all the attached devices. The model does not represent the collisions. Problems of the Collision Detection (CD) were studied in [11].

Each server and workstation has it's own MAC address represented in places **aS***, **aWS***. A switch has separate places for input (**p*in**) and output (**p*out**) frames for each port. It represents the full-duplex mode of work. Bidirected arcs are used to model the carrier detection procedures. One of the arcs checks the state of the channel, while another implements the transmission.

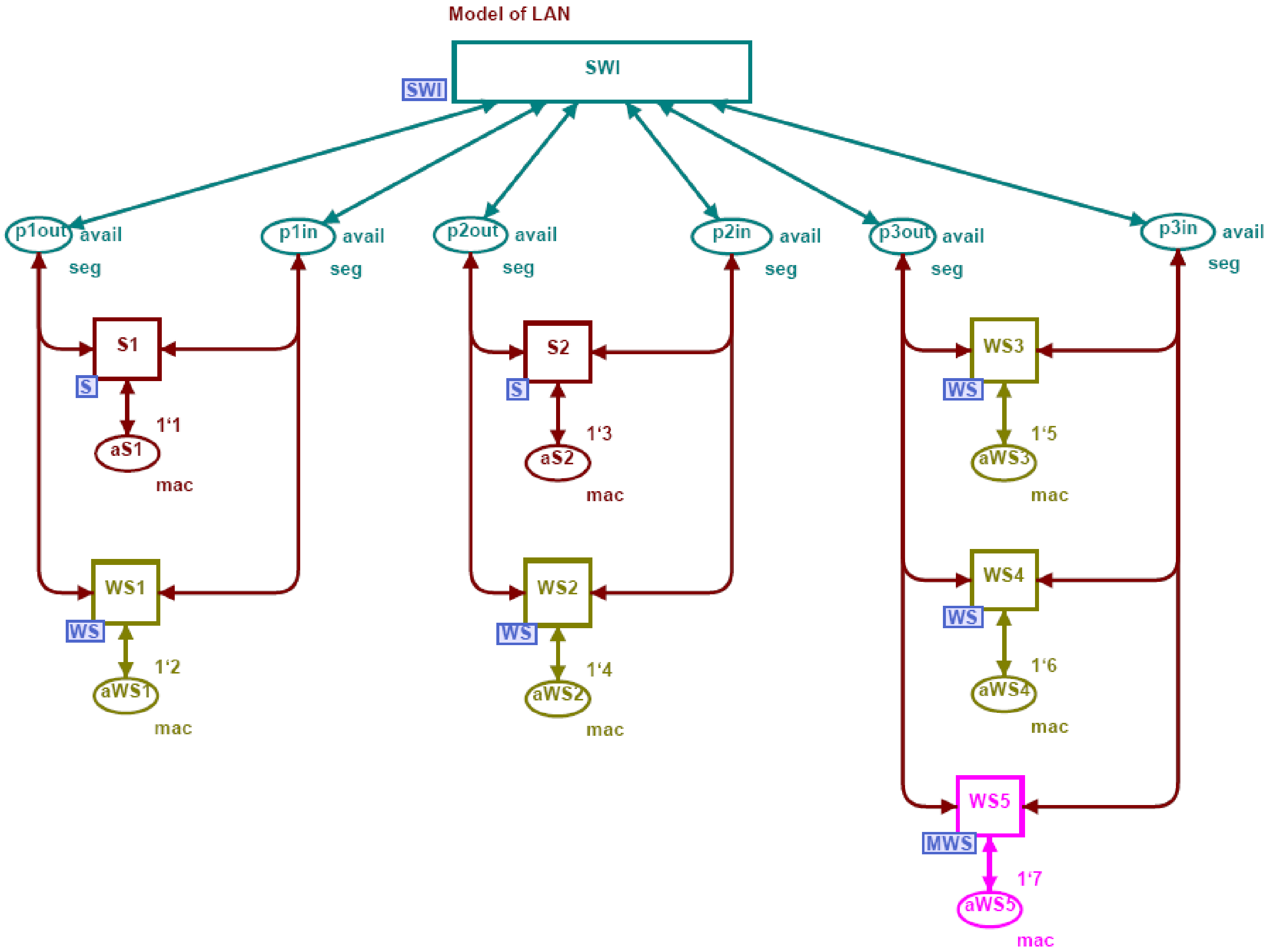

Fig. 2. Model of sample LAN

```
color mac = INT timed;
color portnum = INT;
color nfrm = INT;
color sfrm = product nfrm * INT timed;
color frm = product mac * mac * nfrm timed;
color seg = union f:frm + avail timed;
color swi = product mac * portnum;
color swf = product mac * mac * nfrm * portnum timed;
color remsv = product mac * nfrm timed;
var src, dst, target: mac;
var port: portnum;
var nf, rnf: nfrm;
var t1, t2, s, q, r: INT;
color Delta = int with 1000..2000;
fun Delay() = Delta.ran();
color dex = int with 100..200;
fun Dexec() = dex.ran();
color dse = int with 10..20;
fun Dsend() = dse.ran();
color nse = int with 10..20;
fun Nsend() = nse.ran();
fun cT()=IntInf.toInt(!CPN'Time.model_time)
```

Fig. 3. Declarations

All the declarations of colours (**color**), variables (**var**) and functions (**fun**) used in the model are represented in Fig. 3. The Ethernet MAC address is modelled with integer number (colour **mac**). The frame is represented by a triple **frm**, which contains source (**src**) and destination (**dst**) addresses, and also a special field **nfrm** to enumerate the frames for the calculation of response time. We abstract of other fields of frame stipulated by standard of Ethernet. The colour **seg** represents unidirectional channel and may be either available for transmission (**avail**), or busy with transmission of a frame (**f.frm**). It is represented with a **union** type of colour. Notice that the descriptor **timed** is used for tokens, which take part in timed operations such as delays or timestamps.

The marking of places is represented with multisets in CPN Tools. Each element belongs to a multiset with defined multiplicity, in other words – in a few copies. For instance, the initial marking of the place **aWS2** is **1`4**. It means that place **aWS2** contains 1 token with a value of 4. The union of tokens is represented by a double plus sign (++). Tokens of timed colour have the form **x @ t** which means that token **x** may be involved only after a moment of time **t**. So, notation **@+d** is used to represent the delay with the interval **d**.

4. Model of Switch

Let us construct a model for a given static switching table. We consider the separate input and output buffers of frames for each port and common buffer of the switched frames. The model of switch (**SWI**) is presented in Fig. 4. The hosts' disposition according to Fig. 1 was used for the initial marking of a switching table.

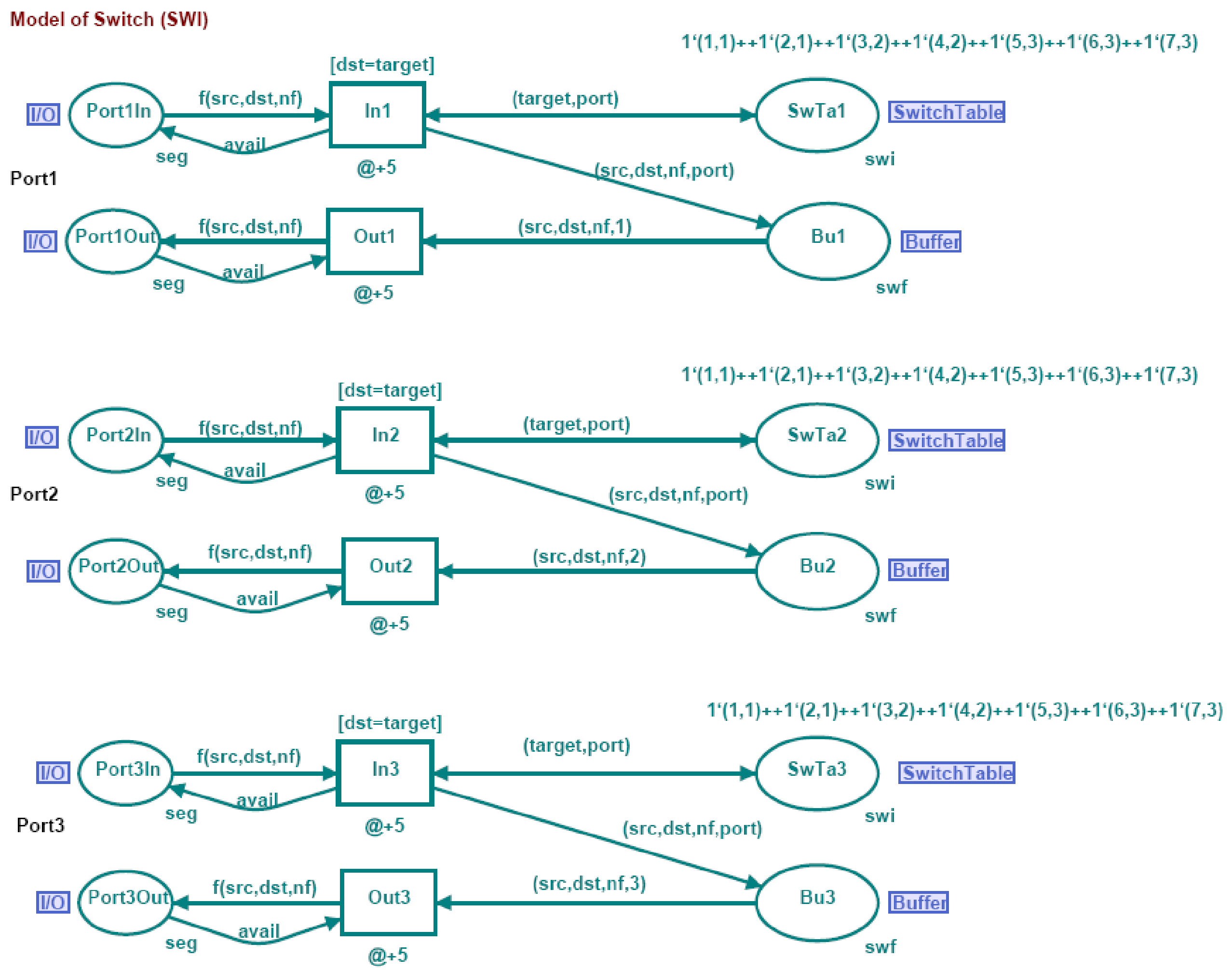

Fig. 4. Model of switch

The colour **swi** represents records of switching table. It maps each known MAC address (**mac**) to the number of port (**nport**). The colour **swf** describes the switched frames, waiting for output buffer allocation. The field **portnum** stores the number of the target port. The places **Port*In** and **Port*Out** represent input and output buffers of the ports correspondingly. The fusion place **SwitchTable** models the switching table; each token in this place represents the record of the switching table. For instance, token **1`(4,2)** of the initial marking means that the host with MAC address 4 is attached to port 2. The fusion place **Buffer** corresponds to the switched frames' buffer. Notice that a fusion place (such as **SwitchTable** or **Buffer)** represents a set of places. The fusion place **SwitchTable** is represented with places **SwTa1, SwTa2, SwTa3**. The fusion place **Buffer** is represented with places **Bu1, Bu2, Bu3**. It allows the convenient modelling of switches with an arbitrary number of ports avoiding numerous cross lines.

The transitions **In*** model the processing of input frames. The frame is extracted from the input buffer only in cases where the switching table contains a record with an address that equals to the destination address of the frame (**dst=target**); during the frame displacement the target port number (**port**) is stored in the buffer. The transitions **Out*** model the displacement of switched frames to the output ports' buffers. The inscriptions of input arcs check the number of the port. The fixed time delays (@+5) are assigned to the operations of the switching and the writing of the frame to the output buffer.

It is necessary to explain the CSMA procedures of LAN access in more detail. When a frame is extracted from the input buffer by transition **In*,** it is replaced with the label **avail**. The label **avail** indicates that the channel is free and available for transmission. Before the transition **Out*** sends a frame into a port, it analyses if the channel is available by checking the token **avail**.

Notice that places **Port*In** and **Port*Out** are contact ones. They are pointed out with an **I/O** label. Contact places are used for the construction of hierarchical nets with substitution of transition. For example, the transition **SWI** in the top-level page of model (Fig. 2) is substituted by a whole net **SWI** represented in Fig. 3. Places **Port*In** and **Port*Out** are mapped into places **p*in** and **p*out** correspondingly.

5. Models of Workstation and Server

To investigate the frames' flow transmitting through LAN and to estimate the network response time it is necessary to construct the models of terminal devices attached to the network. Regarding the peculiarity of the traffic's form we shall separate workstations and servers. For an accepted degree of elaboration we consider periodically repeated requests of workstations to servers with random uniformly distributed delays. On reply to an accepted request a server sends a few packets to the address of the requested workstation. The number of packets sent and the time delays are uniformly distributed random values.

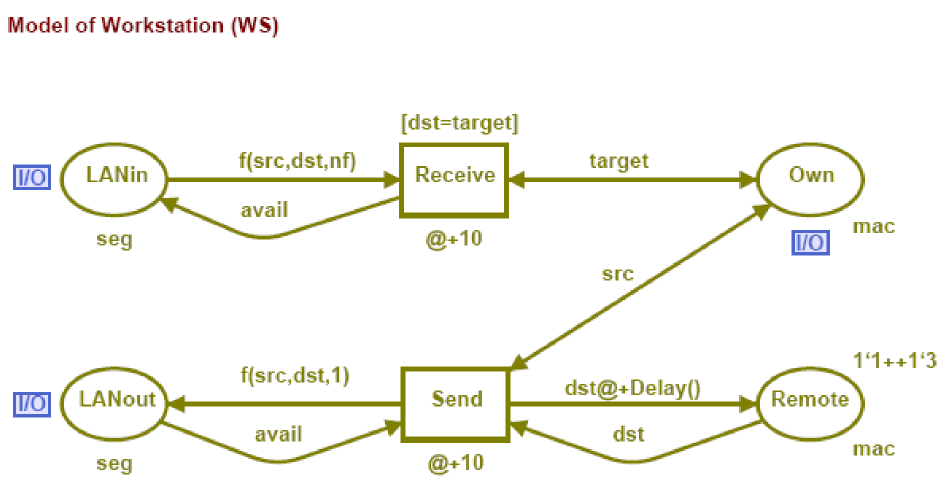

Fig. 5. Model of workstation

A model of workstation (**WS**) is represented in Fig. 5. The places **LANin** and **LANout** model the input and output channels of the local area network correspondingly. The workstation listens to the network by means of transition **Receive** that receives frames with the destination address, which is equal to the own address of the workstation (**dst=target**) saved in the place **Own**. The processing of received frames is represented by the simple absorption of them. The workstation sends periodic requests to servers by means of transition **Send**. The servers' addresses are held in the place **Remote**. After the sending of a request the usage of the server's address is locked by the random time delay given by the function **Delay()**. The sending of the frame is implemented only if the LAN segment is free. It operates by checking place **LANout** for a token **avail**. In such a manner the workstation interacts with a few servers holding their addresses in the place **Remote**.

Notice that the third field of frame, named **nfrm,** is not used by the ordinary workstation **WS**. The workstation only assigns the value of a unit to it. This field is used by a special measuring workstation **MWS**. The copies of the described model **WS** represent workstations **WS1-WS4**. To identify each workstation uniquely, the contact place **Own** is used. This place is shown also in the top-level page (Fig. 2) and contains the MAC address of host.

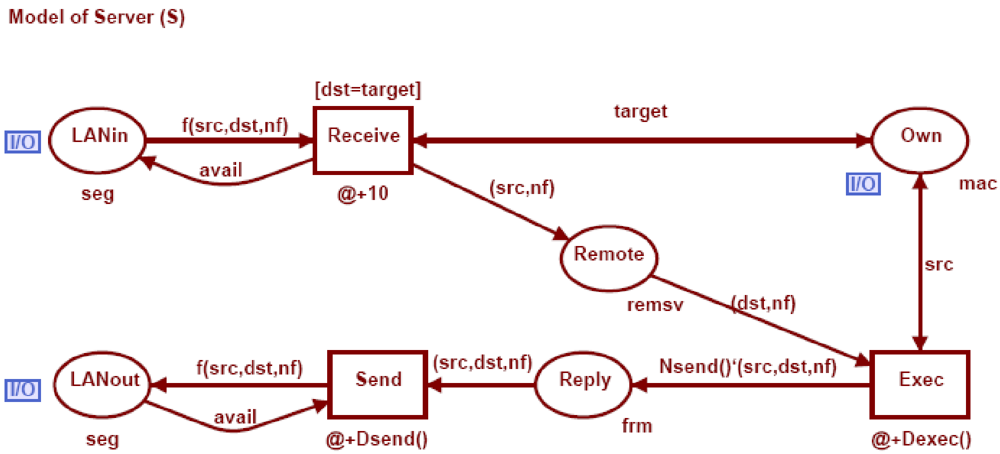

Fig. 6. Model of server

A model of server (**S**) is represented in Fig. 6. The listening of the network is similar to the model of the workstation but it is distinct in that the frame's source address is held in the place **Remote**. The transition **Exec** models the execution of the workstation's request by a server. As a result of the request execution the server generates a random number **Nsend()** of the response frames, which are held in the place **Reply**. Then these frames are transmitted into the network by the transition **Send**. Notice that the request number **nf** is stored in the place **Remote** also. It allows us to identify the response with the same number as the request.

6. Model of Measuring Workstation

A model of the measuring workstation (**MWS**) is represented in Fig. 7. In essence, it is an early considered model of workstation **WS**, supplied with the measuring elements (the measuring elements are drawn in magenta).

Let us consider the measuring elements in more detail. Each frame of a workstation's request is enumerated with a unique number contained in the place **num**. The time, when the request was sent, is stored in the place **nSnd**. The function **cT()** calculates the current value of the model's time. The place **nSnd** stores a pair: the frame's number **nf** and the time of request **cT()**.

The place **return** stores the timestamps of all the returned frames. As the network response time we consider the interval of time between the sending of the request and receiving the first frame of response. This value is stored in place **NRTs** for each responded request. The transition **IsFirst** determines the first frame of response. The inscription of the arc, connecting the transition **IsFirst** with the place **NRTs**, calculates the response time (t2-t1).

A residuary part of the measuring elements calculates the average response time. The places **sum** and **quant** accumulate the sum of response times and the quantity of accepted responses correspondingly. The arrival of a new response is sensed by the place **new** and initiates the recalculation of average response time with the transition **Culc**. The result is stored in the place **NRTime**.

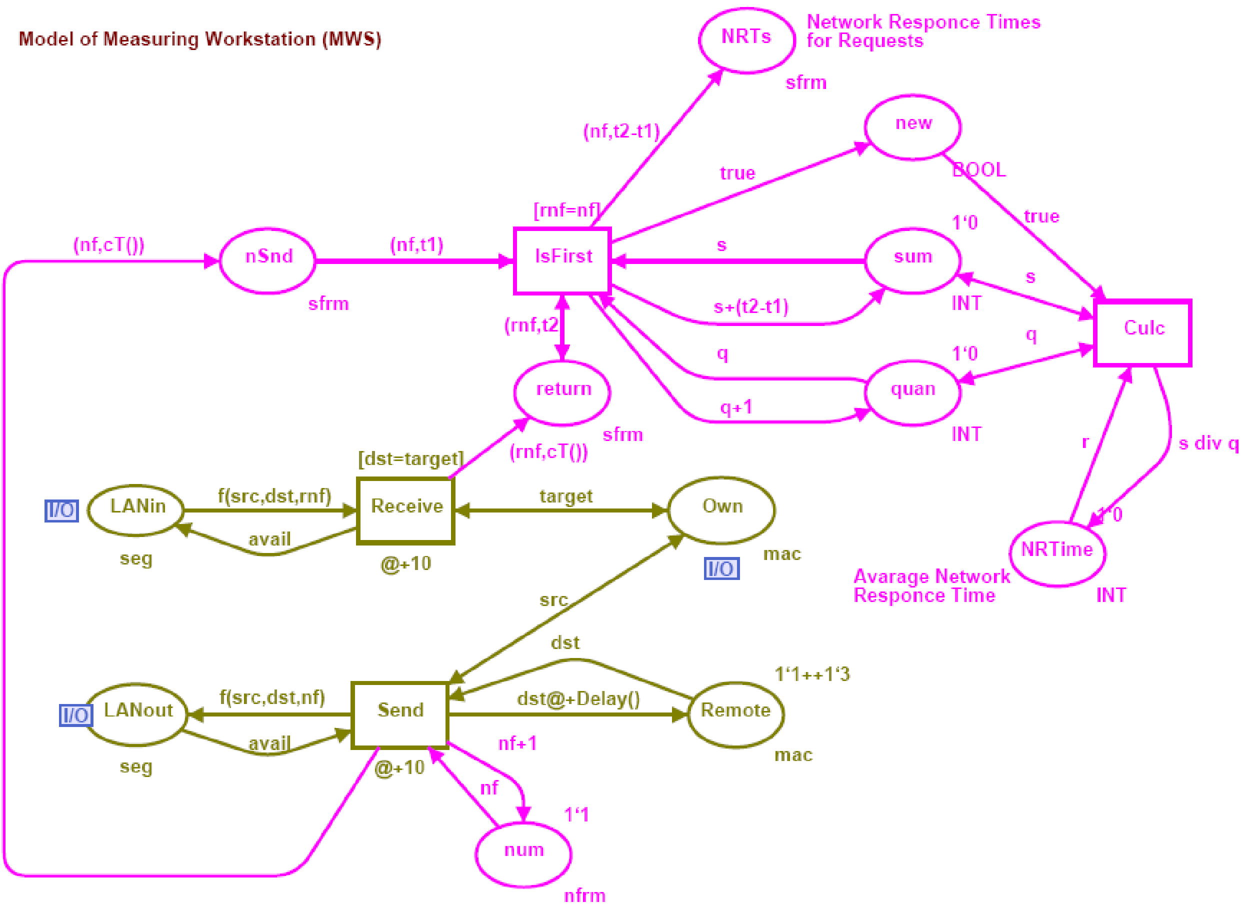

Fig. 7. Model of measuring workstation

7. Evaluation Technique

The model constructed was debugged and tested in a step-by-step mode of simulation. For these purposes the frame generated by the workstation was traced through the network to the server and back. Also we observed the behaviour of the model in the process of automated simulation with a display of net's dynamics – in the mode of the so-called game of tokens. It allows us to estimate the model with a glance at the top-level page and at sub pages during simulation.

To estimate the network response time precisely, rather huge intervals of model time are required. It is convenient for such purposes to use the simulation mode without displaying intermediate marking aimed at the accumulation of statistics.

A snapshot of the measuring workstation model is represented in Fig. 8. The rectangular labels (drawn in bright green) describe the current marking of the simulation system; the circular labels contain the number of tokens. The place **LANin** contains frame (1,5,1). The place **LANout** represents the available state of the channel **avail**. The number of the next request, according to the marking of place **num**, is 7. The place **return** indicates that 83 frames of responses have arrived. The place **NRTs** contains the response times for each of the 6 responded requests. For instance, the network response time for request 5 equals to 235. It should be calculated easily, that the average network response time 389 in the place **NRTime** equals to 2337/6 according to the markings of the places **sum** and **quant**.

8. Parameters of Model

The right choice of time unit for model time measurement is a key question for an adequate model construction as well as the calculation of timed delays for elements of the model. It requires an accurate consideration of the real network hardware and software characteristics.

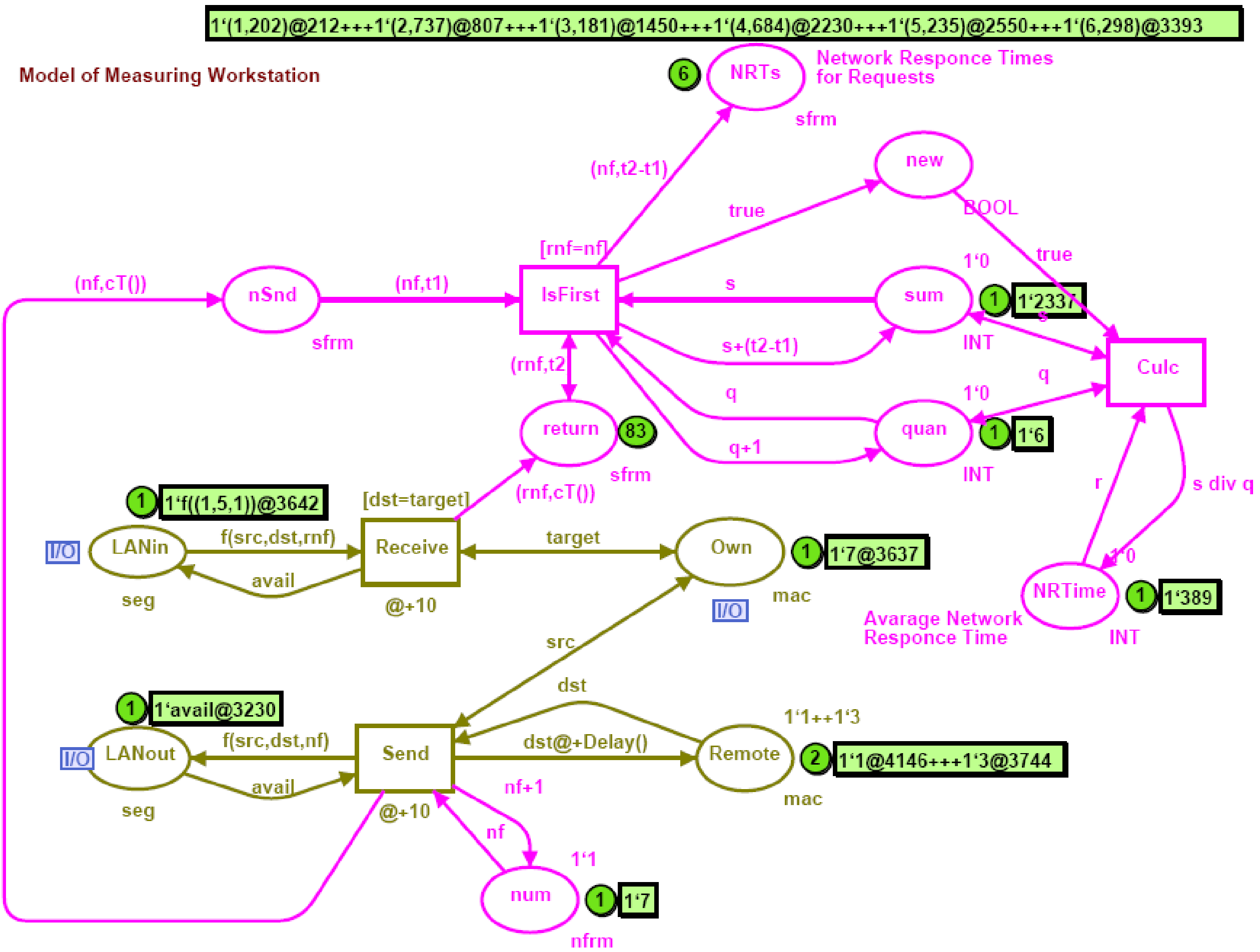

Fig. 8. Estimation of network response time

The scheme shown in Fig. 1 represents a fragment of a railway dispatch centre LAN supplied with special railway CAM software GID Ural [12]. The core of the system constitutes a pair of mirror servers **S1** and **S2**. The workstations **WS1-WS5** are situated in the workplaces of railway dispatchers.

We have to consider the performance of the concrete LAN switch and LAN adapters to calculate the timed delays of transitions **In***, **Out***, **Send**, **Receive**. Moreover, the peculiarities of client-server interaction of GID Ural software ought to be considered for the estimation of such parameters as delay between the requests **Delta** and the time of request execution **dex**. Since the unit of information transmitting through net is represented with a frame, we have to express the lengths of messages in numbers of frames. For these purposes the maximal length of an Ethernet frame equalling 1.5 Kb was chosen.

The types of LAN hardware used are represented in Table 1.

Table 1. Types of hardware

Device	Type
LAN adapter	Intel EtherExpress 10/100
LAN switch	Intel SS101TX8EU
Server	HP Brio BA600
Workstation	HP Brio BA200

In Table 2 the parameters of the model described are represented. LAN switch and adapter operations are modelled with fixed delays so they are small enough in the comparison with client-server interaction times. Moreover, in reliable Ethernet frames of maximal length are transmitted mainly, since the time of frame's processing is a fixed value. Stochastic variables are represented with uniform distribution, which corresponds to Ural GID software behaviour. The smallest timed value is the LAN switch time of read/write frame operation. But for the purposes of future representation of faster equipment we choose the unit of model time (MTU) equalling 100 ns.

Table 2. Parameters of model

Parameter	Variable/Element	Real value	Model value
LAN switch read frame delay	In*	500 ns	5
LAN switch write frame delay	Out*	500 ns	5
LAN adapter read frame delay	Receive	1 ms	10
LAN adapter write frame delay	Send	1 ms	10
Server's time of request processing	Dex	10-20 ms	100-200
Client's delay between requests	Delta	100-200 ms	1000-2000
Length of request		1.2 Kb	1
Length of response	Nse	15-30 Kb	10-20

Thus, the average network response time obtained equals 389 MTU or about 39 ms. This delay satisfies the requirements of train traffic control [12].

9. Conclusion

In the present work the technology of switched local area networks' models development was studied. The usage of coloured Petri nets allows the peculiarity of interaction within the client-server systems to be taken into account. The model reflects the major features of a real-life network. CSMA procedures, full-duplex mode and switching tables were modelled. A special measuring model of workstation was suggested and implemented to estimate the network response time.

The model developed is of enterprise class, so it allows easy and convenient adequate representation of LAN with an arbitrary given topology. The technique described is aimed at real-time applications, requiring the precise estimation of timed delays before implementation.

References

1. Beaudouin-Lafon M., Mackay W.E., Jensen M. et al. CPN Tools: A Tool for Editing and Simulating Coloured Petri Nets. LNCS 2031: Tools and Algorithms for the Construction and Analysis of Systems, 2001, 574-580.
2. Elsaadany M., Singhal T., Lui Ming. Performance study of buffering within switches in local area networks. Proc. of 4th International Conference on Computer Communications and Networks, 1995, 451-452.
3. Jensen K. Colored Petri Nets – Basic Concepts, Analysis Methods and Practical Use. Springer-Verlag, Vol. 1-3, 1997.

4. Hunt R. Evolving Technologies for New Internet Applications. IEEE Internet Computing, 5 1999, 16-26.
5. Peterson J. Petri Net Theory and the Modelling of Systems. Prentice Hall, 1981.
6. Rahul V. LAN Switching. OHIO 2002.
7. Zaitsev D.A., Sleptsov A.I., State Equations and Equivalent Transformations of Timed Petri Nets. Cybernetics and System Analysis, 33, 1997, 659-672.
8. Zaitsev D.A. Subnets with Input and Output Places. Petri Net Newsletter, Vol. 64, April 2003, 3-6, Cover Picture Story.
9. Zaitsev D.A. Switched LAN simulation by colored Petri nets. Mathematics and Computers in Simulation, vol. 65, no. 3, 2004, 245-249.
10. Zaitsev D.A. Invariants of timed Petri nets. Cybernetics and Systems Analysis, no. 2, 2004, 92-106.
11. Zaitsev D.A. Verification of Ethernet protocols // Proceedings of Odessa National Telecommunication Academy, no. 1, 2004, p. 42-48.
12. Zyabirov H.S., Kuznetsov G.A., Shevelev F.A., Slobodenyuk N.F., Krasheninnikov S.V., Krayisvitny V.P., Vedischev A.N. Automated system for operative control of exploitation work GID Ural-VNIIZT // Railway transport, no. 2, 2003, 36-45.

Published in: Proceedings of Fifth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, October 8-11, 2004, Aarhus, Denmark.