

SWITCHED ETHERNET RESPONSE TIME EVALUATION VIA COLORED PETRI NET MODEL

Dmitry A. Zaitsev¹ and Tatiana R. Shmeleva²

Department of Communication Networks,
Odessa National Telecommunication Academy,
Kuznechnaya, 1, Odessa 65029, Ukraine

Web¹: <http://www.geocities.com/zsoftua>, E-mail²: tishtri@rambler.ru

KEYWORDS

Ethernet, Switch, Response time, Colored Petri net, Evaluation

ABSTRACT

The enterprise class model of switched LAN in the form of a colored Petri net is represented. The components of the model are submodels of switch, server and workstation. 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 behavior and accumulation of statistical information, CPN Tools was applied. Hierarchical nets usage allows the convenient representation of an arbitrary given structure of LAN. The influence of such features as limitations of switch's buffer size and dynamic maintenance of switching table were estimated.

INTRODUCTION

The technology of switching (Hunt 1999) 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 (Elsaadany et al. 1995). Petri net models (Peterson 1981) 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 (Zaitsev 2004a) 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 frames' buffers. The model of switch is arranged for technological convenience with fusion places allowing an easy description of an arbitrary number of ports. Moreover, the general model is supplied with a special measuring workstation model that calculates network response time. Further to (Zaitsev 2004d) the influence of such features as limitations of switch's buffer size and dynamic maintenance of switching table are estimated.

It should be noticed that the model is represented with hierarchical colored (Jensen 1997) timed (Zaitsev and Sleptsov 1997; Zaitsev 2004b) Petri nets. For automated composition (Zaitsev 2005) of model and accumulation of statistical information during network behavior simulation, CPN Tools (Beaudouin-Lafon et al. 2001) is used.

At simulation, two major problems arise: to construct an adequate model and to evaluate its characteristics. Several simulating systems provide basic facilities for measurement of average storage capacity and frequencies of actions. But from the practical point of view more complicated characteristics are significant. For example, network response time and number of collisions for Ethernet. To solve this problem special facilities for measurement of models' characteristics are required. Colored Petri net is universal algorithmic system (Jensen 1997). We propose to represent algorithms of measurement as special measuring fragments of model implemented in the same language of colored Petri nets.

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

SWITCHED ETHERNET 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 802.3) is a switch of frames. Logically a switch is constituted of a set of ports (Rahul V. 2002). 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.

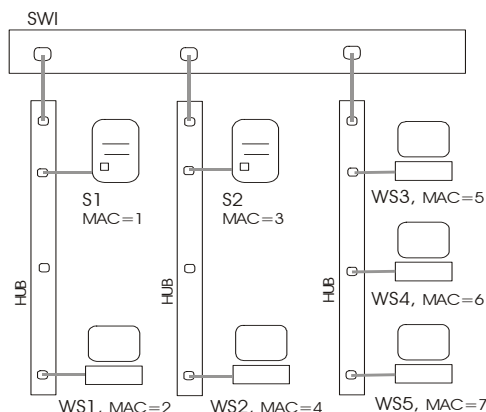


Figure 1: Scheme of railway dispatch center LAN

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.

The scheme shown in Fig. 1 represents a fragment of a railway dispatch center LAN supplied with special railway CAM software GID Ural (Zyabirov et al. 2003). 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.

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. We have to consider the performance of the concrete LAN switch and LAN adapters to calculate the timed delays (further represented with model's 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 equaling 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 modeled 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 behavior. 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) equaling 10 mcs.

Table 2: Parameters of model

Parameter	Real value	Model value	Variable/Element
LAN switch read frame delay	50 mcs	5	In*
LAN switch write frame delay	50 mcs	5	Out*
LAN adapter read frame delay	100 mcs	10	Receive
LAN adapter write frame delay	100 mcs	10	Send
Server's time of request processing	1-2 ms	100-200	Dex
Client's delay between requests	10-20 ms	1000-2000	Delta
Length of request	1.2 Kb	1	
Length of response	15-30 Kb	10-20	Nse

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. It should be noticed that the model is represented with colored Petri net (Jensen 1997) 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. Tokens of a colored Petri net are objects of abstract data types (colors). Transition's fireability rules in colored Petri net involve inscriptions of input arcs, which chose input tokens, and a guard of transition, which constitutes a predicate defined on input tokens. Inscriptions of transition's output arcs constitute constructors of output tokens.

The elements of the 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 modeled 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 (Zaitsev 2004c).

Each server and workstation has its 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.

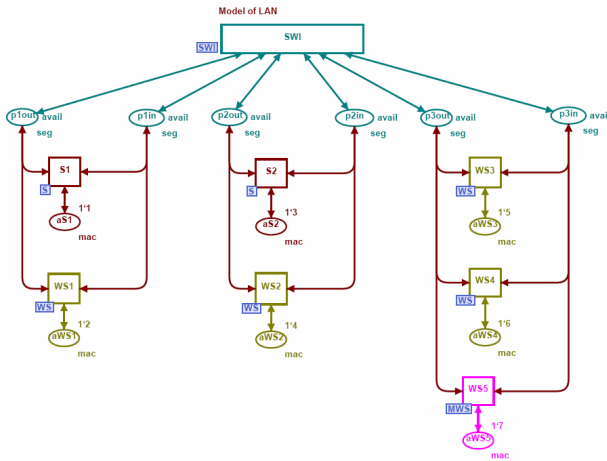


Figure 2: Model of sample LAN

```

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 rmsv = product mac * nfrm timed;
var src, dst, target: mac;
var port: portnum;
var nf, nfm: 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 nse = int with 10..20;
fun Nsend() = nse.ran();
fun cT()=IntInf.toInt(!CPNTime.model_time)
    
```

Figure 3: Declarations

All the declarations of colors (**color**), variables (**var**) and functions (**fun**) used in the model are represented in Fig. 3. The Ethernet MAC address is modeled with integer number (color **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 color **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 color. It should be noticed 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 (Jensen 1997) 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 the value of 4. The union of tokens is represented by the double plus sign (**++**). Tokens of timed color 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**.

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.

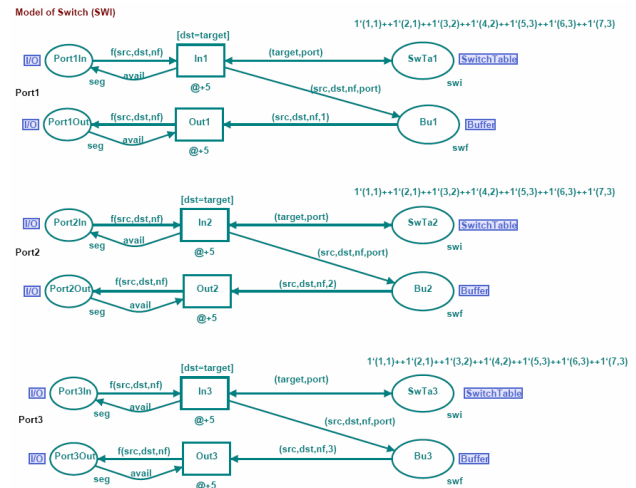


Figure 4: Model of switch

The color **swi** represents records of switching table. It maps each known MAC address (**mac**) to the number of port (**nport**). The color **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. It should be noticed 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 modeling 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**.

It should be noticed 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.

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.

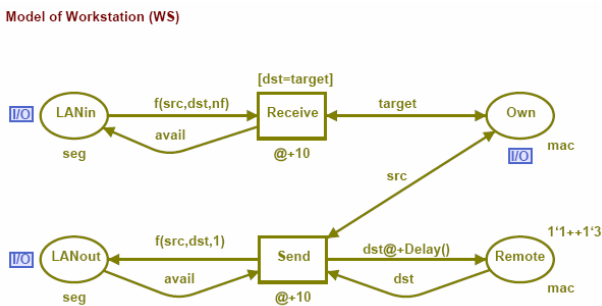


Figure 5: Model of workstation

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**.

It should be noticed 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 a host.

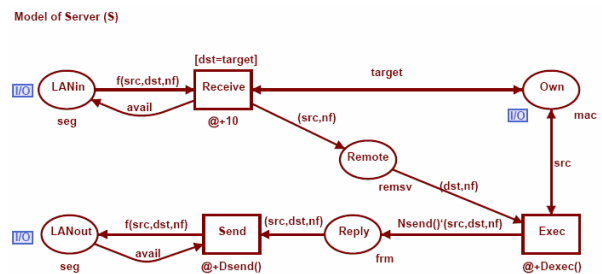


Figure 6: Model of server

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**. It should be noticed 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.

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).

Table 3: Steady-state conditions for network response time

Step	1000	11000	111000	1111000	2111000	3111000
Model time, MTU	2831	24488	245973	3124656	5340611	7547756
Response time, MTU	318	316	307	304	304	304

Table 4: Response time for twice slower switch (In*, Out*: @+10)

Step	1000	11000	21000	121000	221000
Model time, MTU	3306	26978	50536	285136	521713
Response time, MTU	336	800	1284	4477	7692
Queue size, frames	43	96	102	357	498

Table 5: Response time for twice slower network adapters (Receive, Send: @+20)

Step	1000	11000	21000	121000	221000
Model time, MTU	3306	26978	50536	352956	644517
Response time, MTU	336	800	1284	14665	27803
Queue size, frames	43	43	102	3462	6305

BUFFER SIZE LIMITATIONS INFLUENCE

Though model constructed represents a lot of characteristic features of real-life Ethernet, such, for instance, as full-duplex mode, switching table, CSMA procedures, it is simplified enough. There are a few additional peculiarities of switched LAN, which influent more or less essentially on the bandwidth of network. Let us enumerate them. The model of switch does not take into accounting real size of internal buffer of frames and does not implement mechanism of jam messages to avoid overload. Switching table is static whereas the majority of real-life hardware provide dynamic switching table. Random extraction from switching table is implemented. Moreover, we do not consider the timed characteristics of frame transmission through segment of LAN. It is adequate in the case we do not consider broadcasting and multicasting. In model constructed we put frame into the place, modeling the segment, and we are assured that there is a target device in this segment, which will get the frame. For representing of multicasting we have to remove the frame from segment after the elapsing of transmission time.

As for technology of client-server interaction, we consider two-way handshake only. The model may be refined for more complicated algorithms of interaction reflecting the behavior of concrete software. But for algorithm of interaction implemented we use random discipline of extraction for places representing the queues of requests and responses. Moreover, execution of request does not

take into accounting the number of server's processors and issue on unlimited number of processors.

But the complete detailed model of Ethernet LAN is too sophisticated. Therefore, we not go beyond to consider the influence of most significant items of real-life network on response time obtained. At first we modify the model for limited switch's buffer and estimate the network response time under different values of buffer's size. The model of switch's port for limited buffer's capacity is represented in Fig. 9.

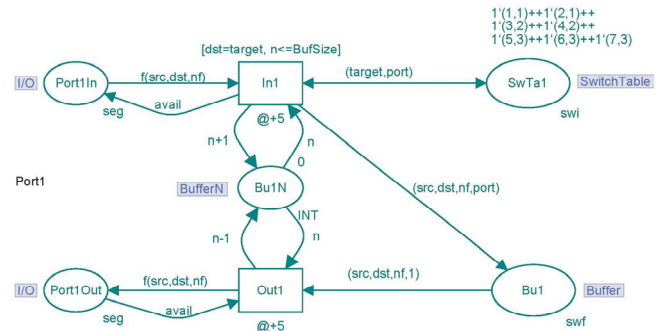


Figure 9: Model of port for limited capacity of buffer

In the comparison with the base model of switch represented in Fig. 2 we added new fusion place **BufferN** represented for Port 1 with place **Bu1N**. This place contains the number of frames in buffer. Transition **In1**, which puts frame into buffer increments the value of **BufferN** and transition **Out1**, which gets frame from buffer - decrements the value of **BufferN**. The guard of transition **In1** checks the size of **Buffer** represented with the constant **BufSize**. The models of another ports are analogous.

The results of network response time estimation under various values of switch's buffer size are represented in Table 6. We may conclude that small buffer sizes influence on response time essentially but more volumetric buffers than of 100 frames usage has no effect and does not decreases the network response time.

Table 6: Influence of buffer size on network response time

Buffer size, frames	2	10	20	100	1000	2000
Response time, MTU	803	312	308	305	304	304

DYNAMIC SWITCHING TABLE INFLUENCE

Static switching tables are used in the networks with the extreme requirements to security. But the administration of static switching table for large networks is laborious enormously. Dynamic maintenance of switching tables is usual practice for the majority of networks. In this case the bandwidth of network is decreased because the broadcast

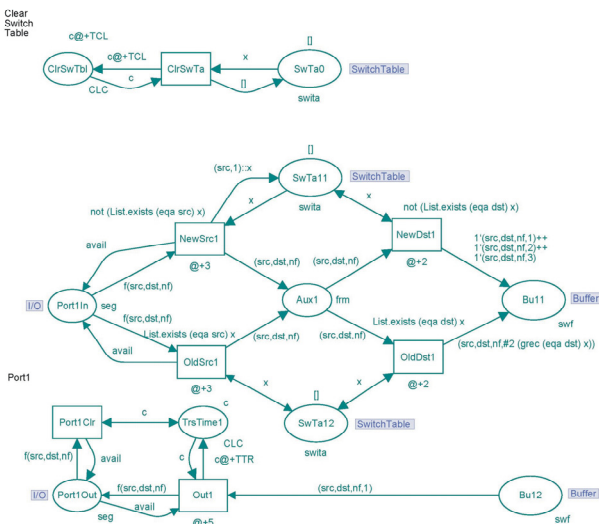
and multicast traffic but network became more flexible and self-adjusting to variable structure.

The task to model dynamic switching table is complicated enough. Let us remind the general technique of dynamic switching stipulated by standards. We pay our attention only to peculiar properties:

- 1) At arrival of frame with unknown source MAC-address switch adds new record corresponding to this address to switching table.
- 2) At arrival of frame with unknown destination MAC-address switch executes multicasting forwarding frame to all the ports. In this case it works like a hub.
- 3) Periodically switch clears its switching table.

First item provides the passive recognition of network structure. Second item allows the avoidance of special active recognition. Periodical reset of switching table provides the adjusting to current structure of network and the set of devices alive.

Implementation of dynamic switching requires not only modification of switch's model but also the total revision of the entire model. Now we have to represent the timed characteristics of frame's transmission at least for channel of segment in the direction from switch. If frame of multicasting will not be removed by destination device we have to remove it at the elapsing of transmitting time. The implementation of the absence checking for token with specific properties requires the usage of list data type (color) of CPN Tools. The model of switch's port for dynamic switching and the model's fragment for periodic cleaning of switching table are represented in Fig. 10.



```

color swita=list swi; var x:swita;
fun eqa a (rr:swi)=((#1 rr)=a);
fun grec prd [] = (0,0) | grec prd (q::r) =if prd(q) then q else grec prd r;
color CLC=unit with c timed;
    
```

Fig. 10: Model of dynamic switching

Let us consider the additional declarations used. Color **swita** is the list of **swi** and models the switching table. List structure is applied to provide the checking of absence of records in switching table with specified MAC-address. Color **CLC** represents a timer. It consists of one timed element **cl**. Function **eqa** provides the comparison of MAC-addresses. Function **grec** gets the record of switching table for which the value of function **eqa** is true.

At arrival frame in **Port1In** one of transitions **NewSrc1**, **OldSrc1** fires and puts frame to auxiliary place **Aux1**. The difference of these transitions' actions consists in that transition **NewSrc1** adds new record to switching table corresponding to source MAC-address. Standard function **List.exists** checks the existence of record with MAC-address equaling to **src**. Function **eqa** provides the comparison of address with first field of record. Then one of transitions **NewDst1**, **OldDst1** fires. The checking of address is implemented in the same way but **dst** field of frame is used. If the destination address is known (**OldDst1**) then frame is moved to buffer and the number of target port is calculated according to switching table. Function **grec** provides the usage of required record; as the number of destination port the second field of record is taken. If the destination address is unknown (**NewDst1**) then multicasting is implemented. The frame arrived is forwarded to all the ports of switch. To avoid a cross influence of modification we save the total time of input frame processing distributing the time amongst two sequential transitions.

At the transmission of output frames to **Port1Out** we set the timer in place **TrsTime1** with value **TTR**. And if the frame will not be removed by terminal device then transition **Port1Clr** removes it after elapsing of transmission time. For simulation the value of **TTR** is chosen equaling to 10 MTU that correspond to 100Mbps speed: $(1,5 \cdot 10^3 \cdot 8) / 10^8$ c.

The models of other ports are analogous. Moreover, model of switch has one common fragment representing the cleaning of switching table. Place **ClrSwTbl** contains a timer token, which starts the cleaning with the aid of transition **ClrSwTa**. Period of cleaning is determined by value of constant **TCL**. Table 7 represents estimations of network response time under various values of cleaning period **TCL**. Frequent cleaning leads to increase of response time, whereas infrequent cleaning with period more than 10^5 practically does not influent on response time. It should be noticed that at **TCL** about 1000 MTU or less the state-stable mode is not approached.

Table 7: Influence of dynamic switching on network response time

TCL, MTU	1000	2000	4000	10000	100000	200000
Response time, MTU	-	512	362	320	304	304

REAL-LIFE LAN MEASUREMENTS

The measurements of response time in the environment of the real-life network of the railway dispatcher center were implemented using sniffer of packets WinDump, which is MS Windows version of well-known Unix sniffer TCPDump. The results have been acknowledged also with the aid of SoftPerfect Network Protocol Analyzer.

WinDump is a command-line tool, which provides writing of Ethernet frames into a file together with time stamps. Then the content of the file may be displayed and analyzed. The following command line provides the writing of frames into file SavedFrames:

```
WinDump -w SavedFrames
```

For analysis of frames' transmission and calculation of response time the following command line was used:

```
WinDump -ttt -r SavedFrames
```

Option -ttt is used for automatically calculation of time interval between frames; option -r provides reading of early saved information from the file SavedFrames. An example of the obtained listing is represented in Fig. 11.

```
000252 IP 192.168.0.158.1172 > 192.168.0.130.139: P 854:917(63) ack 840 win 64957
000854 IP 192.168.0.130.139 > 192.168.0.158.1172: . 840:2300(1460) ack 917 win 64502
000141 IP 192.168.0.130.139 > 192.168.0.158.1172: . 2300:3760(1460) ack 917 win 64502
000029 IP 192.168.0.158.1172 > 192.168.0.130.139: . ack 3760 win 65535
000107 IP 192.168.0.130.139 > 192.168.0.158.1172: . 3760:5220(1460) ack 917 win 64502
000138 IP 192.168.0.130.139 > 192.168.0.158.1172: . 5220:6680(1460) ack 917 win 64502
000024 IP 192.168.0.158.1172 > 192.168.0.130.139: . ack 6680 win 65535
000114 IP 192.168.0.130.139 > 192.168.0.158.1172: . 6680:8140(1460) ack 917 win 64502
000086 IP 192.168.0.130.139 > 192.168.0.158.1172: P 8140:9095(955) ack 917 win 64502
000287 IP 192.168.0.158.1172 > 192.168.0.130.139: . ack 9095 win 65535
000606 IP 192.168.0.158.1172 > 192.168.0.130.139: P 917:980(63) ack 9095 win 65535
000729 IP 192.168.0.130.139 > 192.168.0.158.1172: . 9095:10555(1460) ack 980 win 64439
```

Fig. 11. Dump of frames

Let us consider the information of the dump. The first column contains delays between frames in microseconds, then IP-addresses and port numbers of sender and receiver are printed. After semicolon the details of packet header are printed such as start and finish numbers of transmitting bytes, the length of frame in brackets, acknowledged byte number and the length of window. In above example 192.168.0.158 is the IP-address of workstation and 192.168.0.130 is the IP-address of the server. The port number 139 corresponds to MS NetBIOS TCP service, the port number 1172 is a random port number of client software. In this example the response time for the first request is equal to 854 mcs, for second - 107 mcs, for third 114 mcs.

WinDump was started on the operating GID Ural Workstation. We calculated an average value of response times of individual requests collected on the period of observation about one shift of work, which is equal to 12 hours. Obtained real-life average NRT is equal to 2,83 ms. The error of NRT measurement via Petri net model constitutes about 6% that is good enough result, which proves the applicability of the described technique.

CONCLUSION

In the present work the technology of switched local area networks' models development was studied. The usage of colored Petri nets allows the peculiarity of interaction within the client-server systems to be taken into account. The model reflects the major features of real-life network. CSMA procedures, full-duplex mode and switching tables were modeled. 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. Moreover, the influence of such features as switch's buffers limitations and dynamic switching were studied. The technique described is aimed at real-time applications, requiring the precise estimation of timed delays before implementation.

REFERENCES

- Beaudouin-Lafon M., Mackay W.E., Jensen M. et al. 2001. "CPN Tools: A Tool for Editing and Simulating Coloured Petri Nets". In *LNCS 2031: Tools and Algorithms for the Construction and Analysis of Systems*, 574-580. (<http://www.daimi.au.dk/CPNTools>)
- Elsaadany M., Singhal T., Lui Ming. 1995. "Performance study of buffering within switches in local area networks". In *Proc. of 4th International Conference on Computer Communications and Networks*, 451-452.
- Jensen K. 1997. *Colored Petri Nets – Basic Concepts, Analysis Methods and Practical Use*. Springer-Verlag, Vol. 1-3.
- Hunt R. 1999. "Evolving Technologies for New Internet Applications". *IEEE Internet Computing*, 5, 16-26.
- Peterson J. 1981. *Petri Net Theory and the Modelling of Systems*. Prentice Hall.
- Rahul V. 2002. *LAN Switching*. OHIO.
- Zaitsev D.A. and Sleptsov A.I. 1997. "State Equations and Equivalent Transformations of Timed Petri Nets". *Cybernetics and System Analysis*, no. 5, 659-672.
- Zaitsev D.A. 2004a. "Switched LAN simulation by colored Petri nets". *Mathematics and Computers in Simulation*, 65, no. 3, 245-249.
- Zaitsev D.A. 2004b. "Invariants of timed Petri nets". *Cybernetics and Systems Analysis*, no. 2, 92-106.
- Zaitsev D.A. 2004c. "Verification of Ethernet protocols". *Proceedings of Odessa National Telecommunication Academy*, no. 1, 42-48.
- Zaitsev D.A. 2004d. "An Evaluation of Network Response Time using a Coloured Petri Net Model of Switched LAN". In *Proceedings of Fifth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, October 8-11, Aarhus, Denmark, 157-167. 14
- Zaitsev D.A. 2005. *Functional Petri Nets*. Universite Paris-Dauphine, Cahier du Lamsade 224, Avril, 62p. (<http://www.lamsade.dauphine.fr/cahiers.html>).
- Zyabirov H.S., Kuznetsov G.A., Shevelev F.A., et al. 2003. "Automated system for operative control of exploitation work GID Ural-VNIIZT". *Railway transport*, no. 2, 36-45.