Simulating E6 protocol networks using CPN Tools

Guliayev K.D, Zaitsev D.A., Litvin D.A., Radchenko E.V.

Odessa National Academy of Telecommunications named after A.S. Popov, Ukraine
(www.onat.edu.ua, www.geocities.com/zsoftua)

1. Introduction

Urgency of the research is conditioned by the necessity of new stacks of protocols search which correspond to the requests of the present time. In the conditions of the mass application of Ethernet technology in corporation networks, access networks as well as backbones of communication statements especially in view of wide application of DWDM hardware and solutions "Ethernet over DWDM", protocols IP, TCP, UDP of the most popular stack of protocols TCP/IP become superfluous and duplicate standard features of Ethernet technology.

In [1], new stack of protocols E6 was presented which practically entirely saving application level of TCP/IP annuls protocols TCP, IP themselves substituting them by the total usage of hierarchical system of E6 addresses with the length of 6 bytes situated into the field of standard MAC-address and the usage of slipping window facilities of Ethernet LLC2 for guaranteed delivery of information flows.

For the construction of E6 networks, in [1] was suggested the application of special switching-router E6 (SRE6) which use individual addresses of terminal devices for the solving of the tasks of the switching and aggregated E6 addresses for the solving of the task of the routing. Ethernet frame header contains E6 addresses of receiver and sender and stays invariable during the process of the frame delivery into the network. Moreover, SRE6 does not use other information except of the pair of E6 addresses for the delivery of the frame into the network.

For the confirmation of E6 networks advantages and possibilities of their practical realization, models of E6 networks components were developed in the form of colored Petri nets [2] in the environment of the simulating system CPN Tools [3] on the base of early developed components of IP-networks [4]. Then, in the process of concrete E6 networks modeling which are assembled out of developed components, the efficiency of the new stack E6 was confirmed.

2. Main page of the model

Model was constructed for European information backbone studied in [4] as well as for series of other concrete given networks. Main page of European information backbone model (fig. 1) directly correspond to the structural scheme of the network [4]. Components of the model are switching-routers SRE6 and terminal networks.

At the main page, there are 56 places, 18 transitions and 108 arcs:

- 7 transitions of the type *Substitution transition* are the models of SRE6 (denoted by the names of countries: France, Spain, Sweden, Finland, Germany, Germany1, Ukraine);
- 6 transitions of the type *Substitution transition* are the models of terminal networks (denoted as: Terminal1..Termainal6);
- 2 places of the type *Fusion* are the counter of the total number of received and correctly received packets correspondingly.

Moreover, places with the names of the form **R_jP_kIn, R_jP_kOut** correspond to the points of input and output links or device ports. Transitions with the names **Receive, Transmit** correspond to communication links (channels).

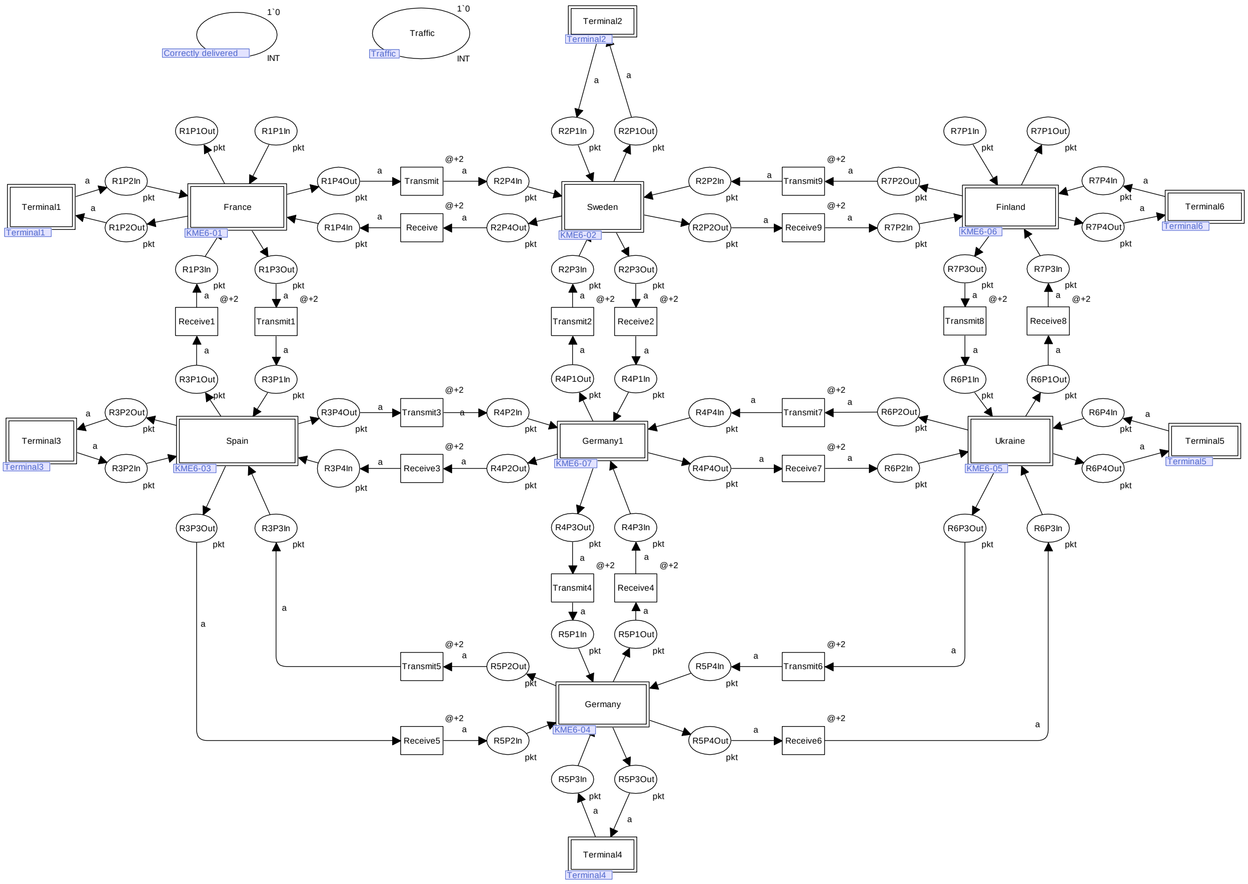

Fig. 1. Main page of the model

3. Model of switching-router E6 (SRE6)

Modeled SRE6 is a “store and forward” device with the routing table key field corresponding to the address of the destination E6 network and such other fields as the network mask, the destination port, the metric. At the time being SRE6 model uses static routing table but the construction of the models using various dynamic routing protocols is planned.

Model of SRE6 is assembled of components **Interface(k)** modeling devices interface.

Model of SRE6 (fig. 2) consists of 14 places, 4 transitions and 20 arcs:

- 8 places of the type *Port* with the names **1in, 1out, 2in, 2out** etc correspond to SR ports and fused with places **R$_j$P$_k$In, R$_j$P$_k$Out** at the main page correspondingly;
- 4 transitions of the type *Substitution Transition* with the names **Interface(k)**, k=[1..4] are the models of SRE6 interfaces;
- 2 places of the type *Socket* with the names **RT** and **RAM** are the models of the routing table and operative memory SRE6 correspondingly;
- 4 places of the type *Socket* with the names **1, 2, 3, 4** contain tokens with numbers of the corresponding interfaces.

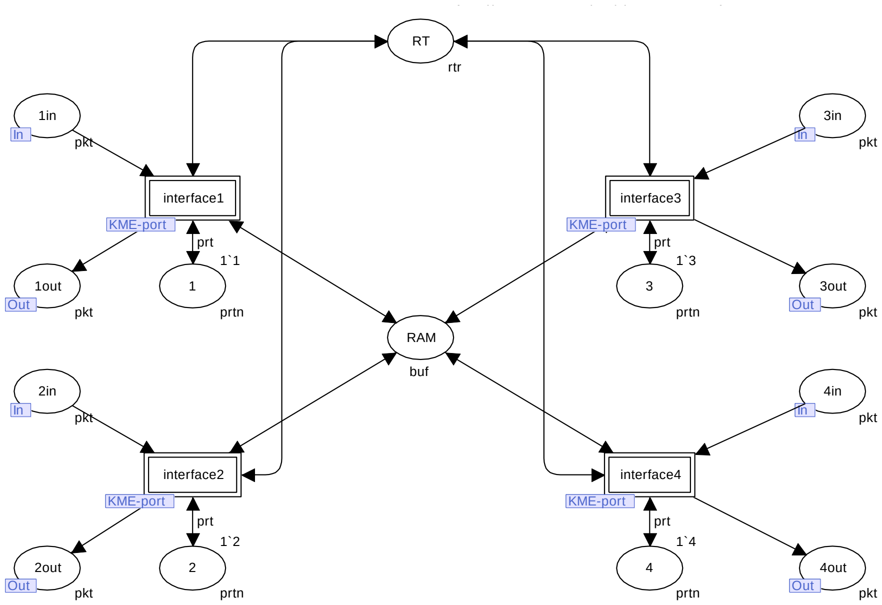

Fig. 2. Model of SRE6

Tokens into the place **RT** correspond to the records of the routing table. For the following example of the token 1`{nw=((15,25,212,225,128,0),33),port=2,metric=0}:

- 1` - number of the tokens with the following content,
- nw=((15,25,212,225,128,0),33) - network address and mask (number of bits),
- port=2 - destination port number,
- metric=0 - metric of the route.

Thus, the network 15.25.212.225.128.0/33 is available via port 2 with the metric 0, so the network is directly attached to the port (interface) 2.

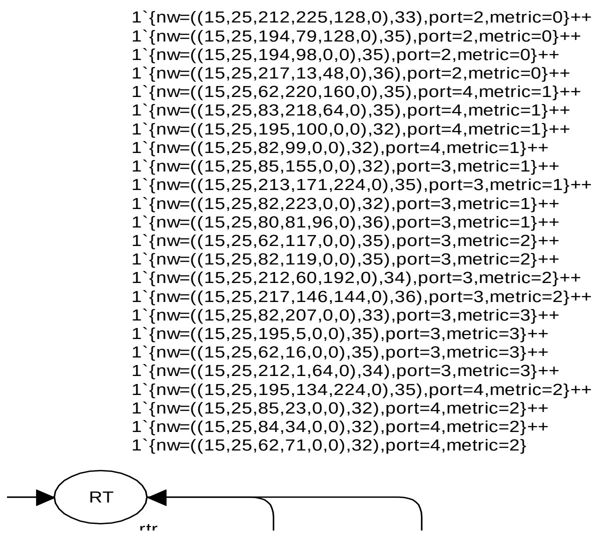

Fig. 3. Routing table of SRE6

4. Model of SRE6 interface

Model of SRE6 interface is represented in Fig. 4. All the interfaces are mapped to the same typical model with the corresponding number of the interface.

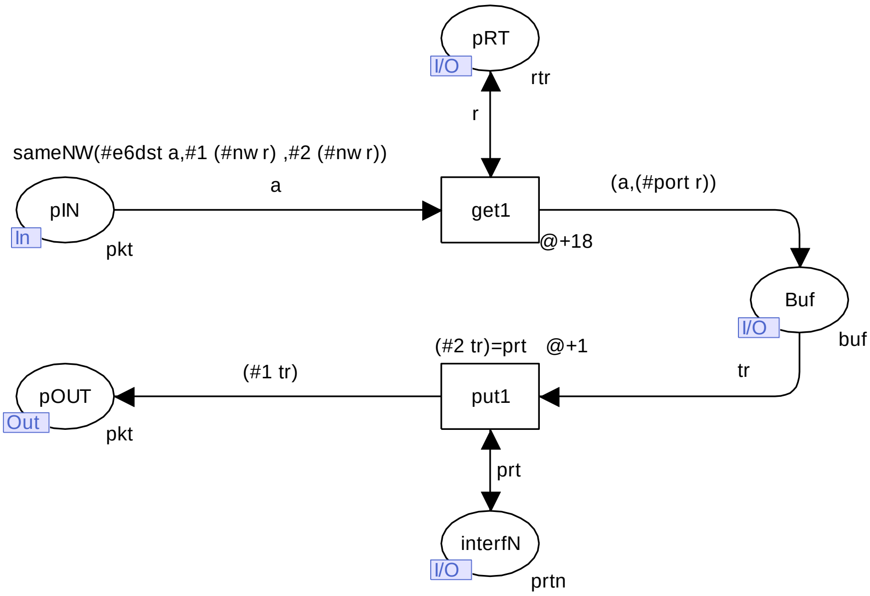

Fig. 4. Model of SRE6 interface (port)

All the places of the model are the contact ones for the connection with the upper level:

- **pIN, pOUT** – onput and output ports correspondingly for the connection with **1In** and **1Out** (as well as **2In**, **2Out** ect) correspondingly;
- **Buf** –RAM which corresponds to the internal buffer of frames;
- **pRT** – corresponds to the routing of SRE6;
- **interfN** – gives the number of SRE6 interface.

Let us consider the function of **get1** transition: at firing it takes token corresponding to received packet from place **pIN**, checks the routing table **RT** and putting the token into place **Buf** adds the tag with the number of the destination port. Special recursive function **SameNW** realizes the search into the routing table.

The colors, variables and functions of the model have the following description:

```
colset e6=product INT*INT*INT*INT*INT*INT timed;
var i,c,k:INT;
colset mask=INT;
colset nwt=product e6*mask timed;
var q:nwt;
colset b=INT timed;
colset pkt=record e6src:e6*e6dst:e6*data:b timed;
colset rtr=record nw:nwt *port:INT*metric:INT  timed ;
var r:rtr;
var p,src,dst:e6;
var a:pkt ;
colset buf=product pkt*INT;
fun pow(x,0)=1 | pow(x,y)=pow(x,y-1)*x;
var tr:buf;
var v:INT;
fun sameNW(a1:e6,a2:e6,m:INT)
fun gene6(q:nwt)
val numpkt=1000;
val DTsrc=50;
val DTdst=200;
colset prtn=INT;
var prt : prtn;
```

5. Model of the terminal network

Model of the terminal network (fig. 5) consists of the generator and the receiver of the packets. The receiver is shown in the upper part of the model, the generator – in the lower.

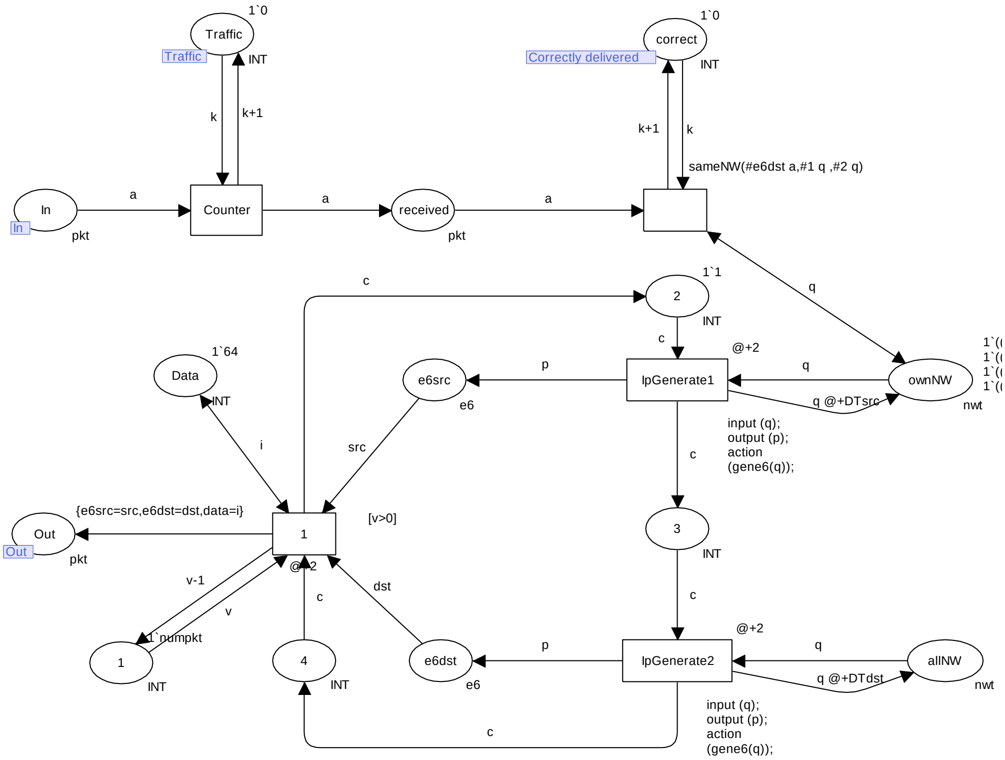

Fig. 5. Model of the terminal network

The generator of the packets consists of 10 places, 3 transitions and 18 arcs where place **ownNW** contains tokens with addresses of the own networks, place **allNW** contains tokens with addresses of the all networks. Place with the names **2, 3, 4** are used for sequential firing of transitions at the generating of the packets. Place **Data** gives the content of the packets. Place **1** contains the total number of the packets which will be generated. Place **e6src** contains token with the current sender address; place **e6dst** contains token with the current sender address. Place **Out** contains the generated packet. At the firing of the transition **IpGenerate1** an arbitrary address is chosen from the table of the own networks addresses **ownNW**; at the firing of the transition **IpGenerate2** an arbitrary address is chosen from the table of the all networks addresses **ownNW.** Moreover, the recursive function **gene6(q)** creates an arbitrary host address into the frames of the chosen network.

4. Debugging and analysis of the model

For the debugging of the model the step-by-step tracing was executed; it was shown that the model works correctly. For the confirmation of E6 networks efficiency the numbers of the delivered and the correctly delivered packets were counted directly in the process of simulation.

For the counting of the delivered and correctly delivered packets the following places **Traffic** and **Corectly** were allocated into the main page which are fused with the places **Traffic** and **correct** of the terminal networks models. For each of the terminal network was put the limit of the generated packets numbers equaling to 1000. Thus, it was generated 6000 by the all terminal networks. So, 6000 packets must be received by the terminal networks.

As the result of simulation model made 73806 steps before it stopped into the final state (no one transition is firable). The values into the places **Traffic** and **Corectly** are equal to 6000 which corresponds to the correct work of the addressing scheme E6 and the delivery algorithms. The results were checked with huge number of packets on protracted intervals of time as well.

5. Conclusions

The type components for E6 network models construction were developed: model of switching-router E6, model of terminal network. Formats of the frame header and the routing table records were described.

The developed components were used for the construction of the models for the series of real-life networks, for instance, European information backbone model. As the result of simulation in the environment of CPN Tools the efficiency of E6 networks was confirmed.

References

1. Vorobiyenko P.P., Zaitsev D.A., Nechiporuk O.L. World-Wide Network Ethernet? // Zviazok (Communications), no. 5, 2007. - p. 14-19.
2. Jensen K. Colored Petri Nets – Basic Concepts, Analysis Methods and Practical Use.- Vol. 1-3, Springer-Verlag, 1997.
3. 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 (http://www.daimi.au.dk/CPNTools).
4. Zaitsev D.A., Sakun A.L. Studying efficacy of MPLS technology using colored Petri nets // Zviazok (Communications). – 2006. – Vol. 65, no 5. – p. 49-55.

Published: Proc. of International Conference on IT Promotion in Asia, August 22-26, 2008, Tashkent (Uzbekistan), P. 203-208.