

УДК 519.74, 681.51

## **Основы построения параметрических моделей Петри коммутируемых сетей**

Д.А. Зайцев

Одесская национальная  
академия связи

Т.Р. Шмелёва

Одесская дирекция  
одесской железной дороги

### **Abstract**

*Zaitsev D.A., Shmeleva T.R. Basics of construction of parametric Petri net models of switched networks. The technique of construction of parametric models of switched telecommunication networks in the form of colored Petri nets was presented. The standard model is invariant with respect to network's topology and consists of a fixed number of elements. The components of the modeled network are workstations, servers, switches and hubs.*

### **Введение**

Раскрашенные сети Петри [1] представляют собой универсальный графический язык для спецификации и анализа асинхронных параллельных систем и процессов. А промышленные моделирующие системы, такие как CPN Tools, Design/CPN, Miss-RdP [2], создают удобную среду для реализации технологий управляемого моделью (model driven) проектирования сложных систем.

При проектировании телекоммуникационных сетей распределенных систем технологического управления возникает задача оперативной оценки временных характеристик. Для этих целей предложено использовать модели коммутируемых сетей [3], собранные из компонентов, представляющих рабочие станции, серверы, коммутаторы и концентраторы [4,5]. Для измерения характеристик моделей разработаны специальные измерительные фрагменты сетей Петри [6,7]. При этом каждая новая структурная схема моделируемой сети требует повторного ввода главной страницы модели, что существенно снижает оперативность выполнения оценок. В [8] представлена параметрическая модель одноуровневой коммутируемой сети топологии "звезда". Однако крупные телекоммуникационные сети, как правило, имеют более сложные топологии, в большинстве случаев древовидной структуры.

Целью настоящей работы является построение типовой параметрической модели коммутируемой сети с произвольным количеством уровней, а

также формулирование основных принципов организации параметрических моделей.

## **Коммутируемые сети**

Построение типовой модели выполним для коммутируемой Ethernet [3]. Основным элементом сети является коммутатор, направляющий прибывший фрейм в порт, к которому подключено устройство назначения. Для адресации устройств используются 6-байтные MAC-адреса (Media Access Control).

Информация о подключении устройств к портам задаётся таблицей коммутации, которая может быть либо статической, либо динамической. Влияние процессов динамического ведения таблиц коммутации на пропускную способность сети изучено в [5], поэтому в настоящей работе рассмотрим статические таблицы.

Среди терминальных устройств по особенностям формирования трафика будем различать рабочие станции и серверы. В качестве протокола взаимодействия рассмотрим двухфазный процесс "запрос-ответ". Интервалы между запросами, времена их исполнения, а также объёмы передаваемой информации будем оценивать случайными равномерно распределёнными функциями. Методика расчёта временных и емкостных параметров модели для заданной спецификации программного обеспечения и оборудования описана в [6,7].

Пример двухуровневой коммутируемой сети представлен на Рис. 1; сеть содержит три коммутатора (SWI), шесть рабочих станций (WS) и два сервера (S).

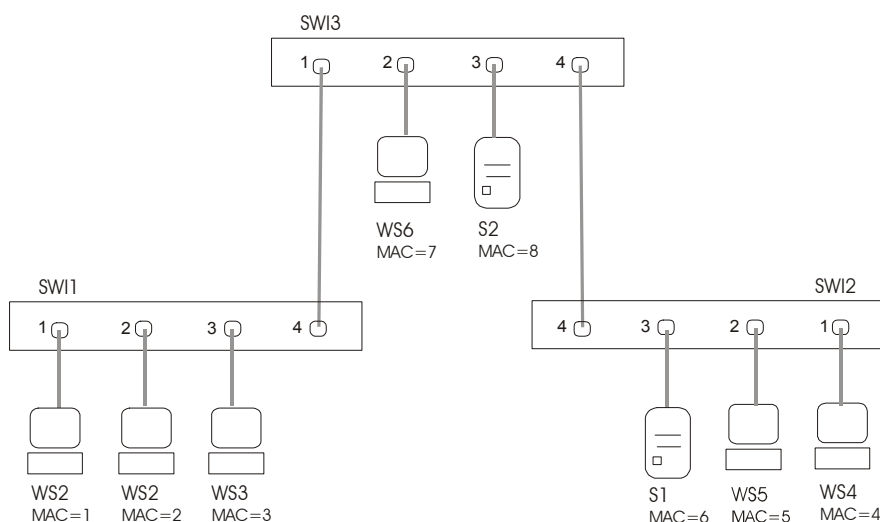


Рис. 1. Коммутируемая сеть диспетчерского центра железной дороги

## Типовая параметрическая модель

Типовая параметрическая модель коммутируемой Ethernet представлена на Рис. 2. Описания типов данных (цветов), переменных и функций приведены на Рис. 3.

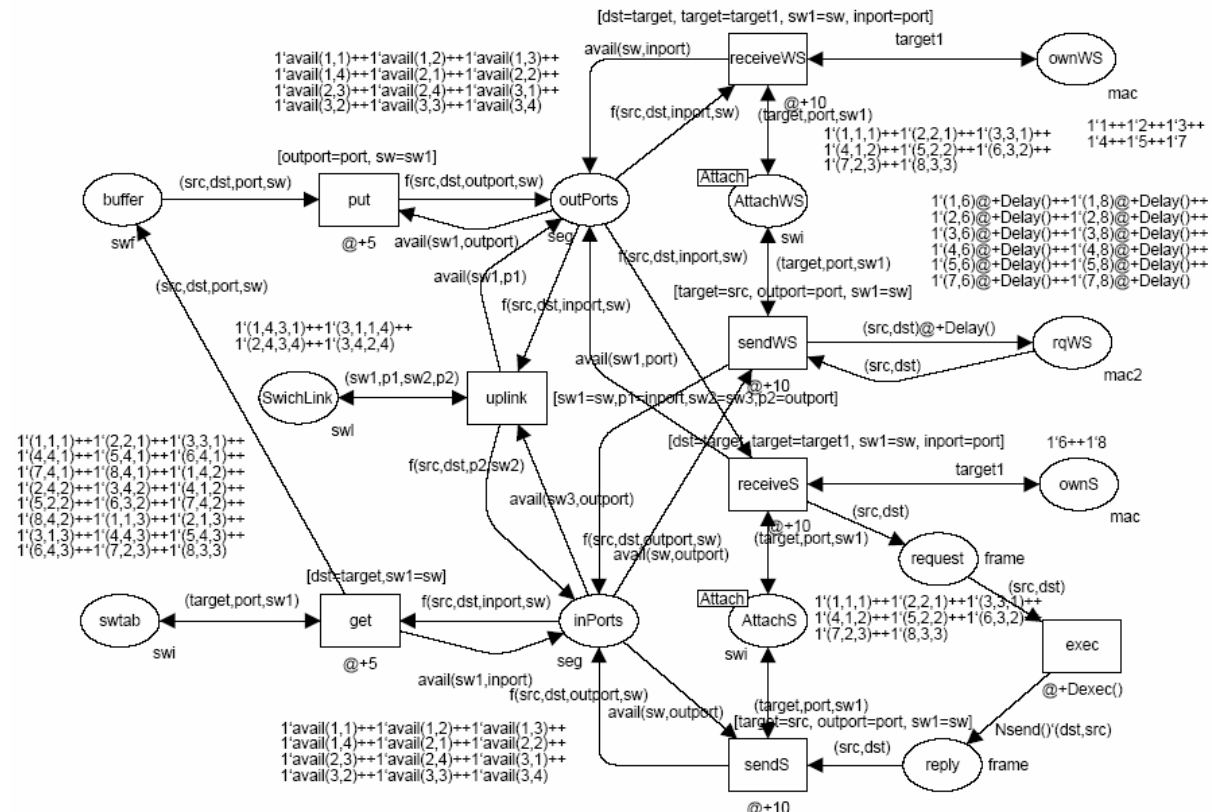


Рис. 2. Параметрическая модель коммутируемой сети

Элементы серверов содержат суффикс  $S$ , элементы рабочих станций – суффикс  $WS$ , элементы коммутаторов не имеют суффикса. Все сегменты локальной сети представлены парой позиций  $inPorts$ ,  $outPorts$ . Для моделирования передачи фреймов от терминальных устройств к коммутатору используется позиция  $inPorts$ ; для моделирования передачи фреймов от коммутаторов к терминальным устройствам – позиция  $outPorts$ . Исключения составляют связи между коммутаторами. Для представления таких связей использован переход  $uplink$ .

При моделировании трафика мы абстрагируемся от содержимого фреймов, рассматривая только адреса отправителя и получателя. Ключевым средством параметризации является абстрактный тэг, ассоциированный с объектами модели. Тэг содержит номер коммутатора и номер порта, идентифицируя, таким образом, сегмент сети. Каждый из сегментов может быть либо свободен, что моделируется специальной меткой  $avail$ , либо занят передачей фрейма  $f$ . Перед передачей фрейма каждое из устройств ожидает освобождения необходимого сегмента, проверяя наличие соответ-

ствующей метки *avail*. При передаче метка *avail* извлекается и замещается фреймом. Приём фрейма представляет собой обратный процесс. Как правило, принимающее устройство проверяет адрес назначения *dst*, извлекает фрейм и замещает его соответствующей меткой *avail*.

Топология сети представлена маркировкой позиций *Attach* и *SwitchLink*. Совмещённая позиция *Attach* (*AttachWS*, *AttachS*) содержит информацию о подключении терминальных устройств, указывая для каждого MAC-адреса номер коммутатора и номер порта. Заметим, что такой способ представления позволяет моделировать несколько терминальных устройств, подключенных к одному порту коммутатора, например, посредством концентратора. Позиция *SwitchLink* содержит описания связей между коммутаторами, указывая номер коммутатора и номер порта для каждого из двух концов линии связи.

```
color mac=INT;
color portnum=INT;
color swch=INT;
color mac2=product mac*mac timed;
color nfrm=INT;
color sfrm=product nfrm*INT timed;
color frm=product mac*mac*portnum*swch timed;
color nseg=product swch*portnum;
color seg=union f:frm+avail:nseg timed;
color swi=product mac*portnum*swch;
color swf=product mac*mac*portnum*swch timed;
color frame=product mac*nfrm timed;
color swl=product swch*portnum*swch*portnum;
var src,dst,target,target1: mac;
var sw,sw1,sw2,sw3:swch;
var port, inport, outport: portnum;
var p1, p2: portnum;
color dex= int with 100..200;
fun Dexec()=dex.ran();
color nse = int with 10..20;
fun Nsend()=nse.ran();
color Delta= int with 1000..2000;
fun Delay()=Delta.ran();
```

Рис. 3. Описания типов, переменных и функций

Таблицы коммутации всех коммутаторов представлены маркировкой позиции *swtab*. Отметим, что дублирование информации связано с тем, что в общем случае каждому коммутатору должен быть известен MAC-адрес каждого из устройств сети. Позиция *buffer* моделирует внутренний буфер коммутатора.

Позиция *rqWS* содержит таблицу запросов рабочих станций к серверам, заданную указанием пар MAC-адресов. После выполнения запроса пара адресов блокируется на интервал времени, заданный функцией *Delay()*. Позиция *ownWS* перечисляет MAC-адреса рабочих станций. Сервер размещает полученный запрос в позиции *request*, выполняет его в те-

чение времени  $Dexec()$  и формирует  $Nsend()$  фреймов ответа в позиции  $reply$ , которые затем передаются в обратном направлении к запросившей рабочей станции, что моделируется изменением порядка MAC-адресов отправителя и получателя в заголовке фреймов.

Для измерения характеристик функционирования локальной сети построенная модель может быть дополнена измерительными фрагментами, описанными в [7,8].

### Имитация динамики модели

Для отладки модели система CPN Tools предоставляет режимы имитации динамики сети с визуальным отображением всех событий, происходящих в текущий момент модельного времени. Обеспечивается пошаговая имитация либо имитация заданного количества шагов. Под шагом подразумевается момент модельного времени, для которого имеется непустой список событий.

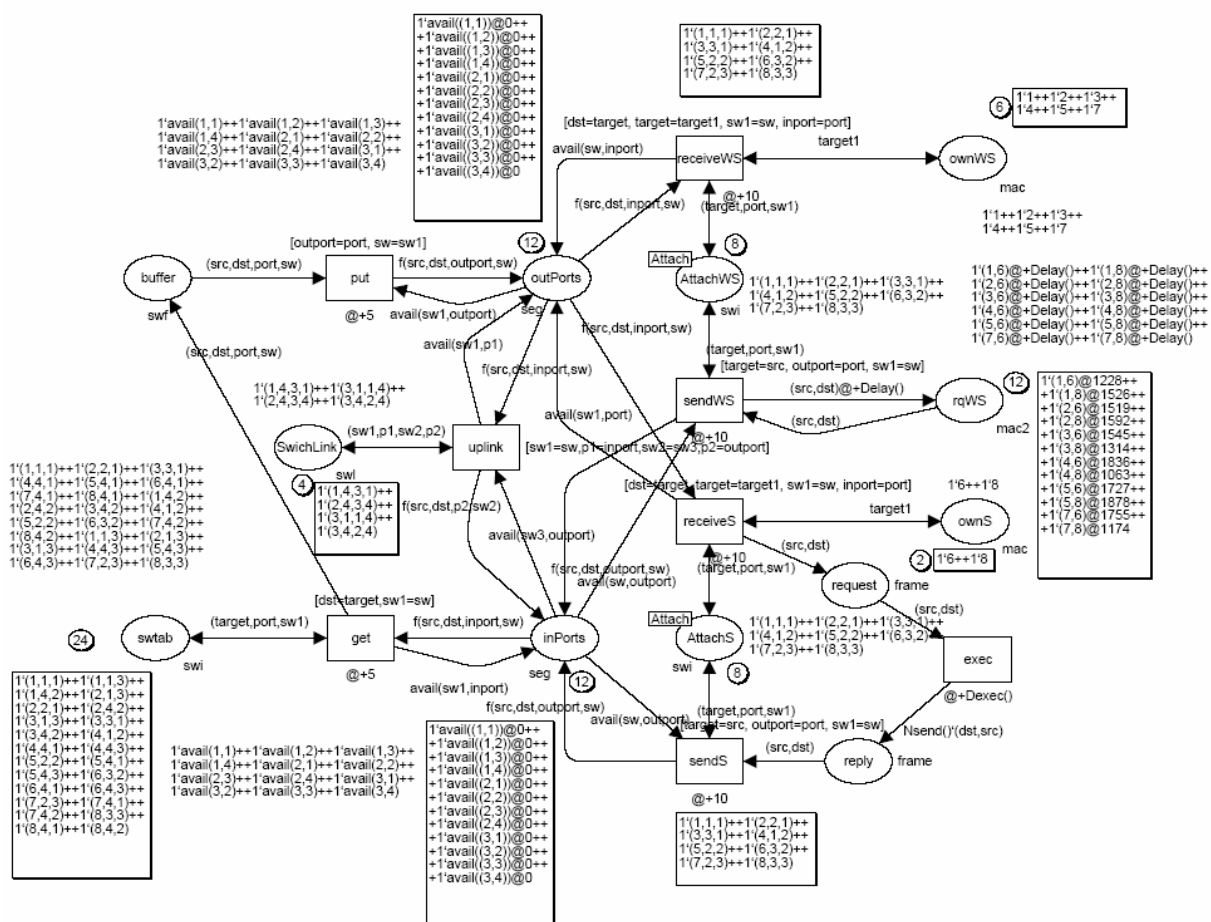


Рис. 4. Начальное состояние модели

Рассмотрим процесс продвижения фрейма запроса от одной из рабочих станций к серверу, а также доставки одного из фреймов ответа сервера

в обратном направлении. На Рис. 4 представлено начальное состояние модели. Отметим, что в соответствии с маркировкой позиции  $rqWS$ , сгенерированной с помощью случайной функции  $Delay()$ , начальный момент модельного времени равен 1063.

Для компактного представления процесса имитации используем трассировочную таблицу (Табл. 1), в которой отображены состояния модели на каждом шаге. Заметим, что кроме указанных в таблице позиций изменяется также маркировка позиции  $rqWS$ ; так как эти изменения связаны только с временными параметрами, рассмотрим состояние указанной позиции после выполнения 32 шагов (Рис. 5).

Таблица 1. Трассировка динамики модели

Шаг	Время	Маркировка позиций				
		InPorts	OutPorts	Buffer	request	reply
0	1063	-	-			
1	1073	$1^f(4,8,1,2)@1073$	-			
2	1078	$1^avail((2,1))@1078$	-	$1^f(4,8,4,2)@1078$		
3	1083	-	$1^f(4,8,4,2)@1083$			
4	1083	$1^f(4,8,4,3)@1083$	$1^avail((2,4))@1083$			
5	1088	$1^avail((3,4))@1088$	-	$1^f(4,8,3,3)@1088$		
6	1093	-	$1^f(4,8,3,3)@1093$			
7	1103	-	$1^avail((3,3))@1103$		$1^f(4,8)@1103$	
8	1174	-	-			$20^f(8,4)@1244$
9	1184	$1^f(7,8,2,3)@1184$	-			-
10	1189	$1^avail((3,2))@1189$	-	$1^f(7,8,3,3)@1189$		-
11	1194	-	$1^f(7,8,3,3)@1194$			-
12	1204	-	$1^avail((3,3))@1204$		$1^f(7,8)@1204$	-
13	1228	-	-			$20^f(8,4)@1244++$ $16^f(8,7)@1362$
14	1238	$1^f(1,6,1,1)@1238$	-			-
15	1243	$1^avail((1,1))@1243$	-	$1^f(1,6,4,1)@1243$		-
16	1244	-	$1^f(1,6,4,1)@1248$			-
17	1248	$1^f(8,4,3,3)@1254$	$1^f(1,6,4,1)@1248$			$19^f(8,4)@1244++$ $16^f(8,7)@1362$
18	1248	$1^f(1,6,1,3)@1248++$ $1^f(8,4,3,3)@1254$	$1^f(1,6,4,1)@1248$			-
19	1253	$1^f(8,4,3,3)@1254++$ $1^avail((3,1))@1253$	$1^avail((1,4))@1258$	$1^f(1,6,4,3)@1253$		-
20	1254	$1^f(8,4,3,3)@1254$	$1^f(1,6,4,3)@1258$			-
21	1258	$1^avail((3,3))@1259$	$1^f(1,6,4,3)@1258$	$1^f(8,4,4,3)@1259$		-
22	1258	$1^f(1,6,4,2)@1258$	$1^avail((3,4))@1258$	$1^f(8,4,4,3)@1259$		-
23	1259	$1^avail((2,4))@1263$	-	$1^f(1,6,3,2)@1263++$ $1^f(8,4,4,3)@1259$		-
24	1259	$1^f(8,4,3,3)@1269$	-	$1^f(1,6,3,2)@1263++$ $1^f(8,4,4,3)@1259$		$18^f(8,4)@1244++$ $16^f(8,7)@1362$
25	1263	$1^f(8,4,3,3)@1269$	$1^f(8,4,4,3)@1264$	$1^f(1,6,3,2)@1263$		-
26	1264	$1^f(8,4,3,3)@1269$	$1^f(1,6,3,2)@1268++$ $1^f(8,4,4,3)@1264$			-
27	1264	$1^f(8,4,3,3)@1269++$ $1^f(8,4,4,2)@1264$	$1^f(1,6,3,2)@1268++$ $1^avail((3,4))@1264$			-
28	1268	$1^f(8,4,3,3)@1269++$ $1^f(8,4,4,2)@1264$	$1^f(1,6,3,2)@1268$	$1^f(8,4,1,2)@1269$		-
29	1269	$1^f(8,4,3,3)@1269++$ $1^f(8,4,4,2)@1264$	$1^avail((2,3))@1278$	$1^f(8,4,1,2)@1269$	$1^f(1,6)@1278$	-
30	1269	$1^f(8,4,3,3)@1269++$ $1^avail((2,4))@1269$	$1^f(8,4,1,2)@1274$		$1^f(1,6)@1278$	-
31	1274	$1^avail((3,3))@1274$	$1^f(8,4,1,2)@1274$	$1^f(8,4,4,3)@1274$	$1^f(1,6)@1278$	-
32	1274	-	$1^avail((2,1))@1284$	$1^f(8,4,4,3)@1274$	$1^f(1,6)@1278$	-

Запрос рабочей станции WS4, отправленный на шаге 1 серверу S2, доставлен серверу на 7-м шаге и выполнен сервером на 8-м шаге, в результате чего сгенерированы 20 фреймов ответа. Начиная с 17-го шага выполняется доставка фреймов ответа в обратном направлении. На 32-м шаге первый из фреймов ответа возвращается к рабочей станции WS4 и поглощается переходом *receiveWS*.

Следует отметить, что Табл. 1 отображает также другие происходящие в модели события, например, выполнение запроса рабочей станции WS6, отправленного на 9-м шаге.

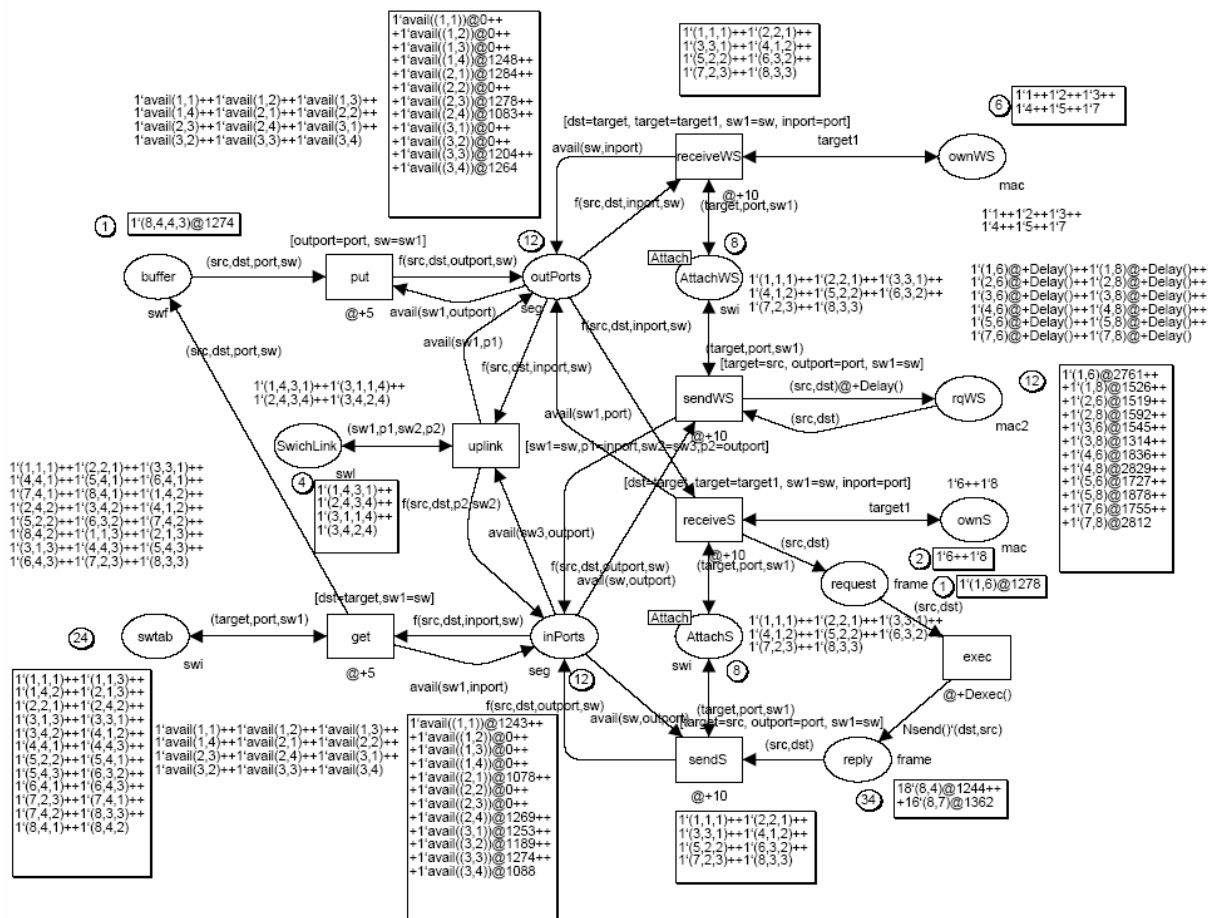


Рис. 5. Состояние модели на шаге 32

Для измерения характеристик функционирования коммутируемых сетей целесообразно применять режим скоростного моделирования, обеспечивающий выполнение заданного числа шагов без визуализации динамики модели. При этом отображается лишь конечное состояние модели. Скоростная имитация позволяет накапливать статистическую информацию на длительных интервалах реального времени моделируемых сетей, обеспечивающих наблюдение стационарного режима функционирования модели. Вопросы измерения характеристик моделей изучены в [7].

## **Принципы параметризации моделей**

Построение параметрических моделей возможно для сетей, собранных из фиксированного множества типовых элементов. Каждый типовой элемент представляется в единственном экземпляре. Вводятся специальные позиции, маркировка которых описывает топологию сети. Динамические элементы, представленные фишками, дополняются специальными тэгами, описывающими местонахождение фишки по отношению к топологической информации.

## **Заключение**

Таким образом, в настоящей работе представлена методика параметризации моделей телекоммуникационных сетей, построенных на основе фиксированного множества типовых элементов. Методика изучена на примере построения параметрической модели Петри коммутируемой Ethernet. Преимуществом параметрических моделей является фиксированная структура для произвольной заданной конфигурации телекоммуникационной сети.

## **Литература**

1. Jensen K. Colored Petri Nets – Basic Concepts, Analysis Methods and Practical Use.- Vol. 1-3, Springer-Verlag, 1997.
2. 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>).
3. Breyer R., Riley S. Switched, fast, and gigabit Ethernet, MacMillan Technical Pub., 1999, 618 p.
4. Zaitsev D.A. Switched LAN simulation by colored Petri nets. Mathematics and Computers in Simulation, vol. 65, no. 3, 2004, 245-249.
5. Зайцев Д.А., Шмелёва Т.Р. Моделирование коммутируемой локальной сети раскрашенными сетями Петри // Зв'язок, № 2(46), 2004, с. 56-60.
6. Zaitsev D.A. An Evaluation of Network Response Time using a Coloured Petri Net Model of Switched LAN // Proceedings of Fifth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, October 8-11, 2005, Aarhus, Denmark, p. 157-167.
7. Зайцев Д.А. Измерительные фрагменты в моделях Петри телекоммуникационных сетей // Зв'язок, № 2(54), 2005, с. 65-71.
8. Зайцев Д.А., Шмелёва Т.Р. Параметрическая модель Петри одноуровневой коммутируемой сети // Труды Одесской национальной академии связи им. А.С.Попова, № 1, 2005.