

ПАРАМЕТРИЧЕСКАЯ МОДЕЛЬ ПЕТРИ ОДНОУРОВНЕВОЙ КОММУТИРУЕМОЙ СЕТИ

PARAMETRIC MODEL PETRI OF SINGLE-LEVEL SWITCHED NETWORK

Аннотация. Построена модель одноуровневой коммутируемой сети инвариантная по отношению к топологии. Модель представлена раскрашенной сетью Петри. Основными параметрами модели являются упакованная матрица топологии сети, таблица запросов рабочих станций к серверам, времена передачи фреймов и обработки запросов. Для отладки модели и сбора статистической информации использована система CPN Tools.

Summary. A model of a single-level switched network invariant with respect to topology was constructed. The model is represented with colored Petri net. Basic parameters of the model are packed matrix of topology, table of workstations' requests to servers, times of frames' transmission and requests processing. For model debug and collection of statistical information CPN Tools system was applied.

Перспективной технологией разработки сложных телекоммуникационных систем является, так называемое управляемое моделью проектирование [1]. Выбор формального языка, удобного как для представления модели, так и для описания проекта, позволяет организовать процесс проектирования как последовательную детализацию первоначальной абстрактной модели. Раскрашенные временные сети Петри [1-3] представляют собой пример такого языка, а система CPN Tools [4] – пример моделирующей системы, реализующей язык сетей Петри. Моделирование является важным этапом построения компьютерных сетей, позволяющим выполнить точную оценку параметров их функционирования. Такая оценка особенно важна для сетей, применяемых в управлении процессами реального времени. Таким образом, построение адекватных моделей Петри компьютерных сетей представляет собой важную научную проблему.

Модели Петри коммутируемых сетей [5] изучены в [6-8]. Абстрактная модель [6] дополнена такими важными особенностями реальных сетей как ведение динамической таблицы коммутации [7], полнодуплексный режим работы [8], измерительные фрагменты [8]. Однако технология [6-8] базирующаяся на использовании множества типовых элементов локальной вычислительной сети (ЛВС), представляющих модели коммутатора, сервера и рабочей станции, требует перекомпоновки модели для каждой новой моделируемой ЛВС. При этом необходимо заново ввести основную страницу модели, соответствующую структурной схеме сети, а также модифицировать модель коммутатора для конкретного количества портов. Указанные операции являются довольно трудоёмкими. Поэтому построение модели, инвариантной по отношению к топологии сети является перспективной задачей, требующей решения.

Целью настоящей работы является построение параметрической модели Петри одноуровневой коммутируемой сети инвариантной по отношению к топологии. Такая модель может быть успешно применена для исследования в кратчайшие сроки параметров функционирования проектируемых локальных сетей.

1. Принципы параметризации модели. В качестве объекта для моделирования выбрана одноуровневая коммутируемая сеть Ethernet [5]. Одноуровневая коммутируемая сеть представляет собой классическую реализацию топологии "звезда" и является характерной для большого числа малых и средних ЛВС. Центральным элементом такой сети является коммутатор либо стек коммутаторов, логически организованный как множество

портов. К каждому порту может быть непосредственно присоединено терминальное устройство, такое как рабочая станция или сервер. Кроме того, в сетях, которые не требуют значительной скорости передачи информации, к порту может быть присоединено несколько терминальных устройств, объединённых с помощью концентратора. Таким образом, топология одноуровневой коммутируемой сети полностью задаётся перечнем MAC-адресов (Media Access Control) терминальных устройств с указанием номера порта, к которому они присоединены. Пример одноуровневой коммутируемой сети диспетчерского центра управления перевозками на железной дороге [8] приведен на рис. 1 а). Коммутаторы обозначены как SWI (switch), концентраторы – как HUB, рабочие станции – WS (workstation), серверы – S (server).

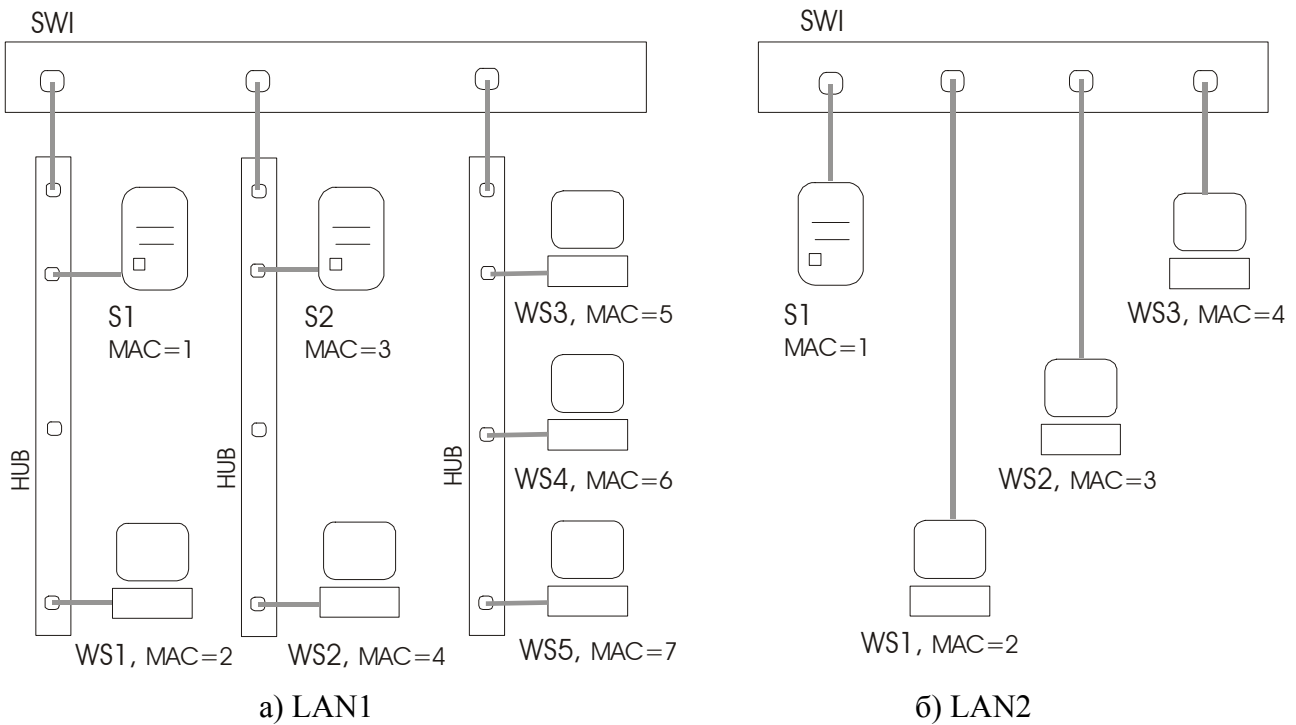


Рисунок 1 – Структурные схемы одноуровневых коммутируемых сетей

Рассмотрим модель коммутируемой Ethernet, представленную в [8]. Модель включает четыре основных страницы: рабочая станция, сервер, коммутатор, топология сети. При этом для каждой новой ЛВС необходимо перекомпоновать модель коммутатора, изображая необходимое число портов. Кроме того, необходимо изобразить заново страницу топологии сети, которая содержит необходимое число серверов и рабочих станций. Указанный процесс является достаточно трудоёмким, так как требует выполнения операций подстановки перехода с отображением портов и сокетов модели [5,8]. По крайней мере, для архитектуры "звезда" можно предложить способ построения модели инвариантной по отношению к топологии ЛВС. При этом саму топологию удобно представить массивом пар (MAC-адрес, порт). Далее будем использовать термины номер порта и номер сегмента ЛВС как синонимы.

Позиции Own модели, представленной в [8] и содержащие MAC-адрес рабочей станции или сервера, при размещении в них нескольких адресов позволяют моделировать одновременную работу нескольких терминальных устройств. Кроме того, каждый из портов коммутатора представлен одним и тем же базовым фрагментом. Таким образом, при обеспечении различия фреймов, поступающих из разных портов, можно представить коммутатор одним базовым фрагментом порта. Аналогичные рассуждения справедливы и для пар позиций, моделирующих сегменты сети. Основным решением, позволяющим осуществить параметризацию модели, является включение специального абстрактного тэга, задающего номер сегмента, по которому передаётся информация, в модельное представление фрейма. Это позволяет моделировать все сегменты локальной сети одной парой позиций,

различая номер сегмента с помощью введенного тэга. Основными параметрами модели являются: топология сети, перечни MAC-адресов рабочих станций и серверов, таблица запросов рабочих станций.

2. Описание параметрической модели. Для параметризации выбрана модель коммутируемой Ethernet, описанная в [8]. Напомним, что модель реализует простейшую схему взаимодействия клиент-сервер: каждая рабочая станция посылает периодические запросы к серверам, состоящие из одного фрейма; сервер исполняет запрос и посылает ответ, состоящий из нескольких фреймов. Периодичность запросов, время их исполнения и количество фреймов ответа являются случайными равномерно распределёнными величинами. Принципы масштабирования модельного времени на основе характеристик оборудования и программного обеспечения описаны в [8].

Параметрическая модель ЛВС представлена на рис. 2; начальная маркировка соответствует ЛВС LAN1, структурная схема которой изображена на рис. 1 а). В модели можно выделить четыре основных части: модель рабочих станций, модель серверов, модель коммутатора, модель сегментов сети. Для различения частей модели использован суффикс наименования их элементов. Позициям и переходам рабочих станций приписан суффикс WS (например, sendWS); серверов – суффикс S (например, sendS). Наименования элементов коммутатора не содержат суффикса. Сегменты ЛВС рассматриваются как порты коммутатора: объединенный выходной порт outPorts, объединенный входной порт inPorts. Отметим, что наименования портов inPorts (входные), outPorts (выходные) выбраны по отношению к коммутатору; для рабочих станций и серверов назначение портов противоположно.

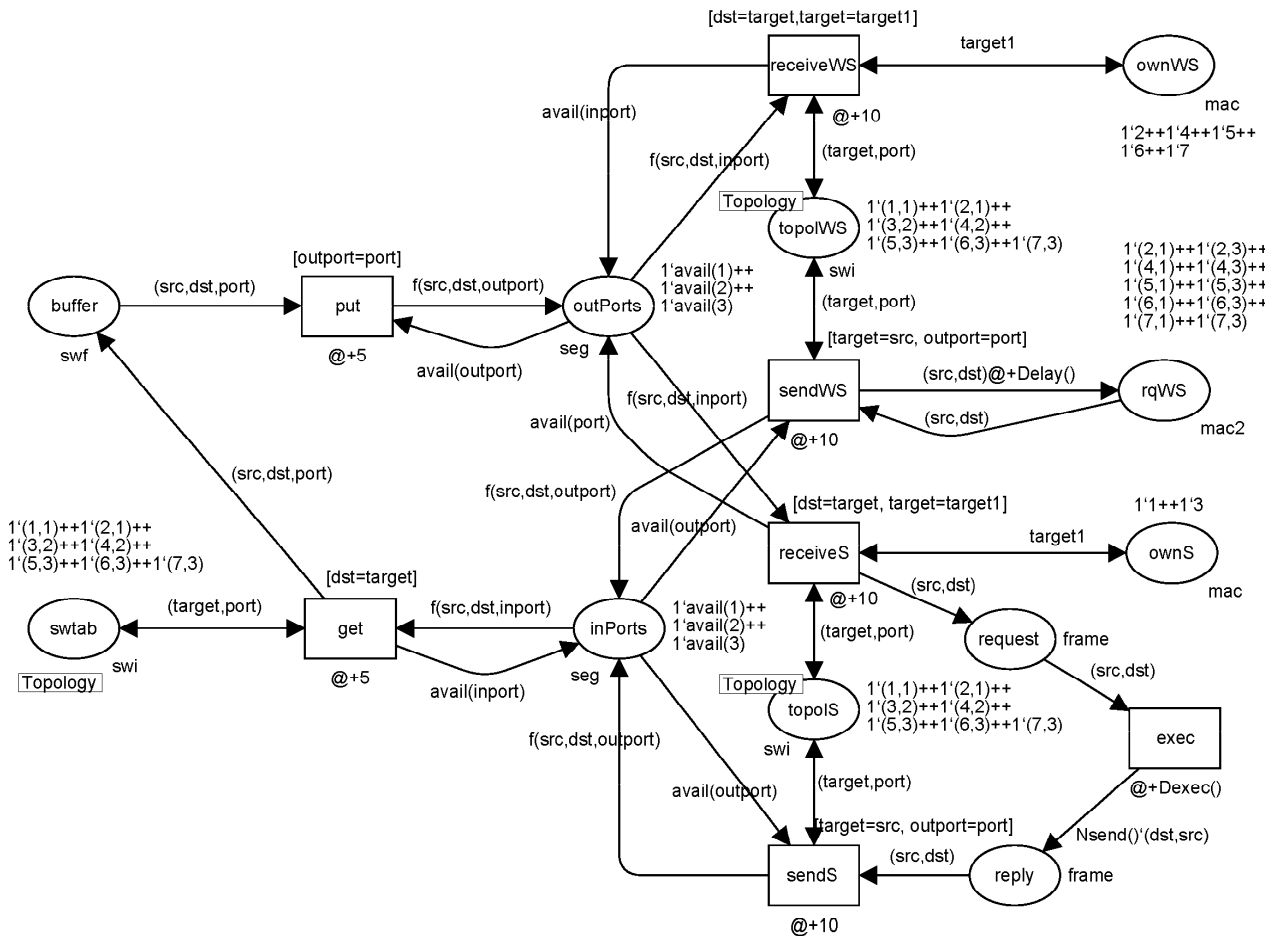


Рисунок 2 – Параметрическая модель одноуровневой коммутируемой сети

Перед тем как изучить назначение элементов модели рассмотрим описания использованных типов (цветов), переменных и функций, приведенных на рис. 3. MAC-адреса (тип `mac`) и номера портов (тип `portnum`) представлены целыми числами. Модельное представление фрейма (тип `frm`) включает адреса отправителя `src`, получателя `dst` и тэг номера сегмента (порта). Рассмотрение фреймов максимальной длины (1,5Кб) позволяет абстрагироваться от передаваемых данных. Сегмент (тип `seg`) представлен объединением типа фрейма `frm` и специальных меток доступности сегментов `avail` таким образом, что каждый из сегментов может быть либо свободным (`avail`), либо занятым передачей фрейма `f`. Топология сети представлена типом `swi`, сопоставляющим MAC-адресу терминального устройства номер порта. Таблица запросов рабочих станций имеет тип `mac2`, задающий пары MAC-адресов: рабочей станции и сервера, к которому она обращается. Тип `frame` представляет физический фрейм без тэга номера сегмента. Тип скомутированных фреймов `swf` фактически эквивалентен типу `frm`. Функции `Delay()`, `exec()`, `Nsend()` задают случайные равномерно распределенные величины в интервалах, указанных их базовыми типами `Delta`, `dex`, `nse` соответственно.

Переход `sendWS` выбирает из таблицы запросов пару MAC-адресов: рабочая станция, сервер и посылает фрейм запроса в сегмент ЛВС. Посылка фрейма возможна только в случае, если сегмент отправителя свободен – имеется фишка `avail(outport)` в позиции `inPorts`. При этом номер сегмента для MAC-адреса отправителя определяется на основе топологии сети из позиции `topolWS`. В позиции сегментов `inPorts` метка доступности `avail(outport)` заменяется фишкой, моделирующей фрейм `f(src,dst,outport)`; заметим, что переменная `outport` представляет собой тэг, позволяющий различать сегменты, по которым выполняется передача информации. После посылки запроса запись таблицы запросов блокируется на интервал времени, представленный функцией `Delay()`.

```
color mac = INT;
color portnum = INT;
color mac2 = product mac*mac timed;
color frm = product mac*mac*portnum timed;
color nseg = INT;
color seg = union f:frm+avail:nseg timed;
color swi = product mac*portnum;
color swf = product mac*mac*portnum timed;
color frame = product mac*nfrm timed;
var src,dst,target,target1: mac;
var port, inport, outport: 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();
```

Рисунок 3 – Описания типов, переменных и функций

Переход `get` выбирает фрейм из входного порта коммутатора и помещает его во внутренний буфер коммутатора, представленный позицией `buffer`. При этом в позицию `inPorts` возвращается метка доступности соответствующего сегмента `avail(inport)`. Значение тэга номера порта присваивается фрейму заново в соответствии с таблицей коммутации `swtab`. Номер порта соответствует сегменту, в котором находится устройство назначения фрейма. Переход `put` выбирает фрейм из внутреннего буфера коммутатора и помещает его в выходной порт `outPorts` в случае наличия фишки `avail(outport)`, моделирующей доступность необходимого сегмента ЛВС.

Переход `receivS` моделирует получения фрейма запроса сервером. При этом параметры запросов сохраняются в позиции `request`. Исполнение запроса моделируется переходом `exes`; время исполнения запроса задано функцией `Dexes()`. В результате фреймы ответа помещаются в позицию `reply`; количество фреймов ответа задано функцией `Nsend()`. Затем фреймы отправляются в сеть переходом `sends`. Передача фреймов ответа аналогична описанной ранее передаче фреймов запроса.

Для инициализации модели необходимо ввести параметры исследуемой сети: MAC-адреса рабочих станций, MAC-адреса серверов, количество входных/выходных портов, таблицу топологии, таблицу запросов. Таблица запросов рабочих станций к серверам сформирована как множество пар MAC-адресов, где первый параметр – адрес рабочей станции, второй – адрес сервера, к которому направляется запрос. Таблица топологии сети сопоставляет известному MAC-адресу устройства номер порта (сегмента локальной сети), к которому он подключен. Начальная маркировка указанных позиций для LAN1 (рис. 1 а) и LAN2 (рис. 1 б) представлена в Таблице 1. Заметим, что начальная маркировка модели, изображённой на рис. 2, соответствует сети LAN1.

Таблица 1 – Параметры модели

Наименование позиции	ownWS	ownS	inPorts, outPorts	Topology (swtab, topolWS, topolS)	rqWS
Описание	Список MAC-адресов рабочих станций	Список MAC-адресов серверов	Входные (выходные) порты коммутатора	MAC-адрес устройства и номер порта, к которому он подключен	MAC-адрес рабочей станции и сервера, к которому он посылает запросы
Содержимое для LAN1	1`2++ 1`4++ 1`5++ 1`6++ 1`7	1`1++ 1`3	1`avail(1)++ 1`avail(2)++ 1`avail(3)	1`(1,1)++1`(2,1)++ 1`(3,2)++1`(4,2)++ 1`(5,3)++1`(6,3)++ 1`(7,3)	1`(2,1)++1`(2,3)++ 1`(4,1)++1`(4,3)++ 1`(5,1)++1`(5,3)++ 1`(6,1)++1`(6,3)++ 1`(7,1)++1`(7,3)
Содержимое для LAN2	1`2++ 1`3++ 1`4	1`1	1`avail(1)++ 1`avail(2)++ 1`avail(3)++ 1`avail(4)	1`(1,1)++1`(2,2)++ 1`(3,3)++1`(4,4)	1`(2,1)++1`(3,1)++ 1`(4,1)

3. Отладка модели. Для отладки модели выполнена пошаговая трассировка прохождения фреймов запросов от рабочих станций к серверам и фреймов ответов в обратном направлении. Последовательность слайдов, представляющих динамику изменения состояния модели изображена на рис. 4. Переменная `Step` указывает номер шага, а переменная `Time` – значение модельного времени.

На шаге 1 фрейм запроса рабочей станции попадает в сегмент ЛВС, в канал, осуществляющий передачу фреймов по направлению к коммутатору. Фрейм сформирован рабочей станцией WS5, MAC-адрес которой равен 7. Фрейм адресован серверу S3 с MAC-адресом, равным 3. Передача фрейма осуществляется в третьем сегменте ЛВС в соответствии со значением тэга номера сегмента (третьего параметра фрейма).

На шаге 2 фрейм запроса попадает во внутренний буфер коммутатора. При этом значение порта, в который он будет направлен, выбранное в соответствии с таблицей коммутации, равно 2. Действительно, терминальное устройство с MAC-адресом равным 3 подключено ко второму сегменту сети.

На шаге 3 фрейм запроса попадает во второй сегмент сети, к которому подключен сервер S2, в канал, осуществляющий передачу фреймов по направлению от коммутатора.

На шаге 4 фрейм запроса принят сервером и размещён в буфере запросов ждущих выполнения. Заметим, что в буфере запросов тэг номера сегмента отсутствует.

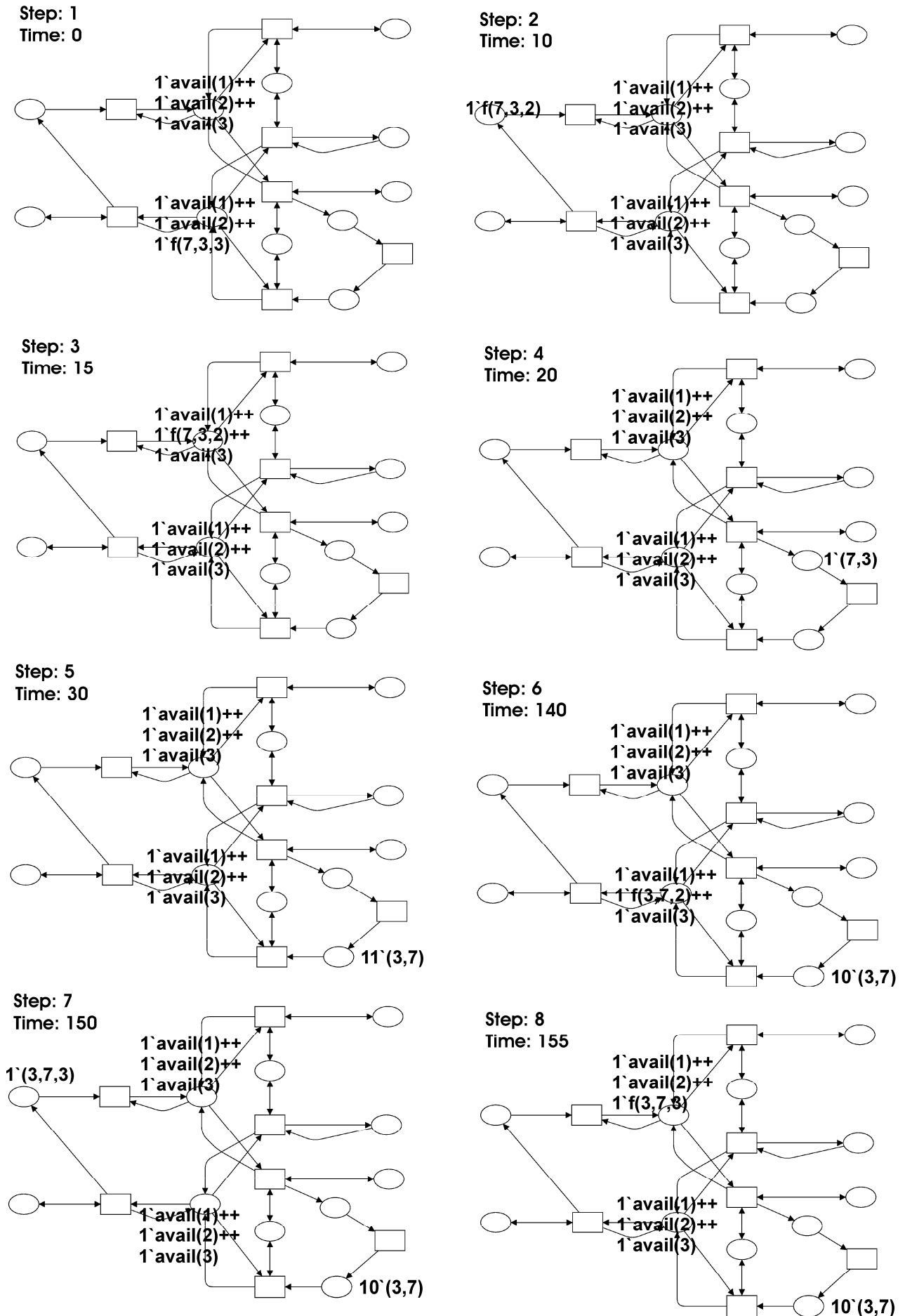


Рисунок 4 – Трассировка прохождения фрейма

На шаге 5 в результате исполнения запроса сервером в буфер ответов сервера помещаются 11 фреймов. Эти фреймы содержат результат выполнения запроса и в дальнейшем будут направлены в адрес запросившей рабочей станции. Заметим, что в заголовке фрейма адрес источника и адрес приёмника поменялись местами, что соответствует обратному направлению передачи фреймов.

На шаге 6 начинается передача фреймов ответа в адрес рабочей станции. Фрейм попадает в соответствующий сегмент ЛВС, в канал, осуществляющий передачу фреймов по направлению к коммутатору. Значение тэга сегмента выбрано в соответствии с таблицей топологии.

На шаге 7 фрейм ответа размещается во внутреннем буфере коммутатора. Значение тэга сегмента выбрано в соответствии с таблицей коммутации.

На шаге 8 фрейм ответа попадает в сегмент ЛВС, в канал, осуществляющий передачу фреймов по направлению от коммутатора. В дальнейшем он будет выбран соответствующей рабочей станцией. Заметим, что выполнена трассировка передачи лишь одного из 11 фреймов ответа; процесс передачи остальных фреймов аналогичен.

Рассмотрим временные характеристики модели. Модельное время на шаге 2 соответствует времени срабатывания перехода `sendWS`, равного 10 единицам модельного времени (Units of Model Time – MTU). Модельное время на шаге 3 равно 15, что соответствует времени получения фрейма коммутатором с помощью перехода `get`. Модельное время на шаге 4 равно 20 и соответствует времени передачи фрейма коммутатором с помощью перехода `put`. Время на шаге 5 определяется временем получения фрейма сервером с помощью перехода `receiveS` и равно 30. Приращение времени на шаге 6 соответствует времени выполнения запроса, заданного случайной функцией `Dehex()`, имеющей равномерное распределение в диапазоне 100..200. Приращения времени на шагах 6, 7 и 8 задаются временами срабатывания переходов `sendS`, `get` и `put` соответственно. Заметим, что количество сгенерированных фреймов ответа, равное 11, является результатом вызова случайной функции `Nsend()`, имеющей равномерное распределение в диапазоне 10..20. Следует отметить также, что после выполнения запроса использование пары адресов (7,3) из таблицы запросов `rqWS` блокируется на интервал времени, заданный случайной функцией `Delay()`, имеющей равномерное распределение в диапазоне 1000..2000.

4. Характеристики модели. Для исследования характеристик модели необходимо наблюдать поведение модели на достаточно больших интервалах времени, обеспечивающих достижение стационарного режима. В соответствии с масштабированием времени, описанным в [8], для модели ЛВС LAN1 диспетчерского центра железной дороги (рис. 1а) значение MTU равно 100 нс. Таким образом, минута реального времени равна $6 \cdot 10^6$ MTU.

Для запуска модели на длительные интервалы времени система CPN Tools имеет специальный режим, обеспечивающий высокую скорость моделирования без графического отображения динамики модели. В результате наблюдения модели на интервалах до одного часа реального времени были определены такие характеристики модели как ёмкость внутреннего буфера фреймов коммутатора, размеры буферов запросов и ответов сервера. Максимальные и средние значения указанных характеристик приведены в Таблице 2. Следует отметить, что наблюдаемые на протяжении указанных интервалов времени статистические характеристики развёрнутой [8] и параметрической модели совпадают.

Таблица 2 – Характеристики модели

Характеристика	Ёмкость буфера коммутатора	Ёмкость буфера запросов сервера	Ёмкость буфера ответов сервера
Позиция модели	buffer	request	reply
Максимальное значение	48	2	20480
Среднее значение	9,14	0,67	2102,39

Для определения более сложных характеристик модели таких, как, например, время отклика сети следует применить измерительные фрагменты, описанные в [8].

В заключение следует отметить, что описанная в настоящей работе параметрическая модель одноуровневой коммутируемой Ethernet инвариантна по отношению к топологии ЛВС. При моделировании новой ЛВС необходимо лишь задать начальную маркировку нескольких позиций, содержащих параметры модели. Таким образом, существенно сокращается время разработки моделей, что особенно важно для процессов управляемого моделью проектирования. Результаты отладки и сравнения статистических характеристик показали, что построенная параметрическая модель по своему поведению идентична исходной развёрнутой модели ЛВС.

Литература

1. Jensen K. Colored Petri Nets – Basic Concepts, Analysis Methods and Practical Use.- Vol. 1-3, Springer-Verlag, 1997.
2. Зайцев Д.А., Слепцов А.И. Уравнение состояний и эквивалентные преобразования временных сетей Петри // Кибернетика и системный анализ.- 1997, № 5, с. 59-76.
3. Зайцев Д.А. Инварианты временных сетей Петри // Кибернетика и системный анализ, Том 40, № 2, 2004, с. 92-106.
4. 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>).
5. Rahul V. LAN Switching // ОНЮ, 2002, 20 p.
6. Zaitsev D.A. Switched LAN simulation by colored Petri nets. Mathematics and Computers in Simulation, vol. 65, no. 3, 2004, 245-249.
7. Зайцев Д.А., Шмелёва Т.Р. Моделирование коммутируемой локальной сети раскрашенными сетями Петри // Зв'язок, № 2(46), 2004, с. 56-60.
8. 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, Aarhus, Denmark, p. 157-167.