

Моделирование коммутируемой локальной сети раскрашенными сетями Петри

Сети, предложенные Карлом Петри для моделирования дискретных параллельных процессов и систем, стали в настоящее время очень популярны в теоретических компьютерных науках и многочисленных приложениях. Исследование локальных и глобальных компьютерных сетей является одной из наиболее динамично развивающихся прикладных областей, в которой сети Петри находят всё более широкое применение.

Для описания реальных объектов используют, как правило, не базовые сети Петри [1], играющие центральную роль в теоретических исследованиях, а различные расширенные модели, такие как раскрашенные, временные, иерархические сети. Теория раскрашенных сетей разработана профессором университета Аархуса Куртом Йенсенем [2], исследованию временных сетей посвящена работа [3].

В работе [6] отмечено, что технология коммутации является перспективной для повышения пропускной способности локальных и глобальных компьютерных сетей. В работе [7] коммутируемая сеть исследована с помощью стохастических сетей с очередями; рассмотрено влияние размеров буфера коммутатора на количество потерянных пакетов и общую производительность сети. В отличие от указанных моделей сети Петри предоставляют средства для более точного описания архитектуры сети и трафика, так как позволяют представить взаимодействие в системах клиент-сервер.

Исследование коммутируемой сети выполнено в настоящей работе средствами моделирующей системы Design/CPN [4] разработанной в университете Аархуса (Дания). Система функционирует в среде ОС Unix и позволяет строить и исследовать промышленные модели реальных объектов представленные раскрашенными временными иерархическими сетями Петри содержащими сотни тысяч элементов. Design/CPN применена в десятках промышленных проектах, наиболее известными из которых являются проектирование интеллектуальных сетей Deutsche Telekom, проектирование системы управления сетью в RC International A/S, проектирование архитектуры новых мобильных телефонов Nokia, разработка системы безопасности и контроля доступа Dalcotech A/S, планирование операций в военно-воздушных силах США. Лицензия на использование системы Design/CPN в исследовательских проектах предоставлена университетом Аархуса Одесской национальной академии связи.

Описание объекта исследования

Основой построения коммутируемой локальной сети является коммутатор пакетов Ethernet. Логически коммутатор представляет собой множество портов [5]. К каждому порту может быть подключен сегмент локальной сети (например, посредством концентратора) либо окончное терминальное устройство – рабочая станция, сервер. Задача коммутатора состоит в перенаправлении поступившего пакета (фрейма) в тот порт, к которому подключено устройство назначения. Применение коммутатора позволяет снизить число коллизий, так как пакет

ретранслируется только в порт назначения, и таким образом повысить пропускную способность сети. Кроме того, повышается качество защиты информации за счёт снижения возможностей подслушивания трафика.

Функционирование сети Ethernet регламентируется стандартами IEEE 802.3. Для адресации используются шестибайтные уникальные MAC-адреса (Medium Access Control). Существенными для работы коммутатора являются поля фрейма, задающие адрес отправителя и адрес получателя. В некоторых коммутаторах предусмотрена также проверка контрольной суммы пакета и специальная обработка различных типов фреймов, например, для виртуальных сетей. Схема локальной сети небольшого офиса построенной с использованием коммутатора представлена на Рис. 1.

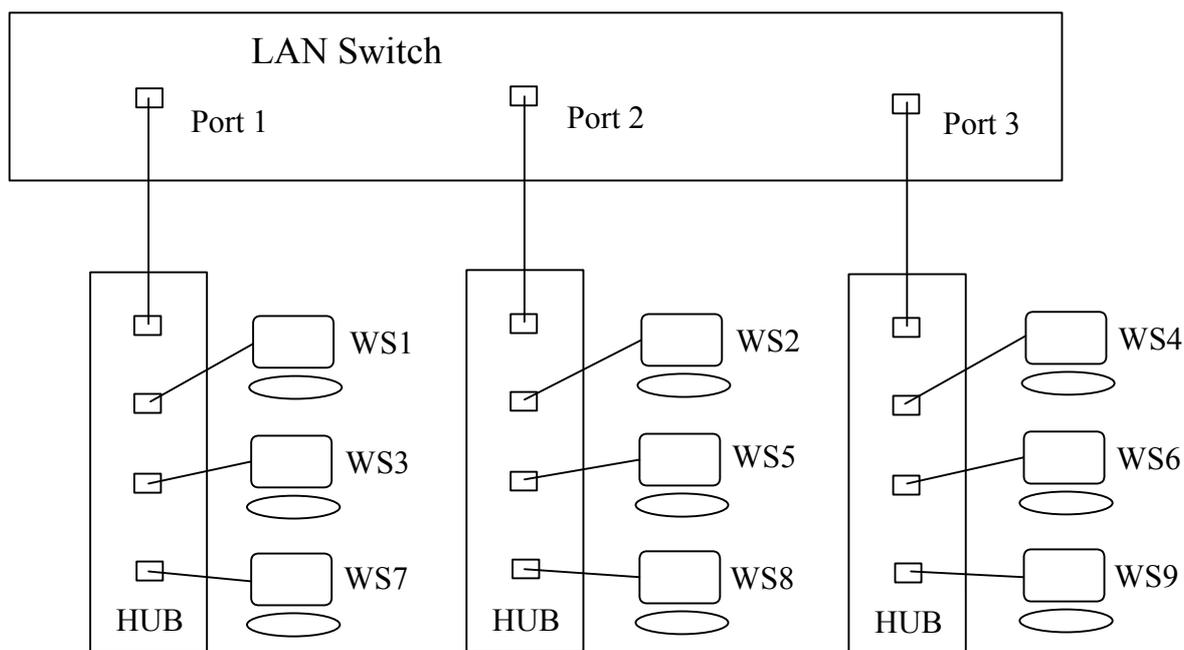


Рис. 1. Схема локальной сети.

Для определения номера порта, на который направляется фрейм, как правило, используется статическая либо динамическая таблица коммутации сопоставляющая каждому известному MAC-адресу номер порта, к которому подключено соответствующее устройство. Статические таблицы коммутации применяются в организациях с повышенными требованиями к безопасности и формируются администратором локальной сети.

Алгоритмы динамического построения таблицы коммутации основаны на прослушивании трафика для нахождения неизвестных MAC-адресов отправителя и создания для этих адресов новых записей таблицы. При обработке неизвестного адреса получателя пакет посылается во все порты коммутатора. Аналогичным образом обрабатываются фреймы широковещательной посылки, имеющие единицы во всех битах адреса получателя. Кроме того, применяется периодическое обновление записей таблицы для поддержания её актуальности и соответствия реальной структуре локальной сети. Фрагмент таблицы коммутации для рассматриваемой сети представлен в Таблице 1.

MAC-адрес	Порт	Время актуальности
1	1	20
2	2	10
3	1	1
4	3	20
5	2	2
6	3	5
7	1	8
8	2	15
9	3	3

Таблица 1. Таблица коммутации.

Следует отметить также ряд дополнительных возможностей коммутатора, являющихся в настоящее время стандартными. Это работа в полнодуплексном режиме, реализуемая с помощью отдельных входных и выходных буферов; контроль потока данных с помощью буферизации и специальных сообщений (jam message); фильтрация пакетов; средства удалённого администрирования с помощью различных протоколов (RMON, SNMP).

Введение в сети Петри

Сеть Петри [1] – это двудольный ориентированный граф с динамическими элементами – фишками. Графическое представление элементов сети Петри изображено на Рис. 3.



Рис. 3. Элементы сети Петри.

Динамика сети представляет собой процесс перемещения фишек в результате срабатывания переходов (Рис. 4). Переход разрешён, если все его входные позиции имеют фишки. Срабатывает произвольный переход из множества разрешённых. При срабатывании переход изымает фишки из своих входных позиций и помещает в свои выходные позиции. Срабатывание перехода происходит мгновенно.



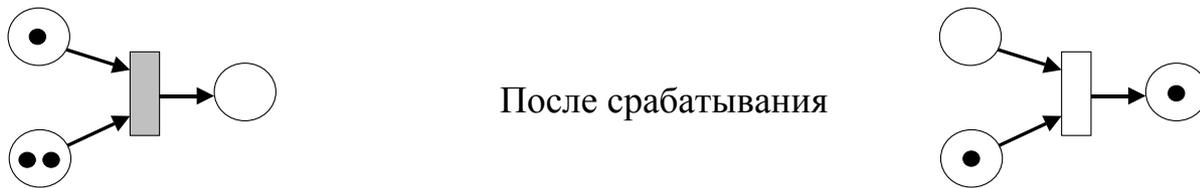


Рис. 4. Срабатывание перехода сети Петри.

Следует отметить, что функционирование сети представляет собой недетерминированный процесс. Исходное описание сети не содержит средства выбора запускаемого перехода из множества разрешённых. Поэтому базовые сети Петри в основном применяют для поиска нештатных ситуаций в множестве различных вариантов поведения моделируемой системы таких как тупики в операционных системах, столкновение транспортных средств в моделях движения на железной дороге, взаимные блокировки при обмене информацией в сети.

Важным преимуществом применения базовых сетей Петри и моделей, полученных с помощью дополнительных структурных либо иных ограничений, является наличие известных аналитических методов их исследования, таких как дерево покрывающих маркировок, инварианты, редукция. Однако низкая изобразительная мощь, отсутствие средств описания индивидуальных характеристик фишек и других элементов сети делает их применение затруднительным при исследовании сложных объектов.

Общая характеристика моделирующей системы Design/CPN

Система Design/CPN включает в себя графический редактор моделей, компилятор языка ML, модуль имитации и модуль анализа. Модель представлена раскрашенной временной иерархической сетью Петри. Предоставляются средства компоновки сложной модели, занимающей несколько листов, с помощью иерархической схемы, а также средства изображения дополнительных графических и текстовых элементов для повышения наглядности. Язык ML предназначен для описания характеристик элементов сети.

Первоначальные попытки различать фишки сети Петри были связаны с присваиванием им характеристик, представленных целым числом: фишки типа 1, фишки типа 2 и так далее. Условия разрешения перехода и результаты срабатывания также различали индивидуальность фишек. Наиболее наглядно индивидуальность фишек для небольшого числа их типов изображалась с помощью цвета: зелёный, красный и так далее. Поэтому сети, различающие характеристики фишек традиционно называют раскрашенными (Рис. 5), даже если фишка представляет собой переменную сложного абстрактного типа данных.

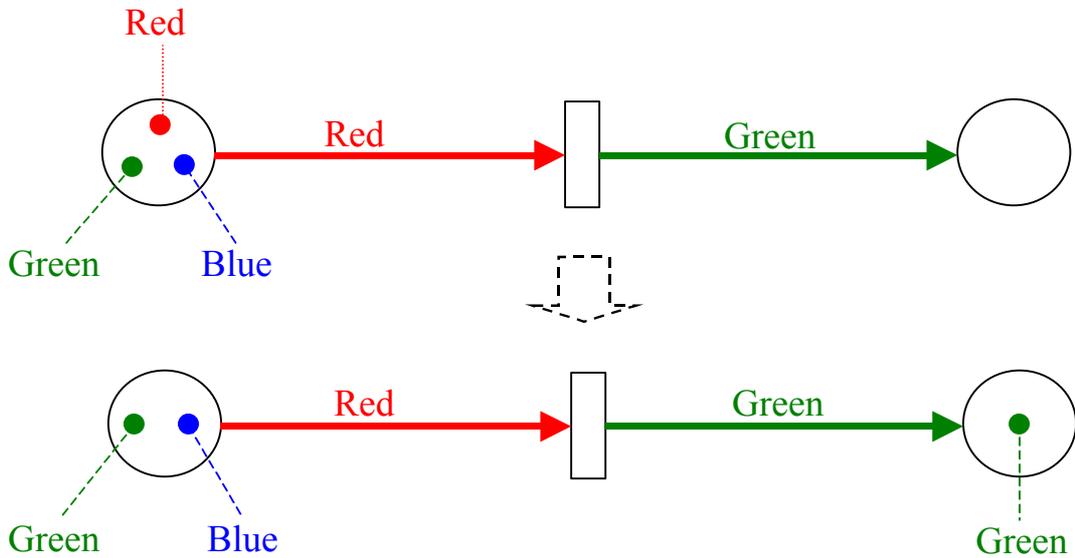
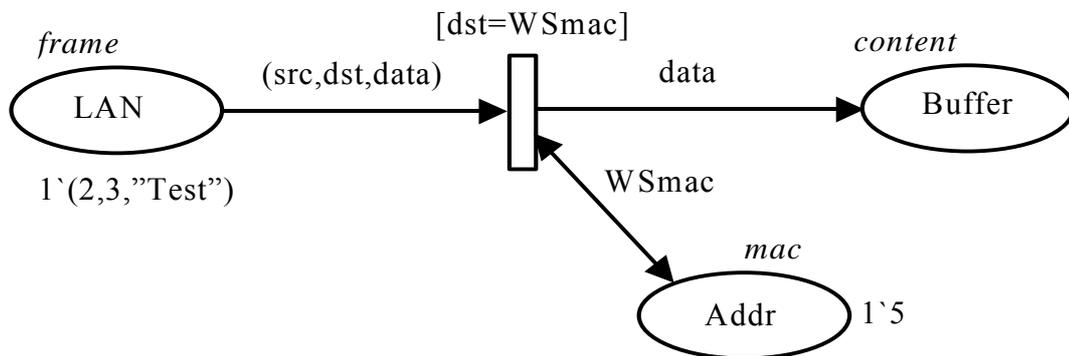


Рис. 5. Простейшая раскрашенная сеть Петри.

В системе Design/CPN элементы сети кроме индивидуальных имён имеют следующие характеристики: позиция – тип данных хранимых фишек; входная дуга перехода – шаблон для выбора входящих фишек; переход – условие возбуждения в форме предиката; выходная дуга – шаблон для формирования исходящих фишек. На Рис. 6 изображен фрагмент сети представляющий процесс выбора фреймов, адресованных рабочей станции с указанным адресом.



```

color mac = int;
color content = string;
color frame = product mac * mac * content;
var src, dst, WSmac : mac;
var data : content;

```

Рис. 6. Выбор фреймов из сети.

Для простоты представления в качестве адресов рабочих станций использованы целые числа, передаваемые данные представлены символьными строками. Фрейм состоит из адреса отправителя, адреса получателя и данных. Выполняется проверка адреса и в случае совпадения данные помещаются в буфер. Наличие фишек в позиции представлено мультимножеством, где числом задаётся кратность, а затем следует описание фишки. Традиционно типы данных в системе носят название цветов (color).

Временные характеристики задаются с помощью выражений вида $@ + Delay$, где значение $Delay$ определяет временную задержку. В простейшем случае временные задержки могут быть сопоставлены переходам сети. В более сложных ситуациях задерживаются индивидуальные фишки, в этом случае время указывается на исходящих дугах.

Для построения иерархических сетей используются подстановка перехода, при которой переход сети заменяется определённой сетью, а также объединение позиций, позволяющее компоновать модель из фрагментов, расположенных на разных листах.

Модель коммутатора локальной сети

Построим модель для заданной статической таблицы коммутации. Будем рассматривать отдельные входные и выходные буферы фреймов для каждого порта, а также общий буфер скомутированных фреймов. Модель коммутатора представлена на Рис. 7. Для тестирования модели выбрано размещение рабочих станций в соответствии с Рис. 1; таблица коммутации изображена на Рис. 2.

Как и в предыдущем разделе адреса представлены целыми числами, кроме того не рассматривается содержимое передаваемых пакетов. Тип данных frm задаёт фреймы сети, тип данных $swch$ представляет записи таблицы коммутации, тип данных $swchfrm$ описывает скомутированные фреймы, ожидающие размещения в выходном буфере. Позиции $PortX In$ и $PortX Out$ представляют входной и выходной буферы порта X соответственно. Позиция $SwitchTable$ моделирует таблицу коммутации; каждая фишка в этой позиции представляет запись таблицы коммутации. Позиция $Buffer$ соответствует буферу скомутированных пакетов. Переходы InX моделируют обработку входящих фреймов. Фрейм извлекается из входного буфера только в случае наличия в таблице коммутации записи с адресом, совпадающим с адресом назначения фрейма; при перемещении фрейма в буфере сохраняется номер порта назначения. Переходы $OutX$ моделируют перемещение скомутированных фреймов в выходные буферы портов. Операциям коммутации и записи в выходной буфер присвоены фиксированные временные задержки.

Из начального состояния:

$Port1 In - 2^{\backslash}(1,5)+3^{\backslash}(3,9)+2^{\backslash}(7,8);$

$Port2 In - 4^{\backslash}(2,6)+3^{\backslash}(5,7)+5^{\backslash}(8,4);$

$Port3 In - 7^{\backslash}(6,8)+2^{\backslash}(4,3)+3^{\backslash}(9,5)$

сеть попадает в конечное состояние, в котором все фреймы находятся в выходных буферах портов:

$Port1 Out - 2^{\backslash}(4,3)+3^{\backslash}(5,7);$

$Port2 Out - 2^{\backslash}(1,5)+7^{\backslash}(6,8)+2^{\backslash}(7,8)+3^{\backslash}(9,5);$

$Port3 Out - 4^{\backslash}(2,6)+3^{\backslash}(3,9)+5^{\backslash}(8,4).$

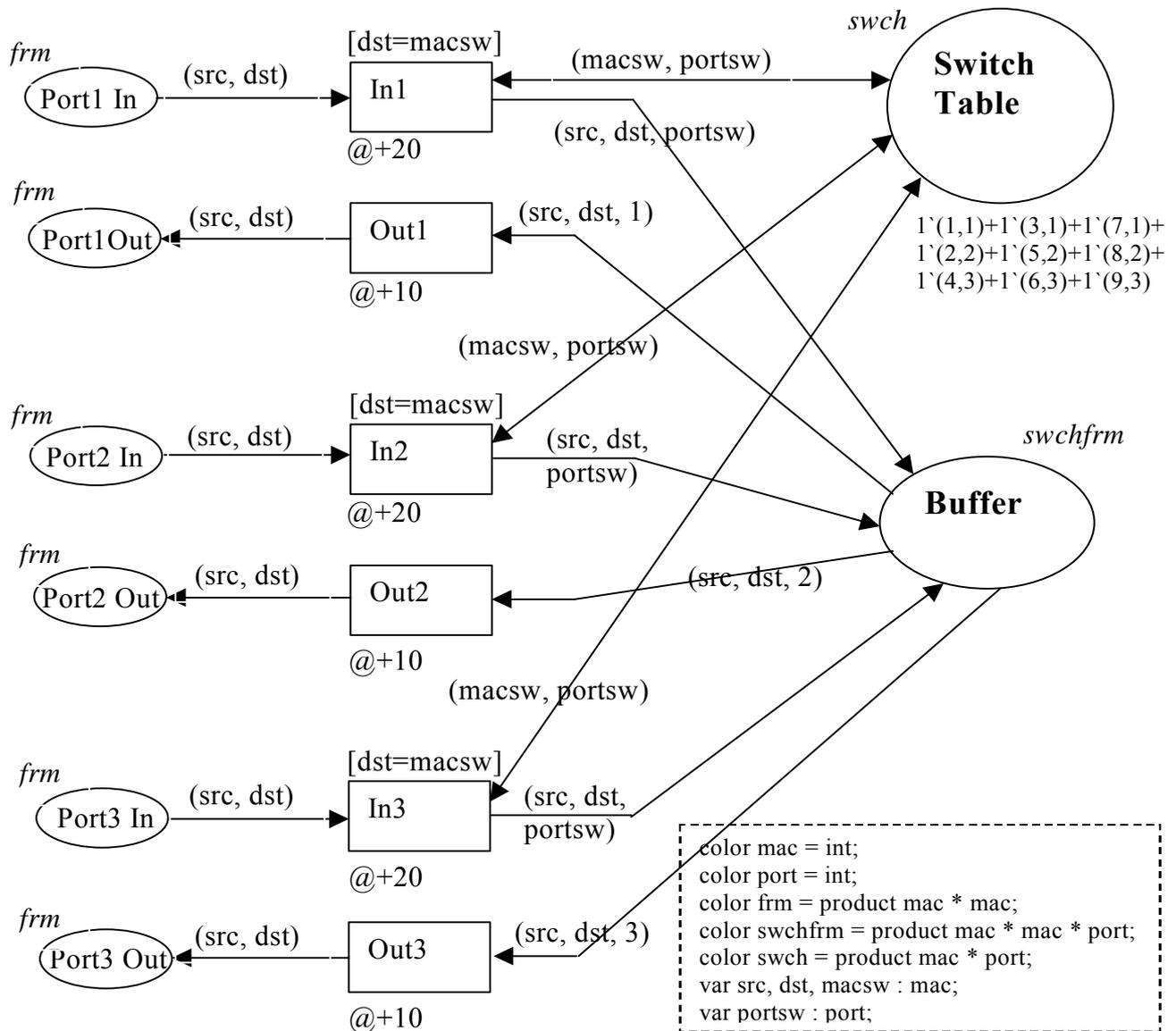


Рис. 7. Модель коммутатора.

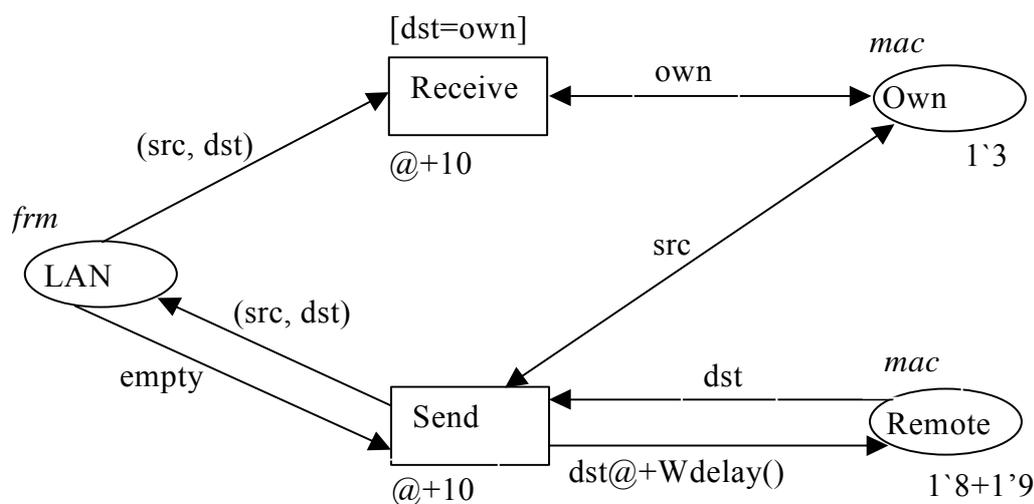
При этом максимальный размер основного буфера не превышает восьми. Процесс перемещения маркеров можно рассматривать в интерактивном режиме. Возможна пошаговая имитация либо указание точек останова модели для контроля её характеристик.

Модели рабочих станций и сегментов локальной сети

Для исследования потоков фреймов, перемещающихся по локальной сети, и решения задачи оптимального выбора буферов построенную модель коммутатора необходимо дополнить моделями терминальных устройств, подключенных к сети. Компоновка общей модели локальной сети может быть выполнена посредством объединения (наложения) позиций.

По особенностям формирования трафика будем различать рабочие станции и серверы. Для принятой степени детализации модели будем рассматривать периодически повторяющиеся со случайными равномерно распределёнными интервалами времени запросы рабочих станций к серверам. Сервер в ответ на

принятый запрос отправляет несколько пакетов в адрес запрашивающей рабочей станции. Количество отправляемых пакетов и временные задержки представляют собой равномерно распределённые случайные величины.

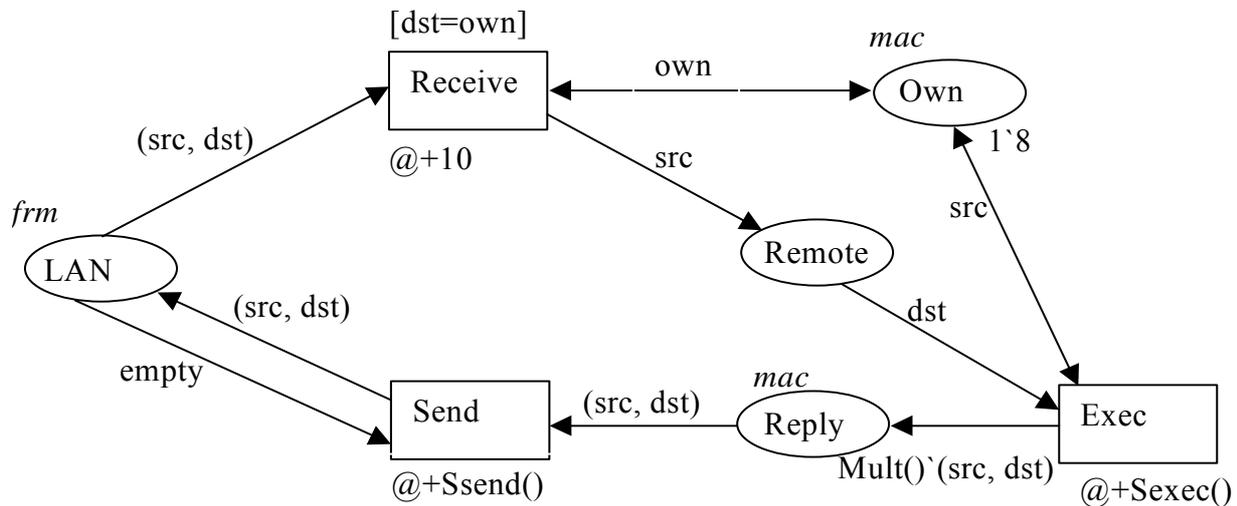


```
color SendDelay = int with 1000..10000 declare ran;
fun Wdelay = ran`SendDelay;
```

Рис. 8. Модель рабочей станции.

Модель рабочей станции представлена на Рис. 8. Позиция *LAN* моделирует сегмент локальной сети, к которому подключена рабочая станция. Станция слушает сеть с помощью перехода *Receive*, который получает фреймы с соответствующим адресом назначения; собственный адрес рабочей станции хранится в позиции *Own*. Обработка получаемых фреймов представлена их простым поглощением. Рабочая станция периодически посылает запросы к одному из двух серверов с помощью перехода *Send*. Адреса серверов хранятся в позиции *Remote*. После посылки сообщения использование адреса сервера блокируется на случайный интервал времени, задаваемый функцией *WSDelay()*. Посылка фрейма выполняется только в том случае, если сеть свободна, что моделируется проверкой позиции *LAN* на отсутствие фишек.

Модель сервера представлена на Рис. 9. Прослушивание сети аналогично модели рабочей станции, отличие состоит в том, что адрес отправителя фрейма сохраняется в позиции *Remote*. Переход *Exec* моделирует исполнение запроса рабочей станции сервером. В результате выполнения запроса генерируется случайное количество ответных пакетов, которые сохраняются в позиции *Reply*. Затем эти пакеты передаются в сеть переходом *Send*.



```

color SendDelay = int with 100..300 declare ran;
color ExecDelay = int with 1000..2000 declare ran;
color MultReply = int with 10..100 declare ran;
fun Ssend = ran`SendDelay;
fun Sexec = ran`ExecDelay;
fun Mult = ran`MultReply;

```

Рис. 9. Модель сервера.

Компоновка общей модели локальной сети выполнена посредством объединения позиций LAN рабочих станций и серверов для каждого из сегментов. Модель коммутатора объединяется с моделью сегментов при помощи дополнительных переходов *ReceiveX* и *SendX*, принимающих фреймы во входной буфер и отправляющих фреймы из выходного буфера соответственно для каждого из портов коммутатора. Построенная модель исследована в интерактивном и автоматическом режимах имитации динамики сети. В качестве оптимального размера буфера выбрано среднее количество фишек в позиции *Buffer*. Для приложений реального времени, где обработка jam-сообщений в случае переполнения буфера может существенно увеличить время отклика, рекомендуем выбирать значения, близкие к максимальным. Интервал модельного времени от отправки до получения фишки рабочей станцией является оценкой времени отклика сети.

Моделирование процессов заполнения таблицы коммутации

Следует отметить, что заполнение таблицы коммутации при массовом трафике в сети представляет собой достаточно редко выполняемые операции и не влияет существенно на общую производительность коммутатора. Поэтому задача выбора оптимального размера буфера коммутатора решена для статической заранее заданной таблицы. Кроме того, как было указано ранее, статические таблицы используются в сетях с повышенными требованиями к защите информации для предотвращения несанкционированных изменений структуры локальной сети. Таким образом, моделирование процессов ведения таблицы коммутации может быть применено для более точного анализа сети, а также для сетей с динамично меняющейся структурой либо частыми включениями рабочих станций.

Процесс заполнения таблицы описанный в первом разделе работы достаточно прост, но его реализация требует использования специальных средств моделирующей системы Design/CPN. Действительно, система не предоставляет стандартные средства для проверки отсутствия в позиции фишки с заданными характеристиками. А такая проверка является необходимой для обнаружения неизвестных MAC-адресов. Разработчики системы предлагают использовать в таких случаях позиции типа списка. Позиция всегда содержит одну фишку, являющуюся списком, в нашем случае, записей таблицы коммутации. Для проверки принадлежности к списку необходимо предусмотреть специальные функции.

В настоящей работе представлен более простой способ проверки, основанный на использовании переходов с различными приоритетами [1]. Проверка в таблице коммутации выполняется переходом с высоким приоритетом. Если таблица содержит необходимую фишку, запускается основной переход. Низкоприоритетный дополнительный переход не имеет дуг, проверяющих таблицу коммутации, он запускается лишь в тех случаях, когда не может быть запущен основной переход. Таким образом, моделируются альтернативные действия, напоминающие часть условного оператора, следующую за ключевым словом “else”.

На Рис. 10 представлен фрагмент модели, содержащий один порт, и иллюстрирующий процесс заполнения таблицы коммутации, а также широковещательную рассылку. Переходы In1Check и FindDst имеют более высокий приоритет по отношению к переходам In1Unknown и UnknownDst. Заполнение таблицы коммутации выполняется переходом In1Unknown, а широковещательная рассылка переходом UnknownDst.

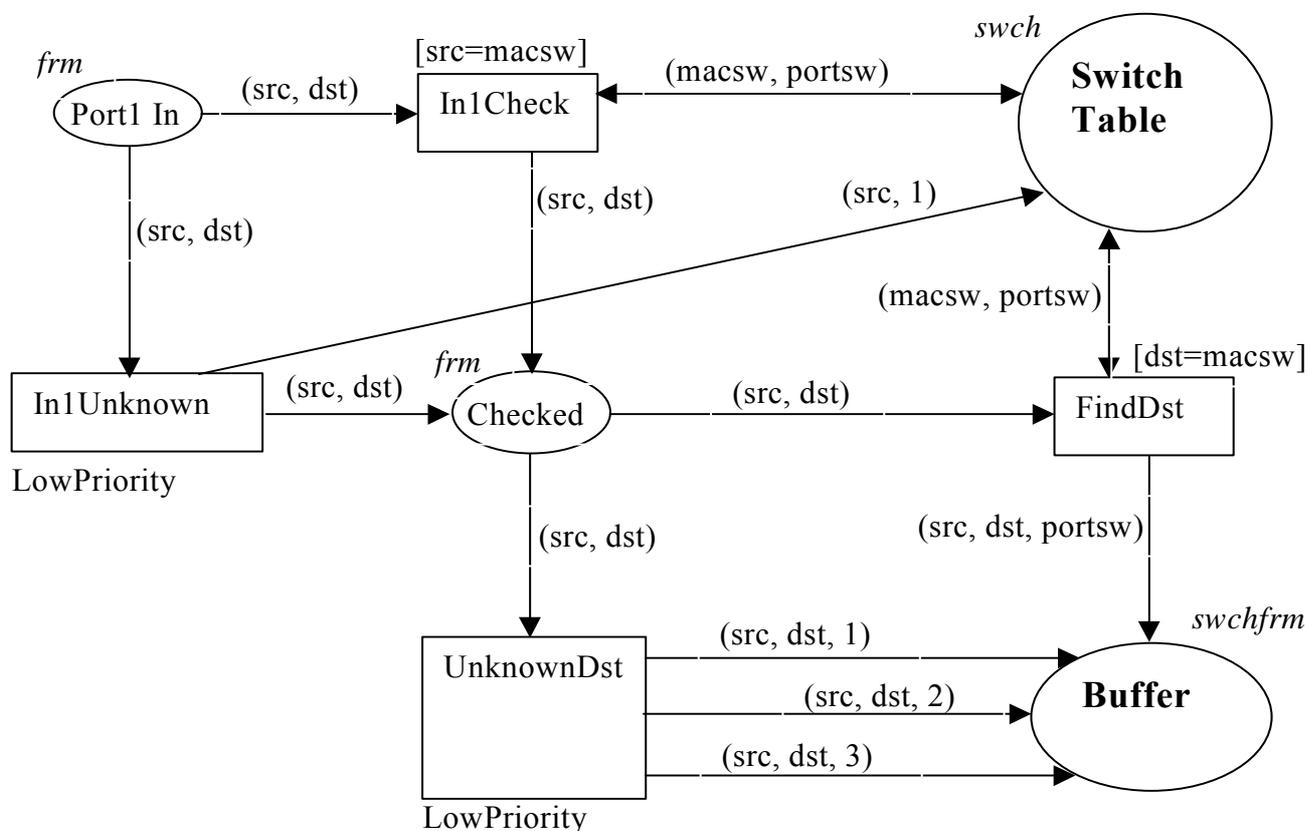


Рис. 10. Заполнение таблицы коммутации

Периодическое обновление осуществляется путём стирания записей таблицы через заданный интервал времени. На Рис. 11 представлен фрагмент модели, на котором в позиции Timer ведётся отсчёт времени отдельно по каждой записи таблицы коммутации. По истечении интервала времени ActualTime переход DeleteRecord стирает соответствующую запись таблицы.

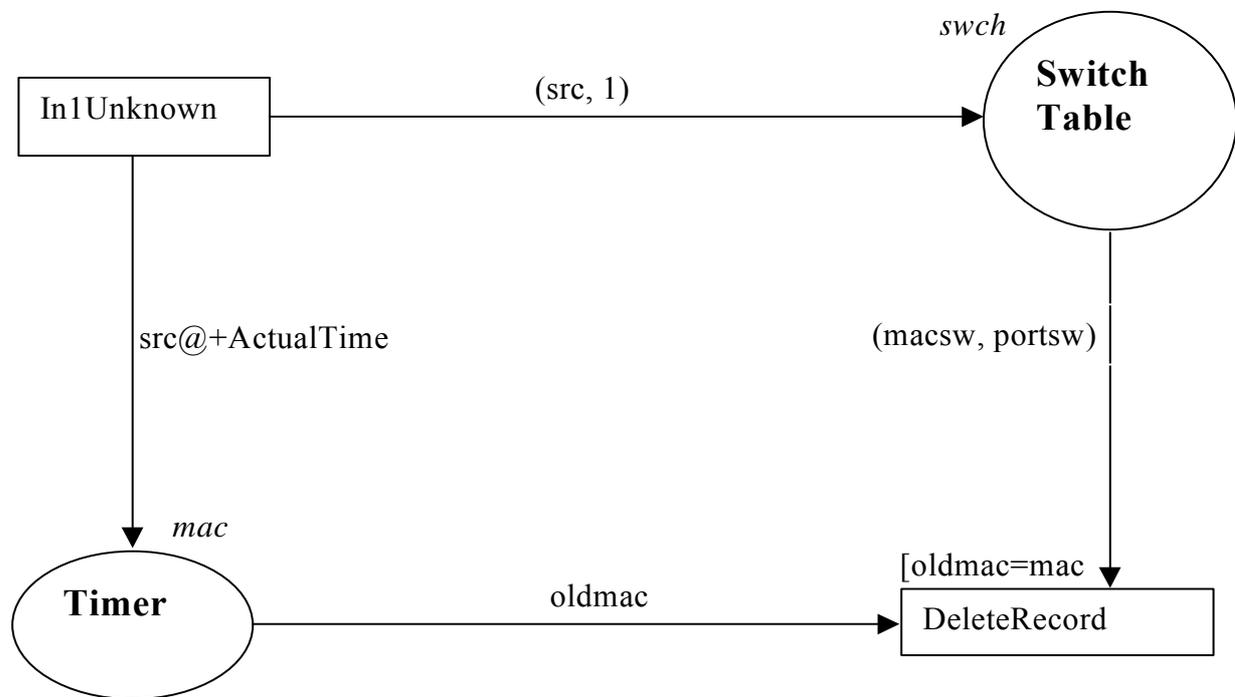


Рис. 11. Обновление таблицы коммутации

Таким образом, в настоящей работе построена и исследована модель коммутируемой локальной вычислительной сети представленная раскрашенной временной иерархической сетью Петри. Для анализа модели использована компьютерная система Design/CPN, с помощью которой решена задача оптимального выбора размера буфера коммутатора и оценки времени отклика сети. Следует отметить, что сети Петри представляют собой мощное и удобное средство исследования инфокоммуникаций, имеющее следующие преимущества: наглядное графическое представление; возможность применять как аналитические так и имитационные методы исследования; набор расширений базовой модели с высокой изобразительной мощностью; средства исследования как поведенческих так и статистических характеристик; возможность комплексной отладки модели объекта и алгоритма управления; средства построения сложных моделей большой размерности; аппаратные реализации процессоров сетей Петри

Список литературы

1. Котов В.Е. Сети Петри.- М. Наука, 1984.- 160 с.
2. Jensen K. Colored Petri Nets – Basic Concepts, Analysis Methods and Practical Use.- Vol. 1-3, Springer-Verlag, 1997.

3. Зайцев Д.А., Слепцов А.И. Уравнение состояний и эквивалентные преобразования временных сетей Петри // Кибернетика и системный анализ.- 1997, № 5, с. 59-76.
4. Albert K., Jensen K., Shapiro R. Design/CPN: A Tool Package Supporting the Use of Colored Nets // Petri Net Newsletter, April, 1989, pp. 22-35.
5. Rahul V. LAN Switching // OHIO, 2002, 20 p.
6. Hunt R. Evolving Technologies for New Internet Applications // IEEE Internet Computing, 1999, № 5, pp. 16-26.
7. Elsaadany A., Singhal M., Ming T. Lui. Performance study of buffering within switches in local area networks // Proc. of 4th International Conference on Computer Communications and Networks, 1995, pp. 451.