

УДК 004.7:004.94

В.Г. Тронин

## ПРИМЕНЕНИЕ РАСКРАШЕННЫХ СЕТЕЙ ПЕТРИ В МОДЕЛИРОВАНИИ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ

**Тронин Вадим Георгиевич**, окончил факультет информационных систем и технологий Ульяновского государственного технического университета. Начальник отдела, заместитель главного конструктора информационной системы предприятия. Специализируется в развитии информационной системы предприятия. Аспирант.

### Аннотация

Описан программный продукт, моделирующий вычислительную сеть на прикладном уровне (ПП МВСПУ) и реализующий раскрашенные сети Петри, собственные математические модели загрузки узлов. Программный продукт разработан с учётом особенностей научно-производственного объединения (НПО) и предназначен для прогноза загрузки, анализа слабых мест в структуре локальной вычислительной сети.

### ВВЕДЕНИЕ

Для научно-производственного объединения характерна уникальная схема процессов управления при реализации цикла НИОКР от замысла, проектирования до изготовления, испытаний опытного образца и до организации мелкосерийного производства наукоемкой продукции. Большой объем доработок, изменений и разнотипность обрабатываемой информации — типичные черты деятельности НПО, которые делают неизбежными высокую сложность информационной системы и сопряжение различных форматов данных и автоматизированных систем.

Главной задачей автоматизации в масштабах предприятия является информационная поддержка производственных, административных и

управленческих процессов (бизнес-процессов), формирующих продукцию и услуги предприятия. Примерная структура информационной системы НПО представлена на рис. 1.

Для моделирования вычислительной сети НПО использована теория сетей Петри. Сеть Петри — это инструмент для моделирования динамических систем. Данная теория позволяет моделировать систему математическим представлением в виде сети Петри. Проанализировав созданную сеть, можно получить информацию о структуре и динамическом поведении моделируемой системы. Несомненным достоинством сетей Петри является математически строгое описание модели, что позволяет проводить анализ с помощью современной вычислительной техники [1-2].



Рис. 1. Примерная структура информационной системы НПО

На общее быстродействие информационной системы в целом влияет небольшое количество параметров конечных устройств и линий связи. Например, для сервера — это быстродействие процессора, дисковой подсистемы, размер оперативной памяти. Каждый тип устройств, линий связи может быть смоделирован как система массового обслуживания с помощью раскрашенных сетей Петри. Каждое бизнес-приложение может быть представлено как генератор пакетов, обрабатываемых системой массового обслуживания.

### 1. Функции информационной системы моделирования вычислительных сетей на основе сетей Петри

На основе математического аппарата сетей Петри реализован ПП МВСПУ. В программном продукте применен пользовательский графический интерфейс как для описания моделей сети Петри, так и для описания структуры вычислительной сети.

ПП МВСПУ в своем составе имеет два основных инструментария: инструментарий сетей Петри и инструментарий моделирования вычислительных сетей.

Функциями инструментария сетей Петри являются:

- представление моделей узлов вычислительной сети на основе сетей Петри;
- дискретное моделирование работы отдельных узлов вычислительной сети и отображение результатов в виде графиков.

Функциями инструментария моделирования вычислительных сетей являются:

- представление топологии вычислительной сети;
- ведение библиотек элементов сети, моделей, программных продуктов;
- экспорт и импорт узлов вычислительной сети через формат XML;
- экспорт топологии вычислительной сети в формат Visio;
- дискретное моделирование работы вычислительной сети и отображение результатов в виде графиков.

### 2. Инструментарий сетей Петри

Одно из основных достоинств аппарата сетей Петри заключается в том, что они могут быть представлены как в графической форме (что обеспечивает наглядность), так и в аналитической (что позволяет автоматизировать процесс их анализа) [3].

Цвет в раскрашенной сети Петри описывает тип передаваемой или хранимой информации [4]. В реализованном ПП МВСПУ используются 3 простых цвета: *mac*, *data*, *load* и один составной — *frame*. Цвет *mac* — адрес узла сети, цвет *data* — элемент данных, передаваемый по сети.

Фрейм (*frame*) содержит:

- адрес источника, адрес получателя (*mac*),
- данные запроса, данные ответа (*data*),
- коэффициенты загрузки запросом процессора, оперативной памяти, жесткого диска, видеопамяти (*load*),
- коэффициенты загрузки ответом процессора, оперативной памяти, жесткого диска, видеопамяти (*load*).

При графической интерпретации сеть Петри представляет собой граф особого вида, состоящий из вершин двух типов — позиций и переходов, соединенных ориентированными дугами, причем каждая дуга может связывать лишь разнотипные вершины (позицию с переходом или переход с позицией). Вершины-позиции обозначаются кружками, вершины-переходы — черточками. С содержательной точки зрения, переходы соответствуют событиям, присущим исследуемой системе, а позиции — условиям их возникновения. Таким образом, совокупность переходов, позиций и дуг позволяет описать причинно-следственные связи, присущие системе, но в статике. Чтобы сеть Петри «оживила», вводят еще один вид объектов сети — так называемые фишки или метки позиций. Переход считается активным (событие может произойти), если в каждой его входной позиции есть хотя бы одна фишка. Расположение фишек в позициях сети называется разметкой сети.

Ограничения для позиции могут задаваться определением максимального числа фишек в позиции и максимального объема информации, передаваемой через определенную переменную.

Порты в описываемом инструментарии сетей Петри служат для организации связи между элементами моделируемой вычислительной сети, линиями связи могут быть соединены только с переходом. В каждой модели элемента вычислительной сети должен присутствовать хотя бы один порт.

Линии связи в рассматриваемом инструментарии могут быть двух видов: однонаправленные и двунаправленные. На рис. 2 представлен фрагмент сети Петри, состоящий из двух позиций (*LAN* и *Buffer*) и одного перехода (*Receive*), однонаправленных линий связи. При срабатывании перехода фишка из позиции *LAN* переходит в позицию *Buffer*.

Для каждой линии связи необходимо определить переменную или набор переменных (соответственно для фишек простого цвета и составного цвета). Поскольку для каждой позиции определяется тип фишек, то при редактировании переменных, относящихся к линии связи, перечень типов соответствует структуре типов для фишек в позиции, из которой исходит данная линия связи.

На рис. 3 представлен фрагмент сети Петри, в котором одна из линий связи является двунаправленной.

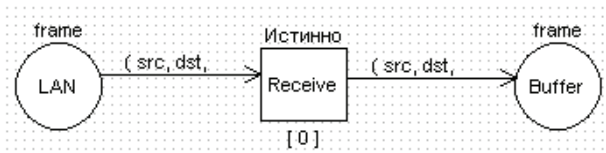


Рис. 2. Вариант применения однонаправленной линии связи

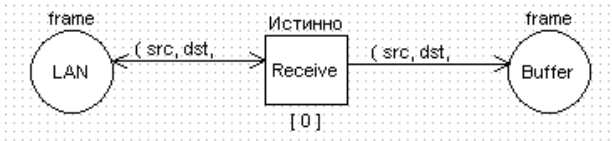


Рис. 3. Вариант применения двунаправленной линии связи

В примере, представленном на рис. 3, при срабатывании перехода Receive фишка попадает в позицию Buffer, при этом в позиции LAN фишка остается (в отличие от варианта с однонаправленной линией связи).

Рассмотрим более подробно процесс срабатывания перехода на примере фрагмента сети Петри, представленного на рис. 4.

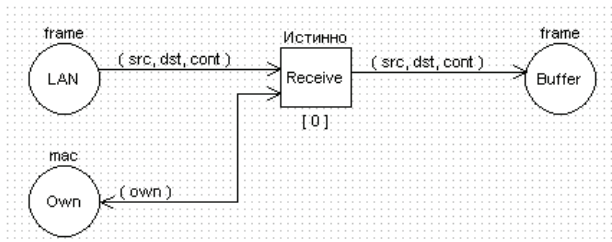


Рис. 4. Примерный фрагмент сети Петри

Данный фрагмент сети позволяет моделировать процесс выборки фреймов из локальной сети сервером. Локальная сеть представлена позицией LAN. В позиции Own хранится локальный адрес сервера, а в позицию Buffer помещается выбранный фрейм. Переход Receive на рис. 4 сработает в том случае, если во входных позициях LAN и Own будет размещаться хотя бы по одной фишке.

Пусть на первом шаге моделирования в позиции LAN находится одна фишка  $\text{frame} = (3, 5, 100)$ , в позиции Own — одна фишка  $\text{mac} = 5$ . Содержимое позиций соответствует заданному начальному значению фишек и представлено на рис. 5.

Разметка сети на втором шаге представлена на рис. 6.

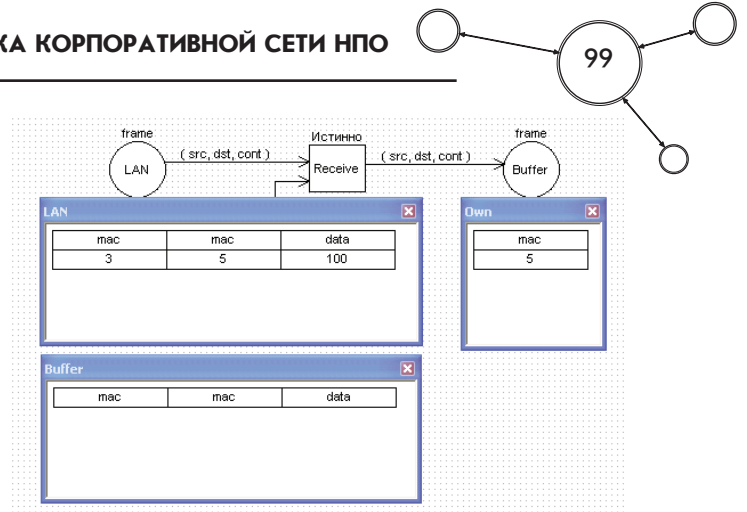


Рис. 5. Разметка сети на первом шаге моделирования

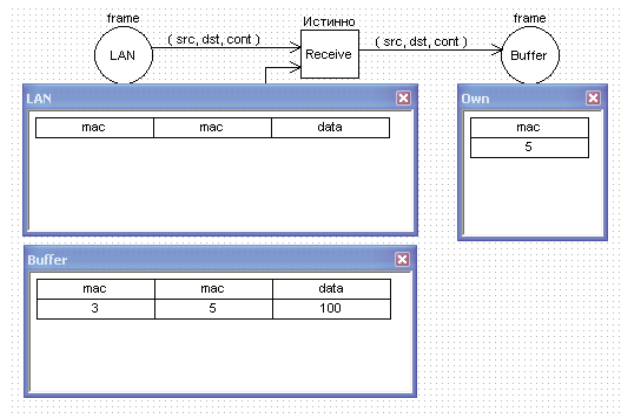


Рис. 6. Разметка сети на втором шаге моделирования

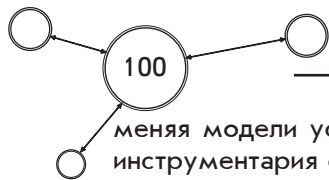
Из рис. 6 видно, что фрейм переместился из позиции LAN в позицию Buffer, а фишка в позиции Own осталась на прежнем месте.

Таким образом, если на определенном шаге моделирования фишка появляется в позиции LAN, то на следующем шаге она переходит в позицию Buffer. Для того, чтобы усложнить задачу и выбирать из позиции LAN только фишки, адресованные моделируемому устройству, необходимо соответствующим образом настроить условие срабатывания перехода.

Величину задержки можно задавать двумя способами: как постоянную (не изменяется во время моделирования) и как функциональную. При выборе функциональной задержки у пользователя есть возможность определить функцию изменения величины задержки и определить параметры выбранной функции.

### 3. ИНСТРУМЕНТАРИЙ МОДЕЛИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

Инструментарий моделирования вычислительных сетей позволяет создавать модель реальной или гипотетической вычислительной сети, при-



меняя модели устройств, созданные на уровне инструментария сети Петри.

Дерево библиотеки элементов рассматриваемого инструментария представлено на рис. 7.

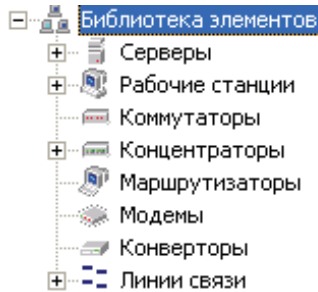


Рис. 7. Дерево библиотеки элементов

В ПП МВСПУ с помощью раскрашенных сетей Петри для каждого типа вычислительной техники создается и заносится в библиотеку отдельная модель. Библиотека содержит набор следующих элементов вычислительной сети: серверы, рабочие станции, коммутаторы, концентраторы, маршрутизаторы, модемы, конверторы и линии связи. Сетевой трафик представлен в виде фреймов, которые генерируются клиентами сетевых автоматизированных систем.

Информационную систему предприятия можно представить как соединенные в одно целое через компьютерную сеть технические средства с функционирующими приложениями (программные продукты, автоматизированные системы), позволяющие автоматизировать производственные процессы организации и являющиеся генераторами трафика [5-6].

Пакет, генерируемый приложением (например, CAD под операционной системой Linux) по

определенному расписанию, характеризуется следующими параметрами:

- размер запроса, разброс запроса, частота запроса;
- размер ответа, разброс ответа;
- коэффициенты загрузки оперативной памяти, процессора, жесткого диска, видеопамати у клиента.

Серверные программные приложения характеризуются следующими параметрами:

- тип сервера (FTP, приложений и т.д.);
- базовая операционная система (Windows Server 2003, Linux и т.д.);
- коэффициент загрузки памяти, жесткого диска, процессора, видеопамати.

В реальной вычислительной сети на одном аппаратном сервере могут функционировать несколько программных серверов (например, FTP-сервер и web-сервер), а ПЭВМ может пользоваться услугами, которые предоставляются разными автоматизированными системами.

Вид модели рабочей станции на основе раскрашенных сетей Петри, созданной в ПП МВСПУ, приведен на рис. 8.

Реализованная схема позволяет учитывать различные варианты устройств (например, ПЭВМ с несколькими сетевыми картами, несколькими процессорами, коммутатор с различным количеством портов), дополнять библиотеку вновь появляющимися типами устройств.

Для вычислительных систем (сервер, рабочая станция) создана библиотека, позволяющая для каждого типа ПЭВМ (сервера) сохранять параметры производительности процессора, жесткого диска, оперативной памяти, видеопамати и размер оперативной памяти.

Для проведения моделирования на уровне

приложений проводится первоначальное заполнение баз данных: клиентских сетевых приложений, программных серверов, вычислительных систем.

В базу данных серверов заносятся данные о программных серверах с коэффициентами загрузки для процессора, памяти, жёсткого диска, видеопамати.

В базу данных приложений заносятся все приложения, с которыми работают пользователи на рабочих станциях, с коэффициентами загрузки для процессора, памяти, жёсткого диска, видеопамати клиента, параметрами среднего размера, разброса, частоты запроса, среднего размера и разброса ответа.

Разработанный програм-

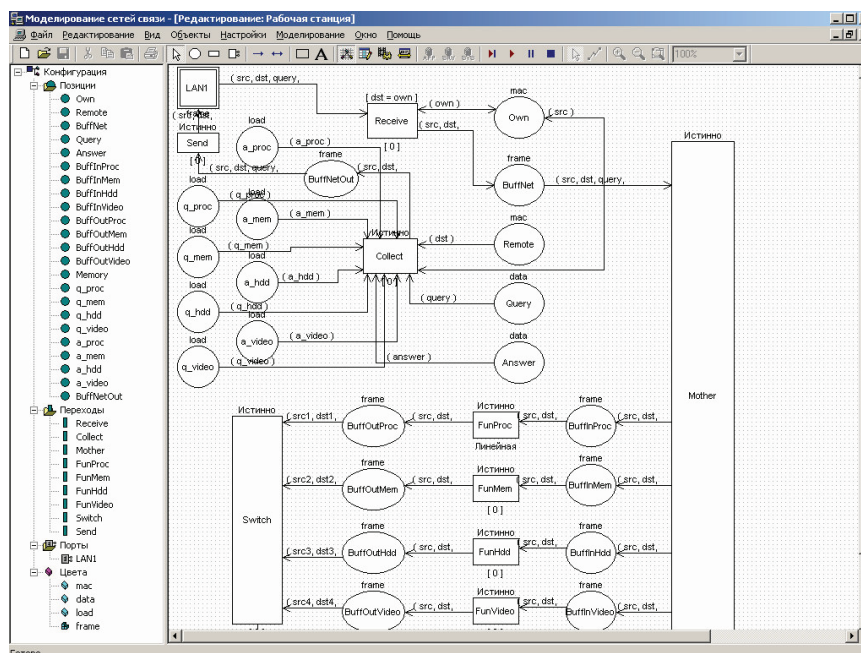


Рис. 8. Вид модели рабочей станции на основе раскрашенных сетей Петри



мный комплекс на основе раскрашенных сетей Петри позволяет выполнить моделирование с использованием естественных для восприятия пользователя объектов - программных приложений и технических средств вычислительной сети (фрагмент сети приведён на рис. 9).

#### 4. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ЗАГРУЗКИ УЗЛОВ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ НА ОСНОВЕ СЕТЕЙ ПЕТРИ

Рассмотрим реализованные в программном продукте модели загрузки. Для каждого из элементов сети вычисляется загрузка с усреднением за определённый интервал времени. Это позволяет в ходе моделирования определять слабые места в вычислительной сети. Для учёта разных единиц измерения (килобайты, мегабиты) вычислены поправочные коэффициенты.

Условные обозначения:

- $t$  - время моделирования;
- $x_m$  (мс) - дискретный шаг моделирования;
- $R_t$  (кБ) - величина запроса/ответа;
- $Dt = 1c$  - интервал усреднения;
- $\bar{O}_j$  (Мбт/с) - производительность узла (эталонная);
- $\bar{V}_j$  (кБ) - объем узла (эталонный);
- $k_u$  (уровень приложений) - коэффициент загрузки узла;
- $k_p$  (технические средства) - коэффициент перерасчета узла.

##### 4.1. Модель загрузки памяти узла

Усредненное значение загрузки памяти эталонного узла приведено на рис. 10 и вычисляется по следующей формуле:

$$V_{\Delta t} = \frac{\sum_{t \in \Delta t} R_t \cdot \tau_M \cdot 10^{-3}}{\Delta t}.$$

Следовательно, загрузка памяти эталонного узла в процентах будет выглядеть следующим образом:

$$Z_v = \frac{V_{\Delta t}}{V_j} \cdot 100\% = \frac{\sum_{t \in \Delta t} R_t \cdot \tau_M \cdot 10^{-3}}{\Delta t \cdot V_j} \cdot 100\%.$$

Загрузка памяти неэталонного узла

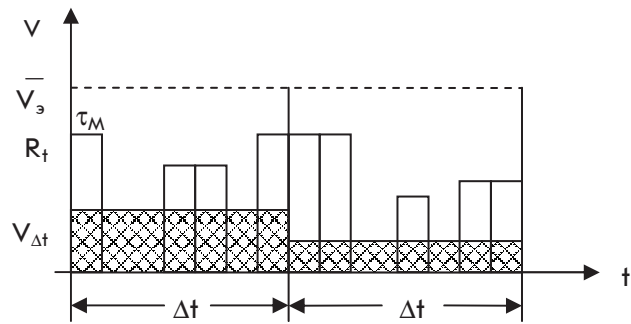


Рис. 10. Загрузка памяти эталонного узла

$$Z_v = \frac{V_{\Delta t}}{V_j \cdot k_p} \cdot 100\% = \frac{\sum_{t \in \Delta t} R_t \cdot \tau_M \cdot 10^{-3}}{\Delta t \cdot V_j \cdot k_p} \cdot 100\%.$$

##### 4.2. Загрузка узла по производительности

Время задержки перехода эталонного узла определяется следующим образом:

$$T_j = \frac{R_t}{V_j \cdot 976,5625},$$

где числовой коэффициент определяет разные единицы измерения (кБ и Мбт/с).

Из этого следует, что количество тактов:

$$N_j = \frac{1,024 \cdot R_t}{V_j \cdot \tau_M}.$$

Далее определяем количество тактов задержки реального узла с реальным приложением:

$$N = \frac{1,024 \cdot R_t \cdot k_u}{V_j \cdot \tau_M \cdot k_p}.$$

Загрузка узла по производительности на основе полученной задержки

$$Z_v = \frac{\sum_{t \in \Delta t} a_t \cdot \tau_M \cdot 10^{-3}}{\Delta t} \cdot 100\%,$$

где  $a_t = \{0, 1\}$ ,

0 — переход не активен в момент  $t$ ,

1 — переход активен в момент  $t$ .

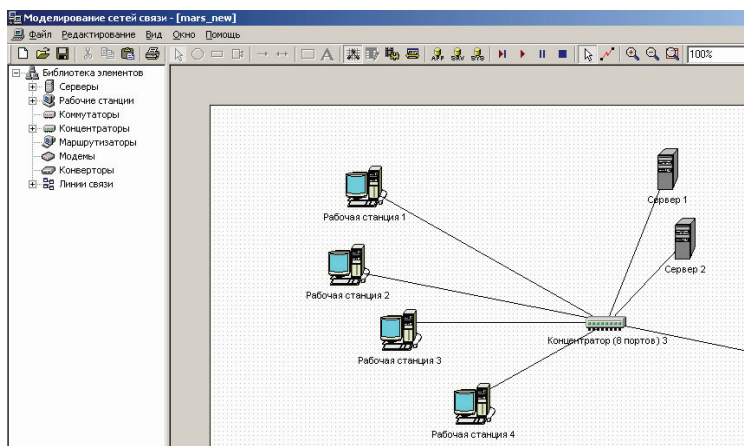


Рис. 9. Фрагмент вычислительной сети

Оценить результаты моделирования можно с помощью графиков и логов (протоколов экспериментов). Система построения графиков привязана к элементам (узлам) вычислительной сети. Имеются две возможности построения графиков: предопределенные графики и настраиваемые пользователем (путем выбора элемента исследования (позиции из списка) и категории исследования (сумма по цвету или число фишек в позиции)).

Логи ведутся в соответствии с заданными параметрами протоколирования и в дальнейшем могут быть обработаны в редакторе электронных таблиц.

По итогам эксперимента, на основании времен превышения порогов загрузки у различных устройств делается вывод о наиболее загруженных элементах вычислительной сети.

#### СПИСОК ЛИТЕРАТУРЫ

1. Бенькович Е.С., Колесов Ю.Б., Сениченков Ю.Б. Практическое моделирование динамических систем. — СПб.: БХВ-Петербург, 2002. — 464 с.
2. Нечепуренко М.И., Попков В.К., Майнагашев С.М. и др. Алгоритмы и программы решения задач на графах и сетях. — Новосибирск: Наука, 1990.

3. Питерсон Дж. Теория сетей Петри и моделирования систем: пер. с англ. — М.: Мир, 1984. — 264 с., ил.

4. Макаров И.М., Назаретов В.М., Кульба А.В., Швецов А.Р. Сети Петри с разноцветными маркерами // Техническая кибернетика. — 1987. - № 6. — С. 101—107.

5. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. — СПб.: Питер, 2001. — 672 с., ил.

6. Закер К. Компьютерные сети. Модернизация и поиск неисправностей: пер. с англ. — СПб.: БХВ-Петербург, 2003. — 1008 с., ил.