

УДК 021.8 + 025.1
ББК 78.34

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ПЕРЕДАЧИ ТРАФИКА В ВЫЧИСЛИТЕЛЬНЫХ СЕТЯХ

Гудов А. М.¹, Семехина М. В.²

(Кемеровский государственный университет, Кемерово)

Рассматривается подход к имитационному моделированию локальных вычислительных сетей (ВС) на основе раскрашенных временных сетей Петри (СП) с очередями. Вводится понятие ролевых функционалов и операций над ними. Приводится пример построения СП для фрагмента локальной сети, предложен общий алгоритм работы имитационной модели.

Ключевые слова: моделирование вычислительных сетей, теория массового обслуживания, имитационное моделирование, сети Петри.

1. Введение

Сегодня к компьютерным сетям как к вычислительной платформе предприятия предъявляются все более жесткие требования надежности, скорости и эффективности. Сеть рассматривается как важный, иногда даже критически важный ресурс, который должен быть использован максимально эффективно при минимальных затратах. Чаще всего к основным параметрам работы сети относят [9]:

- пропускную способность (емкость);

¹ Гудов Александр Михайлович, кандидат физико-математических наук, доцент кафедры ЮНЕСКО по НИТ (good@kemsu.ru).

² Семехина Марина Вячеславовна, аспирант (semehina@kru.ru).

- скорость;
- надежность.

Высокие требования к качеству работы сети обеспечивают интерес к мониторингу и моделированию вычислительных сетей. Средства мониторинга доступны сегодня в широком ассортименте, начиная от системных консольных утилит и заканчивая программными комплексами с возможностью управления и с применением экспертных систем (*HP OpenView NNM* [11], *NetView (Tivoli)* от *IBM*, *Spectrum* от *Cabletron*). В области моделирования ВС дело обстоит иначе, поскольку здесь необходимы не только технические знания, но и понимание теоретических основ процессов передачи и обработки данных, происходящих в вычислительных сетях.

К группе средств, моделирующих работу сетей, можно отнести *ComNet* от *CACI Products Company*, *OPNET* от *OPNET Technologies* (ранее – *MIL3*), *NetCracker*, *OMNeT++*, *NS-2*, *NS-3* и другие (таблица 1).

Каждая из этих систем предназначена для решения конкретных задач разной степени абстракции. Проект *COMNET* позиционировался как система моделирования ВС с развитыми средствами визуализации и модульной структурой, позволяющей облегчить труд системных администраторов. Среди основных подсистем *COMNET* можно выделить блок стохастического моделирования, подсистему быстрого временного анализа *Predictor*, подсистему мониторинга *Enterprise Profiler*, пакет анализа производительности сети *NETWORK*. В настоящее время пакет *COMNET* не поддерживается, компания развивает симуляционный пакет *SIMPROCESS*.

Система *Opnet Modeler* оперирует узлами трех типов – процессорными узлами, узлами-маршрутизаторами и коммутаторами. Узлы могут присоединяться с помощью портов к коммуникационным каналам любого типа, от каналов локальных сетей до спутниковых линий связи. Узлы и каналы могут характеризоваться средним временем наработки на отказ и средним временем восстановления для моделирования надежности сети.

Моделируется не только взаимодействие компьютеров в сети, но и процесс разделения процессора каждого компьютера

между его приложениями. Каналы связи моделируются путем задания их типа, а также двух параметров – пропускной способности и вносимой задержки распространения. Единицей передаваемых по каналу данных является кадр. Пакеты при передаче по каналам сегментируются на кадры. Каждый канал характеризуется минимальным и максимальным размером кадра, накладными расходами на кадр и интенсивностью ошибок в кадрах.

Рабочая нагрузка создается источниками трафика. Каждый узел может быть соединен с несколькими источниками трафика разного типа.

Источники планируемой нагрузки генерируют данные, используя зависящее от времени расписание. При этом источник генерирует данные периодически, используя определенный закон распределения интервала времени между порциями данных. Источники «клиент–сервер» позволяют задавать не трафик между клиентами и сервером, а приложения, которые порождают этот трафик.

К сильным сторонам *OPNET Modeler* можно отнести графические средства построения модели ВС. Система позволяет решать следующие задачи: планирование и оптимизация сети; создание и проверка аналитических моделей сети; создание и проверка протокола связи. К недостаткам относятся отсутствие трассировки моделей и недостаточная гибкость: используется собственная, нерасширяемая библиотека устройств и ограниченный набор отчетов.

Из бесплатных пакетов для задач имитационного моделирования можно отметить такие средства зарубежных разработчиков, как *jist/swans*, *NS-2*, *NS-3*, *GloMoSim*, *Parsec*.

NS-2 (*Network Simulator 2*) – бесплатный пакет, ядро которого реализовано на языке C++. Характерным свойством программного обеспечения *OSS* является возможность модификации ядра программы и гибкая настройка в соответствии с требованиями конкретного пользователя, а также мультиплатформенность – работоспособность существующих версий для *SunOS*, *Solaris*, *Linux*, *FreeBSD*, *Windows 5/98/ME/NT/2000*.

Таблица 1. Сравнительный анализ программных средств моделирования ВС

Продукт	<i>COMNET</i>	<i>NetMaker</i>	<i>OPNet</i>	<i>OMNeT++</i>	<i>SES/Strategizer</i>	<i>NS-2</i>	<i>Net Cracker</i>
Производитель	<i>CACI Products</i>	<i>Make Systems</i>	<i>OPNET</i>		<i>Scientific and Engineering Software</i>		<i>NetCracker technology</i>
Наличие библиотеки устройств	+	+	+	+	+	+	+
Настройка библиотеки устройств	+	+	-	+	+	+	-
Пошаговая трассировка	+	-	-	+	+	+	+
Генерация отчетов	+	+	-	+	+	-	+
Прочие особенности	Возможность ввода данных о трафике в режиме реального времени, простота ввода гипотез о трафике.	Высокая производительность; Наличие библиотек устройств разных производителей;	Удобство проектирования объектов сети, гибкость полученных моделей.	Симулятор дискретных событий, обладает высоким уровнем абстракции. Для моделирования используется	Простота применения Ясность схем сети	Использование двухуровневого программирования (язык <i>C++</i> и <i>OTcl</i>) и объектно-ориентиро-	Простота построения схем сети и настройка параметров сетевых компонент и потоков трафика;

Продукт	<i>COMNET</i>	<i>NetMaker</i>	<i>OPNet</i>	<i>OMNeT++</i>	<i>SES/Strategizer</i>	<i>NS-2</i>	<i>NetCracker</i>
	Возможность тонкой подстройки параметров сети. Наличие дополнительных модулей для различных задач			язык описания структуры модели <i>NED</i> и <i>C++</i> .		ванного подхода обеспечивает гибкую настройку моделей и сценариев их работы.	анимация
Стоимость, долл	от 20 000 за базовую конфигурацию	от 40 000	от 17 000	бесплатно	от 10 000	бесплатно	бесплатно
Недостатки	Сложность внедрения. Трудность восприятия схем сети. Неясность некоторых отчетов, высокая стоимость	Высокая стоимость, Потребность в дополнительном ПО. Требуется дополнительное обучение	Высокая стоимость, недостаточные возможности импорта и настройки существующих компонент продукта	Проблемы с внедрением, сложность обучения пользователей	Потребность в дополнительном ПО. Слабая визуализация.	Слабая визуализация; ограниченность библиотеки компонент; Сложность моделей и анализа результатов	Слабые возможности анализа информации; невозможность импорта данных из реальной сети

Также требует от пользователя навыков программирования на C++ и *OTcl* (*Object oriented Tool Command Language*), который используется в качестве интерпретатора. За счет использования двухуровневого программирования обеспечивается высокая степень настройки модели. Среди слабых мест пакета отмечают также сложность использования *OTcl*, проблемы переносимости кода, ограниченные возможности эффективного использования памяти при моделировании, недостаточная развитость режима трассировки и средств сбора статистики при работе модели, а также слабую визуализацию.

Высокая стоимость, сложность внедрения и интеграции, ограниченные возможности некоторых существующих программных продуктов, в совокупности с существующим спросом, заставляет исследователей искать новые решения для анализа и моделирования ВС [2, 3, 5, 8, 10, 12]. Фрактальные свойства сетевого трафика исследуются в работах Ильницкого С.В. [3] и коллектива американских исследователей (*Taqqi M.S., Wilson D.V., Leland W.E.*) [12]. В Пермском государственном университете разрабатывается распределенная имитационная система *Triad.Net* [5], которая использует трехуровневое описание имитационной модели в формате $M = \{STR, ROUT, MES\}$, где *STR* – слой структур; *ROUT* – слой процедур (рутин), описывающих алгоритм взаимодействия структур, и *MES* – слой сообщений, которыми обмениваются структуры.

Исследователи из Пензенского государственного университета (Вашкевич Н.П., Дубинин В.Н., Зинкин С.А.) [2] формализуют ВС с помощью языка описания сетевых моделей (ЯОСМ) и далее используют статистические методы обработки полученной модели. В ряде прочих работ используется анализ временных рядов, полученных из статистических данных о сетевом трафике, а также аппарат нейронных сетей, например, для решения задачи оптимального распределения источников трафика в корпоративной ВС. Методология гибридного моделирования ВС на основе аналитических методов и дискретных систем подробно рассматривается в работе Ярославцева А.Ф. [10]

Интерес к данной теме со стороны ведущих ИТ-компаний и независимых ученых определяет актуальность исследований в этой области.

В настоящей статье представлен имитационный подход к моделированию ВС на базе сетей Петри, который используется авторами в комплексе с аналитическими методами, что позволяет оптимальным образом решать широкий спектр задач моделирования.

На основе описанного подхода и комплексного использования методов аналитического и имитационного моделирования ВС, дополняющих друг друга, создан прототип информационной системы, **основными задачами** которой являются:

Создание адекватных моделей ВС на основе данных трех типов:

данные о топологии сети;

данные о маршрутах и характеристиках потоков трафика;

данные о вычислительных мощностях сетевых устройств.

Аналитический расчет сетевых параметров для представленных моделей.

Проведение имитационного эксперимента по передаче трафика в рамках построенной модели сети.

Анализ полученных результатов с целью выявления «критичных» участков сети с низкой или нестабильной скоростью обработки/передачи сообщений.

Система позволит оценить ВС на этапе ее проектирования или модернизации, а также будет полезна для оценки эффективности уже функционирующих сетей и выработки рекомендаций для их оптимизации.

2. Информационная модель вычислительной сети

Будем рассматривать вычислительную сеть, состоящую из устройств различных типов, в которой циркулируют пакеты данных. Модель вычислительной сети, в общем виде, можно представить в нотации *RDF* [7] (рис. 1).

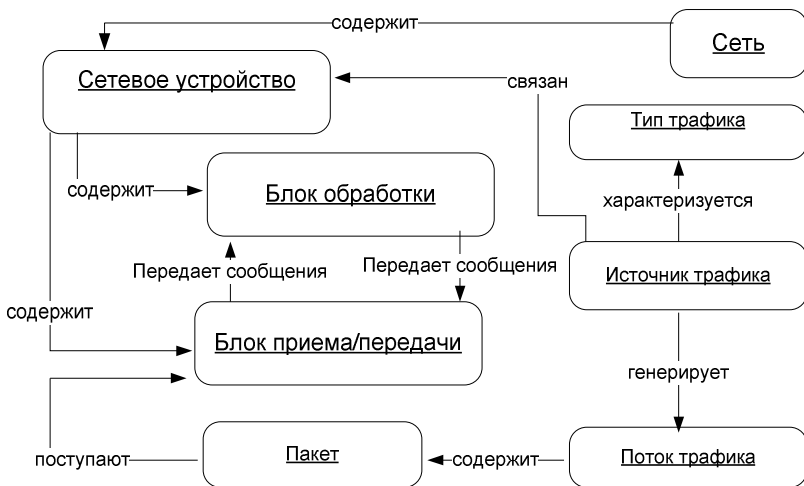


Рис. 1. Информационная модель сети

Здесь представлены основные информационные объекты и связи между ними.

Объект «Сеть» описывает реальную ВС и может включать в себя в качестве подсетей другие объекты этого типа.

Объект «Сетевое устройство» моделирует устройства, входящие в состав сети. Это абстрактный объект, функцией которого является обработка пакетов, т.е. задержка их в устройстве на некоторое время и возможная модификация пакета (например, изменение его типа). Каждый экземпляр объекта характеризуется в первую очередь значением своего параметра «тип». Типами сетевого устройства могут являться: рабочие станции, коммуникационное, периферийное оборудование, каналы передачи данных. Тип определяет конкретную структуру «Сетевого устройства».

Объект «Блок обработки пакетов» характеризует основной элемент сетевого устройства, ответственный за скорость и дисциплину обработки сетевых пакетов. Для однопроцессорных сетевых устройств «Блок обработки» моделирует работу центрального процессора и содержит данные о его мощности (в операциях в секунду). Для многопроцессорных устройств каж-

дый «Блок обработки» характеризует один процессор. Для сетевых каналов передачи данных «Блок обработки» является виртуальным объектом и характеризует скорость передачи данных (байт(бит)/секунда).

Объект «Блок приема-передачи» входят в состав «Сетевого устройства» и служит для описания входа и выхода потока сетевого трафика. Блок приема-передачи для сетевых узлов связан с их сетевыми интерфейсами.

Собственно исследуемый сетевой трафик описывается объектами «Тип трафика», «Источник трафика», «Поток» и «Пакет».

Объект «Тип трафика» содержит информацию о группе потоков сообщений в сети, обладающих общими параметрами. К этим параметрам относятся:

- используемые протоколы;
- службы или приложения, создающие этот трафик (например, служебный, передача видео, *ftp*-трафик);
- приоритет данного типа трафика;
- коэффициент задержки, связанный с обслуживанием данного типа трафика (например, *sql*-запросы могут обслуживаться дольше, чем служебный трафик даже при одинаковом размере пакетов);
- закон распределения, характеризующий количество пакетов, сгенерированных в единицу времени;
- закон распределения, характеризующий размер пакетов, сгенерированных в единицу времени;

Объект «Источник трафика» является отправной точкой нескольких потоков однотипного трафика из одного устройства к нескольким. Источник трафика характеризуется «Блоком приема-передачи» устройства, с которым он связан, «Типом трафика» а также моментом начала генерации сообщений, который может задаваться статически или определяться в момент работы устройства. Все сообщения, сгенерированные «Источником трафика», в общем случае с различной вероятностью поступают к нескольким адресатам (если адресат один, то

и вероятность поступления каждого сообщения к нему соответственно равна 1).

Объект «Поток» служит для общего описания всех сообщений, сгенерированных одним «Источником трафика» и направленных к одному «Блоку приема-передачи» узла по одному маршруту, т.е. через некоторую однозначно определенную в некоторый момент времени последовательность «Сетевых устройств».

Объект «Пакет» описывает один сетевой пакет потока. Детализация до этого уровня необходима только при имитационном моделировании, поскольку ее запуск связан с отслеживанием пути каждого пакета. В результате формируется статистика по трафику в целом.

3. Математическая модель ВС на основе СП

Для моделирования используются иерархические раскрашенные СП с временным механизмом [4, 6], где цвет (тип меток) соответствует типу трафика. Метки могут быть двух типов – «сообщения» и «служебные маркеры», означающие факт занятости ресурса обработкой меток-«сообщений». Количество меток в позициях соответствует состоянию сети, переходы изменяют это состояние, перемещая метки в другие позиции. Время в модели принимается дискретным и измеряется тактами. В течение одного такта метка может совершить не более одного перемещения через разрешенный переход. В классических СП без временных задержек один такт соответствует срабатыванию одного (любого) разрешенного перехода. Но поскольку СП ценны именно для моделирования параллельных систем, в большинстве интерпретаций и расширений СП такт понимается шире – как промежуток, в течение которого срабатывают все разрешенные в начальном состоянии переходы. Каждый сетевой объект (рабочая станция, канал передачи, коммуникационное устройство) моделируется как подсеть Петри.

3.1. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

1. Теоретико-множественное определение СП можно представить в следующем виде [4, 6]:

$$W = \{P, T, I, O, M\},$$

где $P = \{p_1, \dots, p_n\}$ – конечное множество позиций;
 $T = \{t_1, \dots, t_m\}$ – конечное множество переходов сети;
 $\Gamma: T \rightarrow P^*$ – входная функция инцидентности, сопоставляющая переходу множество его входных позиций ($\Gamma(t, p) = 1$, если p – входная позиция для t , $\Gamma(t, p) = 0$ в противном случае);
 $\Gamma^+: T \rightarrow P^*$ – выходная функция инцидентности, сопоставляющая переходу множество его выходных позиций ($\Gamma^+(t, p) = 1$, если p – выходная позиция для t , $\Gamma^+(t, p) = 0$ в противном случае);
 $M = (M(p_1), \dots, M(p_n))$ – вектор распределения меток по позициям (маркировка).

Однако для целей моделирования ВС в нашем случае удобнее ввести следующее определение:

$$W = \{P, T, D, TR\},$$

где P – множество позиций; T – множество переходов;
 D – множество дуг; TR – множество типов трафика; $TR(p)$ – множество типов меток, которые могут находиться в позиции p ;
 $m(tr, p)$ – метка типа tr , находящаяся в позиции p .

Инцидентность в данном определении выражается с помощью дуг, наличие дуги от перехода к позиции равнозначно утверждению о том, что данные переход и позиция инцидентны друг другу следующим образом:

$$d(t, p) \Leftrightarrow I_+(t, p) = 1 \Leftrightarrow I_-(p, t) = 1,$$

$$d(p, t) \Leftrightarrow I_+(p, t) = 1 \Leftrightarrow I_-(t, p) = 1.$$

2. $\forall(t, r), t \in T, r \in TR, \exists q \in R$ – временная задержка на каждом переходе определяется для каждого типа трафика.

3. Введем понятие очереди для типов меток для описания ситуации, когда в момент обработки метки во входную позицию перехода могут поступать новые метки разных типов. Логически метки-«сообщения» должны обрабатываться в порядке прямой или обратной очереди (возможны и иные дисциплины обслуживания). В том случае, когда в данной позиции присутствует только один тип трафика, понятие очереди аннулируется, ибо с точки зрения сети Петри метки одного типа принципиаль-

но не отличаются друг от друга и, таким образом, при наличии нескольких меток во входной позиции и разрешенном переходе перемещается любая метка из них. Если же через позицию проходит смешанный трафик, выбор типа метки для перехода должен учитывать очередность их поступления:

$$\forall p \in P, \forall tr_1, \dots, tr_N \in Tr : M(p) = \{tr_1, \dots, tr_N\}$$

$$\exists \{\{k_i, tr_i\}\}, i = 1, \dots, N : k_i \in Z = t(m(tr, p)).$$

То есть для любой позиции p и разрешенных для нее типов трафика определен набор пар $\{k, tr_i\}$, где k – момент времени поступления метки типа tr_i . Тип следующей метки для обработки выбирается выбором tr_i при минимальном k .

4. Переход является разрешенным, если во всех входных позициях находятся метки и переход не находится в состоянии «занят». Это можно выразить следующим образом:

$$R(t) = 1 \Leftrightarrow (S(t) = 1) \& (\prod M(p_i) > 0),$$

где $p_1, \dots, p_k \in P$ – входные позиции для перехода t ; $R(t)$ – функция «разрешенности» перехода t , а $S(t)$ – функция «незанятости» перехода t .

Переход находится в состоянии «занят» в течение времени обслуживания, после чего переходит в состояние «свободен». Предполагается, что в начальный момент времени работы имитационной модели переход находится в состоянии «свободен» ($S(t) = 1$).

3.2. ПРЕДСТАВЛЕНИЕ ОСНОВНЫХ УСТРОЙСТВ СЕТИ

Функционально каждое сетевое устройство может выступать в нескольких «ролях»: передатчика, приемника или генератора трафика. Причем одно и то же устройство может сочетать в себе несколько ролей, при этом набор ролей носит статический характер, т.е. определяется в момент моделирования устройства и не меняется в течение работы модели. Сеть Петри, выполняющую одну из перечисленных функций, будем называть ролевым функционалом. Это понятие вводится для удобства моделирования сетевых устройств, каждое из которых может выступать в одной или нескольких из перечисленных ролей. При таком подходе модель любого сетевого устройства строит-

ся из таких ролевых функционалов с помощью операций, приведенных ниже.

Формально определение ролевого функционала для нераскрашенной СП можно представить следующим образом:

$$F: \{m(p^{in}_1), \dots, m(p^{in}_n)\} \rightarrow \{m'(p^{in}_1), \dots, m'(p^{in}_n); m''(p^{out}_1), \dots, m''(p^{out}_s)\},$$

где $p^{in}_1, \dots, p^{in}_n$ – входные позиции F ; $p^{out}_1, \dots, p^{out}_s$ – выходные позиции F ; $m(p)$ – количество меток, входящих в позицию p извне; $m'(p)$ – количество меток, которые остаются в позиции p после одного срабатывания всех разрешенных переходов в F ; $m''(p)$ – количество меток, выходящих из позиции p за пределы F после одного срабатывания всех разрешенных переходов в F .

Определим основные ролевые функционалы и представим их в виде простых СП.

Функционал – приемник/передатчик трафика (рис. 2А): элементарная сеть Петри, на основе которой строятся все остальные функционалы. Состоит из двух позиций 1 и 2, перехода t и двух дуг (от позиции 1 к переходу t и от перехода t к позиции 2). Трафик будет поступать в позицию 1 и передаваться в позицию 2, т.е. при наличии метки в позиции 1 метка попадает через переход в позицию 2. С переходом связана задержка обслуживания.

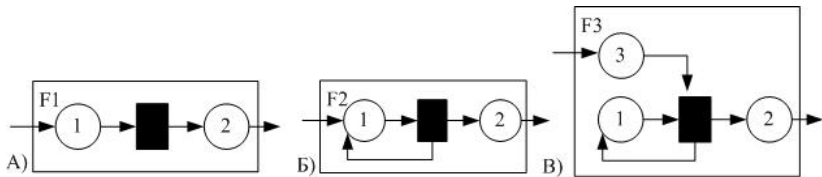


Рис. 2. Ролевые функционалы: А) приемник/передатчик трафика; Б) независимый генератор трафика; В) зависимый генератор трафика

$F(n, 0) = (0, n)$, т.е. при входе в позицию 1 n меток и после срабатывания t во входной позиции останется 0 сообщений, а в выходной позиции 2 появится n меток.

Функционал – генератор трафика независимый вариант (рис. 2Б): момент поступления новых меток в позицию 2 зависит только от временной задержки перехода.

$F(n, 0) = (n, n)$; т.е. при входе в позицию p_1 n меток и после срабатывания t_1 во входной позиции опять будет n сообщений, а в выходной позиции появятся n меток.

Функционал – генератор трафика зависимый вариант (рис. 2В): момент поступления новых меток в позицию 2 зависит от временной задержки перехода и от наличия в позиции 3 метки.

$F(n, 0, m) = (n, 0, m + n)$, однако если для позиций 3 и 2 является разрешенным трафик только типа А, для позиции 1, соответственно, – только трафик типа Б, то, с учетом этих наложенных ограничений, $F(n, 0, m) = (n, 0, m)$.

Операции с сетями Петри:

Для целей моделирования ВС определим (по аналогии с алгеброй сетей Петри, порождающей класс регулярных сетей [4, 6]) следующие операции над элементами сетей A_i – позициями p_j , переходами t_k , дугами d_l ($i, j, k, l \in Z$).

1. **Отождествление позиций** « \sim » с параметрами $(A_1, p_k \in A_1; A_2, p_n \in A_2)$ и аналогично - отождествление переходов « \sim » с параметрами $(A_1, t_1 \in A_1; A_2, t_2 \in A_2)$. Отождествление позиций из сетей A_1 и A_2 дает в результате одну общую сеть A с общей позицией p_{kn} . Запись: $(p_k(A_1) \sim p_n(A_2))(p_{kn})$. При этом по умолчанию определим, что $M(p_{kn}) = M(p_k) \cup M(p_n)$. Если новое наименование позиции (p_{kn}) пропущено, то считаем, что позиция сохраняет наименование из первой сети. Для одновременного отождествления нескольких позиций (переходы аналогично) запись будет выглядеть следующим образом:

$((p_{k1}(A_1) \sim p_{n1}(A_2)); \dots; (p_{kM}(A_1) \sim p_{nM}(A_2)))(p_{k1,n1}), \dots, (p_{kM,nM})$.

Инцидентные отождествляемым переходам и позициям дуги сливаются, если имеют одни и те же входы и выходы.

2. **Добавление дуги** « \rightarrow » с параметрами $(A_1, t_k \in A_1; A_2, p_j \in A_2)$. Запись: $(t_k(A_1) \rightarrow p_n(A_2))$ – для связывания перехода и позиции; запись: $(p_k \rightarrow A_2 t_n)$ с параметрами $(A_1, p_k \in A_1; A_2, t_j \in A_2)$ – для связывания позиции и перехода.

Операция по умолчанию не затрагивает изменения множества типов меток, допустимых для переходов или позиций.

3. **Исключение дуги** « $! \rightarrow$ » с параметрами $(A, d \in A)$, запись: $(! \rightarrow d(A))$. Операция по умолчанию не затрагивает изменения множества типов меток, допустимых для переходов или позиций.

4. **Переименование позиций** « $/$ » с параметрами $(A, p_1 = p'_1)$ и аналогично - переименование переходов « $/$ » с параметрами $(A, t_1 = t'_1)$. Запись – $(p_k/p'_k)(A)$. Переименование может быть полезно для подготовки подсетей к отождествлению при позициях/переходах, имеющих аналогичные имена в объединяемых сетях. По умолчанию переименование никак не влияет на множество типов меток, допустимых для переименовываемых позиций и переходов. Для конструирования иерархических СП, которые моделируют ВС, будем использовать также операции объединения СП и их расщепления. Объединение является отождествлением позиций и переходов в объединяемых СП:

5. **Объединение СП** « $\&$ » с параметрами $(A_1, A_2; p_1, \dots, p_k \in A_1, p'_1, \dots, p'_k \in A_2; t_1, \dots, t_s \in A_1, t'_1, \dots, t'_s \in A_2)$. Объединение в общем случае является групповой операцией отождествления множеств позиций и переходов двух сетей.

$$A' = ((p_1(A_1) \sim p'_1(A_2))(p'_1); \dots; (p_k(A_1) \sim p'_k(A_2))(p'_k); \\ (t_1(A_1) \sim t'_1(A_2))(t'_1); \dots; (t_s(A_1) \sim t'_s(A_2))(t'_s)).$$

6. **Расщепление СП** « $\%$ » с параметрами $(A_1, A_2; t \in A_1, t'_1, t'_2 \in A_2)$, где t – переход из первой сети, который заменяется на вторую сеть в переходах t'_1 и t'_2 . Записывается как $A = A_1(t) \% A_2(t'_1, t'_2)$. Расщепление можно представить через операции « \sim » и « $! \rightarrow$ », обозначив за $d^-_1(t), \dots, d^-_k(t)$ дуги из A_1 , являющиеся входными для t ($d(p_1 t), \dots, d(p_k t)$), и за $d^+_s(t), \dots, d^+_r(t)$ – дуги из A_2 , являющиеся выходными для t ($d(tp_s), \dots, d(tp_r)$).

$$A' = ((t(A_1) \sim t'_1(A_2))(t'_1); (t(A_1) \sim t'_2(A_2))(t'_2)),$$

$$A'' = (! \rightarrow (d^-_1(t'_2), \dots, d^-_k(t'_2))(A'')),$$

$$A = A''' = (! \rightarrow (d^+_s(t'_1), \dots, d^+_r(t'_1))(A''')).$$

То есть СП A' получается отождествлением перехода t исходной СП A и переходов t'_1 и t'_2 сети A_1 . При таком отождеств-

лении вместо дуг $d_1^-(t), \dots, d_k^-(t)$ и $d_s^+(t), \dots, d_r^+(t)$ в новой сети A' появляются дуги $d_1^-(t'), \dots, d_k^-(t'), d_s^+(t'), \dots, d_r^+(t')$, а также $d_1^-(t''), \dots, d_k^-(t''), d_s^+(t''), \dots, d_r^+(t'')$.

Теперь любой объект в модели ВС может быть представлен с помощью вышеописанных функционалов $F1, F2, F3$ (рис. 3) и операций $\sim, \rightarrow, !\rightarrow, /$. Приведем несколько примеров.

Пример 1. Построим СП для рабочей станции с одним сетевым интерфейсом. Пусть рассматриваемый сетевой узел принимает сообщения-запросы, обрабатывает их и отправляет в сеть сообщения-ответы. В этом случае узел является приемником трафика типа α и зависимым от меток типа α генератором трафика типа β (рис. 3а). Получим сеть D путем связывания позиции 1 сети B и позиции 3 сети C' (добавлена дуга α).

$D = p_1(B) \rightarrow p_3(C')$, где $M(B) = \alpha, M(C') = \beta$. $M(D)$ изменим следующим образом: $M(p_1) = M(p_2) = M(p_3) = \alpha; M(p_4) = M(p_5) = \beta$.

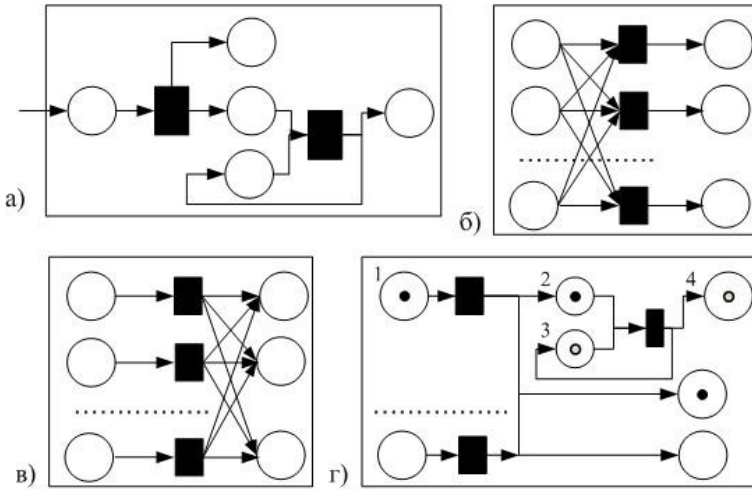


Рис. 3. Модели сетевых устройств: а) рабочая станция; б) концентратор; в) коммутатор-мост; г) маршрутизатор

Пример 2. Узел-концентратор (*hub*) принимает сообщения с одного порта, передает на все. Каждый порт выполняет функцию разветвленного передатчика трафика (рис. 3б). Для каждо-

го i (счетчик портов концентратора) первоначально задается исходная сеть типа A – передатчик трафика (рис. 2а). На каждом шаге цикла по j (счетчик портов) новая сеть типа A объединяется с сетью, полученной на предыдущем шаге, путем отождествления входных позиций и переходов. В итоге для каждого i получается СП, соответствующая одному входному порту с выходными позициями для всех остальных портов. Далее эти сети объединяются путем отождествления соответствующих выходных позиций.

Алгоритм построения устройства типа *hub*:

Для ($i = 1; i \leq \text{NumPorts}; i++$) Цикл
 $A_i = A_{i,1} = A(p_1/p_{1,i}, p_2/p_{2,1});$
 Для ($j = 1; j < \text{NumPorts}; j++$) Цикл
 $A_{i,j+1} = A(p_1/p_{1,i}, p_2/p_{2,j+1});$
 $A_i = (A_i, p_{1,i} \sim A_{i,j+1}, p_{1,i}; A_i, t \sim A_{i,j+1}, t)(p_{1,i})(t_{ij})$
 КонецЦикла по j
 Если ($i = 1$) Тогда
 $A = A_i;$
 Иначе
 $A = (A, p_{2,1} \sim A_i, p_{2,1}; A, p_{2,2} \sim A_i, p_{2,2}; \dots$
 $A, p_{2,\text{NumPorts}} \sim A_i, p_{2,\text{NumPorts}});$
 КонецЕсли
 КонецЦикла по i .

Пример 3. Узел-мост (*switch*) принимает сообщения, анализирует, передает на один/несколько/все порты (рис. 3в).

Пример 4. Узел-маршрутизатор (*router*) принимает сообщения, анализирует и, при передаче из одной среды передачи в другую, преобразует сообщение, передает его на один, несколько или все порты (рис. 3г).

Позиции 1–4 на рис. 3г вместе с переходами моделируют ситуацию перехода пакета в сеть с иной средой передачи (например, входящий в позицию 1 пакет является пакетом *Ethernet*, конечный адресат пакета находится в сети *Token Ring*), в этом случае механизм преобразования следующий:

- пакет из позиции 1 переходит в позицию 2 с задержкой обработки;
- позиция 2 является стоком для сообщений *Ethernet*, после того как для этого пакета сгенерирован новый пакет *Token Ring*;
- в позиции 3 находится метка – пакет *Token Ring*;
- мгновенный переход к позиции 4 становится разрешенным, и в выходной позиции 4 появляется пакет *Token Ring*, за счет того, что для позиции 4 запрещены пакеты типа *Ethernet*;
- если преобразования типов не требуется, пакет сразу переходит от входной позиции к выходной (например, из позиции 1 в 5).

Приведем алгоритм построения СП для маршрутизатора на основе аналогичного алгоритма для концентратора. Отметим, что в данном случае для моделирования связи между разными средами передачи используется сеть типа «приемник» вместо сети типа «передатчик», чтобы не оставлять открытую дугу из выходной позиции. В то же время за счет слияния подсети «приемник» и подсети «зависимый генератор» получаем сеть, передающую трафик.

Для ($i = 1; i \leq \text{NumPorts}; i++$) Цикл

Если (выходной порт j подключен к иной среде передачи, чем входной порт i), Тогда

$$A'_1 = B(p_1/p_{1,i}, p_2/p_{i,3});$$

$$A'_2 = C'(p_1/p_{3,i}, p_2/p_{2,1}, p_3/p_{i,3});$$

$$A_i = A_{i,1} = (A'_1, p_{i,3} \sim A'_2, p_{i,3}); M(p_{i,3}) = M(p_{1,i});$$

$$M(p_{3,i}) = M(p_{2,1})$$

Иначе // среда передачи одинаковая

$$A_i = A_{i,1} = A(p_1/p_{1,i}, p_2/p_{2,1});$$

Конец Если;

Для ($j = 1; j < \text{NumPorts}; j++$) Цикл

Если (выходной порт j подключен к иной среде передачи чем входной порт i), Тогда

$$A'_1 = B(p_1/p_{1,i}, p_2/p_{i,3});$$

$$A'_2 = C'(p_1/p_{3,i}, p_2/p_{2,j+1}, p_3/p_{i,3});$$

$$A_{i,j+1} = (A'_1, p_{i,3} \sim A'_2, p_{i,3}); M(p_{i,3}) = M(p_{1,i});$$

$$M(p_{3,i}) = M(p_{2,j+1})$$

Иначе // среда передачи одинаковая

$$A_{i,j+1} = A(p_1/p_{1,i}, p_2/p_{2,j+1});$$

Конец Если;

$$A_i = (A_i, p_{1,i} \sim A_{i,j+1}, p_{1,i})$$

Конец Цикла по j

Если ($i = 1$) Тогда

$$A = A_i;$$

Иначе

$$A = (A, p_{2,1} \sim A_i, p_{2,1}; A, p_{2,2} \sim A_i, p_{2,2}; \dots;$$

$$A, p_{2,NumPorts} \sim A_i, p_{2,NumPorts});$$

Конец Если

Конец Цикла по i

Пример 5. Полудуплексный канал с потерями (рис. 4А).
 Позиция 8 – содержит метку особого типа – признак занятости ресурса. Переходы t_1^0, t_4^0 – мгновенные, t_2, t_3 – с задержками обслуживания. Позиция 7 – «сток» для недоставленных сообщений.

Прямой алгоритм построения такой сети следующий:

- | | |
|--|--|
| 1. $V_1 = B(t/t_1^0);$ | 11. $W_2 = A(p_1/p_5, p_2/p_6, t/t_2);$ |
| 2. $V_2 = A(p_1/p_2, p_2/p_3, t/t_2);$ | 12. $W_3 = B(p_1/p_8, t/t_4^0);$ |
| 3. $V_3 = B(p_1/p_8, t/t_1^0);$ | 13. $W_4 = B(p_1/p_5, p_2/p_8, t/t_3);$ |
| 4. $V_4 = B(p_1/p_2, p_2/p_8, t/t_2);$ | 14. $W_5 = B(p_1/p_5, p_2/p_7, t/t_3);$ |
| 5. $V_5 = B(p_1/p_2, p_2/p_7, t/t_2);$ | 15. $W' = p_5(W_1) \sim p_5(W_2);$ |
| 6. $V'' = (p_2(V_1) \sim p_2(V_2));$ | 16. $W'' = (t_4^0(W') \sim t_4^0(W_3);$ |
| 7. $V''' = (t_1^0(V') \sim t_1^0(V_3);$ | $p_5(W') \sim p_5(W_3));$ |
| $p_2(V') \sim p_2(V_3);$ | 17. $W''' = (t_3(W'') \sim t_3(W_4);$ |
| 8. $V'''' = (t_2(V'') \sim t_2(V_4);$ | $(p_5(W'') \sim p_5(W_4);$ |
| $p_2(V'') \sim p_2(V_4); p_8(V'') \sim p_8(V_4));$ | $p_8(W'') \sim p_8(W_4);$ |
| 9. $V = (p_2(V''') \sim p_2(V_5);$ | 18. $W = (p_5(W''') \sim p_5(W_5);$ |
| $t_2(V''') \sim t_2(V_5);$ | $t_3(W''') \sim t_3(W_5);$ |
| 10. $W_1 = B(p_1/p_4, p_2/p_5, t/t_4^0);$ | 19. $VW = (p_8(V) \sim p_8(W); p_7(V) \sim p_7(W)).$ |

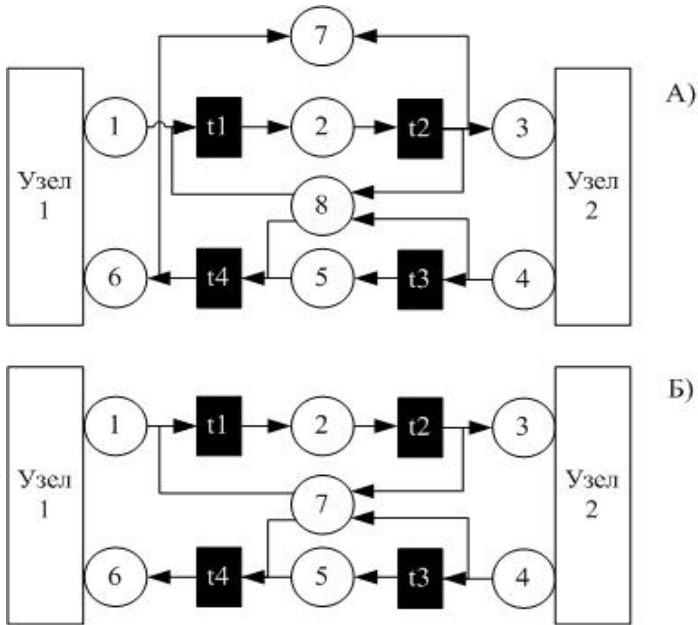


Рис. 4. Каналы связи: А) полудуплексный; Б) полнодуплексный

Можно построить более короткий алгоритм, используя операцию расщепления переходов на подсети, как это сделано в примере 7.

Пример 7. Дуплексный канал с потерями. Строится аналогично полудуплексному, но имеет более простую структуру (рис. 4Б), так как не содержит позиции для меток разделения ресурса и соответствующих связей (дуг и переходов).

Аналогичным образом можно показать конструирование и других типов сетевых устройств.

Приведенные алгоритмы наглядно демонстрируют простоту реализации и имеют линейную сложность для всех представленных типов устройств.

Можно обобщить запись операции отождествления, если опустить параметры отождествляемых позиций и устройств и определить в этом случае отождествление по всем позициям и переходам с идентичным наименованием. Тогда запись

« $C = ((A, t_1; B, t_1); (A, p_2; B, p_2))$ » будет аналогична записи « $C = (A; B)$ » при условии, что все элементы (позиции и переходы) в обеих сетях имеют различное наименование, кроме t_1 и p_2 .

Можно также определить операцию макетирования на основе операции переименования, например, для сети типа $B -$ «приемник»: « $C = ((B, 1/B, p_1); (B, 2/B, p_2))$ », будет аналогична записи « $C = B(p_1, p_2, t)$ ».

1. $V_1 = B(p_1/p_5, t/t_4, p_2/p_6)$;
2. $V_2 = B(p_1/p_5, t/t_4, p_2/p_7)$;
3. $V_3 = B(p_1/p_8, t/t_6, p_2/p_6)$;
4. $V_4 = B(p_1/p_8, t/t_6, p_2/p_7)$;
5. $V_5 = B(p_1/p_6, t/t_3, p_2/p_5)$;
6. $V_6 = B(p_1/p_6, t/t_5, p_2/p_8)$;
7. $V' = (p_5(V_1) \sim p_5(V_2))$;
 $(t_4(V_1) \sim t_4(V_2))$;
8. $V'' = (p_5(V') \sim p_5(V_5))$;
 $p_6(V') \sim p_6(V_5)$;
9. $V^* = (B, t_1) \% (V'', t_3, t_4)$;
10. $V' = (p_8(V_3) \sim p_8(V_4))$;
 $t_6(V_3) \sim t_6(V_4)$;
11. $V'' = (p_6(V') \sim p_6(V_6))$;
 $p_8(V') \sim p_8(V_6)$;
12. $V^{**} = (B(p_1/p_3, t_1/t_2, p_2/p_4), t_4) \%$
 $\% (V'', t_5, t_6)$;
13. $W = (p_6(V^*) \sim p_6(V^{**}))$;
 $p_7(V^*) \sim p_7(V^{**})$.

3.3. АЛГОРИТМ РАБОТЫ ПОСТРОЕННОЙ МОДЕЛИ

Ниже приводится обобщенный алгоритм для реализации процессов передачи сообщений в имитационной модели на основе сконструированной и размеченной СП. Под событием понимается перемещение метки из одной позиции в другую, т.е. множество вида $S = \langle P_1, P_2, T, tr, t \rangle$, где P_1, P_2 – соответственно входная и выходная позиции; T – переход; tr – тип трафика; t – момент времени, в которое произойдет событие.

Общий алгоритм:

- *Определение размера временного шага для модели.*
- *Определение кол-ва шагов N (времени моделирования).*
- *Установка признака модифицированности сети Петри (ПМСП) = 1;*
- *Для $i = 1$ по N Цикл (i – счетчик шагов)*

Проверка модифицированности сети Петри, установка ПМСП = 0/1.

~ Если ПМСП = 1, Тогда

Цикл по всем источникам трафика

Цикл по всем потокам

Вычисление маршрутов

Конец Цикла

Конец Цикла;

~ Конец Если;

Получение списка событий для данного шага.

~ Если шаг нулевой, тогда

Формирование списка событий:

Цикл по всем источникам трафика

Цикл по всем потокам

Вычисление события $S(P_1)$:

Выбор следующего перехода и следующей позиции (из маршрута), расчет момента t ;

Добавление S в список

Конец Цикла;
Конец Цикла;
~ *Иначе Если список пуст и шаг не нулевой*
– *переход на следующий шаг;*
~ *Иначе*
– *Цикл по событиям из списка*
Проверка на разрешенность события (соответствующего перехода)
Если Событие разрешено –
Выполнение:
Формирование следующего события S_{next} :
Выбор следующего перехода и следующей позиции (из маршрута), расчет момента S_{next} .
Добавление S_{next} в список
Конец Если
Конец Цикла;
~ *Конец Если;*
– *Конец Цикла по i .*

4. Общие принципы применения модели

Практическое применение предложенного имитационного моделирования ВС предполагает следующие этапы:

- I. Определение модельной платформы ВС.
- II. Имитация процессов и событий, происходящих в ВС.
- III. Анализ результатов и верификацию моделей.

I. **Определение модельной платформы ВС** опирается на общую *RDF*-модель (рис. 1) и включает в себя следующие задачи:

1. Создание структурной модели ВС:

1.1. Определение топологии ВС, входящих в ее состав подсетей, устройств обработки трафика (рабочих станций, серверов,

периферийного и коммуникационного оборудования), каналов передачи трафика.

2. Создание структурно-функциональных моделей устройств ВС:

2.1. Описание входящих в состав ВС устройств с точки зрения его функций по обработке трафика;

2.2. Выделение в устройстве взаимосвязанных блоков различных типов: приема-передачи трафика, его обработки, генерации.

3. Создание моделей потоков сетевого трафика:

3.1. Описание типов, источников, потоков трафика и характеристик сетевых пакетов;

3.2. Описание маршрутов потоков трафика.

На текущем этапе реализации подсистема имитационного моделирования и анализа ВС СиМАКС позволяет определить таким образом модельную платформу исследуемой ВС.

II. Сетевой трафик, циркулирующий в ВС, смоделированной таким образом, может быть в общем случае, описан **процессами и событиями**, приведенными в таблице 2.

Эти события могут быть детализированы либо напротив, агрегированы и формализованы в зависимости от целей моделирования. Поскольку приоритетным направлением системы СиМАКС является расчет и анализ задержек на обработку и передачу трафика в сети, в ней моделируются следующие процессы и события (таблица 3).

Таким образом, процессы 1 и 2 объединяются, и исключается процесс моделирования разбиения сообщения на пакеты. Также на данном этапе реализации СиМАКС объединяются процессы обработки сообщения в сетевом интерфейсе и на уровне приложения в сетевом устройстве. Такой подход позволяет рассчитать задержки на передачу пакета и на обработку его в сетевом устройстве. Модель допускает в будущем более высокий уровень детализации процессов за счет усложнения структуры сетей Петри для устройств ВС.

III. Анализ результатов. Реализуемая система использует комплекс методов моделирования ВС. Помимо имитационного моделирования применяется аналитическое моделирование, а также вейвлет-анализ временных рядов, полученных путем замера показателей сетевого трафика.

Таблица 2. Общие процессы и события передачи и обработки сетевого трафика

Процессы	События
1. Процесс генерации сетевого сообщения	Определение источника сообщения
	Определение момента генерации сообщения
	Определение адресата сообщения
	Определение характеристик сообщения: размер, приоритет и т.п.
2. Процесс формирования сетевых пакетов из сообщения	Разбиение сообщения на пакеты
	Добавление к каждому пакету заголовков в соответствии с используемыми в сети протоколами
3. Передача сетевого пакета в зависимости от используемой в сети технологии передачи данных (каналов связи, сетевых интерфейсов, коммуникационного оборудования)	Выбор маршрута
	Передача пакета до очередного узла
4. Обработка сетевого пакета	Обработка на уровне сетевого интерфейса
	Обработка на уровне приложения

Таблица 3. Процессы и события передачи и обработки сетевого трафика СиМАКС

Процессы	События
Процесс генерации сетевых пакетов в соответствии с используемыми в сети протоколами	Определение источника пакетов
	Определение момента генерации пакетов
	Определение адресата пакетов
	Определение характеристик пакетов: размер, приоритет и т.п.
Передача сетевого пакета в зависимости от используемой в сети технологии передачи данных (каналов связи, сетевых интерфейсов, коммуникационного оборудования)	Выбор маршрута
	Передача пакета до очередного узла с задержкой передачи
Обработка сетевого пакета	Обработка в сетевом устройстве с задержкой обработки

5. Апробация модели

Приведем результаты практического использования описанного в статье имитационного подхода, который реализуется в программном комплексе СиМАКС (Система моделирования и анализа компьютерных сетей) в Кемеровском государственном университете

Для расчетов использовалась сетевая статистика по протоколу *UDP* за 2,5 часа в учебной аудитории. На основе собранных данных аналитически (метод «анализа средних» [1]) рассчитывались средняя очередь сообщений в каждом узле $L_i(N)$ и среднее время собственно обработки сообщений (без учета ожидания в очереди) $T_i(N)$. В примере оцениваются параметры

рабочих станций и коммуникационного устройства типа мост для однородного трафика.

Таблица 4. Исходные данные замеров

Сетевой узел	Средняя длительность обслуживания с ожиданием
1	4,09812
2	1,31622
3	2,74547
4	7,75227
5	7,46318
6	0,93915

Маршрутная матрица P_{ij} потоков задается следующим образом:

$$\begin{pmatrix} P_{11} & P_{12} & \dots & P_{16} \\ P_{21} & P_{22} & \dots & P_{26} \\ P_{31} & P_{32} & \dots & P_{36} \\ \dots & \dots & \dots & \dots \\ P_{61} & P_{62} & \dots & P_{66} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0,02 & 0,32 & 0 & 0,03 & 0,49 & 0,14 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

В таблице 5 приведены результаты аналитических расчетов.

Таблица 5. Результаты расчетов

Номер сетевого узла	Средний размер очереди сообщений в узле	Среднее время обработки сообщения в узле
1	0,01343	0,24728
2	1,8201	2,09358
3	7,26422	2,67383
4	0,01062	0,13036
5	0,2167	0,162
6	0,67484	1,77427

Результаты имитационного расчета:

Модель ВС на основе СП отражена на рис. 5.

Для получения результатов модель запускалась на выполнение для двух значений времени выполнения: $T=100$, $T=1000$ и $T=5000$ тактов. Для каждого значения модель выполнялась по три раза, после чего рассчитывались средние значения, приведенные в таблице 6.

Данные таблицы 5 показывают относительную погрешность и ее максимальные значения во всех вариантах работы модели ($T=100$, $T=1000$, $T=5000$).

С увеличением количества тактов погрешность уменьшается, и результаты имитационного моделирования сходятся к результатам аналитического моделирования.

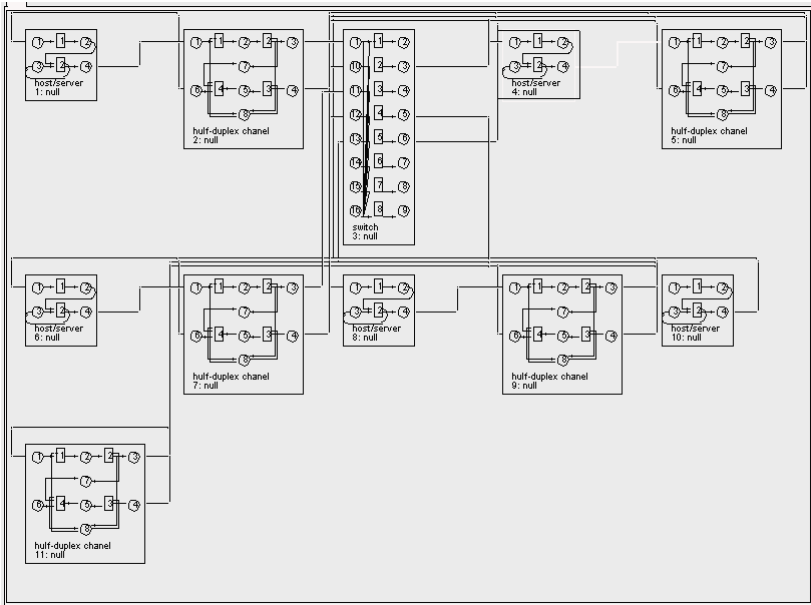


Рис. 5. Имитационная модель ВС

Таблица 6. Результаты работы имитационной модели ($T = 100$)

Номер сетевого узла		1	2	3	4	5	6
Средний размер очереди сообщений в узле	$T = 100$	0,0224	2,171	6,0184	0,0723	0,3601	0,8984
	$T = 1000$	0,012	2,0284	6,849	0,0682	0,2001	0,723
	$T = 5000$	0,0129	1,984	7,191	0,0382	0,221	0,665
Среднее время обработки сообщения в узле	$T = 100$	0,3657	1,8791	2,6733	0,0969	0,1004	2,1321
	$T = 1000$	0,2041	1,9789	2,5309	0,1144	0,1702	1,8802
	$T = 5000$	0,2157	2,089	2,69	0,1312	0,167	1,8247

Таблица 7. Расчет погрешности

№ узла		1	2	3	4	5	6	Max. E (%)
Очередь: отн. погр. (E)	$T = 100$	0,672	0,193	-0,172	0,717	0,662	0,331	72
	$T = 1000$	-0,105	0,114	-0,057	0,151	-0,077	0,071	15
	$T = 5000$	-0,037	0,09	-0,01	0,076	0,019	-0,015	9
Время обработки: отн. погр. (E)	$T = 100$	0,479	-0,103	-0,0001	-0,257	-0,38	0,202	48
	$T = 1000$	-0,174	-0,055	-0,053	-0,123	0,051	0,059	17
	$T = 5000$	-0,128	-0,002	0,006	0,006	0,031	0,028	13

6. Заключение

В работе описан подход к имитационному моделированию архитектуры и передачи данных в вычислительных сетях, на основе модифицированных сетей Петри. Этот механизм целесообразно использовать для сложных задач проектирования сетей и анализа потоков трафика или совмещать описанный подход с аналитическим моделированием (применяя аналитические методы для расчета некоторых характеристик и используя эти результаты для имитационного моделирования).

В работе предложен механизм построения сетей Петри для основных объектов ВС, формализованы функции сетевых объектов в виде примитивных СП – функционалов. Предложены операции над СП, позволяющие легко конструировать модель ВС. Для демонстрации построенной модели, приведен пример построения СП для фрагмента локальной сети, рассмотрены вопросы маршрутизации в имитационной модели, предложен укрупненный алгоритм работы имитационной модели.

Полученные результаты используются для практической реализации программного комплекса СиМАКС (Система моделирования и анализа компьютерных сетей), который разрабатывается в Кемеровском государственном университете и используется для анализа топологии и связностей корпоративной сети.

Литература

1. ВИШНЕВСКИЙ В.М. *Теоретические основы проектирования компьютерных сетей*. – М.: Техносфера, 2003.
2. ДУБИНИН В.Н. *Организация и проектирование интеллектуальных распределенных вычислительных систем с групповыми взаимодействиями* // Вычислительная техника в автоматизированных системах контроля и управления: Межвуз. сб. науч. тр. – Пенза: ПГУ, 1999. – Вып.26. – С. 31–38.

3. ИЛЬНИЦКИЙ С.В. *Работа сетевого сервера при самоподобной нагрузке*. – URL: <http://314159.ru/ilnickis/ilnickis1.pdf>.
4. КОТОВ В.Е. *Сети Петри*. – М.: Наука, 1984.
5. МИКОВ А.И., ЗАМЯТИНА Е.Б. *Система имитации с удаленным доступом* // Материалы третьей междисциплинарной конференции с международным участием (НБИТТ–21), 21–23 июня 2004 г., Петрозаводск. – С. 73.
6. ПИТЕРСОН ДЖ. *Теория сетей Петри и моделирование систем*. – М.: Мир, 1984.
7. *Среда описания ресурса (RDF): Понятия и абстрактный синтаксис*. – URL: http://www.w3.org/2007/03/rdf_concepts_ru/Overview.html (дата обращения 02.04.2009).
8. ТАРАСОВ В.Н. *Вероятностное компьютерное моделирование сложных систем для анализа их производительности*: Дисс. докт. техн. наук. – Оренбург, 2002.
9. ОЛИФЕР Н.А., ОЛИФЕР В.Г. *Средства анализа и оптимизации локальных сетей*. – URL: <http://www.citforum.ru/nets/optimize/index.shtml> (дата обращения 28.04.09).
10. ЯРОСЛАВЦЕВ А.Ф. *Методы и программные средства гибридного моделирования мультисервисных сетей большой размерности*: Дисс. докт. техн. наук. – Новосибирск, 2006.
11. *HP OpenView Network Node Manager*. – URL: http://www.openview.ru/nnm_2.htm (дата обращения 24.02.2009).
12. LELAND W.E., MURAD S. TAQQU, WILLINGER W., SHERMANN R. *Proof of a Fundamental Result in Self-Similar Traffic Modeling* / ACM SIGCOMM'95, San-Fransisco, 1995.

TRAFFIC TRANSFER PROCESSES SIMULATION IN COMPUTING NETWORKS

Alexander Goodov, Kemerovo State University, Kemerovo, Cand.Sc., assistant professor (good@kemsu.ru).

Marina Semekhina, Kemerovo State University, Kemerovo, post-graduate (semekhina@kru.ru).

Abstract: We consider an approach to simulation of local area computing networks. The approach uses colored Petri networks with time mechanism and queues. We offer the notion of role functionals and define operations on them. We also provide an example of Petri net design for the fragment of a local area network and suggest the general algorithm of simulation run.

Keywords: computing networks modeling, queuing theory, simulation modeling, Petri network.

*Статья представлена к публикации
членом редакционной коллегии В. М. Вишневым*

Уважаемые читатели! С радостью сообщаем о выходе Специального выпуска 30.1 «Сетевые модели в управлении» Сборника трудов «Управление большими системами», посвященного применению подходов теории графов к постановке и решению задач управления. Подробная информация о Спецвыпуске, включая полные тексты статей, доступна на официальном сайте Сборника трудов «Управление большими системами» ubs.mtas.ru.