

ПРИМЕРЫ РЕАЛИЗАЦИЙ СИСТЕМ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

Наместников А.М., Тронин В.Г.

В статье приводится обзор двух распространенных систем имитационного моделирования NS2 и GPSS применительно к задачам проектирования вычислительных сетей.

Введение

Имитационное моделирование является весьма широким и недостаточно четко определенным понятием, имеющим достаточно весомое значение для лиц, занимающихся созданием и тестированием практически любых систем, в том числе и информационных. Сама идея имитационного моделирования проста и интуитивно привлекательна [1-3]. Она дает возможность экспериментировать с системами (существующими или предлагаемыми) в тех случаях, когда делать это на реальном объекте практически невозможно или нецелесообразно. В данной статье будут представлены результаты моделирования вычислительных сетей на основе двух систем: NS2 и GPSS.

1. Имитационное моделирование на основе NS2

Система имитационного моделирования NS2. 1996 г. ознаменован началом работ над проектом VINT (Virtual InterNetwork Testbed), организованным DARPA (Defense Research Projects Agency) и реализуемым под руководством целого ряда научных организаций и центров: USC/ISI (University of Southern California / Information Sciences Institute), Xerox PARC, LBNL (Lawrence Berkley National Laboratory) и UCB (UC Berkley). На сегодня основными спонсорами проекта являются DARPA, NSF и ACIRI (AT&T Center for Internet Research at ICSI). В качестве программной реализации был выбран разрабатываемый в University of California с 1989 года пакет network simulator (до 1995 года известный как REAL). Для программного продукта было выбрано имя network simulator 2 (далее – NS2).

Одним из отличительных свойств NS2, с точки зрения гибкости, является мультиоперационность. Полные версии, включающие все функции, на данный момент работоспособны под управлением следующих операционных систем SunOS, Solaris, Linux, FreeBSD, Windows 95/98/ME/NT/2000.

Проектирование имитационной модели телекоммуникационной сети УлГТУ в NS2. В настоящее время университетская сеть включает в себя около 860 хостов (в том числе территориально-распределенных), около 70 серверов внутренней и внешней сети, а также множество сетевых сервисов.

Объекты имитационного моделирования. В роли объектов имитационного моделирования будем понимать телекоммуникационные каналы, узлы, транспортные протоколы и генераторы трафика, образующие сегмент топологии сети УлГТУ в среде сетевого симулятора NS2. Схематично выбранный сегмент представлен на рис. 1.

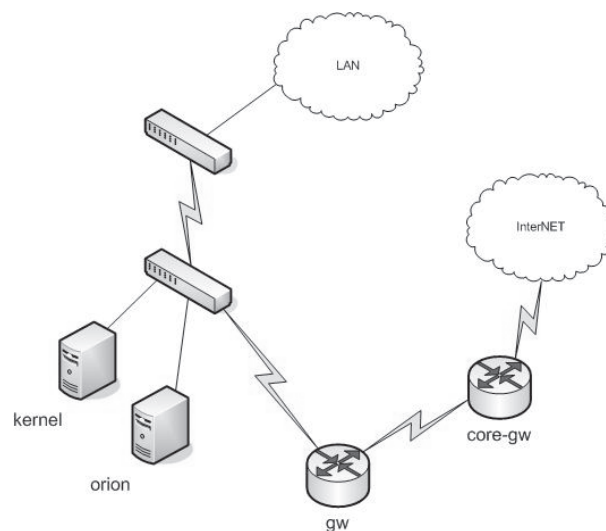


Рис. 1. Сегмент моделирования сети

Выбор произведен по принципу – наиболее важные критичные компоненты корпоративной сети, внимание которым должно уделяться в первую очередь.

Разработка модели в среде NS2. Рассмотрим базовые принципы создания модели в среде NS2. Инициализация модели в среде эмулятора производится командой `set ns [new Simulator]`, которая создает объект класса Simulator (соответственно, объекту ns доступны методы унаследованного класса). Для отражения результатов имитации используются два подхода: запись трэйсов (trace) в файл (данные о пакетах) и (или) запись сценария анимации (nam). Для этого следует использовать метод `open`, например:

```

set tfile1 [open simul.tr w]
$ns trace-all $tfile1
set namf1 [open simul.nam w]
$ns namtrace-all $namf1

```

Таким образом, будут созданы файлы трэйсов и анимации: *simul.tr* и *simul.nam*: соответственно для последующего анализа трафика процесса моделирования, и для анимирования результатов с помощью утилиты *nam*. Для завершения процесса имитации используется метод *finish*, с помощью которого не только закрываются файлы результатов, но и определяются действия, которые необходимо выполнить по окончании моделирования (например – удаление объектов, запуск аниматора, обработка результатов). Запуск и останов имитации производится с помощью методов *\$ns run* и *\$ns "finish"*, соответственно.

Узлы и каналы связи. Для описания узлов имитационной модели используется метод *node* класса Simulator - *set node [\$ns node]*. Для идентификации узлов при визуализации результатов с помощью аниматора NAM возможно использование метода *label* - *\$node0 sample_node*, использование цветовой гаммы реализовано методом *color* - *\$ns color \$i red*. Ниже представлен листинг блока описания узлов разработанной модели:

```

# телекоммуникационный сервер
set kn0 [$ns node]
$ns at 0.0 "$kn0 label kernel.ulstu.ru"
# телекоммуникационный сервер
set or0 [$ns node]
$ns at 0.0 "$or0 label orion.ulstu.ru"
# маршрутизатор
set gw0 [$ns node]
$ns at 0.0 "$gw0 label gw.ulstu.ru"
# корневой маршрутизатор
set coregw0 [$ns node]
$ns at 0.0 "$coregw0 label core-gw.ulstu.ru"
# локальная сеть
set lan0 [$ns node]
$ns at 0.0 "$lan0 label LAN"
# внешняя сеть Интернет
set ext0 [$ns node]
$ns at 0.0 "$ext0 label EXTERNAL"
# коммутатор 3 уровня
set sw0 [$ns node]
$ns at 0.0 "$sw0 label SW-0"
# коммутатор 3 уровня
set sw1 [$ns node]
$ns at 0.0 "$sw1 label SW-1"

```

После создания узлов необходимо описать каналы связи между ними с помощью методов *duplex-link* либо *simplex-link* - *\$ns duplex-*

link \$node0 \$node1 10Mb 10ms DropTail. Метод *duplex-link* описывает двухстороннее соединение, тогда как *simplex-link* описывает однонаправленное соединение (например – спутниковые однонаправленные каналы связи). В симуляторе любая исходящая очередь узла представлена как часть каждого соединения входа данного узла. В описании соединения необходимо указать метод обработки переполнения очереди. Следует заметить, что существует такие опции как Random Early Discard, Fair Queuing, Deficit Round Robin, Stochastic Fair Queuing и DropTail. С помощью визуализатора NAM может быть представлена топология моделируемой сети (рис. 2).

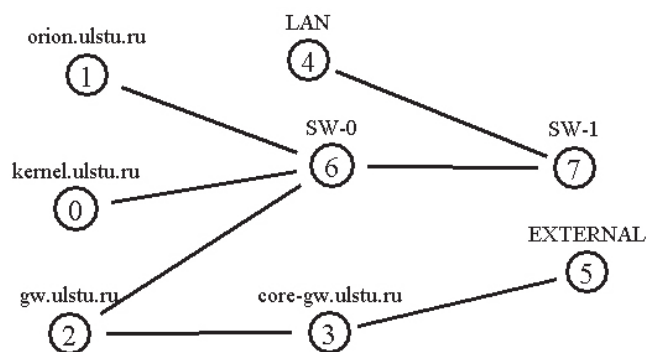


Рис. 2. Топология моделируемой сети, полученная с помощью визуализатора NAM

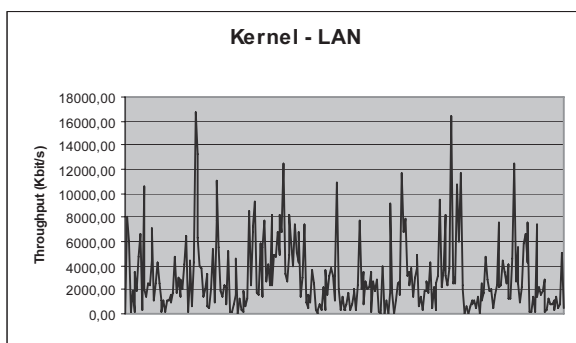
Проведение экспериментов. Для оценки «узких мест» топологии проводились эксперименты, с изменением параметров модели, а именно – характеристик коммуникационных каналов.

Таблица 1. Параметры модели

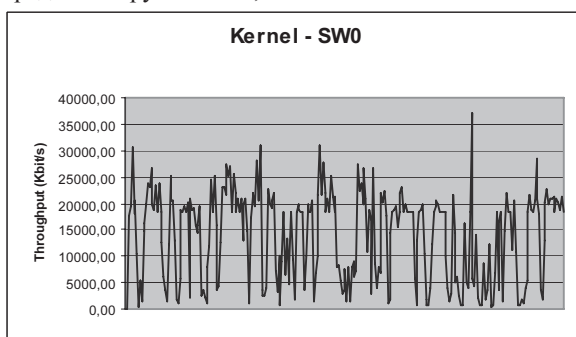
№	Наименование	Узлы	CBR	Среда передачи
1	Канал kernel – sw0	\$kn0 - \$sw0	100Mb/s	100BaseT
2	Канал orion – sw0	\$or0 - \$sw0	100Mb/s	100BaseT
3	Канал gw – sw0	\$gw0 - \$sw0	10Mb/s	10BaseT
4	Канал gw - coregw	\$gw0 - \$coregw0	2Mb/s	E1
5	Канал coregw - UVEN	\$ext0 - \$coregw0	512Kb/s	10BaseT
6	Канал sw0 – sw1	\$sw0 - \$sw1	1Gb/s	1000Base-SX
7	Канал sw1 – ЛВС	\$sw1 \$lan0	100mbit/s	100BaseT

В таблице 1 приведены исходные данные и результаты некоторых экспериментов, проводимых с целью выявления «узких» нагруженных участков сети.

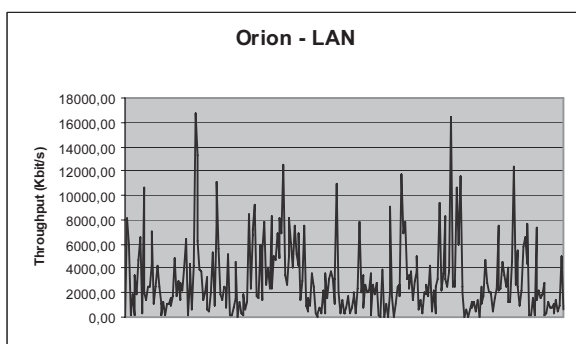
При заданной рабочей нагрузке получаем результаты, показанные на рис. 3.



Теоретическая пропускная способность: 100 Мбит/с
Пиковая нагрузка: 16698,90 Кбит/с
Средняя нагрузка: 3078,31 Кбит/с



Теоретическая пропускная способность: 100 Мбит/с
Пиковая нагрузка: 36971,20 Кбит/с
Средняя нагрузка: 13923,40 Кбит/с



Теоретическая пропускная способность: 100 Мбит/с
Пиковая нагрузка: 16698,90 Кбит/с
Средняя нагрузка: 3133,01 Кбит/с.

Рис. 3. Результаты моделирования

Имитационное моделирование с применением инструментария GPSS

Система имитационного моделирования GPSS. Система GPSS World, разработанная компанией Minuteman Software (США), является мощной средой компьютерного моделирования общего назначения, разработанная для профессионалов в области моделирования. Можно выделить следующие причины популярности пакета:

- наиболее важные классы объектов (требования (транзакты), каналы, накопители и др.) и

их свойства широко используются в реальных вычислительных сетях и т.д.;

- доступно широкое использование анимации;
- пользователи способны легко понять внутреннюю логику и алгоритмы GPSS;
- интерфейс прост и удобен;
- позволяет оперировать непосредственно понятиями имитируемой системы.

Формализованные модели систем

Формализованная модель тонкого клиента.

Тонкий клиент реализует только презентационную логику – прикладной интерфейс для пользователя. Алгоритм взаимодействия модели будет выглядеть следующим образом (см. рис. 4).

На диаграмме видно, что основной функцией для тонкого клиента является – генерация запроса, а, следовательно, основная характеристика – частота генерации запроса.

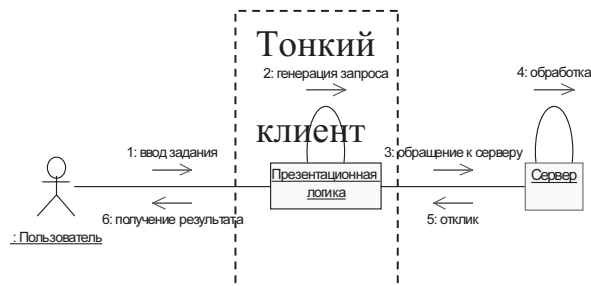


Рис. 4. Модель тонкого клиента

Формализованная модель толстого клиента. Толстый клиент объединяет в себе презентационную логику и бизнес-логику и представляет собой обычную архитектуру персонального компьютера. Ниже приведен алгоритм взаимодействия компонентов данной модели (см. рис. 5).

Основными функциями толстого клиента являются генерация запроса, анализ запроса, обработка запроса. Основные характеристики: частота генерации запроса, время обработки запроса, занятость (выходная переменная).

Свойство занятость является характеристикой компоненты бизнес-логики, поэтому в модели тонкого клиента данное свойство не рассматривалось. Данная переменная показывает загруженность объекта (толстого клиента, сервера) при существующем потоке запросов (наличии определенных бизнес-процессов) и запас бизнес-процессов в сети.

Формализованная модель сервера. Любой сервер характеризуется типом ресурса, к которому он предоставляет доступ. В общем виде, сценарий основного варианта использования можно

представить в виде диаграммы сотрудничества (см. рис. 6).

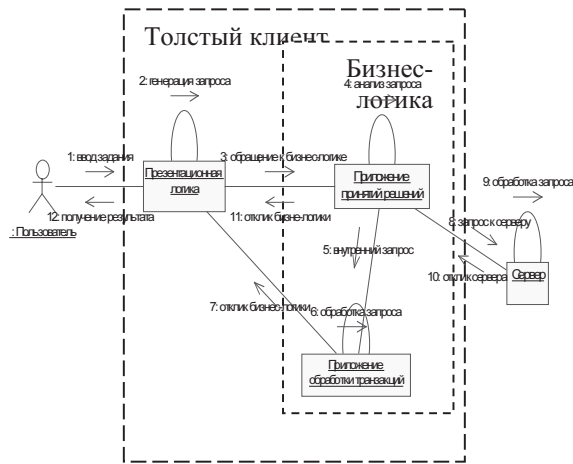


Рис. 5. Модель толстого клиента

Основные функции сервера анализ запроса, обработка отклика запрашиваемого ресурса, занятие ресурса (обработка запроса). Основные характеристики: время отклика ресурса, время обработки запроса, занятость.

Время отклика ресурса зависит от вида сервера (ftp-сервер проверяет параметры авторизации, проху-сервер информация в кэш-памяти или в сети). Время обработки запроса зависит от типа запрашиваемого ресурса, то есть от вида сервера.

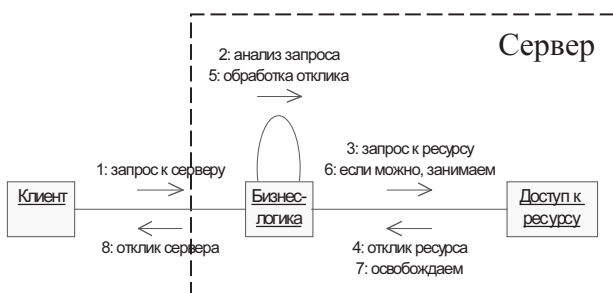


Рис. 6. Модель сервера

В реальных системах оперативной обработки данных используется квантовая технология. Для обслуживания отдельной заявки отводится постоянный квант времени q , достаточный для выполнения нескольких тысяч операций. Если работа была выполнена за время q , она покидает систему. В противном случае она вновь поступает в конец очереди и ожидает предоставления ей очередного кванта процессорного времени.

Результаты моделирования «тонких» клиентов в корпоративной сети

В эксперименте использовались проху-сервер и 20 клиентов. Результаты моделирования сведены в таблице 2.

В таблице 1 представлены только основные характеристики модели. Ниже представлены гра-

фики различных переменных (см. рис. 7), характеризующие данную модель.

Таблица 2. Proху-сервер + 20 тонких клиентов

Переменные	Эксперименты			
	1	2	3	4
Входящие				
Частота генерации запроса (сек)	240	360	24	60
Процент повторного запроса (%)	85	80	90	80
Средний объем запроса категории 1 (байт)	50	20	50	100
Средний объем запроса категории 2 (байт)	500	300	500	400
Быстродействие процессора (байт/с)	600	200	1500	1500
Вероятность нахождения в кэш-памяти ср.значение (%)	60	50	20	40
Вероятность нахождения во внутренней сети (%)	60	30	10	65
Выходящие				
Занятость процессора сервера	0.577	0.556	0.999	0.876
Среднее время ожидания в очереди к процессору	1.09	0.65	1979	185
Занятость операционной памяти	0.001	0.055	0.953	0.446
Занятость устройства внутренней сети	0.328	0.332	0.146	0.926
Среднее время ожидания в очереди к устройству	2.241	6.42	15.26	842
Процент запросов категории 1	0.624	0.568	0.154	0.654
Процент повторов	0	0	4.19	0

Данный эксперимент характеризует ситуацию с небольшой нагрузкой на сеть. Редкая генерация запросов, небольшие объемы запрашиваемой информации, хорошие характеристики сервера, и, следовательно, небольшая загруженность оперативной памяти и средняя загруженность процессора. Загруженность внутреннего устройства получилась выше, так как вероятность попадания на него выше, чем во внешнюю сеть.

Обработка запроса генерируется по нормальному распределению с рассчитываемым средним значением обработки запроса в кэш-памяти, во внутренней сети и внешней.

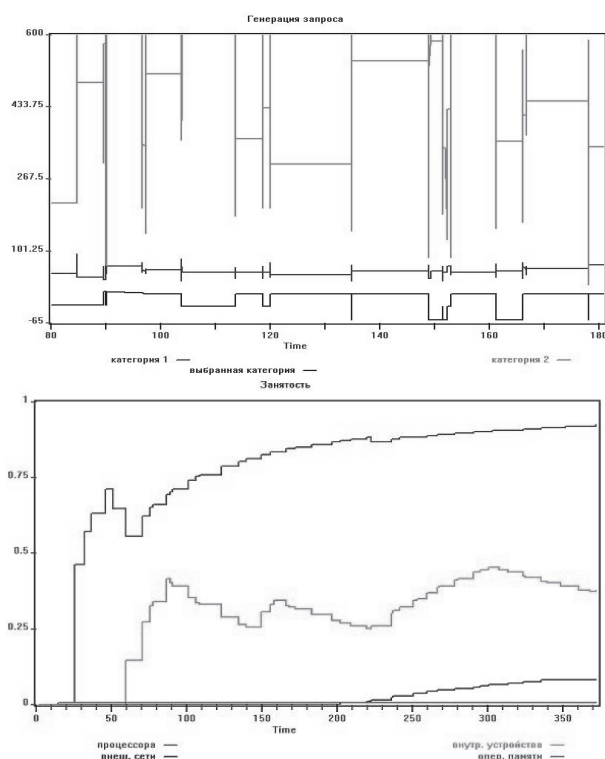


Рис. 7. Результаты моделирования сети, включающей проху-сервер и 20 клиентов

Выводы

На основе вышеприведенного материала можно сделать следующий вывод. Оба рассмот-

ренных инструментария могут быть эффективны для моделирования вычислительных сетей предприятий и организаций. Более гибкой в настройке можно считать систему NS2, поскольку многие функциональные модули проектируются с использованием подключаемых библиотек. Одновременно данная система требует достаточно глубоких знаний в области программирования.

Система GPSS является более простой в применении и не обладает развитыми средствами визуализации и настройки, какие имеются у NS2. Ее можно рекомендовать для построения несложных и неинтегрированных информационных систем моделирования вычислительных сетей.

Литература

1. Шеннон Р. Имитационное моделирование систем: искусство и наука. М.: Мир, 1978. – С. 167-185.
2. Столлингс В. Современные компьютерные сети. 2-е изд. СПб: Питер, 2003. – С. 25-43.
3. Лихтциндер Б.Я., Симонова Е.В. Применение системы имитационного моделирования локальных вычислительных сетей в учебном процессе. Тезисы докладов IV РНМК «Пути и методы совершенствования учебного процесса». Самара, 1995. – С. 12-14.

АСПИРАНТУРА ПГАТИ

Ведет подготовку специалистов высшей квалификации (кандидатов технических и физико-математических наук) по научным специальностям:

- 01.04.03 Радиофизика;
- 05.12.04 Радиотехника;
- 05.12.07 Антенны, СВЧ-устройства и их технологии;
- 05.13.13 Телекоммуникационные системы и компьютерные сети;
- 05.12.13 Системы, сети и устройства телекоммуникаций;
- 05.13.10 Управление в социальных и экономических системах;
- 05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей.

В ДОКТОРАНТУРЕ ПГАТИ подготовка ведется по специальностям:

- 05.12.07 Антенны, СВЧ-устройства и их технологии;
- 05.12.13 Системы сети и устройства телекоммуникаций.

Диссертационный совет Д 219.003.02 принимает к рассмотрению и защите диссертации на соискание ученой степени доктора технических наук и кандидата технических наук по специальностям:

- 05.12.07 Антенны, СВЧ-устройства и их технологии;
- 05.13.13 Телекоммуникационные системы и компьютерные сети;
- 05.12.13 Системы, сети и устройства телекоммуникаций.