

ИСПОЛЬЗОВАНИЕ СПЕЦИАЛИЗИРОВАННОЙ СИСТЕМЫ МОДЕЛИРОВАНИЯ NS-2 ДЛЯ ИССЛЕДОВАНИЯ ТЕЛЕКОММУНИКАЦИОННЫХ СЕТЕЙ

С.В. Храпов

В статье рассматривается проблема моделирования процессов в информационных компьютерных сетях на различных уровнях модели OSI с использованием NS-2.

Введение

Сегодняшний стремительный скачок в развитии телекоммуникационной области однозначно обязан своим появлением компьютерной революции, произошедшей буквально 10 лет назад. Сейчас же мы являемся свидетелями сетевой революции, ибо именно в сети (безусловно, компьютерной) мы находим необходимую информацию, именно в сеть помещаются такие огромные репозитории информации, как Библиотека Конгресса США, библиотека всех печатных работ общества IEEE и т.д. [1]. Не пройдет и десятка лет, как грандиозная часть того, что было создано человеком за всю его историю, будет помещено в сеть. Это лишь пример того, что называется «услуга сети связи». Именно услуги являются той самой силой, которая, с одной стороны, приносит операторам деньги, а с другой стороны – двигает технологию вперед. Если пользователь желает просматривать видеофильмы через сеть, это услуга, которую должен предоставить оператор, а гипотетический производитель оборудования должен, в свою очередь, вкладывать деньги в научно-исследовательскую работу как минимум по следующим направлениям:

- разработка аудио/видео кодеков;
- разработка сетевых протоколов обмена информацией;
- разработка алгоритмов управления перегрузками и т.д.

Мы видим, что услугу с позиции пользователя можно определить как «айсберг», видимую часть которого составляет себестоимость той самой услуги, а огромная подводная часть - затраты производителя и оператора. Известно, что с ростом количества пользователей стоимость услуги резко снижается, т.к. именно пользователям приходится оплачивать весь «айсберг». А насколько велика подводная часть напрямую зависит от используемых методов и технологий оператором и производителем оборудования. В частности, если говорить о сетевых задачах, то необходимость использования компьютерного моделирования разрабатываемой системы очевидна. В нашем примере не последнюю роль в качестве услуги и в формировании ее стоимости будет играть вопрос оптимального распределения ресурсов сети (с точки зрения оператора), а так же используемых технологий (с точки зрения производителя оборудования).

Моделирование динамических систем

Известно, что математическая модель аппроксимирует свойства и поведение исследуемой сети и, как следствие, позволяет решать задачи по оптимизации и ее управлению. Также на модели возможна апробация тех или иных решений, что несравненно дешевле, нежели на реальной системе, и исключает возможные ошибки в ней.

Имитационной является математическая модель, реализованная как программное обеспечение для компьютера и использующая специальные или стандартные языки программирования. При построении подобной модели сети связи могут использоваться как статические, так и динамические модели. При этом под статическими понимаются модели, используемые для исследования состояния сети в заданные моменты времени, например, аналитические методы расчета из теории массового обслуживания, а под динамическими – дискретные стохастические модели, например, процессы генерации

заявок или процессы их обслуживания. Сегодня для решения задач имитационного моделирования сетей связи существует достаточно широкий спектр программных средств – от библиотек функций для стандартных компиляторов до специализированных языков программирования.

Существующее ПО

Сейчас на рынке ПО моделирования сетей связи доступно достаточное количество разноплановых пакетов программ. Если вести речь о многофункциональных пакетах, то среди коммерческих пальму первенства на сегодня занимают следующие:

- COMNET III производства Caci Products Co.;
- BONEs Designer производства Cadence Inc.;
- OPNET Modeler производства Mil3 Inc.

| Производитель и название | Цена | Требования к памяти компьютера | Операционные Системы | Примечания |
|---|----------|--------------------------------|---|--|
| Caci Products Co. COMNET III | \$35.000 | от 32 МБ ОЗУ от 100 МБ HDD | Windows 98/NT/2000/XP SunOS, Solaris | LANs, X.25, ATM, Frame Relay, протоколы маршрутизации IP. Реализация собственного кода на SIMSCRIPT. Анимация. |
| Cadence Inc. BONEs DESIGNER | \$20.000 | от 32 МБ ОЗУ от 80 МБ HDD | SunOS, Solaris, HP-UX | LANs, X.25, ATM, Frame Relay, реализация собственного кода на C++. Анимация. |
| MIL3 Inc. OPNET Modeler | \$40.000 | от 16 МБ ОЗУ от 150 МБ HDD | Windows 98/NT/2000/XP, Solaris, HP-UX | Fixed/wireless LANs, X.25, ATM, Frame Relay, Intelligent Networks, Web caching, http и т.п., реализация собственного кода на C++. Анимация. Исходный код библиотек частично открыт. |
| VINT project Network Simulator version 2 (NS-2) | - | от 8 МБ ОЗУ до 250 МБ HDD | Windows 95/98/ME/ NT/2000/XP, Solaris, SunOS, Linux, FreeBSD HP-UX | Fixed/wireless LANs, X.25, ATM, Frame Relay, Web caching, http, все разновидности TCP и т.п., реализация собственного кода на C++ и tcl/otcl. Анимация. Исходный код полностью открыт. |

Таблица 1. Сравнение ПО моделирования сетей связи

Все перечисленные программные продукты моделирования, в принципе, обладают практически адекватными наборами возможностей, предоставляемых пользователю. Принципиальное различие заключается только в языке программирования, используемом для реализации пользовательских функций и протоколов. В BONEs и OPNET Modeler в качестве базового языка программирования используется C++, в то время как в COMNET – язык SIMSCRIPT. Самым мощным и, как следствие, самым дорогим и емким с точки зрения ресурсов компьютера среди перечисленных является OPNET Modeler, используемый рядом мировых hi-tech корпораций в своих разработках. Сравнение характеристик упомянутого выше ПО моделирования сетей связи приведено в табл. 1.

Проект NS-2

1996 г. ознаменован началом работ над проектом VINT (Virtual InterNetwork Testbed) [2], 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 Berkeley National Laboratory) и UCB (UC Berkley). На сегодня основными спонсорами проекта являются DARPA, NSF и ACIRI (AT&T Center for Internet Research at ICSI).

Главной целью проекта VINT являлось построение программного продукта, позволяющего осуществлять имитационное моделирование сетей связи и обладающего целым рядом характеристик, среди которых – высокая производительность, хорошая масштабируемость, визуализация результатов и гибкость. В качестве основы программной реализации был выбран разрабатываемый в University of California с 1989 г. пакет Network Simulator (до 1995 года известный как REAL). Логично, что для программного продукта было выбрано имя Network Simulator version 2 (далее – NS-2) [3].

NS-2, как и его предшественники, разрабатывался как программное обеспечение с открытым исходным кодом (Open Source code Software - OSS). Такое ПО распространяется бесплатно – без каких-либо ограничений на право использования, модификации и распространения третьими лицами. Таким образом, с точки зрения стоимости NS-2, безусловно, является лидером по сравнению с коммерческим ПО, упомянутым выше. По этой же причине бесплатны и всегда доступны on-line все обновления и дополнения (новые библиотеки, протоколы и т.п.). Еще одним, не менее замечательным свойством программного обеспечения OSS является возможность модификации ядра программы и гибкая настройка в соответствии с требованиями конкретного пользователя. Одним из отличительных свойств NS-2 с точки зрения гибкости является кроссплатформенность. Полные версии, включающие все функции, на данный момент работоспособны под управлением следующих операционных систем:

- SunOS;
- Solaris;
- Linux;
- FreeBSD;
- Windows 95/98/ME/NT/2000/XP.

Для инсталляции полной версии NS-2 необходимо иметь 250 МБ свободного места на диске компьютера и компилятор C++. Существует также упрощенная (компилированная) версия для некоторых ОС, в частности, всех версий Windows, являющаяся не столь гибкой, как полная версия, в частности, невозможно добавлять компоненты, модифицировать ядро и т.п. Однако эта версия очень проста в использовании и не требует глубоких знаний ОС и языка C++. Для функционирования упрощенной версии NS-2 достаточно иметь 3 МБ свободного места на жестком диске компьютера. Но, начиная с версии 2.1b9, полная версия NS-2 может быть скомпонована и ратифицирована под ОС Windows 9x/2000/XP с использованием Cygwin (Cygwin – это Linux-подобная операционная среда под Windows).

Требования к производительности компьютера у NS-2 не столь жестки. В принципе компьютер с процессором 486 может обеспечить приемлемое функционирование даже полной версии NS-2. При необходимости использования NS-2 группой пользователей достаточно иметь инсталлированную полную версию на машине под управлением Unix-подобной ОС. Пользователи могут иметь доступ в режиме терминала к NS-2 и производить необходимые модификации, в том числе и ядра программы, компилируя свою версию в домашнюю директорию. При помощи Nam (Network Animator) возможна анимация полученных результатов.

Архитектура NS-2

NS-2 является объектно-ориентированным ПО, ядро которого реализовано на языке C++. Язык скриптов (сценариев) OTcl (Object oriented Tool Command Language) используется в качестве интерпретатора. NS-2 полностью поддерживает иерархию классов C++ (называемую в терминах NS-2 компилируемой иерархией) и подобную иерархию классов интерпретатора OTcl (называемую интерпретируемой иерархией). Обе иерархии обладают идентичной структурой, т.е. существует однозначное соответствие между классом одной иерархии и таким же классом другой.

Использование двух языков программирования в NS-2 объясняется следующими причинами. С одной стороны, для детального моделирования протоколов необходимо использовать системный язык программирования, обеспечивающий высокую скорость выполнения и способный манипулировать достаточно большими объемами данных. С другой стороны, для удобства пользователя и быстроты реализации и модификации различных сценариев моделирования привлекательнее использовать язык программирования более высокого уровня абстракции. Такой подход является компромиссом между удобством использования и скоростью [4]. В NS-2 в качестве системного языка используется C++, позволяющий обеспечить:

- высокую производительность;
- работу с пакетами потока на низком уровне абстракции модели;
- модификацию ядра NS-2 с целью поддержки новых функций и протоколов.

В качестве языка программирования высокого уровня абстракции используется язык скриптов OTcl, позволяющий обеспечить ряд положительных свойств, присущих языку Tcl/Tk (так как OTcl является объектно-ориентированным расширением языка Tcl):

- простота синтаксиса;
- простота построения сценария моделирования;
- возможность соединения воедино блоков, выполненных на системных языках программирования и простую манипуляцию ими.

```
set ns [new Simulator]

# Создание шести узлов

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]

# Создание связей между узлами

$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail
$ns duplex-link $n3 $n4 0.5Mb 40ms DropTail
$ns duplex-link $n3 $n5 0.5Mb 30ms DropTail

# Установка размера очереди в канале (n2-n3) равного 20

$ns queue-limit $n2 $n3 20
```

Таблица 2. Создание узлов, связей и назначение размера очереди

Объединение для совместного функционирования C++ и OTcl производится при помощи TclCl (Classes Tcl). TclCl является интерфейсом между объектами C++ и OTcl, которым пользуются NS-2 и Nam.

В рассматриваемой архитектуре возможно использование переменных из библиотек C++ объектами OTcl при помощи функции *bind*, в связи с чем переменные C++ могут быть модифицированы через OTcl напрямую. Остальные команды OTcl выполняются путем передачи данных в объект C++ при помощи функции *TclObject::command(int argc, const char*const* argv)*. Для запроса процедуры OTcl из C++ используется класс Tcl. Это может быть полезно, например, при передаче результатов в OTcl из C++. Сценарий моделирования сети связи может быть полностью записан на OTcl, включая параметры линий и узлов, например, задержки, очереди и т.п. Табл. 2 содержит скрипт создания шести узлов и установки связи между узлами.

В случае, если необходимо реализовать какую-либо специфическую функцию, например, дисциплину обслуживания, не реализованную в NS-2 на уровне ядра, используется код на C++. Однако при подобном подходе могут возникнуть неоднозначности. Например, известно, что в NS-2 маршрутизация в основном реализована на OTcl (причем базовый алгоритм Дейкстра – на C++). Если задача состоит в моделировании потоков HTTP, то для работы с каждым потоком достаточно воспользоваться OTcl, при этом обработка на уровне пакетов будет происходить на C++. Реализация однозначно проста. Однако, если количество HTTP потоков в секунду превысит несколько сотен, процедуру работы с потоком лучше перенести в код C++ по причине быстродействия.

Исходная полная версия NS-2 также содержит несколько весьма полезных утилит и средств, например отладчик Tcl, генератор сценариев моделирования и генератор топологии сети. Существует целое множество генераторов топологий, например NTG, RTG, TIRES. В состав же NS-2 входит генератор топологии GT-ITM, при помощи которого можно автоматически создать топологию очень крупной сети без необходимости вручную определять все компоненты сети. Генератор сценариев моделирования, как правило, используется для создания трафика между узлами сети. При моделировании беспроводных сетей этот тип генератора также может быть использован для определения передвижения узлов.

Протоколы

В NS-2 на уровне ядра реализованы почти все известные протоколы сетей связи. Среди наиболее актуальных на данный момент можно отметить следующие: MPLS, IPv6, OSPF, RSVP, протоколы беспроводной связи, web caching и многие другие. Реализовано целое семейство дисциплин обслуживания очередей – RED, WFQ, CBQ, SFQ и т.д. Реализация собственного протокола в рамках NS-2 несложна, если пользователь знаком с принципами построения NS-2, изложенными выше, и с основами программирования протоколов.

К сожалению, весьма слабым местом NS-2 является документация, хотя руководство пользователя существует и всегда доступно в форматах .html и .ps. Это руководство не является исчерпывающим, и некоторые свойства NS-2 в нем не отражены. Поэтому дополнительно к нему рекомендуется использовать другие ресурсы.

Математическая поддержка

Обеспечение сильной математической поддержки позволяет создавать различные виды трафика, начиная от простейшего, подчиняющегося пуассоновскому закону, заканчивая самоподобным. Гибкость архитектуры NS-2 позволяет пользователю реализовывать собственные математические функции на C++.

Модель ошибок

Реализация модели ошибок позволяет моделировать возникновение ошибок на канальном уровне, т.е. искажение или потерю информации. Моделирование возникновения ошибки возможно на уровне битов, пакетов или вероятности в секунду. Возможна также реализация пользовательской модели ошибок.

Визуализация данных

Полная и упрощенная версии NS-2 содержат средство анимации результатов моделирования Nam (Network Animator), реализованное на Tcl/Tk, которое предоставляет графическое воспроизведение проведенного эксперимента – отображение топологии и сети, анимация пакетов, узлов, очередей, различные возможности анализа данных. В качестве входных данных для Nam используются файлы, записанные в процессе функционирования NS-2, т.е. моделирования сети связи.

Заключение

Очевидно, что в настоящее время программный продукт NS-2 является оптимальным средством моделирования телекоммуникационных сетей, по крайней мере, в рамках высшей школы. Проведение экспериментов на базе NS-2 и изучение его архитектуры влечет за собой получение знаний в ряде областей (программирование, телекоммуникации, математика), без которых невозможно существование высококвалифицированного IT-специалиста. На базе NS-2/Nam возможна организация наглядной демонстрации функционирования протоколов и сетевых механизмов, например, влияния дисциплины обслуживания очереди на вероятность потери пакета трафика с разными приоритетами, или сущность различия алгоритмов протокола TCP (slowstart, sliding window, SACK и т.д.).

Весомая часть исследований в области сетей связи в университетах Европы, Азии и США строится на базе NS-2. Почти в каждом университете, готовящем специалистов в области сетей связи, существуют курсы, посвященные NS-2. NS-2 также широко используется и в научно-исследовательских институтах, а в Интернете доступен репозиторий скриптов NS-2 [5, 6].

К сожалению, как показал поиск по ключевым словам с использованием ряда известных поисковых интернет-машин, на территории России NS-2 не используется, или, по крайней мере, используется не широко. Необходимость создания русскоязычной документации на данный момент просто очевидна, а внедрение NS-2 в учебный и исследовательский процессы на территории постсоветского пространства необходимо.

Литература

1. Freericks C. Open Source Standards on Software Process: A Practical Application // IEEE Communication Magazine, April 2001, pp.116–123.
2. Kevin Fall, Kannan Varadhan, and the VINT project, «The ns manual», 2001
3. S. McCanne and S. Floyd, ns Network Simulator. <http://www.isi.edu/nsnam/ns/>
4. Кучерявый Е.А. NS2 как универсальное средство имитационного моделирования сетей связи / Tampere University of Technology, Труды VII международной конференции «Информационные сети, системы и технологии». Минск, 2001.
5. Парамонов А.И. Имитационное моделирование систем и сетей связи. ЛОНИИС, 2000
6. Парамонов А.И. Моделирование сетей связи. ЛОНИИС, 2000.