

# NS2 КАК УНИВЕРСАЛЬНОЕ СРЕДСТВО ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ СЕТЕЙ СВЯЗИ

Кучерявый Е.А.

Tampere University of Technology, Telecommunications Laboratory, P.O.Box 553, FIN-33101, Tampere, Finland, phone: +358 40 7710619, fax: +358 3 3653808, e-mail: yk@cs.tut.fi, <http://www.cs.tut.fi/~yk>

## ПРЕДПОСЫЛКИ

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.. 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.

Т.е. 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).

## 1. МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКИХ СИСТЕМ

Известно, что 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 несравнимо 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 [1,2].

## 2. СУЩЕСТВУЮЩЕЕ ПО

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

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

Все перечисленные программные продукты моделирования, в принципе, обладают практически адекватными наборами возможностей, предоставляемых пользователю. Принципиальное различие заключается только в языке программирования используемого 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 Таблице 1.

## 3. ПРОЕКТ NS2/VINT

1996 год 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) [3].

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 года пакет network simulator (до 1995 года известный как REAL). Логично, что для программного продукта было выбрано имя network simulator 2 (далее – ns2).

Таблица 1. Сравнение ПО моделирования сетей связи.

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	Win 98/NT/2000 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	Win 98/NT/2000, 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 (ns2)	-	от 8 МБ ОЗУ до 250 МБ HDD	Win 95/98/ME/NT/2000, 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.

ns2, как и его предшественники, разрабатывался как программное обеспечение с открытым исходным кодом (open source code software - OSS) [4]. Такое ПО распространяется бесплатно без каких либо ограничений на право использования, модификации и распространения третьими лицами. Таким образом, с точки зрения стоимости ns2 безусловно является лидером по сравнению с ПО упомянутым в разделе 2 статьи – он бесплатен. 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 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.

Для инсталляции полной версии ns2 необходимо иметь 250 МБ свободного места на диске компьютера и компилятор C++.

Существует также упрощенная версия (компилированная) 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 ns2 достаточно иметь 3 МБ свободного места на жестком диске компьютера.

Требования к производительности компьютера у ns2 не столь жестки. В принципе, компьютер с процессором 486 может обеспечить приемлемое функционирование даже полной версии ns2.

При необходимости использования ns2 группой пользователей достаточно иметь инсталлированную полную версию на машине

под управлением Unix-like ОС. Пользователи могут иметь доступ в режиме терминала к ns2 и производить необходимые модификации в том числе и ядра программы компилируя свою версию в домашнюю директорию. Так же при помощи X-сервера возможна анимация полученных результатов (см. ниже).

#### 4. АРХИТЕКТУРА NS2

ns2 является объектно-ориентированным ПО, ядро которого реализовано на языке C++. Язык скриптов (сценариев) OTcl (Object oriented Tool Command Language) используется в качестве интерпретатора. ns2 полностью поддерживает иерархию классов C++ (называемую в терминах ns2 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 ns2 объясняется следующими причинами [5,6]. 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. В ns2 в качестве системного языка используется C++, позволяющий обеспечить:

- высокую производительность;
- работу с пакетом потока на низком уровне абстракции модели;
- модификацию ядра ns2 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 (т.к. 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.

Объединение для совместного функционирования C++ и OTcl производится при помощи TclCl (Classes Tcl). TclCl является интерфейсом между объектами C++ и OTcl, которым пользуются ns2 и nam. Общая архитектура ns2 соответствующая представленным выше подходам приведена на рис.1.

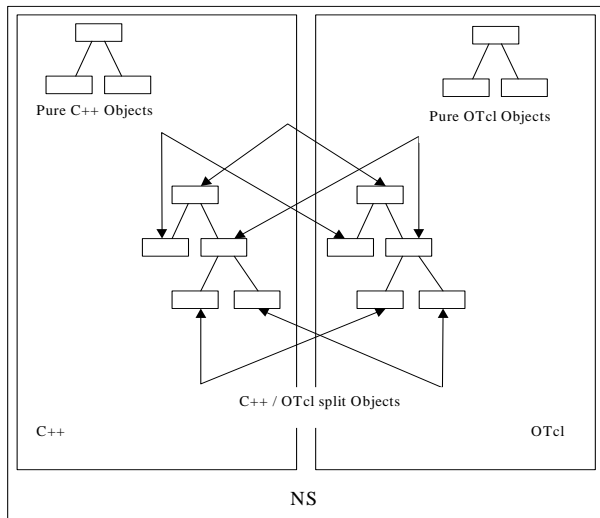


Рисунок 1. Базовая архитектура ns2.

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)*. Процедура проиллюстрирована на рис.2 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. В случае, если необходимо реализовать какую-либо специфическую функцию, например, дисциплину обслуживания не реализованную в ns2 на уровне ядра, то для этого используется код на C++. Однако, при подобном подходе могут возникнуть неоднозначности. Например, известно, что в ns2 в основном маршрутизация 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 быстрейшего действия. Базовые идеи скриптовых языков представлены в [7].

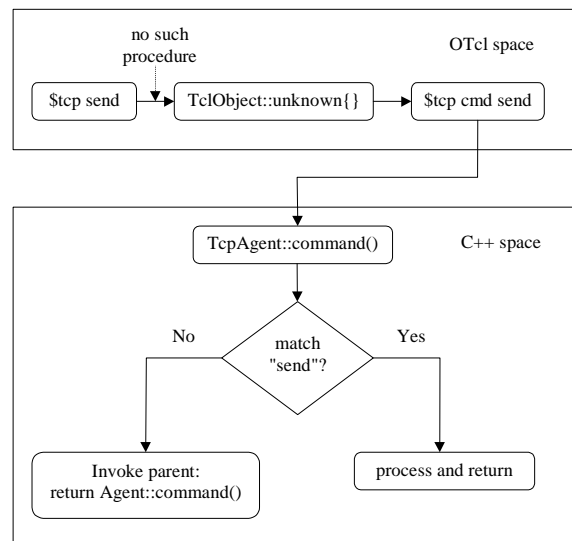


Рисунок 2. Процедура интерпретации OTcl-команды с использованием C++.

Исходная полная версия ns2 также содержит несколько весьма полезных утилит и средств, например отладчик Tcl, генератор сценариев моделирования и генератор топологии сети. Базовая структура каталогов полной версии ns2 представлена на рис.3.

Существует целое множество генераторов топологий, например, NTG, RTG, TIRES. В

состав же ns2 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.

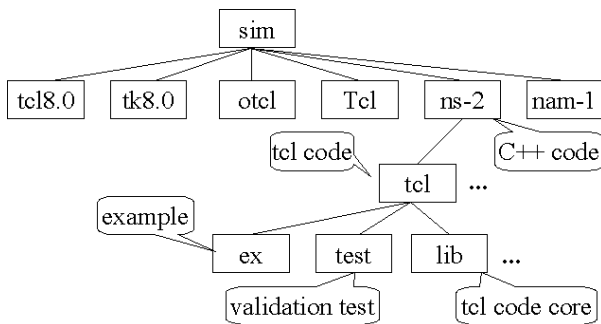


Рисунок 3. Структура каталогов полной версии ns2.

## 5. ПРОТОКОЛЫ

В ns2 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. Так же реализовано целое семейство дисциплин обслуживания очередей: RED, WFQ, CBQ, SFQ и т.д.

Реализация собственного протокола в рамках ns2 не сложна, если пользователь знаком с принципами построения ns2, изложенными в параграфе 4 данной статьи, и с основами программирования протоколов. В первую очередь необходимо написать код C++, в который входит определение форматов пакетов нового протокола (если это необходимо), логики функционирования протокола, необходимых для этого классов C++ и функций *bind* (см. выше) для связи кода с OTcl. Далее модифицируется исходный для ns2 файл *makefile* и производится перекомпиляция ядра. В результате описанных действий пользователь имеет возможность

использовать новый протокол в скрипте сценария моделирования сети связи.

К сожалению, весьма слабым местом ns2 является документация, хотя руководство пользователя существует и всегда доступно в форматах html и ps. Это руководство не является исчерпывающим и некоторые свойства ns2 в нем не отражены. Поэтому дополнительно к нему рекомендуется использовать следующие ресурсы [6,8,9,10].

## 6. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

### 6.1. Математическая поддержка

Обеспечение сильной математической поддержки позволяет создавать различные виды трафика, начиная от простейшего, подчиняющегося пуассоновскому закону, заканчивая самоподобным. Гибкость архитектуры ns2 позволяет пользователю реализовывать собственные математические функции на C++.

### 6.2. 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 так же реализация пользовательской модели ошибок.

## 7. ВИЗУАЛИЗАЦИЯ ДАННЫХ

Полная и упрощенная версии ns2 содержат средство анимации результатов моделирования nam (Network Animator) реализованное на Tcl/Tk [11], которое предоставляет графическое воспроизведение проведенного эксперимента: отображение топологии сети, анимация пакетов, узлов, очередей, и различные возможности анализа данных. В качестве входных данных для nam используются файлы, записанные в процессе функционирования ns2, т.е. моделирования сети связи.

Только полная версия ns2 содержит программное средство Xgraph, позволяющего графически отображать результаты моделирования [12].

## ЗАКЛЮЧЕНИЕ

Очевидно, что в настоящее время программный продукт ns2 является оптимальным средством моделирования сетей связи, по крайней мере в рамках высшей школы. Проведение экспериментов на базе ns2 и изучение его архитектуры влечет за собой получение знаний в ряде областей (программирование, телекоммуникации, математика) без которых невозможно современное существование высококвалифицированного IT-специалиста. На базе ns2/пат возможна организация наглядной демонстрации функционирования протоколов и сетевых механизмов, например, влияния дисциплины обслуживания очереди на вероятность потери пакета трафика с разными приоритетами, или в чем заключается различие 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 ns2. Почти в каждом университете готовящем специалистов в области сетей связи существуют курсы, посвященные ns2 [13,14,15,16,17]. ns2 также широко используется и в научно-исследовательских институтах [18,19], а в Интернет доступен репозиторий скриптов ns2 по адресу [20].

К сожалению, как показал поиск по ключевым словам с использованием ряда известных поисковых интернет-машин, на территории России ns2 не используется, или, по крайней мере, используется не широко. Автором статьи создан интернет-сервер со свободным доступом посвященный ns2, содержащий русскоязычную информацию [6]. Как представляется автору, необходимость подобного информационного ресурса на данный момент просто очевидна и внедрение ns2 в учебный и исследовательский процессы на территории постсоветского пространства необходимо.

## ЛИТЕРАТУРА

[1] Парамонов А.И. «Имитационное моделирование систем и сетей связи», ЛОНИИС, 2000, доступно по адресу <http://www.koulon.sp.ru/modeling/mod.pdf>

- [2] Парамонов А.И. «Моделирование сетей связи», ЛОНИИС, 2000, доступно по адресу <http://www.koulon.sp.ru/modeling1.htm>
- [3] S. McCanne and S. Floyd. ns Network Simulator. <http://www.isi.edu/nsnam/ns/>
- [4] Freericks C. «Open Source Standards on Software Process: A Practical Application», IEEE Communication Magazine, April 2001, pp. 116-123.
- [5] Kevin Fall, Kannan Varadhan, and the VINT project, «The ns manual», 2001
- [6] Сервер с русскоязычной информацией по ns2: <http://www-ns2.da.ru> или <http://yk.atm.tut.fi/ns2/>
- [7] Ousterhout J. «Scripting: High-level programming for the 21<sup>st</sup> century», IEEE Computer, March 1998, #31(3), pp. 23-30.
- [8] Marc Greis' tutorial for the UCB/LBNL/VINT ns2, <http://www.isi.edu/nsnam/ns/tutorial/index.html>
- [9] ns2 related mailing list archive, доступно по адресу <http://www.isi.edu/nsnam/archive/ns-users/webarch/>
- [10] ns2 tutorial pages, доступно по адресу <http://www.isi.edu/nsnam/ns/ns-tutorial/index.html>
- [11] TCL Developer Xchange homepage, доступно по адресу <http://www.scripts.com>
- [12] Xgraph homepage, доступно по адресу <http://jean-luc.ncsa.uiuc.edu/Codes/xgraph>
- [13] Prof. Raj Jain homepage, доступно по адресу <http://www.cis.ohio-state.edu/~jain/>
- [14] Multimedia and computer communications lab, Seoul, Korea, <http://mmlab.snu.ac.kr/>
- [15] Tampere University of Technology, 83950 course, simulation part, доступно по адресу [http://www.cs.tut.fi/kurssit/83950/83950\\_simulointi.shtml](http://www.cs.tut.fi/kurssit/83950/83950_simulointi.shtml)
- [16] Список трудов ns2, доступно по адресу <http://www.isi.edu/nsnam/ns/ns-research.html>
- [17] The Computer Modelling Research Group, University of Bradford, UK, доступно по адресу <http://www.comp.brad.ac.uk/research/performance/HomePage.html>
- [18] Sally Floyd, selected papers, доступно по адресу <http://www.aciri.org/floyd/papers.html>
- [19] Information Science Institute, California, USA, <http://www.isi.edu/>
- [20] Репозиторий скриптов для использования с ns2, <http://www.isi.edu/nsnam/repository/index.html>