

# 基于 NS 的网络仿真\*

## The Network Simulation Based on NS

刘勃兰<sup>1</sup>, 宋玲<sup>1</sup>, 胡小春<sup>2</sup>

Liu Bolan<sup>1</sup>, Song Ling<sup>1</sup>, Hu Xiaochun<sup>2</sup>

(1. 广西大学计算机与电子信息学院, 广西南宁 530004; 2. 广西财经学院, 广西南宁 530001)

(1. School of Comp., Elec. and Info., Guangxi Univ., Nanning, Guangxi, 530004, China; 2. Guangxi College of Finance and Economics, Nanning, Guangxi, 530001, China)

**摘要:** 介绍网络仿真平台 NS (Network Simulation) 仿真原理、仿真方法和一般过程, 结合基于 MobileIP 模型的无线网络与有线网络间的仿真实例, 给出利用 NS 进行网络仿真的方法。仿真结果显示, 通过 NS 能得到很好的网络性能仿真效果。

**关键词:** 网络仿真 无线网络 有线网络

中图分类号: TP391.9 文献标识码: A 文章编号: 1002-7378(2005)S0-0071-04

**Abstract** Introducing the simulation principles, simulation ways and the general process of the network simulation platform-NS (Network Simulator), this paper gives a method of network simulation on NS by a simulation example between a wired network and a wireless network based on the module Mobile IP. The simulation results show a very good simulation effect of network performance that can be gained by NS.

**Key words** network simulation, wireless network, wired network

在网络新技术的研究过程中, 实际网络系统的实现往往是代价最高的或不现实的, 因而网络仿真成为进行网络技术研究的一种基本手段。网络仿真平台 NS (Network Simulator) 正是这样一种针对网络技术的源代码公开的、免费的软件模拟平台, 其协议代码与真实网络应用代码很相似, 仿真结果具有较高的可靠性。网络仿真不仅适用于网络模型的构造和设计, 协议性能的评价和分析, 还适用于网络协议的开发和研究, 以及真实网络的故障诊断。本文在讨论 NS 仿真原理、方法和一般过程的基础上, 给出了基于 MobileIP 模型的无线网络和有线网络的仿真方法。

### 1 NS 的仿真原理、方法和一般过程

NS 是一个面向对象的离散事件驱动的网络模

拟器, 支持多个流行的网络协议, 如 TCP、UDP 和路由调度、拥塞控制算法等<sup>[1]</sup>, 目前的版本是 2.28, 本文的讨论基于这个版本。NS 针对网络仿真已经做了大量的模型化工作, 对网络系统中一些通用的网络实体已进行建模<sup>[2]</sup>。与一般的离散型仿真器相比, NS 的优势在于它有非常丰富的构件库, 而且这些对象易于组合和扩展。用户可以利用这些已有的对象进行少量的扩展, 组合所要研究的网络系统模型, 然后进行仿真。

#### 1.1 NS 的仿真原理

NS 软件包主要包括 Tcl/Tk、Otccl、NS、Tclcl 四部分。软件包各个部分的相互联系如图 1 所示。

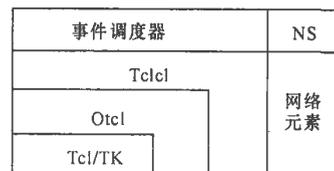


图 1 NS 的内部结构

Otccl 是工具命令语言 Tcl 面向对象的扩展, 是 NS 的仿真描述语言<sup>[3]</sup>。TK 是 Tcl 的图形界面开发工具, 帮助用户在图形环境中开发图形界面<sup>[3]</sup>。NS

收稿日期: 2005-08-13

作者简介: 刘勃兰 (1977-), 男, 湖南人, 硕士研究生, 主要从事计算机网络技术的研究。

\* 广西科学基金项目 (桂科基 0342011) 和广西教育厅科研项目 (桂教科研 [2004]20号) 联合资助。

是整个软件的核心,采用编译和解释双层结构,编译层由 C++ 类组成,它的前端是一个 Otcl解释器。仿真器内核定义了有层次结构的多种类,称为编译类结构;在 Otcl解释器中有相似的类结构,称为解释类结构。用户通过解释器创立新的仿真对象之后,解释器对它进行初始化,与编译类结构中相应的对象建立映射。NS中同时使用两种语言是因为 C++ 语言执行速度快,而修改和编译速度慢,只能用于编译层协议细节的实现;而 Otcl修改快且可用于交互操作,可用于解释层中仿真对象的设置,同时这种分裂对象仿真增强了构件库的可扩展性和可组合性。

NS由事件调度器、网络元素对象库和网络设置模型库三个主要部分组成<sup>[1,2]</sup>。其中调度器是仿真器的重要组成部分,它记录当前时间、调度网络事件链表中的事件。网络元素对象包括节点、链路、代理、业务追踪和数据源等;节点、链路、代理同时继承了 NS Object和事件处理器类,主要用来处理多项事务。

## 1.2 NS的仿真方法和一般过程

用 NS仿真涉及到两个层次:一个是基于 Otcl编程的层次,利用 NS具有的网络元素实现仿真,只要编写 Otcl脚本就可以,而无需对 NS本身进行修改;另一个层次是基于 C++ 和 Otcl编程的层次,如果 NS中没有所需的网络元素,则要对 NS扩展修改,添加所需的网络元素,重新编译,然后进行仿真<sup>[4]</sup>。如果用户已经完成了对 NS的扩展,NS仿真可以分为构件的扩展阶段和仿真阶段,以及对仿真结果的分析阶段,整个模拟过程如图 2所示。

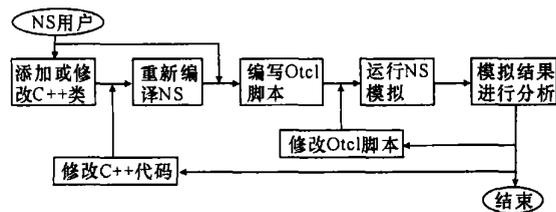


图 2 NS的仿真过程

## 2 基于 NS的网络仿真

本文结合基于 MobileIP模型的无线网络和有线网络之间的实例进行简单仿真<sup>[5]</sup>。仿真过程中建立有 5个 Wired Node 2个 Base Station 和 1个 Mobile Host的拓扑结构。

### 2.1 网络仿真拓扑配置

#### 2.1.1 仿真场景的建立

在基于 MobileIP的无线网络中建立了一个大

小为 1000\* 1000的场景,实现方法是:

```
set topo [new Topography]
$ topo load_flatgrid 1000 1000
```

#### 2.1.2 有线拓扑的建立

在有线网络拓扑中有 5个节点,分别标号为 0~4 其中节点 2 3 4都连接节点 1,而节点 0和节点 1 相连,节点 0还和两个基站 (HA、FA)相连,连接的相对位置在网络拓扑结构的配置中说明。

下面在有线节点间创建链接:

```
$ ns_duplex-link $ node_ (i) $ node(k)
20Mb 2ms DropTail,即在 node(k)和 node_ (i)间建立一条带宽为 20Mbits TTL为 2ms 队列类型为 DropTail的双向链接
```

#### 2.1.3 无线拓扑的建立

无线拓扑主要由 HA和 FA来决定,移动节点 MH在 HA和 FA之间移动,当 MH不在 HA作用范围内时,MH会自动将路由信息转发给 FA,若 MH在 FA作用范围内,数据会发给 MH,如果 MH不在 FA作用范围内,数据将会被丢失。本文在 1000\* 1000的拓扑边界内,用以下语句设定 HA和 FA的坐标:

```
$ HA(FA) set X_ x坐标值
$ HA(FA) set Y_ y坐标值
$ HA(FA) set Z_ z坐标值
```

图 3为有线节点和基站的拓扑图

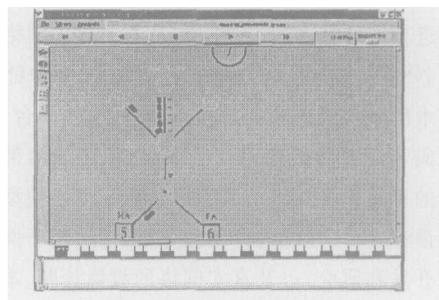


图 3 有线节点和基站拓扑

#### 2.1.4 基站 FA和 HA的配置

NS中基站的配置主要是针对无线节点各构件的设置,构件包括信道、网络接口、无线信号传输模块 MAC类、接口队列、链路层和 ARP地址解析协议等。

基站的设置如下:

```
set opt(chan) Channel/WirelessChannel
;# 信道类型
set opt(prop) Propagation/TwoRayGround
;# 无线传输模块类型
set opt(netif) Phy/WirelessPhy
```

```

;# 网络接口类型
set opt(mac) Mac/802_11;# MAC类型
set opt(ifq) Queue/DropTail/PriQueue
;# 接口队列类型
set opt(ll) LL;# 链路层类型
set opt(ant) Antenna/OmniAntenna
;# 天线类型
set opt(ifqlen) 50;# 接口队列中包的最大值
set opt(mn) 1;# 移动节点的数目
set opt(adhocRouting) DSDV;# 无线路由协议

```

议

### 2.1.5 移动节点 MH的配置

创建并设置移动节点 MH的地址:

```

$ ns_ node-config-wiredRouting OFF
;# 关闭-wiredRouting选项
set MH [$ ns_ node 1.0.1]
set node_(0) $ MH
set HAaddress [AddrParams addr2id [$ HA
node-addr]]
[$ MH set regagent_] set home_ agent _
$ HAaddress

```

### 2.2 协议代理的建立

仿真过程中我们要实现从有线网络向无线网络传送数据,节点 2 3分别和 MH建立 TCP连接,并在上面建立一个 FTP数据流:

```

set tcp1 [new Agent/TCP]
;# 新建一个 TCP Agent
$ tcp1 set class_ 2
;# tcp1设置为类型 2,在 nam中可以设置不同
数据流颜色来区分数据流
$ tcp1 set fid_ 2 ;# 设置流标记
set sink1 [new Agent/TCPSink]
;# 新建一个 TCPSink Agent
$ ns_ attach-agent $ W(2) $ tcp1;# 将 tcp1
代理绑到节点 2上,实现发送 TCP流
$ ns_ attach-agent $ MH $ sink1;# 将代理
绑定到 MH上,负责接收 TCP流并发送 ACK信号
$ ns_ connect $ tcp1 $ sink1
;# 将 tcp1和 sink1连接起来
set ftp1 [new Application/FTP]
;# 新建一个 FTP数据流
$ ftp1 attach-agent $ tcp1
;# ftp1和 tcp1绑定,也就是在 TCP连接上建
立了一个 FTP数据流

```

```

$ ns_ at $ opt(ftp1-start) "$ ftp1 start";#
FTP数据六在 $ opt(ftp1-start)时开始发送数据
$ ns_ at $ opt(stop) 0 "$ ftp1 stop"
;# FTP数据在 $ opt(ftp1-start)时停止发送
数据

```

### 2.3 传输范围的设定

在无线网络中,传输信号的强度一般随着距离增大而迅速的降低,采用不同的传播模型信号衰减公式不一样,本文采用 TwoRayGround类型 一个数据包如果在接收方能正确接收,接收功率必须大于某一个接收功率阈值 NS提供 threshold工具来计算在某种传播模型下,如何设定接收功率阈值来控制无线传输的范围。本文设定无线传输的范围是 300m Threshold工具的调用格式如下:

```

./threshold ht 5.0 hr 0.02 m
TwoRayGround 300

```

### 2.4 设置 Trace对象和 nam对象

为了方便 Trace分析和 nam演示,利用下面的语句对其进行记录和演示:

```

set tracefd [open mobile-ip. tr w]
;# 打开一个 mobile-ip. tr文件来记录仿真过
程的 trace数据,变量 tracefd指向该文件 set
namtrace [open mobile-ip. namw]
;# 打开 mobile-ip. nam文件来记录 nam过程
的 trace数据,变量 namtrace指向该文件 $ ns_
trace-all $ tracefd
$ ns_ namtrace-all-wireless $ namtrace $ opt
(x) $ opt(y)

```

### 2.5 Otcl脚本的运行

在终端中输入命令: ns脚本文件名.tcl

## 3 仿真结果分析

### 3.1 Trace文件分析

Trace文件详细记录仿真过程,根据用户的需要记录仿真过程中任何一个细节。所有对仿真的分析都基于 Trace文件,如传输速率、延迟时间等。Trace文件详细记录了每个节点上发生的事情,如时间信息、流量信息、节点信息等。

Trace文件内容如下:

```

r 2.558871 1 3 tcp 40 - - - - - 3 0.0.
0.0 0.0 2.0 0 13
+ 2.558871 3 1 ack 40 - - - - - 3 0.
0.2 0 0.0 0 0 15

```

第一行记录了节点 3在 2.558871s时收到了从

节点 2 向节点 3 发送的一个 TCP 数据包,包的大小是 40,源地址是 0.0.0,端口号是 0,目标地址是 0.0.2,目标端口号是 0 第二行记录了节点 3 在 2.558871s 时将应答信号 (ACK) 装进队列,准备发送给节点 1 因而通过对 Trace 文件的分析便能了解仿真的全过程

### 3.2 gnuplot 和 xgraph 图形分析

gnuplot 是一个命令驱动的交互式画图软件,其功能就是把数据资料和数学函数转换成容易观看的平面或立体图形,帮助研究者进行数据分析。

xgraph 是 NS 中自带的一个小巧的绘图工具,它可以根据数据文件里的数据绘制出相应的图形,帮助研究者进行数据分析。通过对数据进行直观的观察分析,从中可以更加容易的找出一些规律或问题或是验证模型的正确性,反过来指导研究。图 4 和图 5 是节点 0 和 FA 间数据传输的 gnuplot 和 xgraph 生成图。

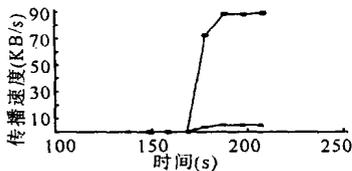


图 4 gnuplot 生成

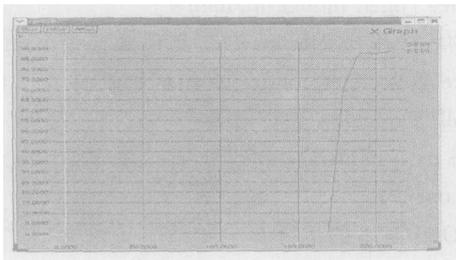


图 5 xgraph 生成

### 3.3 nam 工具分析

nam 是基于 Tcl/Tk 的动画显示工具,常与 NS 仿真配合使用,通过动画向人们展示网络运行情况,

例如网络拓扑、包传输和队列管理等。它通过调用在仿真过程中生成的 mobile-ip.nam 文件实现,调用格式如下:

```
nam mobile-ip.nam
```

通过打开 mobile-ip.nam 文件便可以打开 nam 动画演示窗口,运行过程中的某个时刻的情况如图 3 所示。通过对仿真结果的分析可见,利用 NS 对网络进行仿真,可以得到很好的网络性能仿真效果。

## 4 结束语

NS 作为一个源代码公开的免费开放平台,功能强大,我们的仿真结果表明 NS 确实是一种很好的网络仿真工具。虽然目前对网络仿真的研究虽然已经取得了许多成果,但还有许多问题有待于进一步解决和完善,如 NS 在模拟引擎的并行化、拓扑生成器的可视化以及高效数据采集工具的开发等方面还需要进一步研究和发展的。另外,人们也可以根据需要对 NS 进行二次开发,如拓展它的功能模块使之支持选播通信服务,这也是我们下一步的主要研究任务。

参考文献:

- [1] 徐雷鸣,庞博,赵耀编著. NS 与网络仿真 [M]. 北京:人民邮电出版社, 2003.
- [2] 王晓燕,郑明春. 基于 NS2 的网络仿真研究与应用 [J]. 计算机仿真, 2004 21(12): 128-131.
- [3] Kevin Fall, Kannan Varadhan. NS Manual [EB/OL]. <http://www.isi.edu/nsnam/ns/>, 2002-02.
- [4] Jae chung Mark Claypool. NS by Example [EB/OL]. <http://nile.wpi.edu/NS/>, 2005-01.
- [5] VINT growp. NS tutorial [EB/OL]. <http://www.isi.edu/nsnam/ns-allinone-2.28/ns-2.28/ns-tutorial.html>, 2005-02.

(责任编辑:黎贞崇)