

## NS2 中添加路由协议的研究与实现<sup>\*</sup>

宋 玲<sup>1</sup>, 刘勃兰<sup>2</sup>

(1. 广西大学计算机与电子信息学院, 南宁 530004; 2. 中国移动广西柳州公司, 柳州 545000)

**摘 要:** 在网络路由技术的研究中, 实际网络系统的实现往往是代价很高或不现实的。利用网络仿真器 NS2 对相关网络路由协议进行仿真是一种既有效又经济的研究方法, 但目前的 NS2 还不能支持所有网络路由协议的仿真。本文根据拓展仿真的原理和方法, 对在 NS2 中添加路由协议进行了研究, 添加了移动自组网中基于关联度的路由协议 ABR。节点间数据传输率的测试结果表明, 利用拓展后的 NS2 进行 ABR 协议的网络仿真, 能方便用户对网络性能进行评价。

**关键词:** NS2; 拓展仿真; 移动自组网; 路由协议; ABR

### 1. 引 言

在真实的网络中, 设计和调试网络协议是一件困难且复杂的事情, 仿真成了最佳可供选择的测试、评估和验证手段之一。网络仿真是用计算机程序对通信网络进行模型化, 通过程序的运行模仿通信网络的运行过程。这是一种进行网络技术研究的基本手段, 不仅适用于网络模型的构造和设计、协议性能的评价与分析, 还适用于网络协议的开发与研究以及真实网络的故障诊断。网络模拟器 NS2 (Network Simulator v2) 是一种免费、源代码公开的模拟软件平台, 其协议代码与真实网络应用代码很相似, 仿真结果具有可靠性。NS2 是一个面向对象的离散事件驱动网络模拟器, 支持多种流行的网络协议如 TCP、UDP 和路由调度、拥塞控制算法等<sup>[1]</sup>。相对于一般的离散型模拟器, NS2 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, 易于拓展, 但是对用户的编程能力, 实际网络协议的理解能力要求较高<sup>[2]</sup>。

本文利用 NS2 已有的对象, 组合所要研究的网络系统模型, 在 NS2 中添加了一个单播路由协议进行网络仿真。

### 2. NS2 的仿真原理和拓展方法

#### 2.1 NS2 的仿真原理

NS2 采用编译和解释双层结构, 编译层由 C++ 类组成, C++ 语言执行速度快但修改和编译速度慢, 故只用于编译层协议细节的实现; 前端是一个 Otcl (Object Tool Command Language) 解释器, Otcl 修改快且可用于交互操作, 用于解释层中仿真对象的设置, 这种分裂对象仿真增强了构件库的可扩展性和可组合性<sup>[3]</sup>。仿真器内核定义了有层次的编译类结构; 在 Otcl 解释器中有相似的解释类结构。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。NS2 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。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 (Node)、链路 (Link)、代理 (Agent)、业务追踪 (Trace) 和数据源 (Data Source) 等; 节点、链路、代理同时继承了 NS Object 和事件处理器类, 处理多项事务。

#### 2.2 NS2 的仿真方法

<sup>\*</sup> 本文得到广西科学基金项目 (桂科基0575017) 和广西教育厅科研项目 (桂教科研【2004】20号) 的资助。

#### 【作者简介】

宋玲, 女, 广西南宁人, 副教授; 研究方向: 计算机网络及多媒体通信服务质量。

刘勃兰, 男, 湖南邵阳人, 工程师; 研究方向: 计算机网络。

用 NS2 仿真涉及到两个层次：一个是基于 Otcl 编程的层次，利用 NS2 已有的网络元素实现仿真，只要编写 Otcl 脚本就可以，而无需对 NS2 本身进行修改；另一个层次是基于 C++ 和 Otcl 编程的层次，如果 NS2 中没有所需的网络元素，就要对 NS2 进

行扩展，添加所需的网络元素，重新编译，然后进行仿真。如果用户已经完成了对 NS2 的拓展，NS2 仿真可以分为构件拓展阶段，仿真阶段，仿真结果分析阶段<sup>[2]</sup>。整个模拟过程如图 1 所示。

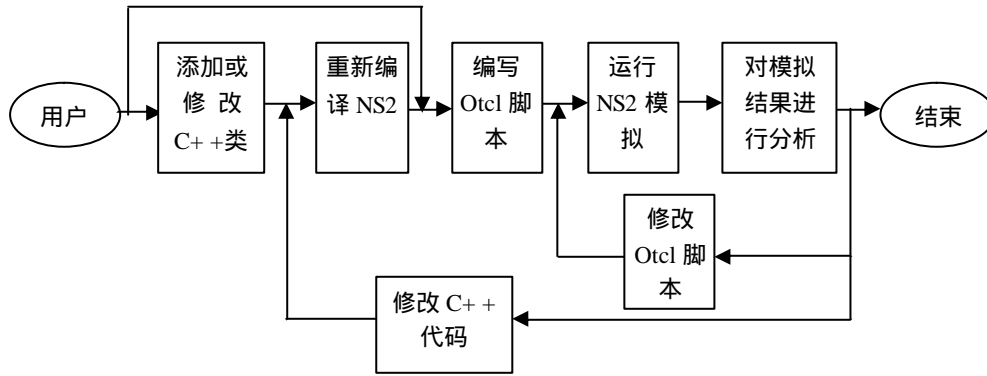


图1 NS2的仿真过程

### 3. NS2 的拓展仿真

#### 3.1 在 NS2 中添加路由协议

本文拟在 NS2 中添加 Ad Hoc 网络中的单播路由协议 ABR<sup>[4]</sup> (Associativity Based Routing)，需要添加 5 个文件( abr.h ,abr.cc ,abr\_pkt.h ,abr\_rtable.h ,abr\_rtable.cc ) 来实现<sup>[5]</sup>。在 abr.h 中定义所需要的定时器和用来实现协议功能的路由代理，其实现以及和 TCL 的挂接在 abr.cc 中；abr\_pkt.h 声明协议需要在网络节点间进行传输的协议包；abr\_rtable.h 和 abr\_rtable.cc 用来定义和实现需要拓展的路由表。这是“物理”架构，“逻辑”上的架构是类，为了实现新的协议需要从 Agent 类继承一个新的 Agent，Agent 描绘端点进行网络层数据包的封装与消耗，在协议的实现中不同的层次都被用到，而且它提供了和 TCL 接口的连接，这样就可以通过 tcl 脚本来控制新的路由协议，可以看出，这是实现路由协议所要编码的一个主要的类。路由代理用来维护内部状态和路由表，内部状态可以用一个新的类或者路由代理内部属性的集合来描述，把路由表看作一个新类：abr\_rtable。当协议需要定期地或在某一事件发生后发送数据包是用类 timer，另外一个很重要的类是 trace，它是书写模拟信息的记录文件的基础。

abr.h 中对 agent 类的说明如下：

```

class abr : public Agent {
    friend class Abr_PktTimer; //协议包定时器
private:
    nsaddr_t    ra_addr_; //路由代理地址
    abr_state   state_; //路由状态
    abr_rtable  rtable_; //路由表
    int         accessible_var_; //访问变量
    u_int8_t    seq_num_; //序列号
protected:
    PortClassifier* dmx_; // 端口分类器，把数据包
    送代理
    Trace* logtarget_; //记录目标
    Abr_PktTimer pkt_timer_; //发送数据包定时器.
    nsaddr_t& ra_addr(); //路由代理地址
    abr_state& state(); //路由状态
    int& accessible_var(); //访问变量
    void forward_data(Packet*); //数据转发函数
    void recv_abr_pkt(Packet*); //协议数据包发送函数
    void send_abr_pkt(); //协议数据包接收函数
    void reset_abr_pkt_timer(); //重置定时器
public:
    abr(nsaddr_t); //abr 协议地址
    int command(int, const char*const*); //Otcl 对象调用 C++
    对象的方法
    void recv(Packet*, Handler*); //接收函数
};
  
```

abr\_pkt.h 的定义：

```
struct hdr_abr_pkt {
    nsaddr_t   pkt_src_;      //产生数据包的源节点
    u_int16_t  pkt_len_;      // 数据包的长度 Packet
    u_int8_t   pkt_seq_num_;  //数据包的序列号
    nsaddr_t&  pkt_src() //返回数据包源
    u_int16_t& pkt_len() //返回数据包长度
    u_int8_t&  pkt_seq_num()//返回数据包序列号
    static int offset_; //数据缓冲区
    static int& offset() //返回数据缓冲区
    static hdr_abr_pkt* access(const Packet* p) //访问协议数据包头
}
```

abr\_rtable.h 对路由表的说明：

```
class abr_rtable {
public:
    void print(Trace*); //输出 trace 文件
    void clear(); //清除路由表
    void rm_entry(nsaddr_t); //删除入口地址
    void add_entry(nsaddr_t, nsaddr_t); //添加入口条目
    nsaddr_t lookup(nsaddr_t); //查看路由表
    u_int32_t size(); //路由表的大小
};
```

此外，对 NS2 模拟器中一些文件进行修改<sup>[6]</sup>，如声明 packet type 的声明文件 common/packet.h，库文件 tcl/lib/ns-packet.tcl 和 tcl/lib/ns-lib.tcl，优先级队列 queue/priqueue.cc，以及编译文件 makefile 等。

### 3.2 ABR 协议的仿真测试

根据文献[7]提出的测试模型和方法对添加的路由协议 ABR 进行测试，设定一个有 50 个节点（编号 0 - 49）的 500m×300m 的模拟场景，节点从这个矩形区域内开始服从速度为（0-20m/s）的随机运动，负载源使用常速率比特流（CBR），采用数据包传输率作为仿真测试的性能指标，节点的配置项如下<sup>[8]</sup>：

```
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(nn) 1 ;# 移动节点的数目
```

```
set opt(adhocRouting) ABR ;#无线路由协议
```

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] ;#移动节点地址
```

```
set node_(0) $MH ;#移动节点地编号
```

## 4. 测试仿真结果

### 4.1 Trace 文件

Trace 文件的功能是详细记录仿真过程，根据用户的需要记录仿真过程中任何一个细节。仿真结束后，留下的唯一记录就是 Trace 文件，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 0 15
```

上面第一行记录节点 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。通过对 Trace 文件的分析能够了解仿真的全过程。

### 4.2 gnuplot 和 xgraph 生成图

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直观的平面或立体图形，帮助研究者进行数据分析。xgraph 是 NS2 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。通过对数据进行观察分析，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 2 和图 3 是两个节点之间数据传输的 gnuplot 和 xgraph 生成图。其他节点间的数据传输率同样可以得到。

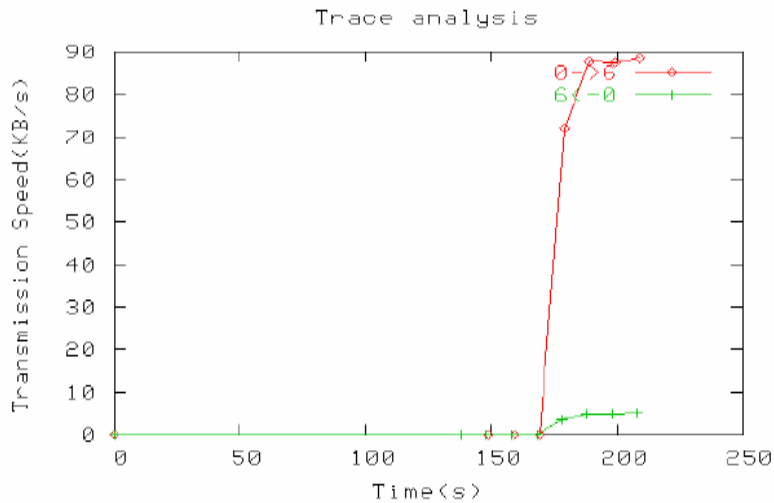


图2 gnuplot生成图

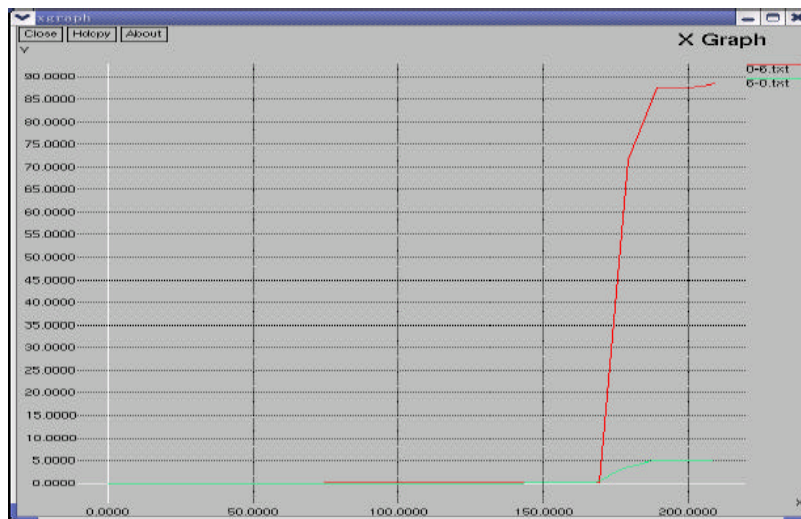


图3 xgraph生成图

## 5. 结束语

根据需要在 NS2 中添加新的路由协议进行网络拓展仿真,得到了良好的仿真效果。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 NS2 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地研究和发展,让 NS2 支持选播通信服务也是一项当前值得研究的课题。

## 参考文献：

- [1] 徐雷鸣, 庞博, 赵耀. NS 与网络仿真[M]. 北京: 人民邮电出版社, 2003.
- [2] J.Mark Pullen.(2000). The Network Workbench: Network Simulation Software for Academic Investigation of Internet Concepts [J]. *Computer Network*, (32): 365-378.
- [3] Kevin Fall & Kannan Varadhan.(2002). NS Manual [EB/OL]. Available from: <http://www.isi.edu/nsnam/>.
- [4] 于宏毅. 无线移动自组织网[M]. 北京: 人民邮电出版社, 2005: 182-187.
- [5] Francisco J. Ros & Pedro M. Ruiz.(2004). Implementing A New Manet Unicast Routing Protocol in NS2[EB/OL]. Available from:

- <http://ants.dif.um.es/masimum/nsrt-howto/html/nsrt-howto.html>.
- [6] Brent B. Welch. 崔凯译. Tcl/Tk 编程权威指南[M]. 北京：中国电力出版社，2002.
- [7] WANG Jian-xin, ZHENG Yuan, JIA Wei-jia.(2003). A-DSR: A Based-DSR Anycast Protocol for IPv6 Flow in Mobile Ad Hoc Networks[A]. In: *IEEE Proc VTC2003*[C]. Orlando, Florida, USA. (9).
- Jae Chung & Mark Claypool. (2005). NS Byexample [CP/OL]. Available from: <http://nile.wpi.edu/NS/>, (2).

## Study and Implementation of Adding Routing Protocols in NS2

SONG Ling<sup>1</sup>, LIU Bo-lan<sup>2</sup>

(1. College of Computer and Electronic Information, Guangxi University, Nanning 530004, China;

2. Liuzhou Subsidiary of China Mobile Guangxi Company, Liuzhou 545000, China)

**Abstract:** To realize an actual network system is always the more costing or unpractical during the research of network routing technology. Simulating relevant network routing protocols by using Network Simulation version2 (NS2) is one of valid and economical research methods. However, current NS2 can not simulate all of routing protocols. According to the extendable simulation principles and methods of NS2, in this paper, adding routing protocols is studied and an Associatively Based Routing (ABR) in Mobile Ad Hoc Networks is added to NS2. The testing result of data transmission speed shows network simulation of routing protocol ABR by extended NS2 is convenient to users to evaluate the performance of network.

**Key words:** NS2; extendable simulation; mobile Ad Hoc networks; routing protocol; ABR

( 责任编辑：Sabrina, Susan, Annie )