

NS 网络流量发生器的研究及实现^①

陈雪林, 吴文铁

(绵阳师范学院 数学与计算机科学系, 绵阳 621000)

摘要: 研究了如何在 NS 中创建新的网络流量发生器的问题. 基于已有 NS 流量发生器的研究, 建立一个新的 VBR 流量发生器模型. 给出了 NS 中移植新流量发生器的过程, 并对 VBR 流量发生器关键代码进行了分析. 最后, 通过仿真实验验证了新建流量发生器的正确性和良好的适应性.

关键词: 网络模拟; 流量发生器; CBR; VBR

Research and Realization of Network Flows Generator Based on NS

CHEN Xue-Lin, WU Wen-Tie

(Department of Mathematics and Computer Science, Mianyang Normal University, Mianyang 621000, China)

Abstract: This paper researches the question of how to create a new network flows generator in NS. After study on the principle and realization of flows generator in NS, we set up a mathematical model for VBR flows generator, and propose a method of migration for a new flows generator in detail, and analyse key programming code in VBR generator. Experiments on the generation and random rate of network flows showed that the VBR generator has effective and is adaptive to network simulation in different environment with different parameters.

Key words: network simulation; flows generator; CBR; VBR

目前, 国际上较成熟的网络模拟软件主要有 OPNET、NS、MATLAB、SPW、QualNet、SSFNet 等. 其中, UC Berkeley 大学开发的 NS(Network Simulator) 由于其源码开放、具有可扩展的网络模型库体系结构等特点, 得到学术界的广泛认可和应用. 研究人员可根据需要, 开发自己的网络组建, 加入到 NS 的模型库之中, 以增强和完善 NS 功能^[1].

NS 中实现了诸如 TCP、UDP 等网络协议, CBR 的模拟以及路由管理机制, 并可用于多播和 MAC 层协议的仿真等. 为了满足新的网络业务模拟的需求, 笔者为 NS 新编写了 VBR 流量发生器模块, 最后通过 OTcl 脚本进行仿真^[2].

1 网络流量发生器的原理

流量发生器对网络中的流量特性进行了描述, 即定义了产生数据分组的规律. 在 NS 中, 创建数据分组

是由 Agent 完成(在不同层上实现各种协议). 流量发生器依据流量模型定时发送信息给 Agent 触发其产生数据分组^[3]. 如果要新建一个流量发生器, 则需要建立其数学模型和确定相应参数.

1.1 NS 流量发生器模型

流量发生器模型的描述重点是其产生数据分组的规律: 每次发送数据的定时时长和数据分组的长度. 其功能是模拟应用程序产生网络流量.

NS 提供了 4 类流量发生器模型: (1)根据指数 on/off 分布产生流量的指数 Expoo 流量模型; (2)根据帕累托 on/off 分布产生流量的 Pareto 流量模型; (3)根据确定的信息速率产生流量的 CBR 流量模型; (4)根据跟踪文件中记录流量的数据产生流量的 Trace 流量模型^[4]. 大部分网络流量使用的是前三种模型中的一种.

对流量模型建模过程中, 使用 I_i 表示第 i 次产生的数据分组大小, 单位为字节; Δt_i 为第 i 次和第 $i-1$ 次产

^① 收稿时间:2012-08-10;收到修改稿时间:2012-09-26

生数据分组的时间间隔, 单位为秒; l 表示某个定值 ($l > 0$), $rate$ 表示数据源的信息速率, 单位为 bit/s. 其中, l_i 和 Δt_i 是模型中需要分析的主要参数. 对于以下所设计的流量模型, 在每一个子流中包长 l_i 是固定的, 所以要确定的主要就是时间间隔参数 Δt_i .

1.2 VBR 流量发生器模型

目前互联网中, 诸如视频数据流这样的应用层应用在流量上表现为可变信息速率的业务, 这类业务可归为 VBR 业务. 我们所要创建的 VBR 流量模型, 就是修改设定模型的 Δt_i 和 l_i 参数, 使其服从均匀分布并在一定范围内随机变化.

定义 1 Δt_i 在取某一定值的基础上添加一个均值为 0 的均匀分布随机抽样值, x_i ($-0.5 < x_i < 0.5$) 为添加的抽样值与原值的比值, $l_i=1$, 令:

$$\text{interval} = \frac{2^3 \times l_i}{rate} = \frac{2^3 \times l}{rate}$$

$$\Delta t_i = \text{interval} \times (1 + x_i), x_i \in \cup(-0.5, 0.5) \quad (1)$$

定义 2 由于 x_i 的均值 $\bar{x} = 0$, 则:

$$\begin{aligned} \frac{\text{rate}}{\text{rate}} &= \frac{\sum_i^k 2^3 \times l_i}{\sum_i^k \Delta t_i} = \frac{\sum_i^k 2^3 \times l}{\sum_i^k \left(\frac{2^3 \times l}{rate}\right) \times (1 + x_i)} = rate \times \frac{k}{k + \sum_i^k x_i} \\ &= rate \times \frac{1}{1 + \frac{1}{k} \times \sum_i^k x_i} \rightarrow rate \times \frac{1}{1 + \frac{x}{k}} (k \rightarrow \infty) = rate (k \rightarrow \infty) \quad (2) \end{aligned}$$

在发送数据分组的时间间隔 Δt_i 添加了均匀分布的随机抽样值之后, 当其发送了大量数组分组后, 其发送速率的统计平均值 $rate$ 仍然为 $rate$ ^[5].

2 网络流量发生器的实现

2.1 VBR 流量发生器移植步骤

1) 参照 NS 中 CBR 流量发生器的实现, 在 NS 安装目录下的 ns-allinone~/ns~/tools/ 子目录中创建 vbr_traffic.cc 源文件, 其分析见 2.2.

2) 在定义消息类型和普通头部的文件 packet.h 中定义新的数据分组类型 PT_VBR. 即在 ns-allinone~/ns~/common/packet.h 中, 定义常量 PT_VBR:

```
static const packet_t PT_VBR = 62;
```

然后, 在 PT_NTTYPE 之前添加消息 VBR 的定义:

```
NAME_[PT_VBR] = "VBR";
```

3) 在 ns-allinone~/ns~/tcl/lib/ns-default.tcl 文件中定义 OTcl 的变量, 以实现将 C++ 变量绑定到 OTcl 变量. 在 CBR 流默认参数定义的后面添加如下代码:

```
Application/Traffic/VBR set rate_ 448Kb
Application/Traffic/VBR set maxpkts_ 268435456
Application/Traffic/VBR set maxsize_ 200
Application/Traffic/VBR set maxsize_ 100
Application/Traffic/VBR set intervaltime_ 200
```

4) 在 ns-allinone~/ns~/Makefile 文件中添加对 vbr_traffic.cc 的编译:

```
tools/expoo.o tools/cbr_traffic.o tools/vbr_traffic.o
```

5) 重新编译并安装 ns, 使新协议生效^[6].

```
make clean; make; sudo make install
```

2.2 VBR 流量发生器代码分析

源代码中体现了 VBR 流量模型的是位于 next_interval 函数中的代码.

```
double VBR_Traffic::next_interval(int &size){
    interval_ = (double)(intervaltime_ <<
3)/(double)rate_;
    double t = interval_;
    if (random_)
    t += interval_ * Random::uniform(-0.7, 0.7);
    if(maxsize_ <= minsize_) size = maxsize_;
    else {
        int i = int(Random::uniform(minsize_, max
size_)); size = i;
    }
    if (++seqno_ < maxpkts_) return(t);
    else return(-1);
}
```

① interval_ 值由 $\Delta t_i = \frac{2^3 \times l_i}{rate} = \frac{2^3 \times l}{rate}$ 确定, 而

intervaltime_ 和 rate_ 的值可在脚本中设置, 否则其默认值在 ns-default.tcl 中设置; 从而 interval_ 的值是一个固定值.

② 通过 uniform 函数给 interval_ 叠加一个随机分量, 将该值赋予变量 t (函数返回值), 这样每次产生的数据分组的间隔将是在区间 $[0.3 * interval_, 1.7 * interval_]$ 中的一个随机值.

③ 在最大和最小分组长度之间随机生成一个整数, 作为数据分组的长度; 如果 maxsize_ 小于或等于

minsize_，则退化为 CBR 流。

产生数据分组的过程，则由 start() 函数调用 timeout() 函数来完成。

```
void VBR_Traffic::start()
{
    init();
    running_ = 1;
    timeout();
}
```

在此，我们没有对 timeout() 函数进行重载，其实现由 TrafficGenerator 类所完成，具体完成以下数据的处理工作。

调用 send() 函数触发数据源 Agent，让 Agent 产生一个数据分组；调用 next_interval() 确定流量发生器模型的 2 个参数：下一次触发数据源的时间和数据分组的大小；确定模型的参数后，得到下一次触发数据源的时间，再启动定时器^[7]。

3 实验验证

建立一个简单的无线网络模型，2 个节点成一字形的拓扑结构。节点 0 向节点 1 发送可变速率 VBR 的数据流（在 UDP 代理上建立 VBR 流），它们在相互的信号覆盖范围内，模拟场景的长宽为 400x200 米，仿真时间为 20 秒。无线节点的物理层的发射功率、信号接收门限等参数采用默认值，在 ns-default.tcl 中有定义。网络拓扑示意图如图 1 所示。

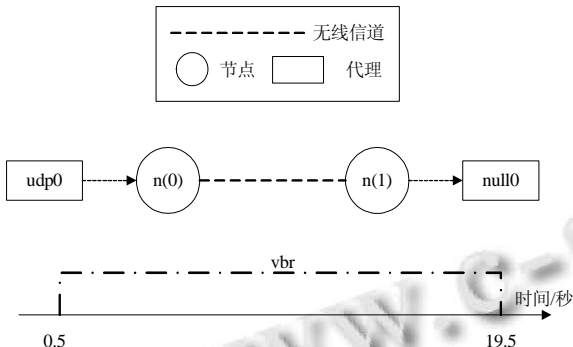


图 1 二节点无线网络拓扑

编写 vbr.tcl 脚本完成以上仿真场景的描述，其中对 VBR 的设置如下：

```
Application/Traffic/VBR set rate_ 120kb
Application/Traffic/VBR set random_ 1
Application/Traffic/VBR set minSize_ 100
Application/Traffic/VBR set maxSize_ 300
```

运行仿真程序 ns vbr.tcl，打开生成的 vbr.tr 文件，截取 Trace 文件的部分内容观察。

```
r 7.512116449 _0 MAC --- 0 CTS 38 [ba4 0 0 0]
s 7.512126449 _0 MAC --- 530 VBR 332 [13a 1 0 800] ----- [0:0 1:0 32 1]
r 7.514783115 _1 MAC --- 530 VBR 274 [13a 1 0 800] ----- [0:0 1:0 32 1]
s 7.514793115 _1 MAC --- 0 ACK 38 [0 0 0 0]
r 7.514808115 _1 AGT --- 530 VBR 274 [13a 1 0 800] ----- [0:0 1:0 32 1]
r 7.515097782 _0 MAC --- 0 ACK 38 [0 0 0 0]
s 7.518020243 _0 AGT --- 531 VBR 222 [0 0 0 0] ----- [0:0 1:0 32 0]
```

图 2 截取 Trace 文件部分内容

由此可见，的确产生了数据分组大小变化的 VBR 包，说明成功实现了对 VBR 流量发生器的移植。

接下来，我们编写一个 vbr.awk 程序，将 Trace 文件中的分组 UID 和分组大小提取出来。

```
BEGIN { }
{
    event = $1; #Event 动作
    node = $3; #Node 节点
    trace_type = $4; #Layer 层次
    uid = $6; #PKTid 分组 UID
    pkt_type = $7; #PKT 分组类型
    pkt_size = $8; #PKT 分组大小
    if(event=="s" && node==0 && trace_
type=="AGT" && pkt_type=="VBR")
        printf("%d %d\n",uid,pkt_size);
} END { }
```

执行命令：gawk -f vbr.awk vbr.tr >> vbr，从而将分组 UID 和分组大小数据单独提取到一个 vbr 文件中。运行绘图程序 gnuplot，输入绘图命令：plot "vbr" with linespoints lw 2 pt 19，将提取的数据绘制到 VBR 流量发生器的曲线图中，如图 3 所示。其中，横轴表示数据分组序号，纵轴表示数据分组的大小。

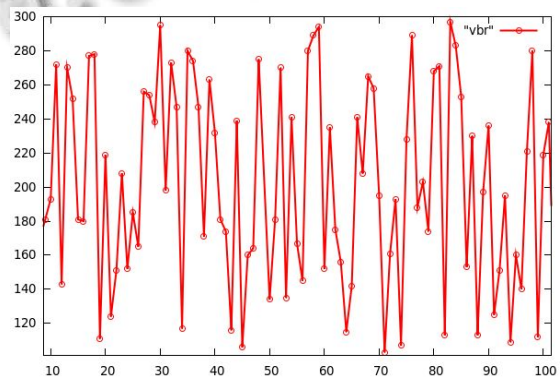


图 3 VBR 流量发生器曲线图

从图 3 中也可看出 VBR 流量发生器产生的数据分组大小和时间间隔确实是在一定范围内的随机值。

(下转第 107 页)

源节点经融合以后的数据作为动态加权信息融合的信息源进行融合,而融合后得到的信息为井下环境的安全状况信息.在 MATLAB 仿真中,平均加和融合,固定加权融合和动态加权融合的数据比较如表 1 所示:

表 1 信息融合算法的方差结果比较

试验次数/ 误差方差	1	2	3	4	5	6	7
平均加和 估计	5.13	3.76	5.69	5.31	4.56	4.27	3.92
固定加和 估计	4.16	3.15	4.61	4.60	3.72	3.31	3.26
方差自 适应加 权融合	3.39	2.56	3.67	4.07	3.25	2.32	2.68

由融合的比较结果可以看出,方差自适应加权融合能有效减少通信量,易于实际应用中的实现,能有效提高无线传感器网络的信息融合精度.

4 结语

本文采用基于无线传感器网络的无反馈分布递阶信息融合算法对矿井环境进行监测,针对无线传感器网络提出了分布递阶卡尔曼滤波信息融合算法,下层源节点采用改进的卡尔曼滤波,上层汇聚节点采用方差自适应的加权信息融合算法,结合 MATLAB 进行了实例仿真.仿真结果表明该方法有很高的信息融合精度和可靠性.

(上接第 172 页)

3 结语

本文介绍了流量发生器的原理,分析了 VBR 流量模型,并给出了 VBR 流量发生器的代码和移植过程.这将有助于我们分析流量发生器模型原理,掌握其实现过程,对于进行网络数据流业务分析的研究人员具有一定实用价值.

参考文献

- 1 李越,钱德沛,何莹.网络仿真器 NS 问题分析及改进方案.系统仿真学报,2005,11:2832-2836.
- 2 杨玉华,刘培宁,刘际炜.NS-2 的仿真模拟技术分析.计算机工程,2005,31(15):110-112.

参考文献

- 1 崔莉,鞠海玲,苗勇,等.无线传感器网络研究进展.计算机研究与发展,2005,42(1):163-167.
- 2 韩鸿泉,朱红松,孟军.无线传感器网络技术.计算机系统应用,2005,14(2):38-41.
- 3 付华,胡雅馨.一种改进的无线传感器网络信息融合技术.计算机系统应用,2010,19(7):183-185.
- 4 何友,王国宏,陆大鑫.多传感器信息融合及应用.北京:电子工业出版社,2000.
- 5 冯秀芳,赵禅禅,贾嘉.无线传感器网络中的信息融合技术研究.太原理工大学学报,2007,38(2):153-155.
- 6 付梦印,邓志红,张继伟.Kalman 滤波理论及其在导航系统中的应用.北京:科技出版社,2003.
- 7 田继善,吴志勤,吴永杰,文成林.多传感器分布式 Kalman 滤波融合算法.河南大学学报,1999,29(2).
- 8 崔平远,黄晓瑞.基于联合卡尔曼滤波的多传感器信息融合算法及应用.电机与控制学报,2001,5(3):204-207.
- 9 李战明,陈若珠,张保梅.同类多传感器自适应加权估计的数据级融合算法研究.兰州理工大学学报,2006,32(4):78-82.
- 10 Kalman RE. A new approach to linear filtering and prediction problems. Trans. of the ASME: Journal of Basic Engineering, 1960,82(1):35-45.
- 11 廖惜春,邱敏,麦汉荣.基于参数估计的多传感器数据融合算法研究.传感技术学报,2007,20(1):193-197.
- 12 姜晓军.基于无线传感器网络的信息融合算法研究[硕士学位论文].郑州:郑州大学,2007.
- 3 Fall K, Varadhan K. The ns Manual, 2008. <http://www.isi.edu/nsnam/ns/doc/index.html>.
- 4 颜昕,李腊元.NS 的仿真机制及协议扩展.武汉理工大学学报,2004,28(2):182-185.
- 5 黄化吉,冯穗力,秦丽姣,陆宏治.NS 网络模拟和协议仿真.北京:人民邮电出版社,2010.186-187.
- 6 杨锦亚,郭虹,于宏毅.NS-2 新功能模块的开发.计算机仿真,2006,23(11):120-123.
- 7 于斌,孙斌,温暖,王绘丽.NS2 与网络模拟.北京:人民邮电出版社,2007.106-109.