

基于 NS2 的网络仿真与性能测试

肖权权,段 迅

(贵州大学 计算机科学与信息学院, 贵州 贵阳 550025)

摘 要:互联网是一个庞大的、复杂的、开放的网络集合。深入了解和优化网络性能并不是一件容易的事。为了方便网络管理者对网络进行管理,探讨了 NS-2 仿真平台体系结构、构建环境和仿真流程,提出了采用 NS2 网络仿真器建立网络模拟场景来获得网络性能参数的方法,通过分析 NS2 网络仿真过程产生的记录文件,利用 awk、gnuplot 等工具获得网络的封包遗失率(Packet Loss Rate)、端到端的延迟(End-to-End Delay)、吞吐量(Throughput)等性能指标呈现了模拟结果。实验表明利用网络仿真求解网络性能参数具有直观可靠、成本低、灵活可靠、避重就轻、易于比较等优点。通过实例分析验证了该方法的有效性。

关键词:网络仿真;遗失率;延迟;吞吐量

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2012)04-0025-04

Network Simulation and Performance Testing Based on NS2

XIAO Quan-quan, DUAN Xun

(School of Computer Science and Information Engineering, Guizhou University, Guizhou 550025, China)

Abstract: The Internet is a huge, complicated and open network. Understanding and optimizing the network performance is not an easy task. In order to make easier for managers to manage the network, discussed NS2 simulation platform architecture, environment and simulation process. Then, used the NS2 to simulation network scenarios to obtain the network performance parameters. Through the analysis of NS2 network simulation process to produce records, use the awk, gnuplot and other tools to obtain the network performance parameters. Experiments show that using NS2 to get network performance parameters has advantages, such as intuitive, reliable, low cost, flexible and reliable, easy to compare. From a example, testing and verifying this method is useful.

Key words: network simulation; loss rate; delay; throughput

0 引 言

随着 Internet 不断快速发展,网络研究人员经常需对网络性能等方面进行研究或优化。目前,对网络性能的研究普遍采用三种方法:分析方法、实验方法和模拟方法^[1]。分析方法由于受到假设限制,结果的有效性和精确性变化较大。实验方法实验成本高、配置不灵活,不适合大规模网络研究。模拟方法采用模拟软件能够在网络协议各层次上进行模拟,近乎真实的反映网络运行情况^[2]。目前,常用的模拟软件有 OPNET、MATLAB 和 NS2。NS2 (Network Simulator, version2)起源于 UC Berkeley 开发的 REAL 网络模拟器,是一款通用的网络模拟器。该方法使用网络模拟器能够实现网络拓扑仿真、通信量仿真和协议仿真,能方便的对网络进行性能测试。通过获得的性能参数,

发现网络运行瓶颈之处,达到改善网络运行状况的目的。

1 性能参数体系

ETF 的 IPPM WG 和 ITU 的 SG13 工作组对网络性能参数进行了标准化定义^[3]。其中 ITU-T 定义的网络性能参数主要内容如下。

(1)封包遗失率(Packet Loss Rate):丢包数在发送包所占的比例。

(2)端到端的延迟(End to End Delay):发送某个封包的时间与对方接收到封包的时间差。影响端到端延迟的因素有很多,包括发送封包的时延,传输时延等因素。

(3)吞吐量(Throughput):单位时间内某节点接受数据量和发送的数据量的总和。

(4)抖动率(Jitter):端到端延迟的变化量,当网络通信量较大时,进入某节点的封包需要排队转发,造成每个封包到达目的节点的时间不一样。这个不同的差异即为抖动率。抖动率越大,表示网络越不稳定。

收稿日期:2011-10-27;修回日期:2012-01-27

基金项目:贵阳市科技计划项目([2008]筑科工合同字第24-7号)

作者简介:肖权权(1985-),男,湖北仙桃人,硕士研究生,研究方向为计算机网络;段 迅,硕士生导师,研究方向为计算机网络。

2 NS2 的仿真模型

NS2 仿真模型通过各种组件来抽象真实的网络。它包括网络拓扑结构模型、应用协议模型和流量模型。

网络拓扑结构模型完成网络节点和链路的配置。在配置的过程中通过调用 Simulator 类的 node - config 过程来配置节点的属性,调用模拟对象的 duplex-link 和 simplex-link 来配置节点之间的双向链路和单向链路。在链路上同时可以配置节点之间的带宽、延迟和队列管理算法。在 NS2 中支持的队列管理算法有 RED 缓冲管理、FIFO 先到先服务和 PFQ 分组公平队列。典型的分组公平队列算法有 WFQ (Weighted Fair Queuing)、SPFQ (Start - Potential Fair Queuing)、CSFQ (Core Stateless Fair Queue) 等。

应用协议模型由代理和辅助模块构成。代理可以实现不同层上的网络协议,也可以用来构建和销毁网络层分组。Agent 是 SplitObject 的一个子类, Agent/Tcp 和 Agent/UDP 分别实现了网络上传输层上的 Tcp 和 UDP 代理。辅助模块负责对代理进行协调和管理。

流量模型由应用层程序产生, Agent 不能产生业务流。应用层程序处于网络层之上,可以分为两大类:应用模拟器和流量产生器。应用模拟器使用在 TCP 代理之上,流量产生器使用在 UDP 代理之上。

●常见的应用模拟器有 FTP 应用模拟器、Telnet 应用模拟器。

(1) FTP 对应的 OTcl 类为 Application/FTP 主要用来模拟大量的数据传输。

(2) Telnet 应用模拟器用于发送分组的时间间隔按照 tcplib 经验分布来发送数据。

●为了模拟网络中的业务流随机产生, NS2 使用流量产生器以不同的概率模型来产生数据。

(1) 指数分布流量产生器按照指数 On/Off 分布产生数据。On 阶段按照固定速度发送分组, Off 阶段不发送分组。

(2) Pareto 分布流量产生器在 On 和 Off 两种状态产生的业务流服从 Pareto, 其它状态服从 On/Off 分布。

(3) 固定比特率流量产生器按照固定的速度来产生业务流。

(4) Trace 文件流量产生器按照一个 Trace 文件来产生数据。Trace 文件中包含两个字段, 第一个字段为分组产生的时间间隔, 另一个字段为下一个产生的分组大小。

3 实际的网络测试

完成 NS2 对网络环境的仿真, 一般需要进行建立

仿真程序、运行仿真结果和对仿真结果进行分析三个阶段。在建立仿真程序时, 用户需要完成创建模拟器对象、设置跟踪文件、创建网络拓扑结构、设置代理和应用层协议和使用模拟器对象的 at 过程设置节点事件与时间的对应关系^[4]。如果问题的定义需要修改源代码, 还需要对 Tcl/Otcl 模拟代码进行编写。程序运行结束后, 根据产生的模拟结果求出我们比较关心的网络性能参数。

3.1 仿真网络拓扑的建立

建立的仿真网络拓扑结构如图 1 所示, 创建了 4 个节点 n0, n1, n2, n3。其中 n0 与 n1 为数据发送端, n3 为数据接收端。各个节点的队列大小设为默认值, 都使用 DropTail 队列管理算法。n0 与 n2 和 n1 与 n2 节点之间的链路带宽为 2Mbps, 延迟为 10ms, n2 与 n3 节点之间的链路带宽为 1.7Mbps, 延迟为 20ms。n2 与 n3 之间为瓶颈链路。n1 与 n3 之间具有一个 udp 联机, n1 上建立了一个 udp 发送代理, n3 之间建立了一个 udp 接受代理。n1 节点在 0.1sec 时开始向 n3 发送 cbr 数据^[5]。n0 与 n3 之间具有一个 tcp 联机, n0 上建立 tcp 发送代理, n3 上建立了一个 tcp 接受代理用来接受 TCP 数据; n0 节点在 1.0sec 时开始向 n3 发送 ftp 数据。

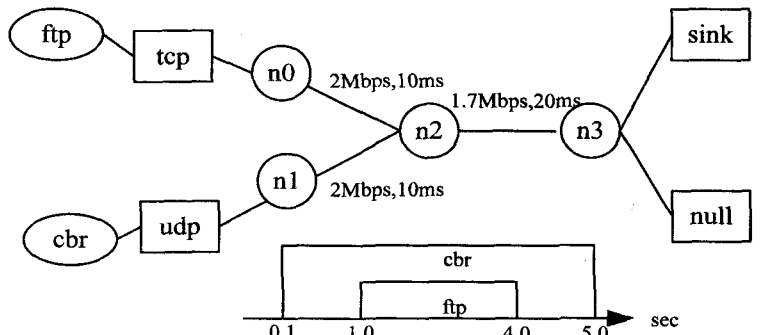


图 1 仿真网络的拓扑结构

3.2 仿真代码

(1) 设置仿真对象和跟踪文件。

```
set ns [new Simulator]
```

...

(2) 创建网络拓扑结构。

```
set n0 [ $ ns node ]
```

```
set n1 [ $ ns node ]
```

...

```
$ ns duplex-link $ n0 $ n2 2Mb 10ms DropTail
```

```
$ ns duplex-link $ n1 $ n2 2Mb 10ms DropTail
```

```
$ ns duplex-link $ n2 $ n3 1.7Mb 20ms Drop-
```

Tail

...

(3) 设置代理和应用层协议。

```

set tcp [new Agent/TCP]
$ ns attach-agent $ n0 $ tcp
set sink [new Agent/TCPSink]
$ ns attach-agent $ d $ sink
set ftp [new Application/FTP]
...

```

(4) 设置节点事件和时间的对应关系。

```

...
$ ns at 0.1 "$ cbr start"
$ ns at 1.0 "$ ftp start"
$ ns at 4.0 "$ ftp stop"
$ ns at 4.5 "$ cbr stop"
...

```

(5) 使用模拟对象的 run 过程开始进行模拟。

```
$ ns run
```

3.3 Nam 动画演示

NS2 提供了对仿真实验结果观察的方式,一种为动态的 Nam,另一种为后缀名为 .tr 的文件。NS2 采用 Nam^[6]对整个通信过程进行动画演示。out.tr 文件记录了网络中每数据包在每个时间内的所有信息。对网络性能参数求解是以 out.tr 文件中的数据为依据^[7]。图 2 是 out.nam 文件执行的过程,显示了节点 n1 向 n3 发送 CBR 数据,当 1.0s 后 n0 向 n3 发送 FTP 数据,网络产生拥塞出现丢包现象,如图 3 所示。

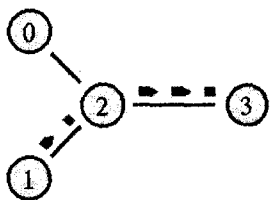


图2 n1 向 n3 发送 CBR 数据

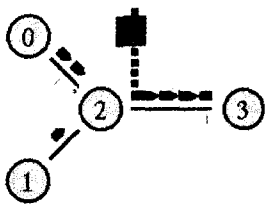


图3 丢包现象

4 网络性能分析

遗失率、延迟、吞吐量等性能参数是体现网络是否高效、可靠的依据^[8]。它们为网络控制和管理、网络异常流量分析、流量拥塞检测提供数据依据。如下将通过上面的仿真实例,分析仿真过程产生的文件数据,利用 awk 程序语言得出包的遗失率、网络端点到端点的延迟和吞吐量。

4.1 CBR 封包的遗失率

许多因素会导致数据包的遗失,如网络拥塞时,节点来不及处理转发的数据,造成缓冲区数据溢出^[9]。当数据传输出现错误时,上层协议会简单地将数据包丢弃。下面使用 awk 脚本语言来就求解网络性能参数,使用编辑器编写代码保存文件为 measure-drop. awk 执行如下命令 \$ gawk -f measure-loss. awk out.tr 输出结果如图 4 所示。

如图显示总共发送了 550 个 CBR 封包,遗失了 8 个封包,封包的遗失率为 0.014545。

```

Administrator@PC2011101722H0D ~/ns-allinone-2.27/ns-2.27/tcl
$ awk -fmeasure-loss.awk out.tr
number of packets sent:550 lost:8

```

图4 封包的遗失率

4.2 CBR 封包端到端的延迟

现在的网络一般是采用分组模式进行数据传输,数据包在传输的过程中需要经历很多节点,当网络数据量较大时,数据包需要在节点内部进行排队依次处理,这样就产生了排队延迟^[10]。在该模式下数据在网络传输中产生的延迟为如下表示:总延迟=发送时延+传播时延+排队时延+处理时延。总延迟可以表示为:

$$OTT(n, s) = \sum_{i=1}^n \left[\frac{s}{b_i} + \frac{d_i}{c} + f_i \right] = \sum_{i=1}^n \left[\frac{s}{b_i} + \frac{d_i}{c} \right] + \sum_{i=1}^n f_i$$

其中, s 为数据包大小, b_i 为第 i 个节点接口的速率, d 为到端线路之间的距离, c 为电磁波在信道上的传播速率,不同的介质传播的速率不同, f_i 为在第 i 个节点排队的时延和处理时延。

编写代码保存在文件 measure-delay. awk 中,执行 gawk 命令,将产生的结果输出到 cbr_output 文件中,利用 NS2 自带的 gnuplot 软件^[11],执行 plot "cbr_output" 得出 CBR 封包端到端的延迟如图 5 所示。

如图 5 所示在 0.1s 时 n1 开始有 CBR 数据发送,此时 n0 无 ftp 数据发送,因此端到端的时延稳定在 0.038s 附近,在 1.0 到 4.0 之间节点 n0 向 n3 发送 ftp 数据,此时由于节点采用 DropTail 队列管理算法^[12], CBR 封包从 n1 端到 n3 端的延时呈现周期性变化状态,当 4.0s 后节点 n0 停止发送 ftp 数据, CBR 封包端到端的延迟又趋于稳定。

4.3 吞吐量

以求 n3 节点在一段时间内的吞吐量为例,观测 n3 节点在通信时间内的吞吐量。使用 gnuplot 输出结果如图 6 所示。

从图 6 可以看出 0s 到 1s 时,只有节点 n1 发送 CBR 数据,节点 n2 吞吐量保持稳定不变,1.0s 到 4.0s 时,节点 n0 开始向 n3 发送 ftp 数据,节点 n2 处工作繁

忙状态,吞吐量总体有所下降,4.0s 后 n0 停止发送 ftp 数据,吞吐量又趋于稳定。

网络通过 NS2 仿真,使虚拟的网络环境具体化,能方便快捷地测试出网络性能参数,这对网络管理者管理网络具有较大地参考意义。

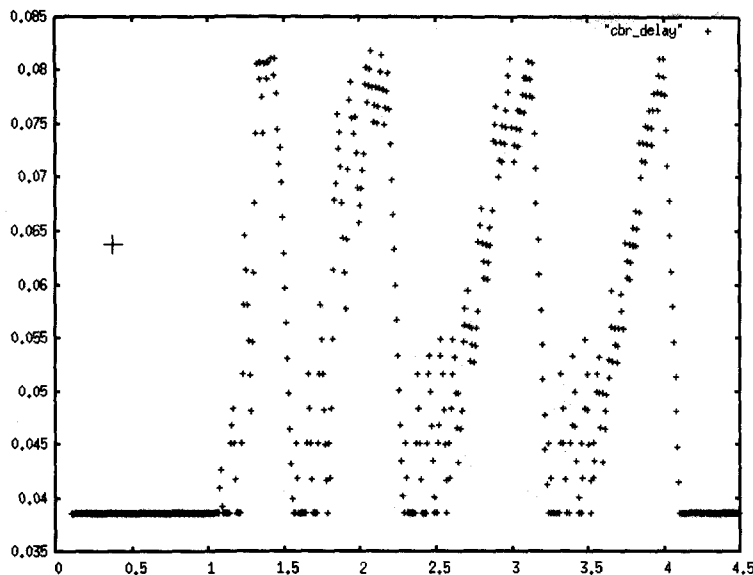


图 5 n2 发送的 CBR 封包端到端的延迟

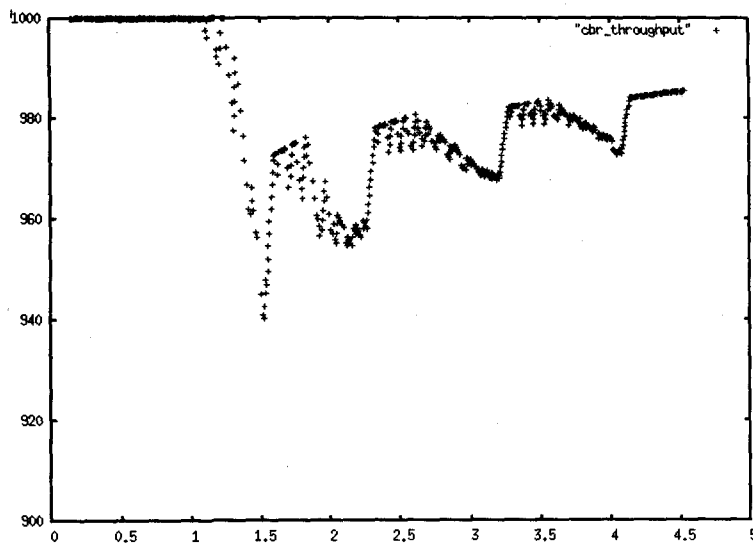


图 6 节点 n3 吞吐量

参考文献:

- [1] 徐雷鸣, 庞博, 赵耀. NS 与网络模拟 [M]. 北京: 人民邮电出版社, 2003.
- [2] 吴仕浩, 林庆华, 胥布工. 网络仿真器 NS-2 及其一个应用实例 [J]. 计算机仿真, 2004, 21(7): 96-98.
- [3] 程涛. 基于 NS-2 的网络拓扑性能研究 [J]. 哈尔滨商业大学学报(自然科学版), 2010, 26(3): 577-582.
- [4] 蔡群英, 黄镇建. 搭建网络实验环境的探索与实践 [J]. 计算机与数字工程, 2009, 37(6): 195-197.
- [5] 柯志亨, 程荣祥, 邓德隽. NS2 仿真实验-多媒体和无线网络通信 [M]. 北京: 电子工业出版社, 2009.
- [6] Floyd S. Promoting the use of end-to-end congestion control in the internet [J]. IEEE/ACM Transactions on Networking, 1999, 7(4): 367-373.
- [7] 陈春梅. 基于 NS2 的网络仿真与性能分析 [J]. 通信技术, 2010, 43(8): 48-50.
- [8] 王明伟, 王奇, 杨洁, 等. 基于 NS-2 的视频流仿真方法研究 [J]. 计算机技术与发展, 2010, 20(2): 117-120.
- [9] 毛跟勇, 刘成安, 兰瑞莉. 料位控制测量的无线网络在 NS2 中的模拟 [J]. 计算机技术与发展, 2010, 20(3): 163-166.
- [10] Fall K, Varadhan K. The NS Manual [M]. [s.l.]: the VINT Project, 2001.
- [11] 李蓬, 黄河. 基于 NS2 的 MPLS 流量工程仿真研究 [J]. 计算机技术与发展, 2008, 18(9): 53-56.

5 结束语

网络仿真是网络安全管理和网络监测技术的重要手段。对于复杂的网络, 进行网络仿真时, 应抽取研究关注的部分进行模拟。从仿真的效果可知, 将真实的

- [12] Dagher J C, Marcellin M W, Neifeid M A. A theory for maximizing the life time of sensor networks [J]. IEEE Transactions on Communications, 2007, 55(2): 323-332.

(上接第 24 页)

- Systems, 2002, 12(3): 311-333.
- [12] 杨玉梅, 刁永锋. 基于 UML 顺序图的 Petri 网建模 [J]. 计算机技术与发展, 2007, 17(10): 130-133.
- [13] 周长红. UML 图的 Petri 网建模 [D]. 青岛: 山东科技大学, 2004.

- [14] 陈志德, 曾凡平. UML 状态图和 Petri 网络在类测试用例生成的应用 [J]. 小型微型计算机系统, 2010, 31(3): 519-522.
- [15] 冯林, 姜浩. 基于时间约束 Petri 网的工作流可调度性分析 [J]. 计算机技术与发展, 2006, 16(11): 34-37.