

基于 Windows 平台的数据包实时传输模拟

王子超, 丛 静, 黄永锋, 潘 乔

(东华大学计算机科学与技术学院, 上海 201620)

摘 要:采用基于 Windows 平台的网络驱动程序接口规范技术,设计并实现一个基于协议过滤的高精度网络模拟器。该模拟器可实时模拟广域网传输链路的各种网络传输条件,包括网络带宽、数据包的丢包、传输延迟及传输错序等。性能测试结果表明,该网络模拟器达到了高精度、低负荷的设计要求。其实现有助于研究多媒体网络应用服务质量、进行网络协议分析以及验证各种网络流量控制算法。

关键词:网络驱动程序接口规范;网络模拟器;传输延迟;带宽;丢包

Simulation on Data Packet Real-time Transmission Based on Windows Platform

WANG Zi-chao, CONG Jing, HUANG Yong-feng, PAN Qiao

(College of Computer Science and Technology, Donghua University, Shanghai 201620, China)

【Abstract】This paper designs and implements a high-precision network emulator based on protocol filtering by employing network driver interface specification technology based on Windows platform. The emulator is capable of emulating various kinds of the network transmission conditions of WAN links in real-time, including network bandwidth, packet loss, transmission delay, and out-of-order transmission. Experimental results indicate that the network emulator achieves high-precision, low-load design requirements. Realization of the emulator can help to study the QoS of multi-media network applications, analyze network protocol, and verify network control algorithms.

【Key words】network driver interface specification; network emulator; transmission delay; bandwidth; packet loss

1 概述

网络模拟是一种网络仿真方法,它为测试 Internet 上的应用系统和技术提供了一个方便、高效的验证与分析方法,具有费用较低、规模较小、使用方便等特点,适用于实验室环境。网络模拟器是一种半实物的网络模拟系统,它构造的虚拟网络和真实网络需要进行同步,其特点是实时性高,模拟器内部规定第 n 秒发生的事件即模拟开始之后外界时钟推进到第 n 秒发生的事件。真实网络具有不稳定、不可重复再现等特点,而网络模拟器可在实验室为网络研究人员提供可控、可再现的网络环境,能加快网络协议和技术的开发速度。

国内外已有的网络模拟器大多是实现在开源操作系统上的。EMPOWER^[1]是基于 Linux 的分布式模拟系统,可模拟多个节点之间的网络环境,并适用于无线网络环境。ENDE^[2]基于 Linux 操作系统,主要实现了网络单程延迟的模拟,其目标是应用于多媒体协议的测试。RplTrc^[3]是一款实现在基于 Linux2.4 内核的高端商务 PC 机上的网络模拟器,主要实现基于 trace 文件延迟发送的功能。网络链路模拟器^[4]是基于 Linux2.4 内核的网络模拟器,实现了基本的丢包、带宽和延迟模拟功能。文献[5]实现了一个半实物网络模拟器,它是基于 Linux 下 NIST Net 软件包开发的,主要用于模拟无线网络。Windows 操作系统的内核代码是不公开的,给基于 Windows 内核的软件开发增加了难度,但鉴于它在国内被广泛使用,本文开发了一个基于 Windows 平台的高精度网络模拟器,不仅能模拟具有常量值的带宽、延迟和丢包等简单网络链路条件,而且可以根据 trace 文件模拟动态的网络传输,使模拟更具有真实性,这对于在不同网络环境下研究、分析应用系统的服务质量及测试各种网络新技术的性能具有重要意义。

2 网络模拟器的体系结构

本文实现的网络模拟器是基于网络驱动程序接口规范(Network Driver Interface Specification, NDIS)框架开发的,由 4 个模块组成(如图 1 阴影部分所示):用户控制模块,模拟引擎,数据包接收模块和数据包发送模块。

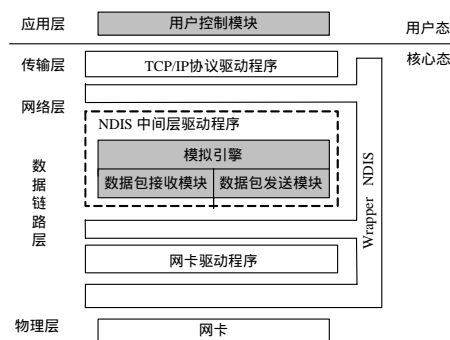


图 1 基于 NDIS 的网络模拟器体系结构

用户控制模块实现于用户态,是用户配置模拟器的接口,它为用户提供选择所需加载的 trace 文件、设置延迟时间长度以及选择数据包协议类型的功能。其他 3 个模块实现于核心

基金项目:国家自然科学基金资助项目(30770589);国家留学科研启动基金资助项目([2008]890 号);东华大学青年科研启动基金资助项目(112-10-0044017)

作者简介:王子超(1984-),男,硕士研究生,主研方向:网络管理;丛 静,副教授、博士;黄永锋,副教授、博士后;潘 乔,讲师、博士

收稿日期:2010-05-28 **E-mail:** wang_zichao000@mail.dhu.edu.cn

态的 NDIS 中间层驱动程序。数据包接收和发送模块分别负责接收和发送数据包,模拟引擎的功能是加载用户控制模块中设置的参数及 trace 文件,并对数据包进行相应的控制操作,以模拟现实网络中的各种传输特性。

模拟引擎作为网络模拟器的核心模块,完成了基本的网络链路模拟功能。如图 2 所示,模拟引擎由加载模块、协议判断子模块、丢包子模块、带宽子模块、延迟子模块和时间戳子模块组成。网络模拟器将接收到的数据包先交给模拟引擎,模拟引擎根据数据包协议类型判断该数据包是否满足用户设置的过滤条件,若满足,该数据包就被交给丢包子模块等待处理,否则立即发送。丢包子模块会判断数据包是否满足用户设置的丢包条件,若满足则将其丢弃,若不满足则交给带宽子模块。经过带宽子模块处理的数据包,最后被存储在延迟子模块的缓冲队列中。加载模块将用户设置的参数(如丢包率、带宽、延迟值)或 trace 文件中的数据传递给各个子模块,最后由时间戳子模块控制数据包离开延迟子模块,并调用数据包发送模块将缓冲队列中已经达到延迟时间的数据包发送给网卡。

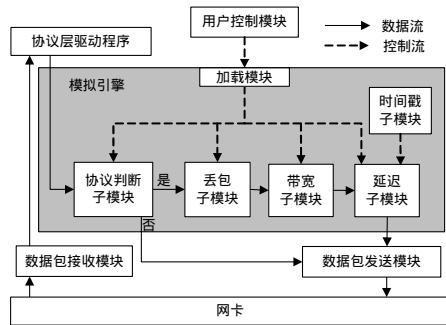


图 2 模拟引擎的功能模块框架

3 网络模拟器实现的关键技术

3.1 高精度延迟的实现

为了使网络模拟器支持实时性的数据包行为,经过实验对比分析,选择利用内核定时器法解决延迟模拟的精度问题。此方法的核心思想如下:将数据包先放入缓存队列,然后利用延迟过程调用(Deferred Procedure Call, DPC)定时器在指定间隔时间后调用发送函数,实现数据包延迟发送。DPC 机制为操作系统提供了在核心态下产生中断并执行系统函数的能力。系统每隔一定时间就执行一个与 DPC 对象相关联的 DPC 例程,产生的定时精度理论上可以达到 100 ns。

在模拟引擎实现中,将时间戳子模块中的周期性内核定时器作为 DPC 对象,发送模块中的发送函数作为 DPC 例程,并将两者进行关联。在延迟子模块中建立一个内存块队列对数据包指针进行缓冲存储和动态维护。当数据包进入缓冲队列时,将被打上入队时间戳并记录。同时,为了保证定时控制的精确性,在时间戳子模块中单独创建内核定时器线程。定时器以 1 ms 为最小周期进行计时,每个计时周期结束后,都读取定时器当前值,一旦有数据包延迟时间到期就调用发送函数,将它们发送出去。

3.2 数据包动态调度的实现

为了模拟现实网络环境下延迟的动态变化,网络模拟器需要实现基于 trace 文件的延迟发送功能。网络模拟器利用折半查找插入法动态维护延迟子模块中的数据包缓冲队列。到达队列的数据包先按照数据包延迟发送时间的预期值从小到大排序后再发送,该方法的实现流程如下:

(1)记进入缓冲队列的数据包为 $p(p \in P, P$ 为数据包缓冲队列),入队时间戳值为 t_s , trace 文件中对应的该包延迟值为 t_d ,则 p 的延迟发送时间的预期值 $t=t_s+t_d$;

(2)采用折半查找插入法,在数据包延迟发送时间的预期值队列 T 中,得到 t 的插入位置 $index$,使 $T[index-1] < t < T[index+1]$;

(3)将 t 插入到 T 中 $index$ 所指向的位置,使队列 T 递增排列;

(4)将数据包 p 插入到 P 中 $index$ 所指向的位置;

(5)定时器每毫秒更新一次后,比较时间戳子模块中的定时器值与 T 中 $T[i](i=0, 1, \dots)$ 值的大小,若相等,则调用发送模块将 $P[i]$ 发送出去,若小于,则继续存储在缓冲队列中。

3.3 数据包丢弃的实现

NDIS 通过一个包描述符(Packet Descriptor, PD)识别数据包,PD 是由一个或多个缓存描述符(Buffer Descriptor, BD)组成的链表,而 BD 中记录了存放数据包的地址和下一个 BD 的地址。当 NDIS_PACKET 通过 NDIS 在各层驱动之间传递时,PD 的所有权临时转移到接收层的驱动中,而 NDIS 要求 PD 只能在相邻层传递,所以,在接收层继续传递数据包之前必须重构一个相同的数据包。

NDIS 的数据包发送流程如下(如图 3 中实线所示):

(1)协议驱动调用 NdisSend 向下层发送数据包 Packet;

(2)NDIS 调用中间层驱动的 MPSendPackets 函数将上层传递下来的数据包构建另一个数据包 MyPacket,同样调用 NdisSend 将此数据包传递到下一层驱动。如果返回 pending,就在步骤(5)中释放 MyPacket 的资源,否则在本函数中立即释放资源;

(3)网卡驱动收到从中间层传递下来的 MyPacket;

(4)网卡驱动调用 NdisSend 发送 MyPacket;

(5)网卡驱动调用 NdisMSendComplete,NDIS 接着调用中间层的 PtSendComplete 函数释放步骤(2)中 MyPacket 的所有资源,并通知上层协议驱动释放步骤(1)中 Packet 的资源;

(6)协议驱动释放 Packet 的所有资源。



图 3 数据包丢弃的实现原理

本文对中间层驱动程序进行修改,在步骤(2)中间层中的 MPSendPackets 函数中完成步骤(5),网卡驱动就无法得到上层传递的数据包从而实现了包丢弃。修改后发送流程如图 3 中虚线所示。

4 网络模拟应用举例

4.1 模拟测试环境

实验拓扑及 IP 地址配置如图 4 所示。网络模拟器运行在装有 Pentium4 1.80 GHz 处理器、512 MB 内存的 PC 机上,安装的网卡都是 RealTek RTL8139 Family 系列。发送端为

ACER 笔记本电脑,接收端是 HP 台式机,以上 3 台 PC 机使用的都是 10/100 Mb/s 自适应的全双工网卡,相互之间使用 10/100 Mb/s 双绞线进行连接,安装的操作系统为 Windows XP Professional。

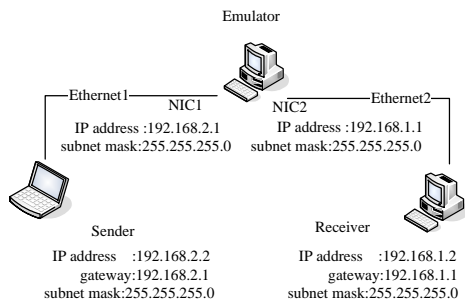


图 4 测试实验的逻辑拓扑

4.2 模拟测试实验

4.2.1 传输延迟模拟

将网络模拟器的协议过滤条件设置为“对 ICMP 包进行延迟模拟测试”。发送端持续 60 s 不断发送 1 470 Byte 长度的 ICMP ECHO REQUEST 数据包至接收端。选用的测试工具是 Fping，可以显示 0.1 ms 分辨率的 RTT 值，测试结果如表 1 所示。将延迟期望值设置为常量，在 5 ms~500 ms 的测试区间中，延迟模拟的相对误差率始终小于 5%，CPU 占用率也保持在 5% 以下。

表 1 传输延迟模拟精确性比较

延迟期望值/ms	延迟测量值			CPU 占用率/(%)
	平均值/ms	标准差/ms	相对误差率/(%)	
5	5.2	0.6	5.0	5
10	10.4	0.6	4.6	4
20	20.1	0.6	1.8	4
50	50.4	0.5	0.8	5
100	100.2	0.5	0.3	4
500	498.7	0.6	0.3	4

加载一个满足标准正态分布的 trace 文件进行动态延迟模拟测试，其中，共 800 个延迟期望值，从 27 ms~197 ms。图 5 显示的是 trace 文件中延迟期望值与测量值的曲线对比，分别用虚线与实线表示，测试中 CPU 占用率始终小于 5%，各点测量结果的相对误差率都小于 2%，因此，图 5 中 2 条曲线基本重合在一起。

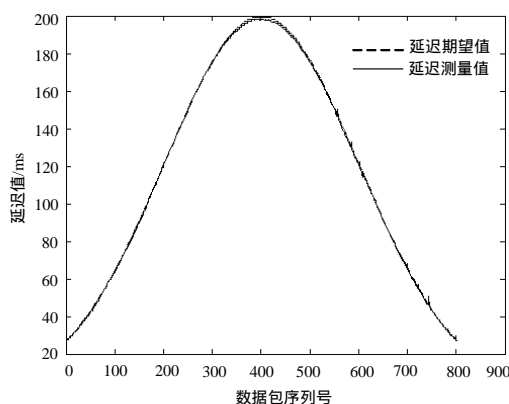


图 5 基于 trace 文件的动态延迟模拟

4.2.2 带宽及丢包传输模拟

对 1 470 Byte 长的 UDP 数据包进行了带宽模拟测试,即将网络模拟器的协议过滤条件设置为“对 UDP 包进行操作”。

实验中测试工具选用 Iperf，带宽的测试范围是 50 Kb/s~4 Mb/s。图 6 是经过测量得到的 60 s 内平均带宽与期望值的对比。可以看出，带宽测量值曲线与期望值曲线很接近，但随着带宽期望值的增大，2 条曲线之间的相对误差率有增大的趋势。这是由于模拟引擎的实现是将带宽期望值转换成延迟值，且受延迟模拟的精度限制以及转换时小数取舍造成的误差影响，因此带宽测量值与期望值会产生偏差，但测试结果显示相对误差率始终在 2.25% 以下。

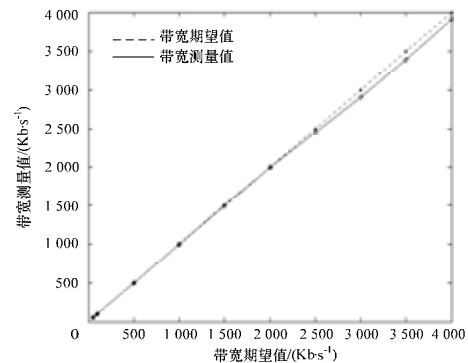


图 6 链路带宽模拟精确性对比

将 UDP 数据包的丢包率期望值设置为 10%~90% 范围的常量值。表 2 中显示的是 10 s 内随机丢包模拟结果的相对误差率，始终保持在 3% 以内。

表 2 网络模拟器丢包率精确性比较

丢包率期望值/(%)	丢包率测量值		
	平均值/ms	标准差/ms	相对误差率/(%)
10	9.9	0.9	2.5
30	29.3	0.8	1.9
50	49.8	0.6	1.2
70	69.9	0.6	0.8
90	89.7	0.7	0.6

4.2.3 数据包错序传输模拟

利用缓冲队列中数据包的重排机制模拟数据包错序现象，并验证网络模拟器对数据包动态调度的有效性。选择加载 trace 文件，该文件中有 2 个数据包的延迟时间较长，使延迟缓冲队列需要按延迟时间期望值进行重新排序。将模拟器的模拟带宽设置为 1 Mb/s，丢包率为 0，发送端利用 Iperf 发送 1 470 Byte 的 UDP 包，接收端设置为 0.1 s 后停止接收数据。图 7 是 Iperf 的结果截图，图中框内的结论显示接收端有 2 个逆序的数据包。这 2 个数据包在进入延迟子模块的缓冲队列时，前一数据包仍未离开，而它们的延迟发送时间的预期值均小于前一数据包，根据折半插入查找法，它们在重排后被提前发送。

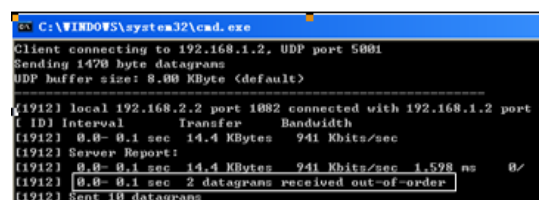


图 7 数据包逆序模拟

5 结束语

本文在 Windows 平台下开发了一个高精度、低负荷的网络模拟器，可用来实时模拟各种网络链路传输条件。所实现
(下转第 147 页)