

# 一种实现端到端网络性能测量的方法

方钟亮 曹争

(东南大学计算机科学与工程系, 南京 210096)

**摘要:** 端到端的网络性能检测在当今的网络中变得日益重要。本文介绍分析了 OWDP 和 RTDP 两种端到端的网络性能的测量原理和使用特点。并在 RTDP 的基础上设计和实现了一个端到端网络性能测量的系统, 本系统实现方便, 灵活性强, 测量的范围广。通过本系统的运行可以很好的反映网络的性能变化。

**关键字:** 服务质量, 往返延迟及丢包协议, 延迟, 抖动

## An implementation of end to end network performance measurement

Fang Zhongliang Cao Zheng

(Department of Computer Science and Engineering, Southeast University, Nanjing 210096)

**Abstract:** Measurement of end to end network performance is very important in today's industry. This paper introduces OWDP and RTDP theory and analyses their characteristics. Then the architecture of measurement system based on RTDP is described in this paper. The system has good flexibility and broad measure scope, which can clearly show the change of network performance.

**Key words:** QoS, RTDP, Latency, Jitter

## 1. 引言

网络性能的测量的主旨就是在网络正常运行的情况下, 通过测试网络的一系列性能参数, 使得网络的用户和网管人员对网络的状况有一个清晰的认识, 从而可以做出网络故障的诊断和资源的合理配置。端到端的网络的性能与网络的服务质量 (Quality of Service, 简称 QoS) 密切相关, QoS 目的就是保证各种数据的传输质量, 这也是网络用户和网络服务提供商 (Internet Service Provider, 简称 ISP) 十分关注的问题。根据 IETF 的 IPPM 工作组定义的反映网络层和传输层的性能的指标包括了网络的连通性, 单向延迟, 往返延迟, 丢包率, 抖动, 乱序等。这些指标有一个共同的特点就是它们会随着时间, 网络的流量或业务的种类的改变而改变。本文就针对端到端的 IP 网络, 分析了网络性能测量的基本原理, 提出了一个性能测量的框架系统, 同时对于网络的往返延迟, 抖动, 丢包等性能指标进行了测量和分析。

## 2. 测量的原理

对于网络性能测量, 目前世界上比较成熟的理论有两种。一种是 OWDP (One way delay and package loss protocol), 另一种就 RTDP (Round trip delay and package loss protocol)<sup>[1]</sup>。OWDP 的基本思想是: 由源端向目的端发送带有时间标志的数据包。在目的端, 当接收到数据包, 就打上本地时戳, 计算出单个报文的时延。若在一定的时间内, 未接收到测试数据则作为该数据包丢失处理并记录。以此来计算其他一些性能参数。用 OWDP 的优点就是测试的精度高, 适用于进行精度要求很高的实时性的检测。但实现的难点是发送端和接受端的时间同步问题, 为了同步必须用到像 GPS 或 NTP 等时钟同步的协议, 因此在发送端和接受端都必须装上各种相应的装置, 还要用到第三方的服务, 来协调彼此的之间的同步。假设测量的一个端点不变, 这对于测量的目标随机性变化较大, 测量范围比较广的端到端的测量, 用 OWDP 就变得相当麻烦了。

RTDP 的基本思想就是通过计算网络数据包的往返延迟来评测网络的其它性能。往返延迟根据 RFC2681<sup>[2]</sup>所定义的是源主机 src 在 T1 时刻将数据发往宿主机 dst, dst 收到数据后马上把收到的数据发回发送主机 src, src 在 T2 时刻收到了从 dst 发回来数据的最后一个字节, 则 RTT 就等于 T2 - T1。丢包的处理就是在指定的时间内, 发送端没有收到从测试的另一端发送的回应信息就作为数据的一次丢包并记录下来。虽然 RTT 不能精确的反映出网络的单向延迟, 因为来回的时间是存在着差异的, 且往返的路径有可能不同, 但是通过计算 RTT 可以有很多有用之处。对于实时交互性比较频繁的网络应用来说 RTT 就能反映出网络的实际的通信的效率了。延迟的变化可以反映出带宽的变化, 越大的延迟说明了传输层带宽的利用就越低了。而延迟的最小值又反映出了网络在最小负载情况下的性能, 超过了最小值的延迟大致表明了网络的拥塞的变化情况。在精度要求不是很高的情况下可以用来估算网络从 src 到 dst 的单向延迟, 及其它一些性能参数等。

**作者简介:** 方钟亮 (1980-), 男, 硕士生; 曹争, 男, 高级工程师。

通过周期性能的长期测试可以反映出这段时间里网络的性能的变化。

与OWDP相比较而言，RTT的计算比起单向延迟的计算要来的简便，计算的时间都是在同一台计算机上，因此不存在时间的同步问题，无需在测量的两个端点处都安装上测量的软件。只要固定一个测量点另一个测量点就可以是任意的选择了，从而使得测量的范围可以扩充到整个Internet。

### 3. 测量的方法

网络测量根据测量点的多少,分为单点测量与多点测量;根据被测量者知情与否,分为协作式测量与非协作式测量;根据测量的方式,分为被动测量和主动测量。被动测量适用于分散在测量网络的各个节点上对经过的报文进行统计,从而对经过该测量点上的流量做出宏观的分析,但这种测量不适合于端至端的测量。在主动测量方式中,通过向网络中发送数据,观察结果来研究网络的行为。通过向网络中发送实际的业务量,利用这些业务量测量反映网络提供给其他用户的服务的参数,主动测量比较适合于端到端测量。

在本文所述的测量系统是基于RTDP理论的属于一种主动的,非协作式的,基于TCP/IP的端至端测量。

### 4. 系统框架

本系统的框架由以下三部分组成:

1) 访问协议:它位于网络的应用层之上,用户通过HTTP的协议,基于WEB的形式访问系统。在访问界面上可以设定各种控制参数,来进行不同要求的测试。同时也是这种形式来查询最后的测试结果。通过HTTP的WEB形式使得系统可以被远程的并发的访问。

2) 测量控制协议:这层协议是基于网络的TCP之上,负责建立和传输控制参数与测试结果。通过该协议,测试端总是开启一个接受服务进程,当控制端的用户设置完测试参数后,首先是建立发送进程,在控制端和测试端以一个固定的TCP端口建立一个可靠的连接,然后将所有的控制信息一定的字符串格式发向测试端。

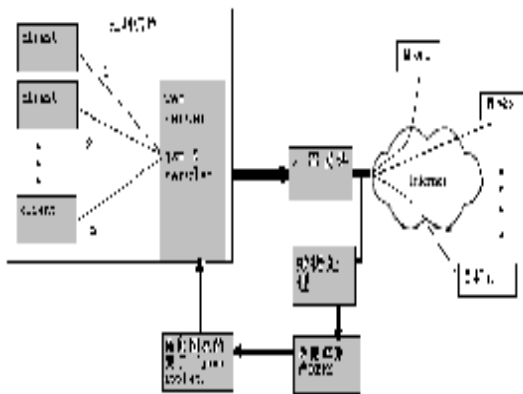
3) 测试协议:这是建立在网络的第三层ICMP协议层之上。ICMP协议的特点是当向某个主机发送ICMP请求回显报文时对方主机会向发送主机发送ICMP回显应答报文。因此测试协议负责在测试端与选定的目标端之间进行发送和接受测试包,并在发送和接受时都盖上时间戳。后续的处理以此来计算一系列的性能参数。整个框架的示意图如图一



图一 系统框架

### 5. 系统的实现

在系统的结构功能上,一共分成了5个模块:控制模块,测试模块,数据库存储管理模块,参数计算模块,数据的显示模块,如图二所示。



图二 系统结构

1) 控制模块是以WEB的形式向用户提供一个控制界面,用户可以在此确定测试端点的地址,发送测试信息报文的数量,发送的时间间隔,发送的队列的长度和报文的长度。本系统选用了RESIN作为支持WEB的服务器,RESIN是一种支持JSP的商用化高效的服务器。用户在WEB页面上输入的所有参数,提交到服务器端后,服务器通过一个JAVA SERVLET的形式将所得的参数转化成一个字符串的流,再通过SOCKET的功能选择固定端口2005将这个流传输到就测试模块所在的主机上。由于JAVA具有良好的可移植性。整个以JSP/SERVLET为基础的控制端同样拥有相当良好的平台移植性。

2) 测试模块是运行于一台LINUX 8.0的工作站之上,测试模块共分成3个部分,一是控制信息的接受部分,二是探测报文的发送部分,三是回送报文的接受部分。控制信息接受部分的实现就是在测试端开启一个SOCKET的接受进程,在2005端口接受控制信息。接受的信息是以字符流的形式传送过来的,因此还需要将数据转换成适当的参数类型,然后将转换后的参数传递给发送部分。

发送部分就是根据所得的控制参数,向一个目标主机以固定的频率和报文长度发送探测报文。在LINUX下发送ICMP报文的主要的程序以C语言实现<sup>[4]</sup>如下:

```
sock=socket(AF_INET,SOCK_RAW,IPPROTO_ICMP); //创建ICMP套接字
...
static u_char outpack[MAXPACKET];
register struct icmp *icp = (struct icmp *) outpack; //定义ICMP头指针
icp->icmp_type=ICMP_ECHO; //设定为ICMP请求报文
icp->icmp_code=0; //设定报文代码,默认为0
icp->icmp_id=发送进程的进程号; //设定报文id号
icp->icmp_seq=发送报文的序列号; //设定报文的序列号
*((long *)icp->icmp_data)=time_now(); //打上发送时间戳
icp->icmp_chksm=ChkSum (icp,sizeof(icp); //计算校验和
...
sendto(sock,buf,sizeof(icp),0,(struct sockaddr *)&disternation,sizeof(disternation)); //报文的发送
```

接收部分则负责的是接受返回的信息,并负责将接受由被测的目标主机返回的数据,从而计算探测报文往返时间RTT,报文的接受报文的步骤如下:

```
for (;;)
{
    cc=recvfrom(sock, packet, len, 0,(struct sockaddr *) &from, &fromlen);
    //从SOCKET读取接受到的报文数据
    if (cc > 0)
        pr_pack( packet, cc, &from ); //进行解包处理
}

```

pr\_pack()函数是一个解包的函数,解包的流程如下:

一 类型的转换,将字符串数据指针转换成IP报文格式的指针,再将指针后移,移动的大小为IP报头长度,将此时的指针再转换成ICMP的报文格式。

二 回应报文的确定, a.判断报文的类型是否为固定的ICMP回应报文类型。 b.确定报文的id项是否为发送进程的进程号。 c.报文序号的确定。满足以上三点的就能说明该报文是测试报文的回应报文。

三 测试报文的往返时间RTT的计算,将本机的时间减去数据报包字段中属性data的值就是所要求的往返时间RTT的值。整个测试模块作为一个常驻进程运行在测试的一个端点主机上,当受到控制信息就会被激活,开启报文的发送和报文的接收的线程。因此系统具有一定的并发的处理能力,可以同时根据不同的用户要求进行不同目标进行检测。

3) 处理模块负责的就是用各种评价算法对测试模块中所得到的数据进行处理,从而得到平均RTT时延,时延抖动,丢包率和网络吞吐量等性能指标,并将这些数据存入数据库。

4) 数据库管理模块主要处理的是将所得的计算结果以一定的格式存储,并提供其它模块统一的数据接口来进行数据的操作,系统所用的是基于 Linux 的 MySQL 数据库,可以用于小型网络数据库。整个数据库共有为有三张表,第一张表存放了各测试阶段的所有的统计的值,包括 RTT 的最大值,最小值,平均值,丢包率,抖动值等。第二张存放了每次测试的 RTT 值和抖动的值,用来作为图像处理的数据。第三表存放了一些对于网络性能预设的标准参数,可以用来作为对实际测量值的一个比较,来查看是否达到了一定的要求。对数据库所有操作都是基于 SUN 公司的 JDBC<sup>[5]</sup>来进行的。

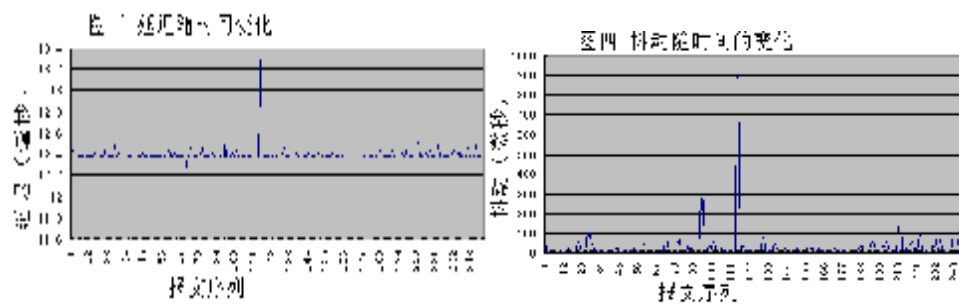
5) 最后的数据显示模块则将数据从数据库中读出以WEB的形式显示出来,同时用到了JAVA APPLET 的技术,将数据绘制成图表的形式显示出来。

## 6. 测量系统的应用

作为测量系统,应该满足各种测量的需求,该系统提供了各种不同的测量的参数设定,其中包括了检测的次数,测试的频度,每个测试报文的长度,每次检测发送的队列长度。对以上参数的不同设定可以进行不同要求的测量,下面列出了三个系统应用的例子。

### 1) 默认的检测

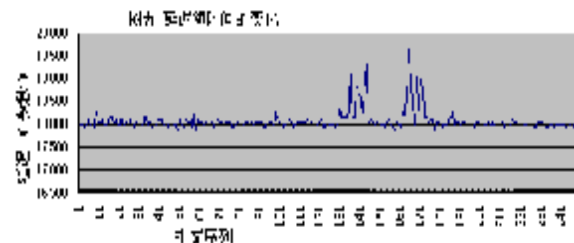
系统默认的测量参数是每一秒钟检测一次,共检测 250 次,每个报文长度为 60 个字节,每次检测发送一个报文。对于以上的默认的参数设置,在网上两台主机之间进行了端点的测试。同时取出了所得的延迟和抖动的数据如下图所示。



图三和图四分别显示出了在连续的 250 秒里网络的延迟和抖动的情况,从上图可以看出两端的之间的延迟基本保持在 12.4 毫秒左右,没有报文丢失的现象,同时网络的最大抖动为 910 微秒,小于 1 毫秒的抖动属于正常情况,说明了网络在测试期间网络处于十分稳定的状态。

### 2) 测试频率的改变

本系统提供了改变报文发送时间间隔的选项,这样就可以调整发送报文的发送频率进行各种测试。图五是对某一热门网站进行的测试结果统计,相隔 12 分钟发送一个测试报文,报文的大小为 1024 个字节



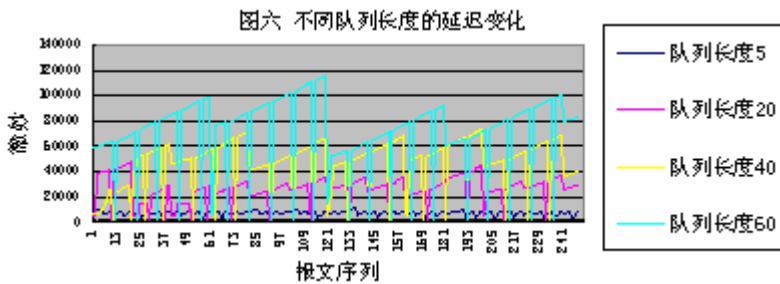
从上图显示的测量结果可以看出,在统计的 250 个报文既 50 个小时里相继报文的延迟没有一个明显的规律变化也没有丢包的现象,在第 130 到 145 和 160 到 174 之间的网络延迟产生了明显的峰值增大的变化,其它的时间里网络的延迟基本维持在 18 毫秒左右,最大的延迟峰值为 19.5 毫秒,超出平均的延迟仅 1.5 毫秒,变化不是很大,但反映出了当时网络拥塞成度的增加。

不同频率的检测,适用于不同要求的测量。对于周期性的检测而言,通常分为每 15 分钟一次的检测来统计一天的端到端之间的情况,每 30 分钟一次来统计一周的性能,每两个小时一次来统计一个月的性能情

况，一天一到两次来统计一年的情况。对于年，月的统计一般是用于长期的网络规划和流量分析,对于天和小时的测量统计则多用于容量的管理如 MPLS tunnel,而如上文所述的每秒钟一次的检测则用于网络实时的监制。网络抖动主要是反映了网络实时的稳定情况，对于长时间的间隔测试意义不大，在此就不讨论了。

### 3) 队列长度的改变

传输队列指的是一次连续发送 n 报文数，这些报文之间没有时间间隔，n 就是队列的长度。各种路由器对队列的处理方法都不一样。通过不同队列的大小对于网络的测试可以显示出路由器对队列处理的一些性能。



图六显示了队列长度分别为 5, 20, 40, 60 的测试情况，其中测试报文的大小都是 1024 个字节，检测的频度为每 5 秒钟一次。从上图可以看出网络的延迟在传输队列长度增加的情况下产生了明显的增加。延迟的变化呈现出周期性锯齿形的变化，每个“锯齿”的长度正好等于一组队列的长度。“锯齿”线性上倾的变化说明了网络在处理同一队列头部报文的延迟较小而在处理到队列尾部的报文时延迟较大。这也反映出同一组报文越往队尾带宽利用就越低，在每一组队尾的处理时网络的拥塞加重了。

在锯齿中间出现了值为 0 的现象，这说明了网络产生了丢包的现象。这和中间路由器的接受缓存大小和转发速度有关。当报文的到达速率大于节点处理报文的速率时就会产生报文排队的情况，节点会以一定的算法来处理排队等待的报文，一般是以 FIFO 的方法进行。当排队等待的队列长度大于节点的接受缓存时就会产生报文的丢弃现象。从上图六可以看出队列长度为 40 和 60 时，在去掉“锯齿”的首尾部分后中间的两个相邻丢包之间的距离在 9 或 10 个报文长度之间变化。在队列长度为 20 的测试结果中每个“锯齿”的丢包出现在队列的中间部位，恰好分成两个大小为 10 的部分。由此可以估算出测试的端到端之间的节点最小的接受缓存大概可以容纳 10 个报文即 10k 个字节左右。

以上是系统的一些不同的应用，可以对各参数的不同设置搭配来对网络进行不同需要的测试，这里就不一一介绍了。

## 7. 结束语

本文实现的基于RTDP的测量系统，根据其运行的情况看，具有良好的稳定性，系统开销小。基于WEB的访问模式使得系统可以远程的控制同时进行并发的测试。通过系统对端到端的测量所得到的一系列性能参数进行可靠性，稳定性，可达性等分析，可以使得网管人员对网络进行及时的维护管理，保障服务质量。同时周期性的测量数据可以用作对网络动态预测的基础数据，可以用在广域网上大规模的计算调度。因此本系统具有很大的实用价值。

## 参考文献

- [1] 凌军,杨桂文,刘羽中,曹阳. Internet网络性能实时监测关键技术研究. 武汉大学学报(理学版), 2001 47(3) 324-326.
- [2] RFC 2681 A Round-trip Delay Metric for IPPM
- [3] 张宏莉,方滨兴,胡铭曾,姜誉,詹春艳,张树峰 Internet测量与分析综述. 软件学报 2003 14(1): 110-116.
- [4] 张威 Linux网络编程教程 北京希望电子出版社
- [5] 最新JAVA2入门与应用 中国铁道出版社