

一种基于分段测量的 Internet 端到端性能参数估计方法

马震远^{1,2} 周 杰^{1,2} 陈 楚² 张 凌^{1,2}

(华南理工大学计算机学院 广州 510640)¹

(华南理工大学广东省计算机网络重点实验室 广州 510640)²

摘 要 传统的基于测量的 Internet 性能参数估计要求测量点是端到端路径的起点和终点,这一方面使得测量和估计带有很大盲目性,增加了网络流量;另一方面会导致测量方法难于实施,估计结果不准确。提出了基于分段测量的思想,对 ITU-T 定义的延迟、延迟变化、丢包率和错误率 4 个参数进行测量和估计。由于测量是分段进行的,每次测量值都可应用于多条端到端路径的性能参数估计,因此降低了盲目性并大大减少了网络流量。为验证本方法的性能,基于 PlanetLab 平台设计和实现了生成器、性能参数测量机和结果收集器。通过在全球的 100 多个节点上的部署和为期一个月的运行证明了该方法便于实施,并且具有较高的准确性。

关键词 性能预测,分段测量,延迟变化,PlanetLab

中图法分类号 TP393.06

文献标识码 A

Segmental Measurement Based Approach to Estimate Internet End-to-end Performance Parameter

MA Zhen-yuan^{1,2} ZHOU Jie^{1,2} CHEN Chu² ZHANG Ling^{1,2}

(School of Computer Science, South China University of Technology, Guangzhou 510640, China)¹

(Communication and Computer Network Laboratory of Guangdong Province, South China University of Technology, Guangzhou 510640, China)²

Abstract The traditional measurement based method for estimating Internet performance parameters requires measurement points to be the start and the end of an end-to-end path. Such condition can make measurement aimless and increase network traffic. Meanwhile the estimation may be hard to implement, and the results may also be less accurate. This paper proposed a novel approach based on segmental measurement to estimate delay, delay variation, loss rate and error rate, which were defined by ITU-T. As each measurement can be used to several end-to-end path performance parameter estimations, the aimless of measurement was avoided and the network traffic was cut down significantly. Three module, including packet and path generator, parameter measurement machine and result collector, were designed and implemented based on PlanetLab. Its feasibility was illustrated by deploying on more than 100 world wide nodes. And the running data collected by result collector proved its accuracy.

Keywords Performance estimation, Segmental measurement, Delay variation, PlanetLab

基于测量的 Internet 端到端性能参数估计是一个重要的研究课题,有着广泛的应用价值。通过准确地估计性能参数,服务器可以控制业务流的发送速率和缓冲区大小。例如,使用 VoIP 进行通话时,如果端到端的延迟变化过大而预留的数据缓冲区不足,那么声音在某一段时间可能没有被发送,而在另外一段时间里又集中被发送出去,对方听到的将会是不连贯甚至是发生变形的声音。准确地估计端到端的性能参数,还可以有效地构建组播树。例如,时下流行的第一人称的射击类游戏,需要为参与游戏的用户构建组播树,当端到端延时超过 50ms 时,用户就会有明显的停顿感,而每减小 10ms 的延时变化用户射击的真实感就会提升一个档次。

1 端到端性能参数

Internet 本质上是一个尽力而为的包交换网络,在 Inter-

net 上提供端到端服务质量保证可等效为保证包流在端到端之间传输的性能参数满足一定的要求。国际上的两大组织 ITU-T^[1,2] 和 IETF^[3] 有多个工作组致力于端到端性能参数方面的研究,其中 ITU-T 的 SG-12 工作组在下一代互联网 QoS 的提案中^[1],用延迟(TD, Transfer Delay)、延迟变化(DV, Delay Variation)、丢包率(LR, Loss Rate)和错误率(ER, Error Rate)4 个参数定义了包交换网络的服务质量类别。

表 1 给出了不同服务质量类别对应的 4 个性能参数值, U 是指该类别对此参数不作要求。其中 C0 和 C1 可用于对延迟和延迟变化都敏感的服务,例如 VoIP, VTC 等。C2—C4 可应用于对延迟变化不敏感而对传输延迟敏感的服务,例如事务数据处理。C5 是 Internet 基本的尽力而为服务。C6 和 C7 除了要求延迟和延迟变化外,还提高了对丢包率和错误率的要求,适用于实时性要求高的交互类服务。

到稿日期:2009-05-08 返修日期:2009-07-28 本文受国家 973 计划(No. 2009CB320505),国家科技支撑计划(No. 2008BAH37B08)资助。

马震远(1980—),男,博士生,主要研究方向为 IP-QoS, anycast 通信方法, E-mail: Mazy@scut.edu.cn; 周 杰(1964—),男,博士,副教授,主要研究方向为高性能网络技术; 陈 楚(1980—),男,博士生,主要研究方向为分形几何理论、网络流量模型; 张 凌(1962—),男,教授,博士生导师,主要研究方向为计算机网络管理与安全、下一代高性能计算机网络。

表1 ITU 服务质量类别定义^[2]

性能参数	QoS 类别							
	C0	C1	C2	C3	C4	C5	C6	C7
TD	100ms	400ms	100ms	400ms	1s	U	100ms	400ms
DV		50ms		U		U		50ms
LR			1×10^{-3}			U	1×10^{-5}	
ER			1×10^{-4}			U	1×10^{-6}	

针对上述性能参数,目前基于测量的估计方法主要分为两类:基于测量包时间间隔的 PGM(Probe Gap Model)估计方法^[4]和基于测量包速率的 PRM(Probe Rate Model)估计方法^[5]。PGM 方法是通过计算测量包的输出间隔和输入间隔之差得到路径的性能参数。而 PRM 方法主要是通过观察测量包的输入输出时间间隔的变化来估计路径的性能参数。这两种方法以及其它方法^[6]都必须测量完整的端到端路径,也就说两个测量点必须是端到端路径的起点和终点。这大大限制了性能参数估计的应用范围,因为通常端到端路径会跨越多个节点,这些节点可能属于多个运营商网络,网络状况随时间和突发流量不断变化,经常性的探测任意两个端节点之间的网络状况是低效的且盲目的。而在通信开始前通过端到端测量的方式获得某一时刻的网络状况,会增加通信的时间开销,增加网络负担。我们注意到,通常相邻的两个节点比较容易进行测量和实时监控,因为相邻的测量点往往位于同一个运营商,采用同样的实现技术和管理手段。如果端到端的性能参数可以被拆分为多段路径性能参数的组合,问题将大大简化。下文以分段性能参数来估计 ITU-T 定义的端到端路径的性能参数。

2 分段性能参数合成

设从起点 A 到终点 B 的端到端路径需要经过中间节点 $N_1, N_2, \dots, N_{n-1}$, 记 P_1 是起点为 A 终点为 N_1 的路径, P_i 是起点为 N_{i-1} 终点为 N_i 的路径, P_n 是起点为 N_{n-1} 终点为 B 的路径。考察对象为包流 S, 包含 $S_1, S_2, \dots, S_x$ 共 x 个数据包。记 TD_P, DV_P, LR_P, ER_P 分别是路径 P 的延迟、延迟变化、丢包率和错误率。ITU-T 关于端到端路径上包流的延迟定义是^[2]: $S_1, S_2, \dots, S_x$ 通过路径 AB 所耗费延迟时间的数学平均值, 因此端到端延迟可用如下公式计算:

$$TD_{AB} = TD_{P_1} + TD_{P_2} + \dots + TD_{P_n} \quad (1)$$

对于丢包率和错误率, 由于数据包在任意分段路径发生丢失或错误后都不可恢复, 因此:

$$LR_{AB} = (1 - LR_{P_1}) \times (1 - LR_{P_2}) \times \dots \times (1 - LR_{P_n})$$

$$ER_{AB} = (1 - ER_{P_1}) \times (1 - ER_{P_2}) \times \dots \times (1 - ER_{P_n})$$

由于丢包率和错误率一般很小, 因此可以有如下近似:

$$LR_{AB} \approx LR_{P_1} + LR_{P_2} + \dots + LR_{P_n} \quad (2)$$

$$ER_{AB} \approx ER_{P_1} + ER_{P_2} + \dots + ER_{P_n} \quad (3)$$

ITU-T 关于端到端路径上包流延迟变化的定义为:

$$DV_{AB} = \text{upper}(TD_{AB}) - \min(TD_{AB}) \quad (4)$$

其中, $\text{upper}(TD_{AB})$ 是指从 A 到 B 路径上包 $S_1, S_2, \dots, S_x$ 延迟的 99.9% 分位数, $\min(TD_{AB})$ 是最小延迟。相比较 IETF^[3] 以及其它文献^[4,7] 对传输延迟变化的定义, ITU-T 将延迟变化定义在一个时间段内, 消除了网络上小概率事件对延迟变化的影响, 因而更加准确地反映了包流的抖动程度。

用式(1)~式(3)可以通过分段延迟、分段丢包率和分段错误率来较为准确地估计端到端的总体延迟、丢包率和错误

率。但按照定义对延迟变化的估计就比较困难, 下面采用概率的方法由分段延迟变化来近似估计总体延迟变化。

3 延迟变化上界估计

考察包流 S 的 x 个包, 设 $t^S_j, 1 \leq j \leq x$ 为包 S_j 在 AB 间传输所需要的时间, 设 $T = \{t^S_j | j=1, 2, \dots, x; t^S_j \leq t^S_{j+1}\}$, 首先来估计式(4)中的 $\text{upper}(TD)$ 。

设 t^S_i 为包 S_i 通过路径 P_i 所需的时间, $1 \leq i \leq n$, 明显有 $t^S_i = t^S_{i1} + t^S_{i2} + \dots + t^S_{in_i}$, 将 $T_i = \{t^S_{ij} | j=1, 2, \dots, n_i\}$ 视为随机变量, 则有:

$$T = T_1 + T_2 + \dots + T_n \quad (5)$$

这里需要强调的是, 我们并没有对 T_i 可能的分布情况进行任何假设。Ramsey 证明了任意随机变量标准化后其上界都可以通过偏度近似估计^[8], 且偏度越接近于 0 估计值越准确。将 T 标准化后, 有如下近似:

$$\frac{t^S_j - \mu}{\sigma} \approx Z + \frac{\gamma}{6}(Z^2 - 1) \quad (6)$$

其中, μ 是 T 的均值, σ^2 是 T 的方差, γ 是 T 的偏度, Z 是标准正态分布。由式(6)可得:

$$Z \approx \left(\frac{1}{\gamma} \sqrt{9 + 6\gamma \left(\frac{t^S_j - \mu}{\sigma} \right) + \gamma^2} - \frac{3}{\gamma} \right) \quad (7)$$

同理对路径 P_i 有:

$$Z \approx \left(\frac{1}{\gamma_i} \sqrt{9 + 6\gamma_i \left(\frac{t^S_{ij} - \mu_i}{\sigma_i} \right) + \gamma_i^2} - \frac{3}{\gamma_i} \right) \quad (8)$$

其中, μ_i 为 T_i 的均值, σ_i^2 为 T_i 的方差, γ_i 是 T_i 的偏度, 设 ω 和 ω_i 分别为 T 和 T_i 的三阶中心距则:

$$\omega = \gamma \cdot \sigma^3 \quad (9)$$

$$\omega_i = \gamma_i \cdot \sigma_i^3 \quad (10)$$

由于分段路径可以由用户自行选取, 不同的分段路径位于不同的运营商, 它们之间是相互独立的, 因此认为随机变量 $T_i, i=1, 2, \dots, n$ 之间也是相互独立的, 从而有:

$$\omega = \omega_1 + \omega_2 + \dots + \omega_n \quad (11)$$

设 X_p, t, t_i 分别为 Z, T, T_i 在概率 99.9% 时的分位数, 由于 Z 是标准正态分布, 因此 $X_p \approx 3.09$ 为常量。将式(7)~式(10)带入式(11)可得:

$$t \approx \left(\sum_{i=1}^n \mu_i + x_p \sqrt{\sum_{i=1}^n \sigma_i^2 + \frac{\sum_{i=1}^n t_i \sigma_i^2}{\sum_{i=1}^n \sigma_i^2}} \right) - \left(x_p \frac{\sum_{i=1}^n \sigma_i^3}{\sum_{i=1}^n \sigma_i^2} + \frac{\sum_{i=1}^n \mu_i \sigma_i^2}{\sum_{i=1}^n \sigma_i^2} \right) \quad (12)$$

按照定义 $\text{upper}(TD_{AB})$ 是 AB 间端到端延迟的 99.9% 分位数, 因此可以用式(12)来近似估计。 $\min(TD_{AB})$ 是端到端延迟的最小值, 用如下经验公式近似:

$$\min(TD_{AB}) \approx k \sum_{i=1}^n \min\{t^S_{ij} | j=1, 2, \dots, n_i\}, k > 0 \quad (13)$$

可以看出, 仅需要知道包流通过分段路径 i 时延迟时间的均值 μ_i 、方差 σ_i^2 、99.9% 分位数 t_i 和最小值 $\min\{t^S_{ij} | j=1, 2, \dots, n_i\}$ 就可以估计端到端的总体延迟和延迟变化。下面通过 PlanetLab 平台对上述方法进行验证。

4 基于 PlanetLab 的实验与分析

4.1 模块设计

PlanetLab^[9] 是一个基于 Internet 的分布式实验床。其基本思想是: 在全球范围内招募志愿主机, 主机间通过 Internet 相互连接, 在应用层上形成覆盖网, 用户可将需要测试的协议

或其它应用程序发布于覆盖网的多台主机上并运行,从而获得 Internet 上的实验结果。PlanetLab 的突出贡献之一是改进和扩展了 Red hat 操作系统,建立了 Slice 机制。通过为用户分配 Slice,将主机的各种资源虚拟地分配给用户,并向用户提供称之为 PLCAPI 的使用接口,使得每台主机的资源管理对用户透明。PLCAPI 提供了一整套应用发布、结果收集以及错误处理机制,使用者可以方便地将实验程序发布,并在全球范围的主机上运行。由于 PlanetLab 采用完全基于 Internet 的结构,因此可以为实验者提供完全真实的环境。截止到 2008 年底,PlanetLab 在全球的 50 多个国家有超过 1000 个节点,大部分节点连接当地主干网络,真实地反映了节点所在网络的运行状况。

验证实验分为 3 个相对独立的模块,包括生成器、性能参数测量机和结果收集器,均以 Python 2.6 实现。生成器模块包含两个功能:一是按照要求生成随机路径,二是生成测量所需的包流序列。工作原理如下:首先通过 CoMon^[10] 获取 PlanetLab 当前可用的活动节点,本次实验固定选取 100 个节点生成全连接的网络拓扑图。从中随机选择一条路径做为本次实验对象,路径长度控制在 3~10 之间。然后生成器为该路径生成包流 S,其固定包含 1000 个包,包长为 2k 字节,然后向每个包中添加路由信息,指定每个包要通过的完整路径。再生成 S 的副本 Q,用于探测端到端延迟时间。最后生成器启动两个独立的进程向第一个路由节点发送 S 和 Q,发送速率固定为 100ms,以 UDP 协议发送。发送的同时于本地文件中为 S 和 Q 记录独立的当前时间戳。生成器是一段独立的程序,部署在每一个 PlanetLab 节点上,通过操作系统的 cron 机制每隔一定时间启动一次。

性能参数测量机在接收到生成器生成的包流后开始工作,并部署在每个 PlanetLab 节点上。为叙述方便,设 C, D, E 是包流传送路径上相连的 3 个节点。节点 D 的测量机工作流程如图 1 所示。需要说明的是,当测量机 D 收到流 S 的某个包时,若 D 不是终点则为该包记录时间戳,然后主程序要同时启动两个独立的线程,分别完成向下一跳发送该包,将副本 S' 发送给上一跳节点,如图 1 中 * 所示。 t_{CD}^S 和 t_{DC}^S 分别表示流 S 的第 j 个包在 CD 间往返的开始和结束时间,两者之差即为流 S 的第 j 个包在节点 C 和 D 之间的 RTT 值。 t^Q 表示流 Q 的第 j 个包在起点和终点间的 RTT 值,图 1 中省略了包序号 j。

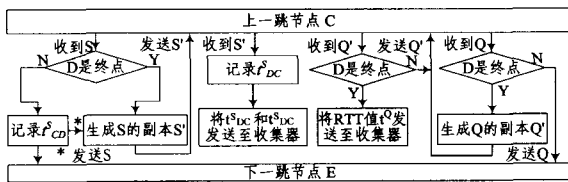


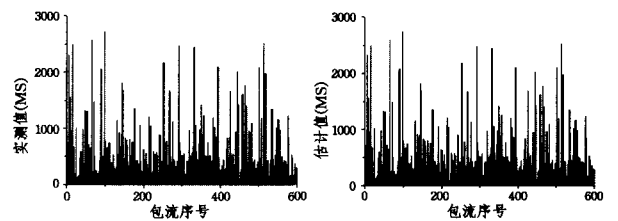
图 1 节点 D 的测量机工作流程图

结果收集器是一个全球可访问的 Web 应用服务器,由一段 Servlet 程序负责收集 PlanetLab 节点的数据。这里并没有使用数据库而是仅向本地硬盘写入节点生成 RTT 数据文件,这样做虽然增大了后期数据的处理量,但提高了效率。最初的版本由于节点多并发数据量过大,导致本地硬盘 I/O 出错,为此修改了测量机的实现,先对 RTT 数据进行压缩,然后发送至收集器。

4.2 数据分析

本实验通过向分布在 63 个地区的 102 台主机发布应用程序来实现数据发送和收集。实验从 2009 年 1 月 15 日开始运行至 2009 年 2 月 26 日止。期间由于 PlanetLab 节点的调整中断了一周,有效运行时间为 35 天。运行的最大规模为 102 个节点,由于部分节点故障,实际的节点数在 98 个左右。数据分析前先对收集到的数据进行整理,将端到端延迟超过 3 秒钟的数据剔除。

图 2 是随机选择的 600 组包流的延迟数据,包含了端到端路径长度从 3 到 10 的所有情况。图 2(b)给出了以式(1)估计总体延迟的情况。需要说明的是由于 PlanetLab 没有提供精确的时间同步机制,因此在生成器和测量机中都以 RTT 来代替单向延时,这里和下文的延迟都是指将 RTT 值除 2 后得到的近似值。可以看出与图 2(a)的实测值吻合得相当好。但由于处理时间损耗,预测值稍微偏大。对于丢包率和错误率,由于 PlanetLab 只能提供应用层的接口,而 Internet 的数据链路层和网络层都具有纠错机制,因此实验数据中的丢失率和错误率很小,在此不做讨论。



(a) 端到端延迟实测值; (b) 端到端延迟估计值

图 2

为了说明不同路径长度下本估计方法的性能,对延迟变化按照路径长度进行分类统计,每种路径随机选取 500 次测量数据作为统计样本。横坐标为测量机中由流 Q 和 Q' 获得的测试结果 t^Q ,也就是实测的延迟变化值。纵坐标是测量机中由流 S 和 S' 获得的分段延迟变化按照式(12)和式(13)计算的近似值,其中对式(13)取 $k=3/4$ 。若估计值与实测值相吻合,则样本的每一组数据都应接近图中斜率为 1 的实线。

由图 3 可以看出绝大多数延迟变化在 100ms 以内,最大值在 300ms 左右,随着延迟变化的增大预测值的偏差略有增大。左右的预测延迟变化值几乎都是小于实测值的,这是因为一方面,式(12)较好地估计了延迟变化的上界,另一方面,对最小延迟的经验式(13)缩小了端到端延迟的实际最小值。

从路径长度为 5, 7 和大于 7 的结果(如图 4~图 6 所示)中可以看出,随着路径长度的增大延迟变化也不断增大,但估计值基本上小于实测值,无论路径长度如何变化,延迟变化在 100ms 以内时估计值都与实测值吻合得较好,超过 100ms 时估计值与实测值的误差也有很大概率保证在 20% 以内,同时无论何种情况,式(12)和式(13)都很好地控制延迟变化的上界,对本实验涉及到的端到端路径估计的精确度都比较高。

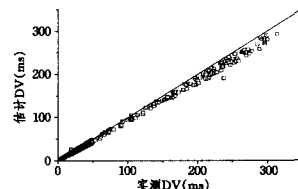


图 3 路径长度为 3 的延迟变化 QQ 图

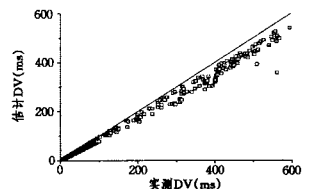


图 4 路径长度为 5 的延迟变化 QQ 图

(下转第 148 页)

重配置处理器定制指令设计是一个典型的剖析驱动优化过程,根据设计需要,本文提出了一种新的代码剖析方法 AID-prof,实现了剖析结果的平台无关性和静态分析与动态分析的无缝契合,并在该方法的基础上构建了自动化的定制指令识别过程 CID。实验显示,利用代码剖析,CID 识别产生的定制指令可使应用程序获得可观的性能提升。

参考文献

- [1] GNU. Profiling a Program: Where Does It Spend Its Time? [EB/OL]. <http://sourceware.org/binutils/docs-2.16/gprof/>
- [2] Seward J, Nethercote N, Weidendorfer J. Valgrind 3.3-Advanced Debugging and Profiling for GNU/Linux applications [M]. Network Theory Ltd
- [3] Srivastava A, Eustace A. ATOM: A System for Building Customized Program Analysis Tools [C]//Proc. of ACM SIGPLAN Conference on Programming Language Design and Implementation, Orlando, Florida, 1994; 196-205
- [4] Karuri K, Faruque M A A, et al. Fine-grained application source code profiling for asip design [C]//Proc. of the 42nd Annual Conference on Design Automation, Anaheim, California, USA, 2005; 329-334

(上接第 140 页)

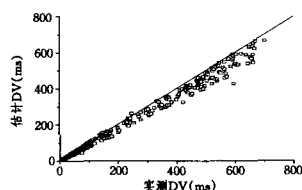


图5 路径长度为7的延迟变化 QQ 图

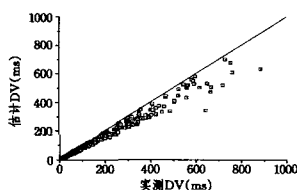


图6 路径长度为7~10的延迟变化 QQ 图

结束语 本文提出了用分段性能参数来估计端到端性能参数的方法,重点解决了由分段延迟变化来准确估计端到端延迟变化的问题。通过在 PlanetLab 上部署生成器和测量机证明了本方法的有效性和准确性。本研究为 Internet 上的服务质量保证提供了一种全新的手段。例如:将 Internet 划分为若干分段,每个分段都维护一系列实时测量参数,以很少的字节描述包流通过本自治域的性能参数。任意两个节点通信时,首先发送信令询问端到端路径上每个分段的当前性能参数,当信令到达通信目的节点时即可计算当前链路所能提供的服务质量,此时接收端可以按照一定的策略选择开始或终止通信。对带宽的讨论超出了本文的讨论范围,调整包长和发送速率可以进一步研究带宽对延迟变化和其他性能参数的影响,这是下一步的研究重点。

最后,感谢普林斯顿大学的 Stephen Soltesz、浙江大学的 PlanetLab 工作组,以及其他对本文的实验工作提供宝贵资源和提出宝贵意见的同行。

参考文献

- [1] Internet protocol data communication service-IP packet transfer

- [5] Wilson R P, French R S, Wilson C S, et al. SUIF: An Infrastructure for Research on Parallelizing and Optimizing Compilers [J]. SIGPLAN Notices (S0362-1340), 1994, 29(12): 31-37
- [6] Smith M D. Machine-SUIF: A Research Infrastructure for Profile-driven and Machine-specific Optimizations [EB/OL]. <http://www.eecs.harvard.edu/hube/software/software.html>
- [7] Atasu K, Pozzi L, Ienne P. Exact and Approximate Algorithms for the Extension of Embedded Processor Instruction Sets [J]. IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems (S0278-0070), 2006, 25(7): 1209-1229
- [8] Lattner C, Adve V. LLVM: A Compilation Framework for Life-long Program Analysis & Transformation [C]//Proc. of the 2004 International Symposium on Code Generation and Optimization, Palo Alto, California, USA, 2004; 75-86
- [9] Lee C, Potkonjak M, Mangione-Smith W H. Mediabench: A Tool for Evaluating and Synthesizing Multimedia and Communications Systems [C]//Proc. Int. Symp. Microarch., 1997; 330-337
- [10] Guthaus M R, Ringenberg J S, Ernst D, et al. MiBench: A Free, Commercially Representative Embedded Benchmark Suite [C]//Proc. IEEE 4th Ann. Workshop Workload Characterization, 2001; 3-14

and availability performance parameters [DB/OL]. ITU-T Recommendation Y. 1540, 2005

- [2] Network Performance Objectives for IP-based Services [DB/OL]. ITU-T Recommendation Y. 1541, 2006
- [3] Demichelis C, Chimento P. RFC3393: IP Packet Delay Variation Metric for IP Performance Metrics [EB/OL]. RFC Editor United States, 2002
- [4] Kapoor R, Chen L J, Lao L, et al. Capprobe: A simple and accurate capacity estimation technique [J]. ACM SIGCOMM Computer Communication Review, 2004, 34(4): 67-78
- [5] 刘世栋, 张顺颐, 邱恭安, 等. 一种基于端到端测量的路径性能参数估计算法[J]. 电子与信息学报, 2007, 29(7): 1617-1621
- [6] Ismail M N, Zin A M. Comparing the Accuracy of End-to-End Network Performance Measurement Testbed and Simulation Model for Data Transfers in Heterogeneous Environment [C]//Proceedings of the Second International Conference on Modeling & Simulation, Singapore, 2008; 124-131
- [7] Armolavicius R. Simple Approximations of Delay Distributions and Their Application to Network Modeling [J]. Lecture Notes in Computer Science, 2007, 4516(1): 507-518
- [8] Ramsey C B. A Note on the Normal Power Approximation [J]. ASTIN Bulletin, 1991, 2(1): 147-150
- [9] PlanetLab [EB/OL]. <http://www.Planet-Lab.org/>, [4. 10 20 08]
- [10] Park K S, Pai V S. CoMon: a mostly-scalable monitoring system for PlanetLab [J]. ACM SIGOPS Operating Systems Review, 2006, 40(1): 65-74