基于组播的网络延迟测试*

卢光辉，孙世新，邵子立，张艳

(电子科技大学 计算机科学与工程学院，四川 成都 610054)

E-mail: luguhu_11@263.net

http://www.uestc.edu.cn

摘要：分组延迟极大地影响了网络应用的整体性能，确定延迟的原因和位置至关重要. R. Caceres 等人提出了比其他网络性能测试与分析方法要好的端到端基于组播的网络性能测试与分析法，但这类方法还有一些不足之处. 主要从减少测试分组数和降低计算复杂度两个方面改进了该测试分析法，从而既减轻了带给网络的负载，又使测试分析更加高效、适用. 实验结果表明，这种改进的测试与分析方法更能有效地获取网络延迟等信息.

关键词：组播；分组延迟；复杂度；端到端测试；性能分析

中图法分类号：TP393 **文献标识码**：A

分组延迟对网络的整体性能有极大的影响，确定延迟的原因和位置等有利于设计、控制和管理网络. 网络测试分析可分为内部法和外部法. 内部法直接获取分组延迟、丢失等信息. 外部法通过获取端到端的分组延迟、丢失等信息来确定网络内部的延迟、丢失、拥塞状况等. 内部法有许多不足之处，容易加重网络负载、可扩展性不好、不同的网络产品难于协调一致等. 而外部法不需要网络内部各个结点协调一致，只需要边界端点处的延迟、丢失等信息. 在使用外部法的测试与分析技术中，文献[1,2]通过实验了解端到端的性能现象；文献[3～5]基于单播的方法来测试与分析，扩展性不好，在 N 个结点的的网络上，最坏时带给网络的负载为 $O(N^2)$；虽然 mtrace[6]是一种基于组播的测试工具，最坏时带给网络的负载为 $O(N)$，但它要求所有的路由器对其提出的要求作出响应.

基于以上原因，R. Caceres, Presti F Lo[7,8]等人提出了端到端的基于组播的延迟、丢失分布推导分析法. 它具有可扩展性好、不需网络内部结点上的信息等优点，但有两个不足之处：(1) 未能解决怎样控制从源结点发送的测试分组数的问题；(2) 没有根据问题的具体特点设计出有效地降低时间复杂度的方法.

本文解决了该方法的以上两个不足之处. 实验结果表明，本文提出的改进的测试分析法不仅能准确地控制测试分组数，还使时间复杂度大为降低，从而既减轻了带给网络的负载，又使该测试分析法更加高效、适用.

1 模型及框架

按照文献[7,8]的方法，由物理网络构造出逻辑组播树，使得除根结点和叶结点以外，每个结点至少有两个子结点，测试分组从根结点以组播的方式发送出去. 令 $\tau=(V,L)$ 表示此逻辑树，V 是所有结点的集合，L 是所有有向连接的集合，根结点记为 0，$U=V\backslash\{0\}$ 为中间结点及目的结点的集

* 收稿日期：2000-02-15；修改日期：2000-05-30

作者简介：卢光辉(1971－)，男，四川邻水人，博士生，主要研究领域为计算机网络技术及应用；孙世新(1941－)，男，湖北孝感人，教授，博士生导师，主要研究领域为计算机科学理论，并行算法及其应用；邵子立(1973－)，男，河北石家庄人，博士生，主要研究领域为网络，并行处理；张艳(1973－)，女，河北邯郸人，博士生，主要研究领域为网络，并行处理.

合,结点 j 的所有儿子的集合记为 $d(j)$,其父结点记为 $f(j)$. 如果 j 是 k 的子孙,则记为 $j<k$,"$<$"是 V 上的一个偏序关系. 根结点是源结点,所有叶结点 R 是目的结点. 对于任意 $k\in V$,D_k 是取值在扩展的正实数集 $\mathrm{R}^{+}\cup\{\infty\}$ 上的随机变量,它表示分组从 $f(k)$ 到 k 的时延,∞ 表示分组在此连接上丢失. $D_0=0$,假设 D_k 是相互独立的,记 $Y_k=\sum_{k<j}D_j$,它表示分组从根结点到结点 k 的传输时延. 把 D_k 的取值范围进行离散化,使其在 $\{0,q,2q,\ldots,i_{\max}q,\infty\}$ 上取值,Y_k 在 $\{0,q,2q,\ldots,i_{\max}ql(k),\infty\}$ 上取值,其中 q 为离散尺寸,$i_{\max}$ 为离散数. D_k 的分布记为 α_k,$\mu_k=1-\alpha_k(0)$ 为连接 k 的利用率. Y_k 的分布为 A_k,$A_k(i)=\sum_{j=0}^{i}\alpha_k(j)A_{f(k)}(i-j)$,令 $A_0(0)=1$. $(\tau,k)=(V(k),L(k))$ 表示以第 k 个结点为根的子树,$R(k)=R\cap V(k)$. $\Omega_k(i)$ 表示随机事件 $\{\min_{j\in R(k)}Y_j\leqslant iq\}$,其概率为 $Y_k(i)$. (τ,α) 称为延迟树,如果 $\forall\ k\in U$ 均有 $\alpha_k(0)>0$,则称此延迟树为标准延迟树.

2 基于多播的延迟分布估计及性质

现在考虑从根结点以组播的方式发送 n 个分组. 本文的主要目的是仅由端到端的测试数据 $(Y_{k,l})_{k\in R,l=1,2,\ldots,n}$ 分析出网络内部各连接上的延迟特性.

定理 1[8]. 令 $\Delta=\{\alpha=\alpha_k(i):\alpha_k(0)>0,\sum_{i\leqslant i_{\max}}\alpha_k(i)\leqslant 1,k\in U,i\in\{0,1,\ldots,i_{\max}\}\}$,$\Theta=\{\gamma=\gamma_k(i):\exists\ \alpha\in\Delta|\gamma=\Gamma(\alpha),k\in U,i\in\{0,1,\ldots,i_{\max}\}\}$,则 Γ 是从 Δ 到 Θ 的连续可微且有反函数的双射.

以后用 $\hat{\gamma}$ 来估计 γ,其中 $\hat{\gamma}$ 的表达式见式(8),用 $\hat{\alpha}_k(i)=(\Gamma^{-1}(\hat{\gamma}))_k(i)$ 来估计 $\alpha_k(i)$.

α 的表达式. 当 $i=0$ 时,$A_k(0)$ 由式(1)解出.

$$(1-\gamma_k(0)/A_k(0))=\prod_{d\in d(k)}(1-\gamma_d(0)/A_k(0)),k\in U\backslash R\text{ 时};\quad \gamma_k(0)=A_k(0),k\in R\text{ 时}. \tag{1}$$

$$\beta_k(0)=\gamma_k(0)/A_{f(k)}(0),k\in U. \tag{2}$$

当 $i>0$ 时,$A_k(i)$ 式(3)和式(4)解出.

$$\gamma_k(i)+A_k(0)\Big\{\prod_{d\in d(k)}\Big[1-\Big[\gamma_d(i)-\sum_{j=1}^{i-1}\beta_d(i-j)A_k(j)-\beta_d(0)A_k(i)\Big]/A_k(0)\Big]-1\Big\}+$$

$$\sum_{j=1}^{i-1}A_k(j)\Big\{\prod_{d\in d(k)}[1-\beta_d(i-j)]-1\Big\}+A_k(i)\Big\{\prod_{d\in d(k)}[1-\beta_d(0)]-1\Big\}=0,\ k\in U\backslash R, \tag{3}$$

$$A_k(i)=\gamma_k(i)-\sum_{j=0}^{i-1}A_k(j),\ k\in R, \tag{4}$$

$$\beta_k(i)=\Big\{\gamma_k(i)-\sum_{j=1}^{i}A_{f(k)}(j)\beta_k(i-j)\Big\}/A_{f(k)}(0),\ k\in U. \tag{5}$$

把所有的 $A_k(i)$ 都解出来以后,$\alpha_k(i)$ 即可求出.

$$\alpha_k(i)=\begin{cases}A_k(0)/A_{f(k)}(0), & i=0\\ \Big[A_k(i)-\sum_{j=1}^{i}A_{f(k)}(j)\alpha_k(i-j)\Big]/A_{f(k)}(0), & i=1,2,\ldots,i_{\max}\end{cases}. \tag{6}$$

3 延迟分布的测试与分析

本节主要解决两个方面的问题:(1) 何时停止发送测试分组;(2) 如何根据式(1)~(6)的具体特征,有效地降低时间复杂度.

由于传输时延是网络传输中的确定部分,在考虑网络的性能时没有必要考虑,因此在计算时要

减去这部分.

$$\hat{Y}_{k,l}=Y_{k,l}-\min_{m\in\{1,2,\ldots,n\}}Y_{k,m},k\in R\text{ 时};\quad \hat{Y}_{k,l}=\min_{j\in d(k)}\hat{Y}_{j,l},k\in V\backslash R\text{ 时}. \tag{7}$$

$$\hat{\gamma}_k(i)=n^{-1}\sum_{m=1}^{r}1_{\{\hat{Y}_{k,m}\leqslant iq-q/2\}},\ i=0,1,\ldots,i_{\max}, \tag{8}$$

其中 $i_{\max}=|(\max_{k\in R}\max_{m\in N_k(n)}\hat{Y}_{k,m})/q|$, $N_k(n)=\{m\in\{1,2,\ldots,n\}|Y_{k,m}<\infty\}$.

测试分组数的控制. 现假设发送了 n_1 个测试分组以后,由 γ_1 估计出的延迟分布为 α_1,再发送 m_1 个分组以后,由 γ_2 估计出的延迟分布为 α_2,为了要能准确地测试出延迟分布 α,必须要求 α_1 与 α_2 相差小于某个给定的精度,从定理 1 可知,函数 Γ 及 Γ^{-1} 是连续可微的,它们的 Jacobi 矩阵非奇异,因此,只要 γ_1 与 γ_2 相差小于某个给定的精度就行.

结论. 给定精度 $\sigma>0$,首先根结点发送 n_0 个分组,计算出 γ_0,然后每发送 m 个分组就计算一次对应的 γ 的值并和前一次 γ 的值进行比较,则发送有限个测试分组以后,相邻的两个 γ 的差一定小于给定的精度 σ,并且存在一个正数 ε,当 σ 充分小时 ε 也充分小,使得 α 的前后两个值的差小于 ε.

证明:设发送了 n 个分组以后计算出 γ 的值为 γ_1,再发送 m 个分组后计算出 γ 的值为 γ_2. 对于任意 $\gamma_k(i)$,由式(8)知,可令其前一个值为 l/n,后一个值为 $(l+r)/(n+m)$, $0\leqslant r\leqslant m$. 考察前后两个值的差 $|l/n-(l+r)/(n+m)|=|(ml-nr)/(n(n+m))|\leqslant m/(n+m)$,当 $n\to\infty$ 时,$\gamma_k(i)$ 前后值之差为无穷小,故发送有限个测试分组以后,相邻的两个 γ 的差一定小于给定的精度 σ.

另外,根据定理 1,对于给定的 σ,一定存在一个正数 ε,使得 α 的前后两个值的差小于 ε,并且当 σ 充分小时 ε 也充分小. 结论成立. □

因此,按照该测试分组控制法进行性能测试时,只要预先给定的精度 σ 足够小,最终获得的延迟分布 α 就能满足误差控制的要求.

从而该测试分组控制法既保证了测试的正确性,又减少了时间开销及避免加重网络负载,而文献[7,8]通过求出延迟分布 α 再来估计结果的正确性,计算量太大,且可能会加重网络负载.

快速求解延迟分布 α.

定理 2. $0<c_i<1,i=1,2,\ldots,d,d>1,\sum_i c_i>1,h(x)=\prod_i(1-c_ix)-(1-x)$,则 $h(x)$ 是 $[0,1]$ 上的严格凸函数,$h(x)=0$ 在 $(0,1)$ 上有唯一的根,且关于 c_i 连续可微.

证明:根据 c_i 的条件,$h''(x)>0,0\leqslant x\leqslant 1$,因此 $h(x)$ 是 $[0,1]$ 上严格凸函数. $h(0)=0,h(1)>0,h'(0)<0$,因此,$h(x)=0$ 在 $(0,1)$ 上有唯一的根,显然此根关于 c_i 连续可微. □

由文献[7,8]可知,当 $i>0$ 时,$A_k(i)$ 是 $\#d(k)$ 次多项式的第 2 个最大的根($\#d(k)=2$ 时的简单情形这里不再讨论),若采用 Durand-Kerner 等算法,计算复杂度为 $O(\#d(k)^2)$. 下面的定理 3 把求解 $A_k(i)$ 的复杂度降为 $O(\#d(k))$,且具有二阶收敛速度. 下述引理易于证明.

引理. $f(x)$ 在 $[a,b]$ 上二阶连续可导且 $f''(x)>0,x\in(a,b],f''(a)\geqslant 0,f'(a)<0,f(a)>0,f(x)=0$ 在 (a,b) 上存在实根. 则以 a 为初始点的 Newton 法求出的必然是与 a 相邻的第 1 个根.

定理 3. $f(x)=\gamma_k(i)+A_k(0)\left\{\prod_{d\in d(k)}\left[1-\left[\gamma_d(i)-\sum_{j=1}^{i-1}\beta_d(i-j)A_k(j)-\beta_d(0)x\right]/A_k(0)\right]-1\right\}+$

$$\sum_{j=1}^{i-1}A_k(j)\left\{\prod_{d\in d(k)}[1-\beta_d(i-j)]-1\right\}+x\left\{\prod_{d\in d(k)}[1-\beta_d(0)]-1\right\}=0,$$

其中 $\#d(k)>2$,则由 $f''(x)=0$ 的第 1 个最大根为初始点的 Newton 法求出的解即为 $A_k(i)$,计算复杂度为 $O(\#d(k))$,且具有二阶收敛速度.

证明:令 $x=A_k(i)+yA_k(0)$,则 $f(x)=0$ 化简为

$$g(y)=\sum_{l=1}^{\#d(k)} y^l \sum_{b\in B,\sum b_m=\#d(k)-1} \prod_{m\in\{1,\ldots,\#d(k)\}} (1-\beta_{d_m}(i))^{b_m}\beta_{d_m}(0)^{1-b_m}+$$

$$y\left\{\prod_{d\in d(k)}[1-\beta_d(0)]-1\right\}=0,$$

其中 $b=\{b_1,b_2,\ldots,b_{\#d(k)}\}$,$B=\{0,1\}^{\#d(k)}\backslash\{0\}^{\#d(k)}$. 显然 $g(y)$的二次到$\#d(k)$次项的系数均大于0,一次项系数小于0,常数项为0,且 $g'(0)<0$,$g^{(r)}(y)>0$,$1<r<\#d(k)+1$,$y\geq 0$,因此 $y=0$ 为 $g(y)=0$ 的第2个最大的根,即 $A_k(i)$为 $f(x)=0$ 的第2个最大的根. 根据 $g(y)$的性质可知,$f^{(r)}(x)>0$,$1<r<\#d(k)+1$,$x\geq A_k(i)$,$f(x)$在$[A_k(i),\infty)$上严格上凸,根据引理,只需找到一个 $x_0<A_k(i)$,使得 $f(x)$在$[x_0,\infty)$上满足引理的条件即可证明该定理.

根据$f(x)$的性质,$f''(x)=0$ 的第1个最大的根,也即 $f'(x)$的第1个极小值点为所求的 x_0,而 $f'(x)$的第1个极小值点又可由以1为初始点,具有二阶收敛速度的变尺度法求出. 而 Newton 法也具有二阶收敛速度. 因此,整个计算过程具有二阶收敛速度.

$A_k(i)$的迭代求解主要是计算大量的多项式的值,用 Horner 法计算,其复杂度为 $O(\#d(k))$(直接计算的复杂度为 $O(\#d(k)^2)$),因此求解 $A_k(i)$的复杂度为 $O(\#d(k))$. □

根据定理2、定理3及引理可知,对于任意 $A_k(i)$,其计算复杂度为 $O(\#d(k))$,且具有二阶收敛速度,然后再用式(6)即可求出延迟分布 α,从而整个测试分析过程的计算复杂度为 $O(n\times i_{\max}\times\sum\#d(k))$,而其他方法的时间复杂度为 $O(n\times i_{\max}\times\sum(\#d(k))^2)$.

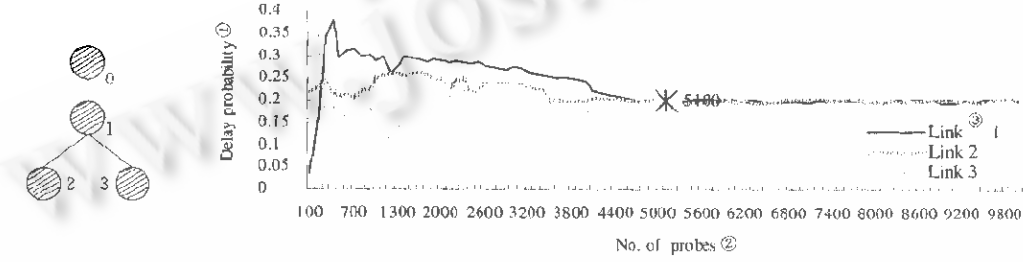
延迟分布测试过程. 整个测试分析过程如下:

```
PROC main(){
    发送 n0 个测试分组;
    发送 m 个测试分组;
    计算前后两组 Ŷ 的值;
    While (前后两组 Ŷ 差的绝对值>σ)
      {
        发送 m 个测试分组;
        计算新的 Ŷ;
      }
    Find_y(1);
    for (i=0,1,...,i_max) infer_delay(1,i);
  }
PROC find_y(k){
    for (j∈d(k)){
    Ŷ_j=find_y(j);
      for (m=1,2,...,n)
        Ŷ_k[m]=min(Ŷ_k[m],Ŷ_j[m]);
    }
    for (i=0,1,...,i_max)
        γ̂_k(i)=n^{-1} Σ_{j=1}^{n} 1_{(Ŷ_k[j]≤iq+q/2)};
    return (Ŷ_k);
PROC infer_delay(k,i){
    if (i==0) Â_k(i)=solveAK1(k);
    else Â_k(i)=solveAK2(k,i);
        β̂_k[i]={γ̂_k[i]-Σ_{j=1}^{i} Â_{f(k)}[j]β̂_k[i-j]}/
                Â_{f(k)}[0]; ‖
        α̂_k[i]={Â_k[i]-Σ_{j=1}^{i} Â_{f(k)}[j]α̂_k[i-j]}/
                Â_{f(k)}[0];
    for (j∈d(k)) infer_delay(j,i);
  }
PROC solveAK1(k){
    X=Ŷ_k[0]/Â_k[0];
    由式(1)以 X0=1为初始值,用 Newton 法计算
    出 X,用 Horner 法计算多项式的值;
    Â_k[0]=Ŷ_k[0]/X;
    return (Â_k[0]);
  }
PROC solveAK2(k,i){
    以 X^(0)=1为初始点,用变尺度法求出定理5中
    f'(X)的第1个极小值点 X0,用 Horner 法计算
    多项式的值;
    由式(4)以 X0为初始点用 Newton 法计算出
    Â_k[i],用 Horner 法计算多项式的值;
    return (Â_k[i]);
  }
```

4 实验结果

文献[7,8]采用采用两种实验方法来验证测试分析法的正确性:一种是模型模拟,用随机过程产生分组的延迟及丢失;另一种是用 NS(network simulator)[9]. 测试分组和网络中的 TCP/UDP (transfer control protocol/user datagram protocol)竞争网络资源,从而引起测试分组的延迟和丢失. 两者都把估计结果与实测结果进行比较. 本文也采用这两种实验方法,为了便于分析对比,实验时不仅采用本文的测试分析法,也采用了文献[7,8]的方法.

模型模拟. 考虑如图1(a)所示的拓扑结构,各路径上的延迟相互独立,取值为$\{0,1,\infty\}$,$n_0=1000$,$m=100$. 实验结果如图1(b)所示,其中 $\hat{\alpha}_k(1)$收敛到 $\alpha_k(1)$,$\alpha_k(0)=0.79$,$\alpha_k(1)=0.2$,$\alpha_k(2)=0.01$.



(a) Network topology (a) 网络拓扑结构

(b) Experimental result (b) 实验结果

Fig. 1
图1

从实验结果可以看出,当测试分组数 $n=5100$时,估计值 $\hat{\alpha}$ 与实际的延迟分布 α 已吻合,即使继续发送测试分组,结果也不变,即 $n=5100$为文中的测试方法的最终发送分组数,从而避免了文献[7,8]中的测试分组的不确定性. 这不仅降低了整个测试时间(采用文中的快速计算方法),而且避免了进一步加重网络负载.

TCP/UDP 模拟. 考虑如图2所示的拓扑结构,内部网络带宽为5Mb/s,传输时延为50ms,边界连接带宽为1Mb/s,传输时延为10ms,采用 FIFO(first in first out)调度策略. 根结点以20KB/s 的速度发送长度为40字节的测试分组. $n_0=2000$,$m=400$,$d=0.1$ms.

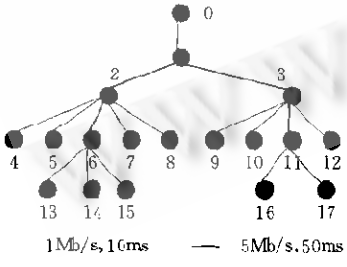
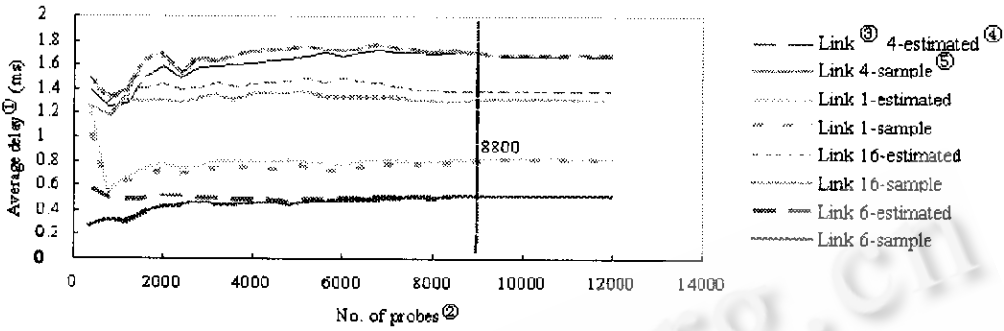


Fig. 2 Network topology
图2 网络拓扑结构

由于延迟分布过于复杂,不便于表示,这里用延迟的平均值来评价文中测试分析法的正确性(仅列出了具有代表性的几条连接). 其主要结果如下:(1) 采用文中的测试分析法的总的时间开销为48秒,而其他方法至少为62秒;(2) 图3表明了当发送了8 800个测试分组时即可停止发送,估计值与实测值基本一致,即使继续发送测试分组,结果也不变,但和文献中实验结果一样,由于连接16的深度较大,因此估计值与实测值存在一些差距.



①平均延迟,②测试分组数,③连接,④估计值,⑤抽样测试值.

Fig. 3 Estimated average delay contrasts to measured average delay

图3 估计平均延迟与和实测平均延迟对比

References:

[1] Paxson, V. End-to-End Internet packet dynamics. IEEE/ACM Transactions on Networking, 1999,7(3):277~292.

[2] Carter, R. L., Crovella, M. E. Measuring bottleneck link speed in packet-switched networks. Performance Evaluation, 1996,27(8):297~318.

[3] Felix: independent monitoring for network survivability project. ftp://ftp.bellcorc.com/pub/mwg/felix/index.html.

[4] IPMA: Internet performance measurement and analysis project. http://www.merit.edu/ipma.

[5] MINC: multicast inference of network characteristics project. http://gaia.cs.umass.edu/minc.

[6] Mtrace: print multicast path from a source to a receiver. ftp://ftp.parc.xerox.com/pub/netresearch/ipmulti.

[7] Cáceres, R., Duffield, N. G., Horowitz, J., *et al*. Multicast-Based inference of network-internal loss characteristics. IEEE Transactions on Information Theory, 1999,45(7):2462~2480.

[8] Lo., P. F., Duffield, N. G., Horowitz, J., *et al*. Multicast-Based inference of network-internal delay distributions. 99-55, CMPSCI, UMass, 1999.

[9] NS: Network simulator. http://www.mash.cs.berkeley.edu/ns.

Multicast-Based Measurement of Network Delay*

LU Guang-hui, SUN Shi-xin, SAO Zi-li, ZHANG Yan

(*College of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu* 610054, *China*)

E-mail: luguhu_11@263.net

http://www.uestc.edu.cn

Abstract: Packet delay greatly influences the overall performance of network application, so it is very important to identify the cause and location of delay. Among the many performance measurement and analysis methods, the multi-cast based inference method advanced by R. caceres etc. is better than others. But it still has some shortcomings. In this paper, a progressed method is advanced, it reduces probes and has lower time complexity, so it not only reduces loading brought to network and is more efficient. Experiments results showed that the method in this paper is more efficient to capture the delay distribution.

Key words: multicast; packet delay; complexity; end-to-end measurement; performance analysis

* Received February 15, 2000; accepted May 30, 2000