面向单向延迟测量的时钟同步技术研究

林容容　丁伟　程光

（东南大学计算机科学与工程系，南京 210096）

（江苏省计算机网络技术重点实验室，南京 210096）

E-mail: rrlin@njnet.edu.cn

摘 要　为测量网络传输中的单向延迟等性能参数以准确提供和分析网络的性能状况，时钟同步的精度要求已非时钟同步协议（NTP）所能完成。本文对近年来出现的各种针对此问题的时钟同步技术研究和实现展开综述性陈述，并在此基础上提出 Altair & Vega（A&V）方法。该方法修正了 Moon 方法中理论推导的欠妥之处，建立两主机之间同步模型，能提供高精度的相对时钟偏移修正。

关键词　网络测量　时钟同步　相对时钟偏移　单向延迟

文章编号　　**文献标识码** A　　**中图分类号** TP393

Research on Clock Synchronization Technique for One-way Delay Measurement

Lin rongrong　　Ding wei　　Cheng guang

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

Abstract: Network Time Protocol (NTP) cannot meet the accuracy requirement of clock synchronization when measuring the metric parameters, which are valuable for the precise analysis and optimize of the network performance, such as one-way delay of the network transmission. Thus, this dissertation summarizes the latest techniques and research in recent years and compares the existing Synchronization Technique, and on the base of which, it proposed a Altair & Vega meathod. This meatod has correct the design weakness of Moon meathod and establish the clock synchronizaion model of two host, which can provide the precious correction for relative clock offset.

Key words: Network Measurement, Clock Synchronization, Relative Clock Offset, One-way Delay

引言

计算机时钟一般是以振荡电路或石英钟为基础，每天的误差达数秒，经过一段时间的累积就会出现较大的误差。不准确的计算机时钟对于网络结构以及其中的应用程序的安全性会产生较大的影响，尤其是那些对时钟是否同步比较敏感的网络指令和应用程序。在大型网络中，我们使用网络时间协议（NTP）[1]，但随着网络技术的不断发展，许多度量参数——如单向延迟[2]的测量需要毫秒级的同步精度。因而必须对时钟的偏差进行进一步修正。

本论文将重点阐述在单向延迟等高精度测量中，时钟同步问题的解决方案及其比较分析，并进一步提出未来的研究方向。

1 网络时钟概念定义

网络时钟同步算法涉及到各种基本概念，现将时钟同步算法中涉及的各种概念定义概括如下：

1) 真实时钟（true clock）：在描叙时钟 C 这一术语时，“真实时间”的概念由系统时钟所报告的时间来表示。真实时钟 C_t 定义为：$C_t(t)=t$ 且 $P_t=\phi$。

2) 偏移（offset）：指系统报告的时间与真实时间的差。设：C_a、C_b 为两时钟，C_a 的偏移为 $C_a(t)-t$。$t\geq 0$ 时，C_a 相对于 C_b 的偏移为 $C_a(t)-C_b(t)$。

3) 偏差（skew）：指时钟的频率与真实时钟频率之差。在 t 时刻，时钟 C_a 相对于 C_b 的偏差为 $C'_a(t)-C'_b(t)$。

4) 时钟同步（synchronized）：对于两时钟而言，由于偏移和偏差都不是真实时钟的，故以相对偏移和相对偏差为标准来讨论。在某一时刻时钟同步是指，时钟间相对的偏移和偏差都为 0。

2 现有网络时钟同步技术

时钟同步问题由来已久。可以说自网络出现后，随即便有了同步网的概念。随着网络技术的发展及虚拟主机和局域网的出现，主机之间的时钟同步精度的要求也越来越高。网络时间协议（NTP）的出现能很好地解决这一问题。但面对单向延迟等端到端测量的精度需求，NTP 技术也不能适用，因而又产生了一系列针对更高精度时钟同步技术的研究与实现。

2.1 NTP 技术

网络时间协议 NTP（Network Time Protocol）是由美国德拉瓦大学的 David L. Mills 教授于 1985 年提出，除了可以估算封装在网络上的往返延迟外，还可估算计算机时钟偏差，从而实现在网络上的主机校时。

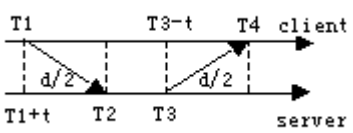

图 1　NTP 同步体系原理图

NTP 使用国际标准时间 UTC 提供准确的时间来源进行同步。其基本方法为：通过报文传送，首先由客户端发起同步请求，在请求报文中记录下当前机器的时间戳 T1。当服务器端接收到此报文时，立即记录下接收时的时间戳 T2。服务器端接收到报文后，向客户端发送一个应答报文，并记录应答时间戳 T3。客户端收到此应答报文，也立即记录该时间戳 T4。

基金项目：国家 973 基础科学研究发展计划项目（编号：2003CB314803）资助。

作者简介：林容容（1981－），硕士，研究方向：网络测量，网络行为学。丁伟（1962－），教授，研究方向：网络管理，网络测量，网络行为学。程光（1973－），博士，讲师，研究方向：网络测量，网络行为学。

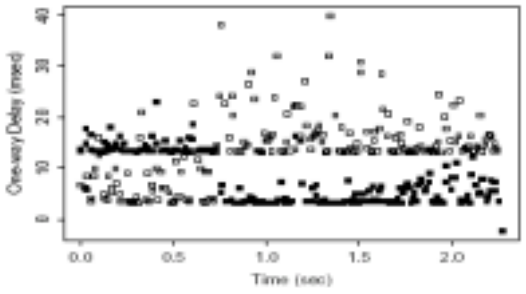

图 2 往返两方向上的单向延迟测量结果

图 1 说明了这四个参数的意义及它们之间的关系。设：t 为服务器和客户端之间的时钟偏移；d 为两者之间的往返时间。因为 T2=T1+t+d/2; T2-T1=t+d/2；T4=T3-t+d/2；T3-T4=t-d/2；所以 d=(T4-T1)-(T3-T2) ； 故 最 终 得 到 时 钟 偏 移 t=((T2-T1)+(T3-T4))/2。

NTP 提供的时间校正的精确度在 LAN 上约小于 1 毫秒，在 WAN 上几十毫秒左右。但随着技术的不断发展，这种精度的同步已逐渐不能满足网络测量研究的需求。

2.2 线性回归算法和分段最小算法

实验发现单向延迟的测量值基本是呈线性模型的,所以在单向延迟的估计计算以修正其中时钟的偏移量时，首先想到的必然是线性回归方法。但遗憾的是，单向延迟的测量中常常会遇到偏离拟和直线较远的点，因此线性回归算法不能很好的修正时钟偏差。

分段最小算法考虑到：要修正系统时钟的偏移量（如图 2 所示），必须尽量排除网络其他状况（如拥塞，排队等）对时钟偏移测量的影响，所以需考虑的理应是那些低于拟和所得直线的点，即：端系统及路由并不繁忙时，时戳报文很快得到响应和传递的情况下得到的单向延迟。

算法首先将单向延迟的测量结果分段，从每一段中选出其最小值。然后将这些最小值合并，拟和出一条直线段，即：以“降噪”的 OTTs 值拟和出的直线段为基础计算时钟偏差的估计值。同样，这种方法也只是一种粗略的估计和计算，同步精度不高。

2.3 Paxson’s 方法

Paxson 的研究[3]是基于 TCP 报文传输的静态数据，对往返两个方向上的 OTT（one-way transit time）进行的研究。由于在 TCP 连接中，SIN，FIN 等报文由于没有携带任何数据，故必须和携带数据的报文区分开。同时，单向延迟会受到报文所经路径上的路由器拥塞程度等因素的影响。Paxson 同步算法的具体步骤可简略概括为：

a) 将测得的两个方向上的两组 OTT 延迟时间都分为 $\sqrt{N}$ 断,从每段中选出最小的延迟，即“降噪”(de-noised) 的 OTTs 值；

b) 从“降噪”OTTs 值拟和得到的直线段中找出其斜率的中值。如果斜率中值为负，假设 OTTs 的变化趋势为逐渐减少的（这里我们假设变化趋势为逐渐减少）；

c) 检测所得到的“降噪”的 OTTs 最小值的量是否足以显示第 2 步中提及的逐渐减少的趋势；

d) 若通过了检测，则将那些最小值中的中值取出，即为时钟偏差的斜率的估计值。否则，算法将认为无时钟偏差。

2.4 Moon 方法

与 Paxson 方法所不同的是：Paxson 方法基于往返路径的延迟测量，而 Moon 方法[4]的测量基于单向的延迟测量体系，并提出了新的单向延迟直线处理思路。

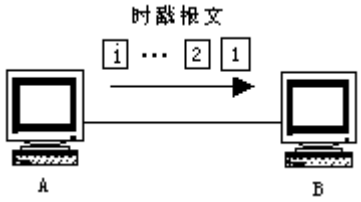

图 3 Moon 方法测量体系

图 3 所示为该方法的测量体系。图中 A 主机为服务器端（即发送方），B 主机为客户端（即接收方）。Moon 方法以接收方 B 的时钟为真实时钟进行同步。通过这一测量体系，可以得到的量有：从第 1 个报文离开 A 主机到第 i 个报文离开 A 主机的时钟间隔 $\tilde{t}_i^{\,s}$ ；从第 1 个报文到达 B 主机到第 i 个报文到达 B 主机的时钟间隔 $\tilde{t}_i^{\,r}$。由 $\tilde{t}_i^{\,s}$ 和 $\tilde{t}_i^{\,r}$ 可计算得到延迟值 $\tilde{d}_i$。

设 A 主机与 B 主机间的实际的端至端单向延迟值表示为 $\bar{d}_i$。Moon 方法研究发现： $\bar{d}_i = \tilde{d}_i - (\alpha - 1)\tilde{t}_i^{\,s} + \bar{d}_1$。令：$\hat{\alpha}$ 和 $\hat{\beta}$ 分别为 α 和 $\bar{d}_1$ 的估计值，则延迟值中所需要去除的偏差 $\tilde{d}_i$ 为 $\hat{d}_i = \tilde{d}_i - (\hat{\alpha} - 1)\tilde{t}_i^{\,s} + \hat{\beta}$。

Moon 方法使用一种新的线性算法逼近测量得到的延迟值，以估算 α 的值，从而消除其中的延迟偏差。即：满足如下两个约束条件的要求的直线：

$$\tilde{d}_i - (\hat{\alpha} - 1)\tilde{t}_i^{\,s} + \hat{\beta} \geq 0, \qquad 1 \leq i \leq N \qquad \text{式 (1)}$$

$$\min\left\{\sum_{i=1}^{N}(\tilde{d}_i - (\hat{\alpha} - 1)\tilde{t}_i^{\,s} + \hat{\beta})\right\} \qquad \text{式 (2)}$$

2.5 Li Zhang,方法

Li Zhang 方法之前的种种时钟修正方法都是建立在系统中没有其他时钟校正机制干扰的假设下的。Li Zhang 时钟同步方法[5][6]提供了在具有其它时钟校正情况下的基于凸包计算的时钟偏移修正算法。

Li Zhang 方法定义了三个目标函数：(1) 使测量得到的点在 y 轴方向上与所得直线的距离和最小（亦即 Paxon 的目标函数）；(2) 使测量得到的点连成的曲线与所得直线形成的区域面积最小；(3) 使落在所得直线上的测量得到的点的数目最大。

目标函数的意义在于描述欲求的时钟偏差直线是如何逼近测量所得的各延迟值的。该直线的求解通过凸包方法实现。所谓凸包，如式 3 的定义可知，N 个点的凸包其实是一个能够将所有的点包围的分段的直线段。

$$co(\Omega) := \left\{ x \mid x = \sum_i \lambda_i v_i, \lambda_i \geq 0, \sum_i \lambda_i = 1, v_i \in \Omega \right\} \qquad \text{式 (3)}$$

当系统中可能存在瞬时的时钟校正时，Li Zhang 方法采用启发式算法思路将所测得的延迟值在时间轴上分为若干区间。比对各个区间内的直线，将相似的合并为一个区间，从而找出发生时钟校正的时刻。在最终以时钟校正时刻定位的各个区间内使用前述算法即可进行时钟同步工作。

当系统时钟存在速率变化时，Li Zhang 方法在每组单向延迟测量值中取出一个点，计算引入此点后的斜率是否使得原先的斜率有较大的差异。若有，则该点是一个转折点；否则不是。判断出转折点后，即可在以转折点为界的各个直线段中利用凸包计算算法为每一个单向延迟值更新时钟偏差。

3 Altair & Vega（A&V）方法

综合以上方法优缺点，本文提出 Altair & Vega 同步方法（简称，A&V 方法）。该方法在往返两个方向的报文序列里找出各自所网络状况影响最小的测量值，以此为依据，建立两主机时钟同步的模型。

在图 3 所示测量体系中。图中 A 主机为服务器端（即发送方），B 主机为客户端（即接收方）。该方法以接收方主机 B 的时钟为真实时钟对发送方主机 A 的时钟进行同步。其中设：

C_s：发送端时钟；

C_r：接收端时钟；

N：时戳报文包数；

l_i ：以 C_r 为标准的第 i 个时戳报文离开发送端的时间，i = 1，2，…，N；

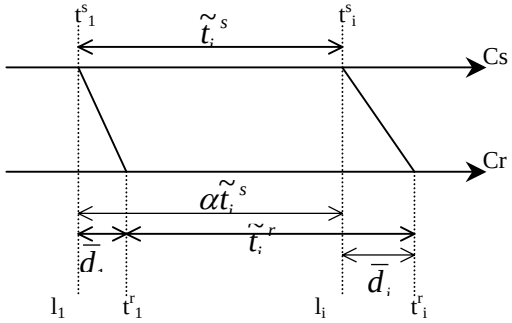

图 4 时钟示意图

t_i^s：以 C_s 为标准的第 i 个时戳报文离开接收端的时间，i = 1，2，…，N，$t_i^s = C_s(l_i)$；

t_i^r：以 C_r 为标准的第 i 个时戳报文到达接收端的时间，i = 1，2，…，N；

d_i：根据第 i 个时戳报文计算得到的端到端单向延迟值，i = 1，2，…，N。在时钟实现同步的情况下，d_i 可由下式计算得到：

$$d_i = t_i^r - t_i^s \qquad \text{式（4）}$$

通过图 3 所示的测量体系，可以得到：从第 1 个报文离开 A 主机到第 i 个报文离开 A 主机的时钟间隔 $\tilde{t}_i^s$ ；从第 1 个报文到达 B 主机到第 i 个报文到达 B 主机的时钟间隔 $\tilde{t}_i^r$ 。Moon 方法的不合理之处在于其用时钟间隔 $\tilde{t}_i^s$ 和 $\tilde{t}_i^r$ 计算得到延迟值 $\tilde{d}_i$ ，并以此为依据实现时钟的同步。而实际测量中的单向延迟 $\tilde{d}_i$ 应由式（4）得到。由时钟间隔推导的模型显得失之偏颇。

由此，本文沿用 Moon 方法的符号体系进行如下推导。

记 $\Delta(l_1, l_i, C_r)$ 为以 C_r 为标准时钟，l_1 到 l_i 内的时间值，则设：

$$\Delta(l_1, l_i, C_r) = l_i - l_1 = \alpha\Delta(l_1, l_i, C_s) = \alpha t_i^s \qquad \text{式（5）}$$

测量得到的单向延迟值为：

$$\tilde{d}_i = t_i^r - t_i^s \qquad \text{式（6）}$$

而真实的主机间的单向延迟值为：

$$\bar{d}_i = \Delta(l_i, t_i^r, C_r) = t_i^r - l_i \qquad \text{式（7）}$$

由于测量的延迟中包含了真实的延迟和相对时钟偏移两部分的值，故可以得到相对时钟偏移 offset 为：

$$offset = \tilde{d}_i - \bar{d}_i \qquad \text{式（8）}$$

代入式（6）、式（7）得：

$$offset = t_i^r - t_i^s - (t_i^r - l_i)$$

代入式（5），有 $t_i^s = \tilde{t}_i^s - t_1^s$ 并化简得：

$$offset = \alpha\tilde{t}_i^s + l_1 - (\tilde{t}_i^s + t_1^s)$$

又 $\bar{d}_1 = l_1 - t_1^r$，故

$$offset = (\alpha - 1)\tilde{t}_i^s + (t_1^r - \bar{d}_1) - t_1^s$$

由图 4 易知：$t_1^s = t_1^r - \tilde{d}_1$，最终得到：

$$offset = (\alpha - 1)\tilde{t}_i^s + \tilde{d}_1 - \bar{d}_1 \qquad \text{式（9）}$$

故：可以用线性模型测量主机间的相对时钟偏移。

通过单方向上的时戳报文传输可以很快得到同步模型的一次项系数 $(\alpha - 1)$，记为 s。但常数项很难估计。

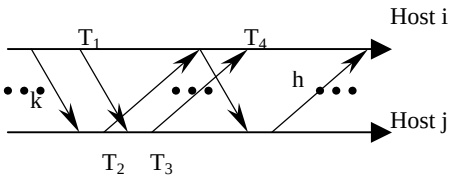

图 5 A&V 方法报文序列

考虑两主机间的单向延迟 OWD＝$T_{propagation}$+$T_{transmit}$+T_{queue}。其中，$T_{propagation}$ 为光速传输延迟，其大小由物理传输路径长度除以传输速度得到。对于同链路传输的报文来说，$T_{propagation}$ 是一样的。$T_{transmit}$ 是发送数据所花费的时间，由数据分组大小除以带宽得到，因而在一组同样大小的报文传输中也是近似相等的。T_{queue} 是网络内部的排队延迟。在一组大小相同，路径相同的时戳报文的传输过程中，它是影响报文单向延迟变化的主要原因。

在图 5 所示的报文序列中，记：其中任一往返传输中，由 i 主机到 j 主机的单向延迟测量值（包含有主机间相对时钟偏移量）$\triangle T_{ij}' = T_2 - T_1$；由 i 主机到 j 主机的单向延迟测量值 $\triangle T_{ji}' = T_4 - T_3$；$\triangle T_{ij}$ 为由 i 主机到 j 主机的单向延迟真实值；$\triangle T_{ji}$ 为由 j 主机到 i 主机的单向延迟真实值。设：在 i 主机到 j 主机的报文发送序列中，在第 t_k 时刻测量到其中“最小的”单向延迟测量值，该时刻 i 主机与 j 主机间的相对时钟偏移为 f_k ;在 j 主机到 i 主机的应答报文发送序列中，在第 t_h 时刻测量到其中“最小的”单向延迟测量值，该时刻 j 主机与 i 主机间的相对时钟偏移为 f_h。则可以得到下式：

$\triangle T_{ij}' = \triangle T_{ij} - f_k$

$\triangle T_{ji}' = \triangle T_{ji} + f_h$

$f_k - f_h = s * (t_k - t_h)$

在链路相同，报文大小相同，T_{queue} 的影响最小的情况下，$\triangle T_{ij}$ 和 $\triangle T_{ji}$ 的值应接近相等。构造函数：

$F(f) = (\triangle T_{ij} - \triangle T_{ji})^2$

$= [(\triangle T_{ij}' + f_k) - (\triangle T_{ji}' - f_h)]^2$

$= (\triangle T_{ij}' - \triangle T_{ji}' + f_k - f_h)^2$

$= [\triangle T_{ij}' - \triangle T_{ji}' + s * (t_k - t_h) + 2f_h]^2$

令 $F'(f) = 0$，得：

$f_h = [\triangle T_{ji}' - \triangle T_{ij}' - s * (t_k - t_h)] / 2$ 式（10）

f_h 即为同步模型的常数项。

图 6 所示为采用该 A&V 方法得到的单向延迟测量值与时钟同步模型图。在该图中可以明显看出单向延迟的测量值呈线性模型，这是由于该测量值中包含了两主机相对时钟偏移的缘故。经修正后的单向延迟值如图 7 所示，

为评价该修正后的单向延迟值的准确性，将修正后的两个方向上的单向延迟相加与实际的 RTT 值进行了比较。结果平均误差只有 0.02ms，最大误差不超过 0.107ms。

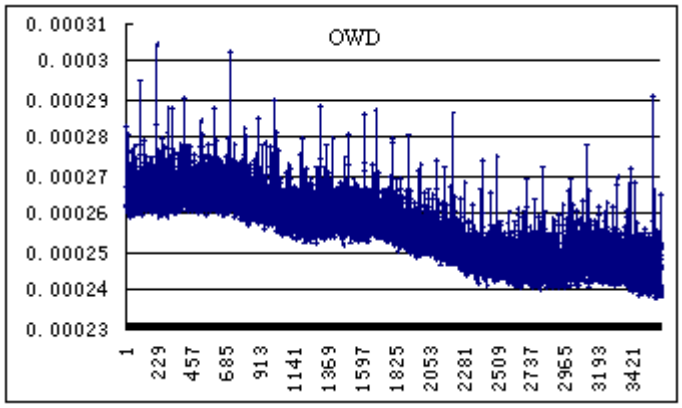

图 7 修正后的单向延迟值

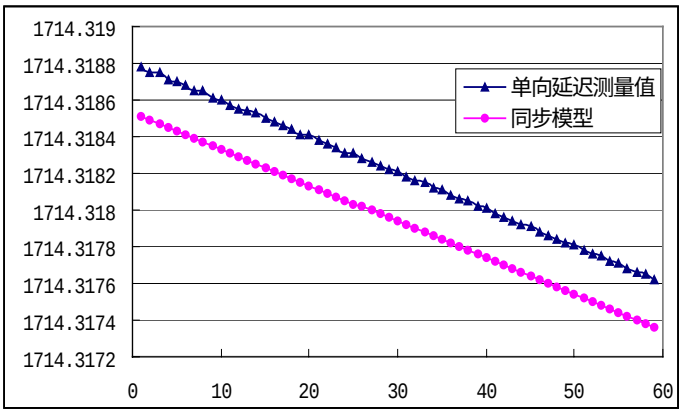

图 6 单向延迟测量值与同步模型

4 各种方法性能比较

各种方法比较看来，当同步过程中没有外界对时钟的强制调整的情况下，线性回归方法的方法和分段最小方法则较易受到网络拥塞状况的影响，并非很好的时钟同步方法。Paxson 方法的测量中，误差会随着时钟偏移的变大而变大，不够稳定。Moon 方法和 Li Zhang 方法在计算复杂度、稳定性及测量精度方面基本一样，只是 Li Zhang 方法显得更加直观一些。然而 Li Zhang 方法的凸包集合点在实际计算中十分不均匀，给同步测量带来很大误差。

另外，Li Zhang 方法能够测量到系统中的强制的时钟调整点，并进行时钟偏差测量和同步，即便在时钟偏差的速率改变时也能以高精度完成时钟同步工作。

A&V 方法对 Moon 方法的理论模型推导做了修正，并在此基础上建立了新的同步模型，其同步精度较高，但仍处于两主机的实验阶段。

5 总结和未来工作

从上述各种时钟同步方法的介绍中可以看出，各种方法都是建立在对含有时钟偏差的单向延迟的测量基础上对其中的时钟偏差进行计算分析从而实现网络时钟的同步。各种方法的不同首先是由于同步方法所采用的逼近延迟直线的方式及其处理方法的不同而产生同步误差的差异。未来的同步研究仍必须致力于更准确的延迟直线的逼近方式的选择和研究。

另一方面，Li Zhang 方法虽然实现了在不仅有时钟的强制调整，而且时钟偏差速率可能发生变化的系统（如由 NTP 同步的系统）中的时钟同步任务，但在实际情况中，如果网络负载极其严重，时钟信号可能发生延迟响应时，系统的时钟偏差速率变化不稳定，在这种情况下如何解决时钟同步问题也是未来研究的方向之一。

最后，当前对于时钟同步的各种研究基本针对单向延迟的测量展开，仍处于两主机的同步研究阶段。虽然 NTP 工具可以很好的提供多台主机间的时钟同步，但其精度太低，不适用于单向延迟等协同数据的测量。高精度的多系统时钟同步将是极具挑战性的工作。

参考文献

[1] Mills, D., “Network Time Protocol (Version 3) Specification, Implementation and Analysis”, RFC 1305, March 1992.

[2] J. Mahdavi, M. Mathis, “Framework for IP Performance Metrics”, RFC2330, May 1998.

[3] Vern Paxson, “On calibrating measurements of packet transit times,” in Proceedings of SIGMETRICS’98, Madison, Wisconsin, June 1998.

[4] MOON, S.B., SKELLEY, P. AND TOWSLEY, D., Estimation and Removal of Clock Skew from Network Delay Measurements. In Proceedings of the IEEE INFOCOM Conference on Computer Communications, page 227-234, March 1999.

[5] Li Zhang, Zhen Liu and Cathy Honghui Xia, Clock Synchronization Algorithms for Network Measurements. In IEEE, 2002.

[6] Kostas G. Anagnostakis, Michael Greenwald, Raphael S. Ryger, cing: Measuring Network-Internal Delays using only Existing Infrastructure. In IEEE INFOCOM, 2003.