

一个消除单向时延测量中时钟频差和时钟重置的新方法

王洪波,林宇,金跃辉,程时端
(北京邮电大学网络与交换国家重点实验室,北京 100876)

摘 要: 单向时延测量对于分析和评价网络端到端性能具有重要的意义. 主机之间的相对时钟频差和时钟重置会给单向时延测量引入不容忽视的误差. 本文提出了一个基于模糊聚类分析的算法来检测并消除这些误差. 大量实验表明:与同类算法相比,该算法有更好的准确性和鲁棒性. 而且此算法时间复杂度为 $O(N)$.

关键词: 网络测量; 单向时延; 相对时钟频差; 时钟重置; 模糊聚类分析

中图分类号: TN393.1 **文献标识码:** A **文章编号:** 0372-2112 (2005) 04-0584-06

A New Approach for Removing Clock Skew and Resets from One-Way Delay Measurement

WANG Hong-bo, LIN Yu, JIN Yue-hui, CHENG Shi-duan

(State Key Laboratory of Networking and Switching, Beijing University of Posts & Telecommunications, Beijing 100876, China)

Abstract: Packet one-way delay traces are important measurements for analyzing and evaluating end-to-end performance of the network. These measurements can be quite inaccurate due to the relative clock skew, i. e., the frequency difference between two clocks, and clock reset which is the abrupt adjustment of clock at end systems. In this paper we introduce a new algorithm to estimate and remove the relative clock skew and clock resets from measurements based on the fuzzy clustering analysis. Numerical experiments demonstrate that this algorithm is more accurate and robust than existing algorithms. Additionally, the new algorithm has time complexity of $O(N)$.

Key words: network measurement; one-way delay; relative clock skew; clock reset; fuzzy clustering analysis

1 引言

近年来,网络测量已成为研究互联网特性、评价网络性能的重要手段和方法. 而端到端单向时延是反映网络路径特性基本而重要的测量指标. 为此, IETF 的 IPPM (IP Performance Metrics) 工作组已制定了相应的 RFC^[1,2].

数据包从一主机经网络到另一主机所经历的时间为单向时延. 其测量的基本方法是发端主机发送数据包时在数据包上记录发送时间,收端主机收到数据包时在数据包上记录到达时间,并利用这两个时间来计算单向时延及其相关值. 称到达时间与发送时间之差为测得时延,如果收发两端主机的时钟完全同步,则测得时延即为真实时延. 然而,在实际测量中,两主机时钟的运行速率通常是不同的,这种速率上的不同叫两主机的相对时钟频差 (relative clock skew), 本文简称时钟频差; 另外,在测量时,主机时钟有时会发生瞬时调整,称之为时

钟重置 (clock reset). 这些情况都会使测得时延存在误差. 本文要解决的问题就是从测得时延中去除这些由非精确时钟可能引入的误差. 这个问题实质上是关于时间的测量问题,正如 IPFM 框架 RFC 2330^[3]指出:“时间测量处于许多测量指标的核心地位”,对于单向时延尤为如此,因而此问题是单向时延测量的关键问题.

当然,理想方法是同步所有测量主机的时钟. 数据网中常采用 NTP^[4]进行时钟同步,但由于其精确性不稳定且影响时钟速率等缺点^[3]而不建议在网络测量中使用. 以全球定位系统 (GPS) 提供精准外部时钟源是一种技术上可行的方案,但由于其较大的经济和空间代价,不宜在大规模、可扩展的测量系统中采用^[5]. 同样,文献[6]中的精确时钟方案由于对软件的特殊需求也存在可扩展性缺陷而不能实用. 因此在当前技术条件下,单向时延的测量往往在测量主机时钟不同步的情况下进行.

收稿日期:2004-07-20;修回日期:2005-01-20

基金项目:国家自然科学基金(No. 90204003);国家 973 计划(No. 2003CB314806);国家 863 项目(No. 2002AA103063;2003AA121220);教育部博士点基金(No. 20010013003)

在事先不知时钟频差、时钟重置的情况下,我们仍然可以从测量数据包含的信息中估计出时钟频差和时钟重置。记测量数据的集合为 $V = \{v_i = (t_i, d_i) : i = 1, 2, \dots, N\}$, 其中 t_i 是第 i 个数据包离开发送主机时根据发送主机时钟记录的时间, d_i 是第 i 个数据包的测得时延。如图 1 所示, 我们把测量数据绘制在二维坐标系中, 其中 x 轴是 t_i , y 轴是 d_i 。从图中可发现时间段 $[15000s, 17000s]$ 内的所有点都处于一条斜率非零的直线之上, 在其它时间段也有相同的现象。这是由时钟频差导致时延随时间线性

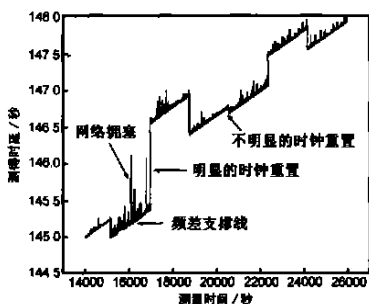


图 1 单向时延 (互联网测量实验数据)

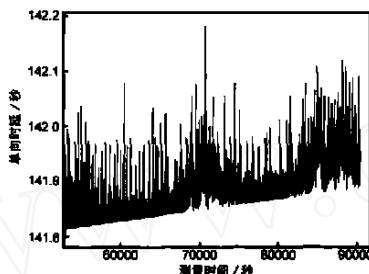


图 2 具有长时间拥塞的情况

增长。因此估计出了这种直线的斜率也就得出了时钟频差, 此直线称为频差支撑线。时钟重置的发生会使估计时钟频差更为困难, 它反映在图 1 中就是如 17000s、20500s 等时刻处时延的突然上升或下降, 这种时刻称为时钟重置点。准确地检测时钟重置并非易事, 因为网络的暂时拥塞也会使得时延发生突增 (见图 1), 检测时钟重置时必须把这种情况和时钟重置区分开。

更具挑战性的是在拥塞持续一段时间情况下, 时延值会出现先上升后下降的现象, 如图 2 所示, 检测时钟重置时也必须区分出时钟重置和拥塞引起的时延上升或下降。

本文首次使用模糊数学来处理时钟重置的检测问题, 有效提高了检测并消除时钟频差和重置的准确性和鲁棒性, 为解决同类问题提供了新的思路和方法。后续部分安排如下: 第二节介绍相关工作和我们的新思想; 第三节描述新算法; 第四节通过网络测量实验数据进行算法验证、比较及分析; 第五节总结全文。

2 相关工作和我们的创新

针对时钟频差问题相关文献 [7~10] 已经提出了一些方法。特别地, 在文 [10] 中, Zhang 等人总结了已有算法 [7~9] 优缺点, 并基于凸分析 [13] 给出了凸包方法 (convex-hull approach), 此方法在测量路径上有临时拥塞时工作得也很好。

对于只存在时钟频差的情况下, 文献 [9] 与 [10] 提出的算法都能较好估计出时钟频差。但是, 文献 [7, 8, 10, 11] 均指出时钟重置的同时存在性。因此在做误差分析时, 必须同时考虑时钟频差和时钟重置的可能性, 否则, 会得出错误的结果。

检测时钟重置的方法可分成两类, 一种是利用测量路径前向和后向两个方向上的单向时延数据来确定时钟重置点,

另一种则只利用一个方向上的单向时延数据。文献 [8, 11] 的算法和文献 [10] 中的 Marching 算法属于前者, 由于许多环境或应用只能得到一个方向上的单向时延数据, 因此这类方法有很大的局限性。文献 [10] 的分而治之 (Divide And Conquer) 算法 (下文记为 DAC 算法)、文献 [12] 及本文中的算法属于后一类方法。DAC 算法优点是时间复杂度低 ($O(N)$)。然而, 当应用于我们实际网络测量获得的实验数据时, 我们发现此算法还存在不足: 一方面, 对于一些实际发生的时钟重置, 此算法往往检测不出来, 我们称之为“漏检”; 另一方面, 在网络高负载情况下, 此算法往往把由于网络拥塞造成的相对较大的时延抖动误认为是时钟重置, 我们称之为“误检”。在本文第四节中, 我们会详细讨论此问题。最近, 文献 [12] 提出了基于时间序列分段技术的算法, 其优点是能在线实时检测时钟重置, 缺点是其中的分段数量的估计算法缺乏准确性。上述文献中的算法都试图使用确定性的数学方法和公式来计算某种确定的数值, 以此来判断时钟重置 (调整) 的发生, 由于各种噪声和非确定性的影响, 这些算法共同的缺点是在判断时钟重置时会过于“绝对”, 从而造成“漏检”和“误检”。

若用肉眼观察图 1, 立刻就能判断出所有的时钟重置, 以 16000 秒到 18000 秒间的数据为例, 在时刻 17000 秒前后的数据“绝大部分”可“明显地分为两类”, 正是基于这个特性的把握我们才确信在 17000 秒发生了时钟重置。而在 20500 秒处的时钟重置幅度虽较小, 但仍有这样的本质特征。以上描述中“绝大部分”、“明显地分为两类”均是模糊概念, 但我们人类正是运用这些模糊概念进行判断从而做出准确的结论。对于这种模糊问题, 我们认为使用模糊数学 [14, 15] 来解决是合适的。

基于以上新思路, 本文使用模糊聚类分析 [15, 16] 并结合文献 [10] 来估计和消除时钟频差和重置。下文称此方法为基于模糊数学算法 (fuzzy-based algorithm), 简称 FBA。

3 基于模糊数学的时钟重置检测和消除算法

在实际中, 网络拥塞是经常出现并且是时变的, 反映到单向时延上就是拥塞时时延大于无拥塞时单向时延, 如图 1 所示。这些拥塞时的时延数据对于分析时钟频差和重置来说是一种噪声。但是, 除网络路径重要改变情况外, 噪声总表现为单向时延的增长 [8]。这意味着, 在有时钟频差的情况下, 可使用最小滤波来去除噪声。文 [8] 就是采用此方法去噪声的。然而, 在实际中时钟频差是普遍存在的, 测得时延会随时间的推移而线性增长或下降, 如果直接采用最小滤波会引入误差。因此我们并没有采用文 [8] 直接进行滤波的方法, 而是先去除了时钟频差的影响, 再进行滤波。

我们的算法用四个步骤来消除时钟频差和时钟重置: 首先, 分析原始测得时延数据, 得到其频差估计值, 并以该值对原始测得时延数据进行初步纠斜处理。第二步, 采用最小滤波对上一部纠斜后的数据进行去噪声。第三步, 在前步得到的去噪数据上, 使用基于模糊数学的方法检测出所有时钟重置点。最后, 找出全局最优时钟频差值, 并纠正原始测得时延数据。

3.1 频差估计和初步纠斜

主机时钟的速率在测量时间内变化极小可认为固定不

变,因而本文假设两端测量主机时钟频差为常数,这与其它文献[7~10]是一致的.又由于在通常情况下,主机时钟不会频繁的发生时钟重置,我们可以假设存在时间长度 w ,在此时间长度内最多有一次时钟重置.

令测量数据的测量总时间为 T (即 $t_N - t_1$),从第一个测量点开始每 $w/2$ 测量时间段划分为一个测量子集,若最后一段数据的测量时间不足 $w/2$,则不记为测量子集.假设共划分为 n 个子集,对前 m ($m = n$ 或 $m = n - 1$, m 为偶数)个子集分别求频差值(用文[9]的方法或文[10]中的凸包方法),得到 m 个频差值的集合 $A = \{v_1, v_2, \dots, v_m\}$;由于在任意时间段 w 内最多发生一次时钟重置,因此 v_{2x-1}, v_{2x} ($x = 1, 2, \dots, m/2$)二者之中最多有一个值因时钟重置影响而与真实时钟频差值 偏差较大,从而在集合 A 中一定有超过 50% 的元素的值比较接近且分布在 值附近;因此我们可以以集合 A 的所有元素为对象进行模糊聚类分析以得到元素个数超过 $m/2$ 、元素值最为接近的集合 A ,以 A 的元素来计算频差估计值 e :

$$e = \frac{1}{|A|} \sum_{i \in A} v_i, A \subset A \quad (1)$$

假设 $\{B_i | i \in I\}$ (I 为指标集)是集合 A 的一个分类,即

$$B_i \subset A, \bigcup_i B_i = A, \text{ 并且 } \forall t, s \in I, s \neq t, B_i \cap B_s = \emptyset \quad (2)$$

记分类结果中对象最多的类为 B_{\max} ,即 $(\forall i \in I) |B_{\max}| \geq |B_i|$.

为了求得 A ,我们采用下面的算法.

(1) 建立模糊相似矩阵 $R_1 = (r_{ij})_{m \times m}$,

$$r_{ij} = \begin{cases} 1, & i = j \\ 0, & i \times j < 0 \\ \frac{\min(|v_i - v_j|)}{\max(|v_i|, |v_j|)}, & i \times j > 0 \end{cases} \quad (3)$$

其中 r_{ij} 是两个不同测量子集中频差值的相似系数.

(2) 求 R_1 的传递闭包得到模糊等价矩阵 R_1^* .记 R_1^* 的所有元素组成的集合为: $E = \{r_{ij}^* | r_{ij}^* = 1, i, j \in m\}$.假设此集合中的元素排序如下: $r_1 > r_2 > \dots > r_{|E|}$.

(3) for $i = 1$ to $|E|$

由 R_1^* 计算 R_1^* 的 截矩阵 R_1^* ;

求 R_1^* 对应的分类 $\{B_i | i \in I\}$;

如果 $|B_{\max}| > |A|/2$, 则 $A = B_{\max}$; 计算 e ; 结束;

得到频差估计值 e 后,再对测得单向时延数据做初步校正得到纠斜数据集:

$$v_i = \{v_i = (t_i, d_i), i = 1, 2, \dots, N\}, d_i = d_i - e(t_i - t_1) \quad (4)$$

图 3 是对一个单向时延测量数据初步纠斜后的结果(部分时间段内结果).

3.2 去噪声

在上一步得到的纠斜数据集上,进行去噪声处理.主要思想是把数据集 分成若干子集段并通过保留每子集中单向时延值最小的数据来进行去噪声.记测量数据的发送端平均发送测量包的速率为 v (个/秒),计算:

$$L_i = w / \sqrt{w \cdot v}; f = \lceil T / L_i \rceil \quad (5)$$

把前一步中得到的纠斜数据集 划分成 f 个子集 $P_j =$

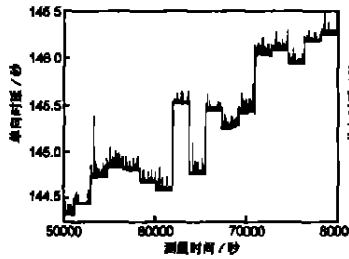


图 3 初步纠斜后的数据

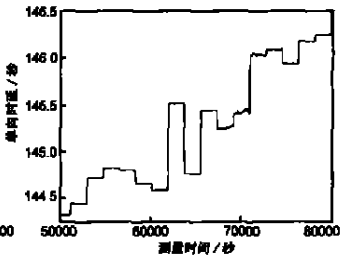


图 4 去噪声后的数据

$\{v_{i_j}, v_{i_{j+1}}, v_{i_{j+2}}, \dots, v_{i_{j+1}-1}\}$, 其中 $j = 1, 2, \dots, f; v_{i_1} = v_1, v_{i_{j+1}-1} = v_N$, 使得对所有 P_j 有 $t_{i_{j+1}-1} - t_{i_j} = L_i$.

使用式(5)进行子集划分使得在任意时间段 w 内去噪声后的数据个数为原来数据个数的平方根,这样做既达到了去除噪声目的,又为后续步骤减少了数据输入量.

划分子集后,再从每个子集 P_j 中选出时延值最小的测量点 $v_s (v_s \in P_j, (\forall v_t \in P_j) d_s < d_t)$, 记为 v_j ; 以这些数据组成新的数据集 $\{v_j, j = 1, 2, \dots, f\}$, 我们称之为去噪数据集.

图 4 是对图 3 中的数据进行去噪声后的结果图.

3.3 确定时钟重置的次数和重置时刻

我们使用“检测窗口”的概念来检测在一定时间段内是否发生时钟重置.检测窗口的大小设为 w .图 5 显示了一个在检测窗口内的去噪数据的例子,容易观察到:此检测窗口内的单向时延值以时刻 74525 秒为分界明显的分成两类,一类的单向时延值基本上分布在 146.08 秒附近,另一类则处于 145.93 秒附近,可以判断在 74525 秒附近发生了时钟重置(已经由日志文件的记录所证实).因此,我们能采用模糊聚类分析来识别时钟重置.

但是如果时钟重置点发生在检测窗口左右边界附近,则较难确定时钟重置.为此,我们设计了滑动检测窗口机制来消除检测窗口位置对算法的影响.如图 6(a)所示,定义在区间 $[t_{sw}, t_{ew}]$ 内的单向时延数

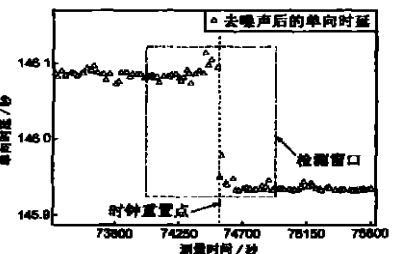
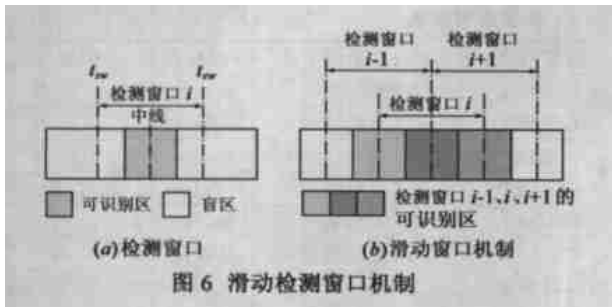


图 5 检测窗口内聚类分析

据为一个检测窗口,其中 t_{sw} 和 t_{ew} 分别是检测窗口的开始和结束时刻.定义检测窗口中间的 50% 部分为可识别区,两边各 25% 区域为“盲区”.只有时钟重置发生在可识别区才能被检测出来.在检测时钟重置时,首先把检测窗口放在测量数据的测量开始时刻;在一个窗口内完成时钟重置的检测后,总是把检测窗口向前滑动 $w/2$ 大小;然后在新窗口内进行检测;如此等等直到测量数据的末尾.如图 6(b)所示,窗口 i 的盲区由窗口 $i-1$ 和窗口 $i+1$ 的可识别区所覆盖,因此不会对时钟重置漏检(除测量最开始和结束的时间段外).

剩下的问题是如何在检测窗口内检测时钟重置.检测规则如下:如果分类结果同时满足以下三个条件: 检测窗口



内超过 75% 的点被分成两类、：每一类内点的数量都超过检测窗口内点的数量的 25%、：两类以某一时点左右分开,则判定有时钟重置发生,否则,没有时钟重置. 参数 75% 和 25% 是由“盲区”的大小决定的. 前两个条件用来判断是否“明显”分为两类,第三个条件进一步刻画时钟重置本质,从而能与长时拥塞等现象区别开来. 检测过程可由以下算法描述:

时钟重置检测算法:

假设落在检测窗口的单向时延测量的去噪数据为: $v_j = (t_j, d_j), 1 \leq j \leq f$. 模糊聚类分析应用到时延集合: $D = \{d_1, d_2, \dots, d_f\}$. 记分类结果中具有元素最多的类为: $B_{\max} = \{d_{m1}, d_{m2}, \dots, d_{mp}\}, 1 \leq m1 < m2 < \dots < mp \leq f, p = |B_{\max}|$; 具有元素次多的类为: $B_{\text{sub}} = \{d_{s1}, d_{s2}, \dots, d_{sq}\}, 1 \leq s1 < s2 < \dots < sq \leq f, q = |B_{\text{sub}}|$.

(1) 以 D 中元素为对象建立模糊相似矩阵 $R_2 = (r_{ij})_{f \times f}$:

$$r_{ij} = \begin{cases} 0, & d_i \times d_j < 0 \\ \frac{|d_i| \wedge |d_j|}{|d_i| \vee |d_j|}, & d_i \times d_j > 0 \end{cases}, d_i, d_j \in D$$

(2) 求的 R_2 闭包得到等价矩阵 R_2^* . 记 F 为 R_2^* 所有元素的集合,对 F 的元素进行排序得到: $|_1 > |_2 > \dots > |_{|F|}$.

(3) for $i = 1$ to $|_{|F|}$

```

{ 计算  $R_2^*$  的 截矩阵  $R_2^*$ ;
  得到对应于  $R_2^*$  的分类  $\{B_i | i = 1, \dots, |F|\}$ ;
  if ( $|B_{\max}| > 0.75 |D|$ ) goto (3);
  if ( $|B_{\text{sub}}| > 0.25 |D|$ ) and ( $|B_{\max}| + |B_{\text{sub}}| > 0.75 |D|$ )
  { // 明显分为两类
    if ( $mp < s1$ ) {  $t_{\text{sor}} = t_{mp}; t_{\text{eor}} = t_{s1};$  goto (4); }
    else if ( $sq < m1$ ) {  $t_{\text{sor}} = t_{sq}; t_{\text{eor}} = t_{m1};$  goto (4); }
    else goto (5); // 两类非左右分开
  }
}

```

(4) 有时钟重置发生在时间段 $[t_{\text{sor}}, t_{\text{eor}}]$; 以在区间 $[t_{\text{sor}}, t_{\text{eor}}]$ 内的原始测得数据为输入,使用按文献[10]定理 7 设计的线性查找算法计算出时钟重置时刻 t_r ;end;

(5) 此检测窗口无时钟重置发生;end;

3.4 获得最优频差

在以上步骤中假设我们获得各时钟重置时刻: $\{t_{r1}, t_{r2}, \dots, t_{rR}, 1 < r1 < r2 < \dots < rR < N\}$, 它们把原始测量数据划分成 $R+1$ 个子集:

$$1 = \{v_i = (t_i, d_i), i = 1, 2, \dots, r_1 - 1\}$$

...

$$k = \{v_i = (t_i, d_i), i = r_{k-1}, r_{k-1} + 1, \dots, r_k - 1\}, \quad (1 < k < R + 1)$$

...

$$R+1 = \{v_i = (t_i, d_i), i = r_R, r_R + 1, \dots, N\}$$

对于每个子集 $j (j = 1, 2, \dots, R + 1)$, 使用文献[9]提出的线性规划方法或文献[10]中提出的凸包方法而得到频差值 j . 定义最优时钟频差为:

$$= \frac{1}{R+1} \sum_{j=1}^{R+1} \quad (7)$$

然后,按如下公式纠正测量时延值(即消除频差)

$$d_i = d_i - (t_i - t_1) \quad i = 1, 2, \dots, N \quad (8)$$

并且按下面公式求每个子集的 j :

$$j = \min(d_{r_{j-1}}, d_{r_{j-1}+1}, \dots, d_{r_j - 1}),$$

$$j = 1, 2, \dots, R + 1 \quad \text{其中}, r_0 = 1, r_{R+1} = N + 1 \quad (9)$$

再对每个子集的测量时延值纠正如下(消除时钟重置造成的时间偏移):

$$d_i^{\text{correct}} = \begin{cases} d_i, & i = 1, 2, \dots, r_1 - 1 \\ d_i - (t_i - t_1), & i = r_1, r_1 + 1, \dots, r_2 - 1 \\ \dots \\ d_i - (t_{r_{R+1}} - t_1), & i = r_R, r_R + 1, \dots, N \end{cases} \quad (10)$$

最后消除时钟频差和时钟重置后的数据集为:

$$v_i^{\text{correct}} = (t_i, d_i^{\text{correct}}), i = 1, 2, \dots, N \quad (11)$$

图 8 显示的是纠正了的单向时延测量数据(与图 3 来源于同一测量数据).

4 实验与分析

我们把算法应用于实际互联网测量数据,并与同类算法进行了比较和分析.网络测量实验进行于 2003 年和 2004 年的 3、4 月间,分别称之为测量 I 与 II,与此相应,获得了两个单向时延数据集;该实验通过发送定长 UDP 数据包来测量单向时延,发送的 UDP 数据流为泊松(Possion)流.表 1 是两次测量的概况,其中的有效测量天数是指在测量期间除去维护或故障时间后的等效天数,有效测量数据量定义为所有单向路径上测得可用数据的测量时间(小时)的总和,其单位为“路径小时”.

表 1 测量实验概况描述

测量	测量日期范围	测量点个数	有效测量天数	有效测量数据量
	2003.3.17 ~ 2003.4.28	8	14	2956
	2004.3.6 ~ 2004.4.10	8	31	6547

两次测量共使用 10 个测量点,测量点的细节如表 2.从表中可以看出,各个测量点所处地理位置、所属网络、接入方式存在一定的差异,这就决定了它们间形成的端到端路径在网络负载、带宽、转发路由器数目等方面形成了多样性,因而在这些路径上测得的单向时延数据也会具有多样性和代表性.

而且,决定主机时钟特性的主机硬件和操作系统在两次测量中也具有一定的多样性.所以,在该测量实验中期望获得具有一定多样性和代表性的实验数据是合理的,我们确信用这些数据来进行算法的检验、评价和比较也应该具有客观性.

测量时两端主机的其中一台通过操作系统内 cron 的设置每 24 小时发生 48 至 50 次的时钟重置(测量点 、 、 、 可进行时钟调整),而且在日志文件中记录发生时钟重置的时刻.人工产生时钟重置的原因如下:首先,能通过日志记录下时钟重置的确切时刻和次数,从而能用于验证算法输出结果的正确性.其次,能在较短的时间内观察到较多时钟重置,有效缩短整个测量实验的时间.

下面使用该实验的测量数据,对我们的算法 FBA 进行验证,并与同类算法进行比较,最后分析 FBA 的时间复杂度.在只利用单方向测量数据检测时钟重置的算法中,文献[10]中的 DAC 算法是目前同类算法中较好的算法.它的时间复杂度较低,检测结果较为准确,且在一定量的实际测量数据上验证过.因此我们选择它作为我们算法的比较对象.

A. 准确性

根据日志文件,我们比较了两算法的准确性.比较时我们使用了三个度量指标:时钟重置的漏检数或漏检率、时钟重置的误检数或误检率以及总的正确率.比较结果列于表 3,其中的时钟重置总数为测量 、 在各自有效测量数据量中发生的时钟重置总数.结果表明:FBA 的漏检率和误检率都明显比 DAC 算法的低,而对于正确率,前者要比后者有显著的提高.

表 3 算法准确性比较

测 量	算 法	时钟重置总数	误检数(率)		漏检数(率)		正确检测个数	正确率
			误检数	误检率	漏检数	漏检率		
	DAC	5450	345	6.33 %	295	5.41 %	4810	88.26 %
	FBA		25	0.46 %	4	0.07 %	5421	99.47 %
	DAC	13094	1023	7.81 %	898	6.86 %	11173	85.33 %
	FBA		71	0.54 %	10	0.08 %	13013	99.38 %

现在分析两个算法出现准确性差异的原因.一方面,由于时钟频差估计的凸包法的缺陷,当时钟频差为正值(负值)时,如果时钟重置发生在靠近子集的左边界(右边界),则 DAC 算法中往往会出现时钟重置的漏判.图 7 是 DAC 算法对节点到节点的单向路径上 2003 年 4 月 27 日测得的单向时延数据的纠正结果图,其中有四个时钟重置漏检发生.作为对比,图 8 是使用 FBA 的结果图,其中没有时钟重置漏检情况,这是由于 FBA 使用了滑动检测窗口机制来避免

表 2 测量点情况表

测量点编号	地理位置	所属网络	接入互联网方式	操作系统	所属测量
	北京邮电大学(北京)	CERNET	以太局域网	RedHat 8.0	、
	清华大学(北京)	CERNET	以太局域网	FreeBSD	、
	北京大学(北京)	CERNET	以太局域网	RedHat 7.3	、
	南开大学(天津)	CERNET	以太局域网	Solaris 8	、
	西安交通大学(西安)	CERNET	以太局域网	RedHat 7.3	、
	河北大学(保定)	CERNET	以太局域网	RedHat 7.3	、
	中科院声学所(北京)	中国科学院网	以太局域网	RedHat 7.2	、
	郑州市	中国网通	ADSL	RedHat 7.3	、
	石家庄市	中国网通	DDN 专线	RedFlag3.0	、
	北京市	中国网通	ADSL	Solaris 8	、

漏检.另一方面,DAC 算法没有对网络引入的噪声进行处理,因此 DAC 算法更易被网络拥塞所“欺骗”,从而产生误检,而 FBA 采用最小滤波对测量数据进行去噪声,大大减小了误检的产生.

此外,我们还考察了两个算法确定时钟重置点的准确性.我们发现由 FBA 和 DAC 算法(如果 DAC 正确检测出了时钟重置)确定的时钟重置点是准确的,与系统日志中记录的时间相同.

B. 鲁棒性

由于客观原因,测量 和测量 所使用测量点有两个测量点不同.因而,两次测量所测的路径及其时延特性有所差别.从表 3 可以看出,FBA 对于测量 I 及 II 的正确率变化很小,而 DAC 算法关于两次测量的正确率却相差近 3 个百分点.这说明 FBA 对路径变化来说鲁棒性优于 DAC 算法.

另一方面,DAC 算法对其输入参数“容限值”非常敏感:此值若过大,则时钟跳变较小的时钟重置情况就会被漏检;若过小,网络拥塞就会被过多的误检成时钟重置.至于 FBA,由于引入了模糊数学的处理方法,不存在象“容限值”那样的硬性参数,因而能保证较好的鲁棒性.

C. 算法时间复杂度

由于 FBA 中第三步的时间复杂度比前两步高,所以第三步的时间复杂度决定了整个算法的时间复杂度.假设 N 为单向时延测量数据的个数,测量数据的发送端平均发送测量包的速率为 v (个/秒).则每个检测窗口的去噪数据个数为 \sqrt{wv} ,由模糊数学定理,每次重置点检测中求解模糊等价矩阵的复杂度为 $O(\sqrt{wv})^3 \log \sqrt{wv}$,而进行分类的复杂度为 $O((\sqrt{wv})^3)$,所以每次重置点检测的复杂度为 $O(\sqrt{wv})^3 \log \sqrt{wv}$;整个算法需

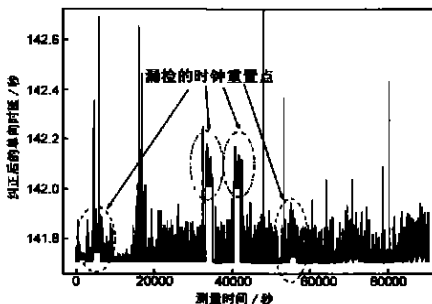


图 7 DAC 算法纠正结果

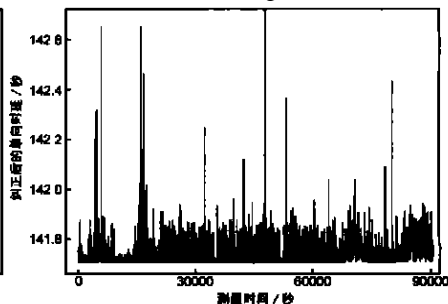


图 8 FBA 的纠正结果

进行 $(2N/wv) - 1$ 次重置点检测,因而整个算法的时间复杂度为 $O(cN)$,其中 c 为常数 $\sqrt{wv} \log \sqrt{wv}$. 因此我们的算法与 DAC 算法的时间复杂度是相近的.

5 总结及未来的工作

本文提出了一种利用模糊聚类分析从单向时延测量数据中去除时钟频差和时钟重置的新算法. 经过大量的实验数据验证和分析,与现存同类算法如 DAC 算法相比,新算法更准确、更鲁棒,而且时间复杂度为 $O(N)$,与 DAC 的复杂度相近. 在单向时延测量中引入时钟相关误差的另一原因是时钟速率调整(如 NTP 进行同步时),准确、有效地纠正有时钟速率调整的单向时延测量数据是更为复杂和困难的问题. 虽然文献 [10, 12] 都提出了有关此问题的算法,但是与检测时钟重置时相同,仍存在由于判断过于“绝对”而产生“漏检”或“误检”的缺点. 这也是我们下一步要做的工作.

参考文献:

- [1] G Almes, S Kalidindi, M Zekauskas. A One-Way Delay Metric for IPPM, RFC 2679 [S]. Internet Engineering Task Force, September 1999.
- [2] C Demichelis, P Chimento. IP Packet Delay Variation Metric for IP Performance Metrics (IPPM), RFC 3393 [S]. Internet Engineering Task Force, November 2002.
- [3] V Paxson, G Almes, J Mahdavi, M Mathis. Framework for IP performance metrics, RFC 2330 [S]. Internet Engineering Task Force, May 1998.
- [4] D Mills. Network Time Protocol (version 3) : Specification, implementation and analysis, RFC 1305 [S]. Internet Engineering Task Force, March 1992.
- [5] V Paxson, A Adams, M Mathis. Experiences with NIMI [A]. Passive & Active Measurement : PAM2000 [C]. Hamilton, New Zealand, April 2000. <http://pam2000.cs.waikato.ac.nz/pdf/papers/P006.pdf>
- [6] A Pástor, D Veitch. PC based precision timing without GPS [A]. In Proc ACM SIGMETRICS 2002 [C]. June 2002. 1 - 10.
- [7] Vern Paxson. Measurements and Analysis of End-to-End Internet Dynamics [D]. Ph D thesis, University of California, Berkeley, 1997.
- [8] Vern Paxson. On calibrating measurements of packet transit times [A]. In Proc. ACM SIGMETRICS '98 [C]. Madison, Wisconsin, June 1998. 11 - 21.
- [9] S B MOON, P SKELL Y, D TOWSLEY. Estimation and removal of clock skew from network delay measurements [A]. In Proc. IEEE INFOCOM '99 [C]. March 1999. 227 - 234.
- [10] Li Zhang, Zhen Liu, Cathy Honghui Xia. Clock synchronization algorithms for network measurements [A]. In Proc IEEE INFOCOM 2002 [C]. June 2002. 160 - 169.
- [11] R S Ryger. fixclock: Removing Clock Artifacts from Communication Timestamps [R]. Technical Report DCS/ TR-1243 Yale University, March 2003.
- [12] Wang JF, Yang JH, Zhou HX, Xie GG, Zhou MT. Detecting clock dynamics in one-way delay measurement [J]. Journal of Software, 2004, 15 (4) : 584 - 593.
- [13] R T ROCKAFELLAR. Convex Analysis [M]. Princeton Univ. Press, 1970. 3 - 94.
- [14] Dubois D, H Prade. Fuzzy sets and systems: theory and applications [M]. New York: Academic Press, 1980. 9 - 149.
- [15] 李洪兴, 汪培庄. 模糊数学 [M]. 北京: 国防工业出版社, 1994. 1 - 101.
- [16] F Höppner, F Klawonn, R Kruse, T Runkler. Fuzzy cluster analysis: methods for classification, data analysis, and image recognition [M]. Chichester Great Britain John Wiley & Sons Ltd, 1999. 5 - 113.

作者简介:

王洪波 男, 1975 年 12 月生于河北省唐县, 北京邮电大学网络与交换国家重点实验室博士研究生, 主要研究方向: 网络测量与管理, 业务量工程, IP 报文分类技术等. E-mail: hbwang@bupt.edu.cn

林宇 男, 1976 年 12 月生于福建省浦城, 博士, 北京邮电大学副教授, 主要研究方向: 互联网服务质量管理与测量、无线自组织网络及传感器网络等.

金跃辉 女, 1965 年 3 月生于黑龙江省哈尔滨, 北京邮电大学副教授, 主要研究方向: 互联网传输层协议及优化、网络测量、移动 IP 技术、MPLS 等.

程时端 女, 1940 年生于上海, 北京邮电大学教授, 博士生导师, 主要研究方向: 互联网性能分析与服务质量控制、网络测量、传感器网络、业务量工程等.