

基于环状拓扑的网络时钟同步*

朱海婷 韩 林容容

(东南大学 计算机科学与工程学院, 江苏 南京 210096)

摘 要: 时钟同步是很多网络应用的基本要求. 文中设计实现了基于环状拓扑的分布式多主机时钟同步系统. 该系统使用面向单向延迟测量的 Allair & Vega 方法支持多台主机相互之间的时间同步, 使用动态环状逻辑拓扑的控制方案, 利用特有的环状拓扑上的累积误差提高系统的精度, 得到了高精度的相对时钟偏移修正方案. 在实验环境下进行的测试结果表明, 该方法利用累积误差修正可以得到毫秒级的同步结果.

关键词: 时钟同步; 环状拓扑; 单向延迟; 相对时钟偏移

中图分类号: TP393

文献标识码: A

计算机时钟是基于各自独立的时钟源, 两台机器每天的误差达数秒. 随着分布式计算和网络技术的发展, 不准确的时钟对于网络结构和其中的应用程序的安全性会产生较大的影响. 时钟同步是网络环境下应用的基本要求.

时钟同步在不同的应用需求下有不同的解决方案. 依靠系统管理员使用命令手动来调节各个系统的时钟来支持同步, 是最简单但误差在秒级以上的的方法. 大型网络中, 现在广泛使用网络时间协议(NTP), 它可以满足一般的应用需求. 但由于不同主机的时钟振荡频率不一, 即使采用 NTP 同步协议, 也很难在一段相对较长的时间里在精度上满足同步的要求. 测试实验发现 5min 内两台主机之间的不同步大约为 30ms^[1]. 随着网络技术的不断发展, 许多度量参数, 如单向延迟的测量需要毫秒级的同步精度, 这样的精度需要其他的解决方案来对时钟偏差进行进一步修正. 林容容等^[1]对 NTP 同步后的时间做了进一步的同步实验, 以使其能够符合单向延迟测量的需要. 但是, NTP 的同步精度还依赖于运行 NTP 服务的结点之间的稳定性和对称性. 另一方面,

在网络中, 并不是所有的主机都支持 NTP 服务, 而且使用全球定位系统作为一级服务器的标准, 也会增加成本. 因此, NTP 更适用于支持有绝对时钟要求的应用.

现有的从软件上进行多主机时钟同步的技术包括基于树状模型的 NTP^[2-3]、点对点(P2P)网状模型的无分级时钟协议(CTP)^[4], 以及其他如基于星型模型的自适应时钟同步系统(SACS)^[5]. 本研究主要针对在相对时钟精度上要求较高的应用需求, 即多主机系统中主机与标准时间的同步要求不高, 但对各个主机之间的时间的同步要求较高. 比如面向单向延迟的测量中, 要求各个主机之间时间同步有毫秒级精度; 又如新的网络测量技术网络层析技术, 也对各个主机采集数据的时间同步有一定的要求^[6]. 为此, 本研究采用了环状拓扑结构, 分析和设计了一个面向环状拓扑的多主机时钟同步系统, 并在实验环境下进行了测试. 该系统的特点有: (1)构造拓扑环, 即网络上需要同步的主机之间建立一个环; (2)利用环状逻辑拓扑所形成的回路来统计累积误差, 从而提高多主机同步时的同步精度.

收稿日期: 2007-06-28

* 基金项目: 国家“973”计划资助项目(2003CB314804); 华南理工大学广东省计算机网络重点实验室开放基金资助项目(CCNL200706)

作者简介: 朱海婷(1983-), 女, 博士生, 主要从事网络测量研究. E-mail: htzhu@njnet.edu.cn

1 同步模型

1.1 环同步模型

在各个需要同步的主机之间建立起一个逻辑上的环,各节点之间平等,如图 1 所示,本文称它为同步环。环上每一个节点代表一个同步主机,称作同步节点。同步环上,给每一个同步节点指定一个编号,从 N_0 到 N_m ,共 $m+1$ 个同步节点。对于任意一个节点 N_i ($0 \leq i \leq m$),它与其他同步节点间的同步关系为:节点 N_i 以节点 N_{i+1} 的时间为标准时钟,把它称为 N_i 节点向 N_{i+1} 节点同步,记为 $N_i \rightarrow N_{i+1}$,在图 1 上表示为一个带方向的箭头。

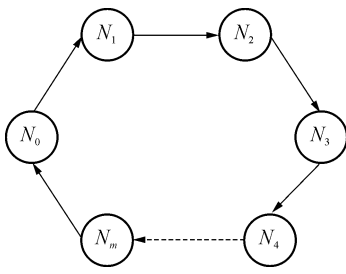


图 1 逻辑环状拓扑结构
Fig 1 Logically ring-topology structure

假设 t 时刻,在 N_i 节点的时间值为 t_i 。由于 $N_i \rightarrow N_j$ ($0 \leq i < j \leq m$ 且 $j = i + 1$) 在 N_i 节点上向 N_j 节点同步修正后的时间值记为 t_{ij} 。若有节点 N_k ($0 \leq j < k \leq m$),则在 N_i 节点上向 N_j 节点上的同步数据 N_{jk} 同步修正后的时间值记为 t_{ijk} 。图 2 为各节点累积同步的示意图。其中箭头的方向表示时钟同步关系,箭头所指方向为标准时钟(时间值)。

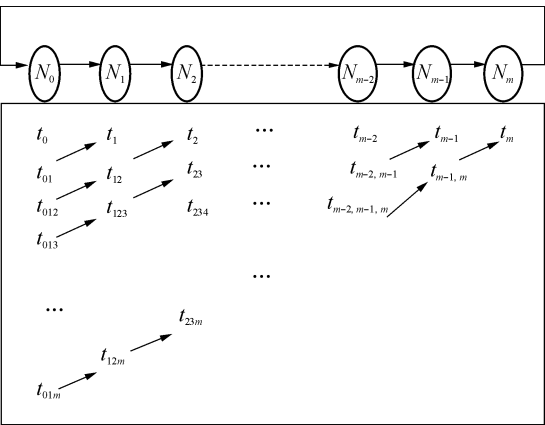


图 2 逻辑环状拓扑上的同步模型
Fig 2 Synchronization model on logical ring topology

设计中,在环上的两个主机之间同步使用的是基于时钟线性模型的 Altair & Vega 方法^[7]作为时

钟同步模型。Altair & Vega 具体算法的过程主要分为两步:首先,对输入的数据集进行凸包集合运算,找出凸包集合元素组(用栈实现);然后在该凸包集合元素中找出能满足目标条件的目标元素,即可得同步时钟模型。

1.2 模型的分析

本研究设计的目标是一个能够稳定运行在网络环境中,基于环状拓扑且具有自主维护拓扑能力的多主机时钟同步系统,为分布式网络环境下的各主机提供准确的相对时间戳。其两个要素分别是环状拓扑结构的控制维护和面向延迟的时钟同步。因此从功能上,需分别满足同步环操作和同步操作两大块的应用需求,系统的结构如图 3 所示。下面分别讨论。

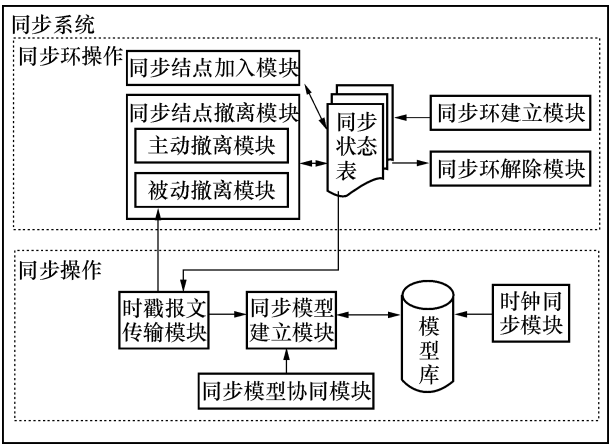


图 3 同步系统功能模块
Fig 3 Function modules of synchronization system

1.2.1 环操作

环操作包括环的建立和解除、节点的加入和撤离。其中撤离方式有:(1)节点的主动提出撤离;(2)节点的被动撤离,即网络故障或者节点突然崩溃等原因造成的无法通信。本研究只考虑了故障-停止型故障。

实际中,环上的节点分为“普通同步节点”(以下简称同步节点)和“基点”两类。任何同步节点都具有成为“基点”的能力。“基点”的主要功能是:(1)统计系统的全局同步状态得到同步一周后的累积误差;(2)在同步节点的主动加入过程中,计算和判断该点加入位置和加入后的同步关系;(3)在基本的环境操作——环的建立和解除中,担当如同令牌环中的“监控器”一样的角色,管理环上的一些会涉及到全局同步状态的操作。为了保证系统的健壮性,在基点失效的时候也能正常工作,需要选择其他的

结点对全局信息进行备份. 这样可保证在基点发生故障时, 备用结点可以代替它进行日常维护.

借鉴 chord系统的 finger表, 本设计采用使用简化了的邻居表, 即环上的一个结点上只保存附近左右两个邻居结点的信息. 本研究定义了简单的环状拓扑的维护协议, 用消息进行传递, 在各个主机之间进行相应的环的维护工作. 由于环上的每个结点都是独立的, 功能也都是相同步的, 每个结点都有可能成为基点 (因为原基点可能撤离同步环), 而基点也同样具有环上普通结点的功能. 因此, 在定义消息报文的标识时, 对相同功能模块里的完成同一功能的若干消息给出了不同的标识, 以便接收的结点给出对应的操作. 环上的各种操作动作都由相应的消息报文内容触发.

同步环的建立过程如下:

```
if(不是基点)
{发送建环请求; }
else 建初始全局状态表;
while (本地表未建完) {
    if(收到建环请求 EST_H) {
        更新全局表;
        查找全局表, 若有, 则取出其左右邻居信息;
        回复更新的同步表消息 UPD 以帮助结点
        建立本地表;
        发送更新的同步表消息 UPD 以帮助其邻
        居完善邻居表; }
    if(收到更新的同步表 UPD)
    {按照消息内容建立或更新本地表; }
}
```

基点选择生成备份结点;

基点通知备份结点备份全局表 BAK;

```
if(收到备份全局同步表消息 BAK)
```

```
{备份全局表; }
```

// 建立环完成

环同步结点加入过程如下:

发送加入环 INS_H 的请求;

```
while (主动加入状态) {
    if(收到询问谁是基点 Q_BAS消息)
    {答复当前基点的信息; }
    if(收到加入环的请求) {
        更新全局表;
        根据全局表中距离项的计算并确定该结
        点的左右邻居;
```

回复插入回应 RE_NS消息以帮助结点
建立本地表;

通知备份结点备份全局表 BAK; }

```
if(收到插入回应消息 NS_H) {
```

按照消息内容建立或更新本地表;

加入结点发插入回应消息以帮助其邻居
完善邻居表; }

```
}
```

```
if(收到备份全局同步表 BAK)
```

```
{备份全局表; }
```

// 加入环的过程完成

同步结点的撤离过程:

```
while (没有收到退环或撤离信号) {
```

```
if(收到撤离信号) {
```

```
    if(撤离结点是基点)
```

发送重建全局表消息 GLO 给备份结点以
重建基点;

```
    else{
```

更新全局表;

回复主动退出消息 RE_AWD 以通知该结
点撤离;

通知备份结点备份全局表 BAK; }

退出;

```
}
```

```
if(收到主动退出帮助消息 AWD_H)
```

回复主动退出消息以帮助其邻居完善邻
居表;

```
if(收到主动退出回复消息 RE_AWDBAK)
```

主动退出回复 (RE_AWD) 以帮助其邻居
完善邻居表;

```
if(收到备份全局同步表消息 BAK)
```

备份全局表;

```
if(收到重建全局同步表消息 GLO) {
```

重建基点;

发送备份全局表消息 BAK, 重建备份结点; }

```
}
```

被动撤离的过程和主动撤离有类似的地方, 但之前多了一个询问和判断结点是否失效的步骤.

最后, 同步环退出的过程为: 由收到结束同步信号的结点发出帮助退出消息 (DM_H) 给基点. 基点收到退出消息 (DM_H) 后转发回复退出消息 (RE_DM) 给其右邻居结点. 其他结点收到回复退出消息 (RE_DM) 后依次转给右邻居结点并退出同步环.

直到基点收到回复退出消息 (RE_DM), 完成退出.

1.2.2 同步操作

由于 Allair& Vega方法是面向单向延迟数据的, 同步操作需要完成对时戳数据的采集和分析 (详细过程见参考文献 [7]), 从中得出时钟同步的模型, 继而以该模型为依据进行时钟同步.

1.2.3 同步环建立流程

同步过程是首先启动时戳报文的传输模块, 从同步邻居状态表中获取本地结点的同步对象信息, 采集时戳报文, 对时戳数据进行分析计算, 从而建立结点间的时钟同步模型. 当结点在一定时间间隔内没有同步信号时, 该结点会被认为是失效了的, 其邻居将迫使其退出时钟同步环的同步工作. 因此, 在时戳传输中, 设置计时器, 计时器满期时, 结点被认为失效, 时钟同步模块的工作将提前结束, 该结点将退出同步环. 如果结点正常, 则进行正常的同步模型建立和计算工作. 各同步结点的同步流程如图 4 所示.

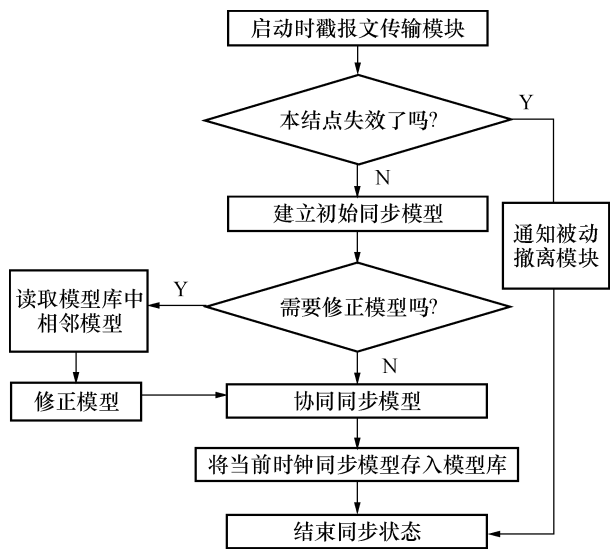


图 4 同步环建立流程

Fig 4 Setup procedures of synchronization ring

1.2.4 模型的误差分析

在该同步环上, 可以得到节点 N_i 到 N_k 的累积同步误差, 将该值用符号 IER 表示. IER_{ik} 表示结点 N_i 和 N_k 间的累积同步误差.

$$IER_{ik} = (t_{ijk} - t_{ij}) / (k - i - 1) \quad (j = i + 1) \quad (1)$$

在实际测量中, 无法同时得到一个全局时间 t 时刻下在各不同节点的时刻值 $t_0, t_1, \dots, t_m$. 此时, 设在 t_{ijk} 和 t_{ij} 所在的同步路径下得到的相对时钟偏移分别为 f_{ijk} 和 f_{ij} , 则 N_i 到 N_k 的累积同步误差可由下式计算得到

$$IER_{ik} = (f_{ijk} - f_{ij}) / (k - i - 1) \quad (2)$$

设 $(t_1, T_1), (t_2, T_2), \dots, (t_n, T_n)$ 为一组使用 Allair& Vega方法测得的样本, 考虑回归模型为

$$D = a + bt + \varepsilon \quad (\varepsilon \sim N(0, \sigma^2)) \quad (3)$$

假设观测的时钟偏移 D 满足与时间 t 成线性关系 (一次项系数为常数 b , 常数项系数为 a , ε 为测量误差), 并且含有均值为 0 方差为 σ 的正态分布误差的分布. 在通过显著性检验的前提下, 设 $\hat{D}_0 = \hat{a} + \hat{b}t_0$ 为式 (3) 得到的在 t_0 处的观察值 $D_0 = a + bt_0 + \varepsilon_0$ 的一个估计. 给定置信度 $1 - \alpha$ 可以得时钟偏移 D_0 的预测区间^[8-10] 为 $(\hat{D}_0 - \delta(t_0), \hat{D}_0 + \delta(t_0))$, 其中:

$$\delta(t_0) = \frac{t_0}{2} (n - 2)^* \sqrt{\frac{\sum_{i=1}^n (D_i - \hat{D}_i)^2}{n - 2} \left[1 + \frac{1}{n} + \frac{(t_0 - \bar{t})^2}{\sum_{i=1}^n (t_i - \bar{t})^2} \right]} \quad (4)$$

Allair& Vega方法在两结点间进行时钟同步时, 同步的误差 $\delta(t_0)$ 不仅和取到的时戳样本观察值以及置信水平有关, 与 t_0 也有关. 当取得的时戳样本观察值和置信水平给定后, 同步误差就只依赖于 t_0 , 估计的预测区间的长度越短, 估计预测的也就越精确. 除了回归算法产生的系统误差以外的误差是难于测量的; 但当多结点构造成逻辑环时, 结点间的系统误差由于递归同步过程, 得到累积, 从而便于测量和修正. 若依据同步环的结点 N_i 到结点 N_k 的同步路径上的测量结果对同步路径上的各结点进行修正, 则可得到该路径上的同步误差为

$$\Delta y_0 = 2\delta(t_0) - IER_{ik} \quad (5)$$

2 系统测试和结果

2.1 实验环境

测试环境由四台 PC 主机搭建, 主机名、各主机的 IP 地址及其拓扑连接结构如图 5 所示, 其中一台

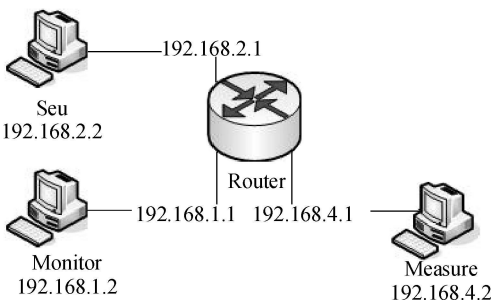


图 5 实验环境

Fig. 5 Test environment

名字为 Router 的机器作为路由器, 其他三台机器分别配置在三个网段, 他们之间的通信由 Router 转发.

本设计实现的系统在由此 4 个服务器结点组成的逻辑同步环上进行.

2 2 实验结果

2 2 1 同步结果

在环同步系统中, 建立同步模型的时间间隔是由用户设置的, 测试中, 本研究选用 5 m in 为同步时间间隔的长度, 测量了 1 h 内环同步系统的同步情况. 图 6 是以这 4 个结点构建的同步环为拓扑基础, 对各主机以 5 m in 粒度为时间间隔, 对此时间内的时戳进行同步和环同步修正后, 一个时间粒度内在结点 Measure 上得到它与 Monitor 之间的单向延迟值. 由公式 (4) 得出的预测区间值近似值与实际计算结果之间的误差平均约为 $5.15069\text{E-}05\text{ s}$ 修正后两个方向上的单向延迟值相加与实际的 RTT 值进行比较后, 平均误差只有 $2.137841\text{E-}04\text{ s}$ 图 6 表明修正预测模型误差很小, 比现有的模型更加准确.

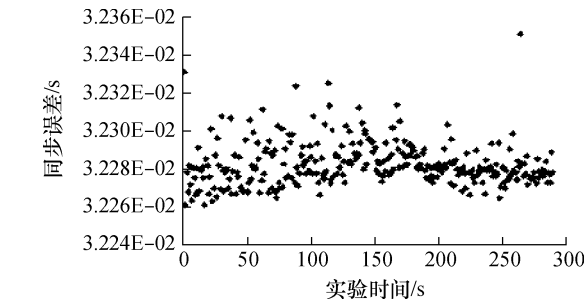


图 6 5 m in 粒度同步后的单向延迟
Fig 6 Single trip delay after synchronization in each 5 minutes

图 7 是持续 1 h 时间内, 以 5 m in 粒度为时间间隔进行同步后得到的单向延迟值. 模型的线性状态均良好. 对图 7 中 12 个 5 m in 的分段模型进行计算和分析后, 发现模型中的参数 k 值的变化范围比较小, 但参数 b 的变化比较明显 (主要存在于每一个 5 m in 分段之间), 其 R^2 测度值^[4]为 1.000 001, 说明是符合线性模型的. 从图 7 中还可以明显看出测量得到的延迟值分段分布. 这表明各时间间隔内以各自的时钟同步模型进行同步后的时戳值在边界上是不连续的. 这是本系统设计时没有考虑到的一个问题. 但由图 7 可知, 图 7 中各段数据之间有明显的 discontinuity; 由于各段的数值波动均在毫秒级, 要进一步减小误差, 必需消除这个波动, 这个问题需在以后的研究中解决. 当然, 在本系统中, 若不希望出现这种

间断性的不连续时戳, 可在配置同步时间间隔中选择比较大的间隔时间. 因为在时间间隔内得到的修正时戳是连续的.

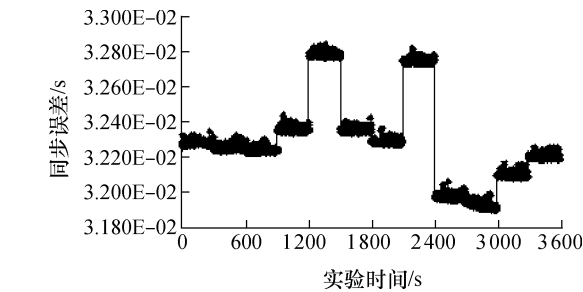


图 7 1 h 内主机 monitor 上测得的单向延迟
Fig 7 Single trip delay on host monitor during 1 hour

2 2 2 误差分析

在与图 6 相同的数据集里 ($\text{IER} = 0.103\text{ m s}$), 根据对环同步方法的误差估计公式 (5) 计算在各时间间隔内的同步误差统计如图 8 和表 1 所示 (取置信系数为 95%. 由于其中的参数 $n = 300$ 较大, 采用了标准正态分布代替 t 分布进行计算): 对于相对时钟偏差, 即模型参数 k 项的变化情况, 分别采用了 300 1 800 3 600 7 200 s 为时间粒度对该数据集进行分析统计. 结果显示, 时间粒度取得越大时, 相对时钟偏差的变化越稳定.

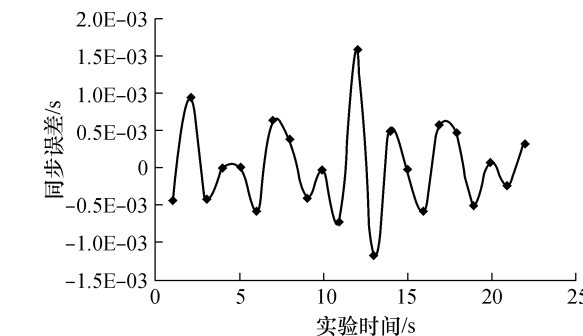


图 8 模型计算引起的误差
Fig 8 Errors caused by the model calculation

表 1 同步误差统计			
Table 1 Statistics of synchronization errors			
误差	最小值	最大值	平均值
$\delta(t_0)$	0.018992	0.019087	0.019024
Δy_0	0.037882	0.038071	0.037946

通过 24 h 的同步实验测试同步环累积误差的变化情况可知, 累积误差值均在 0.5 m s 左右, 结果如图 9 所示. 由图 9 可以看出, 4 个结点构成的环上进行同步时, 环同步可以修正到毫秒级的同步误差.

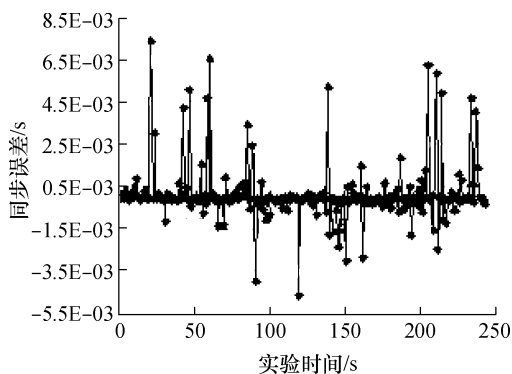


图 9 累积误差

Fig 9 Inherited errors

3 结语

本研究提出了一个基于环状模型的分布式同步系统, 包括设计和测试两个方面. 本系统在结点间的同步模型建立之后, 利用逻辑的环状同步拓扑, 计算该同步环上的时钟累积误差, 对环上各结点的同步模型进行协同修正, 以在全局范围内提高同步精度, 同时可以保证精度的同步. 对于本系统的同步结果而言, 结点间使用 Allair& Vega同步方法后再使用环同步进行进一步的修正, 实验测试结果得到的误差可以达到毫秒以下, 一般在 0.5ms 左右.

参考文献:

- [1] 林容容, 程光, 丁伟. 基于 NTP 协议的时钟同步系统 [C]// 第五届海峡两岸信息技术研讨会论文集. 南京: 东南大学, 2003: 288-294.
- [2] Mills D. Network time protocol (version 3): specification,

- implementation and analysis RFC 1305 [S]. Menlo Park Network Information Center SRI International 1992.
- [3] 清华大学网络研究中心. 网络时间协议简介 [EB/OL]. [2004-10-12]. <http://www.time.edu.cn/nntp.htm> 2003.
- [4] Omer Gurewitz, Israel Cidon, Moshe Sidi. Network time synchronization using clock offset optimization [C]// Proceedings of the 11th IEEE International Conference on Network Protocols Atlanta: IEEE, 2003: 212.
- [5] Zhao Ying, Zhou Wanlei, Elia J. Lanham, et al. Self adaptive clock synchronization based on clock precision difference [C]// Proceedings of the Twenty-sixth Australasian Computer Science Conference on Conference in Research and Practice in Information Technology Adelaide: Australian Computer Society, 2003: 16-18.
- [6] Rui Castro, Mark Coates, Gang Liang, et al. Network tomography: recent developments [J]. Statistical Science, 2004, 19(3): 499-517.
- [7] 林容容, 丁伟, 程光. 面向单向延迟测量的时钟同步技术研究 [J]. 计算机工程与应用, 2005(34): 127-130.
- Lin Rong-rong, Ding Wei, Cheng Guang. Research on clock synchronization technique for oneway delay measurement [J]. Computer Engineering and Application, 2005(34): 127-130.
- [8] 邓勃. 分析测试数据的统计处理方法 [M]. 北京: 清华大学出版社, 1995.
- [9] 曹振华, 赵平, 胡跃清. 概率论与数理统计 [M]. 南京: 东南大学出版社, 2001: 211-216.
- [10] 唐象能, 戴俭华. 数理统计 [M]. 北京: 机械工业出版社, 1994: 165-169.

Network Clock Synchronization Based on Ring Topology

Zhu Hai-ting Ding Wei Lin Rong-rong

(School of Computer Science and Engineering Southeast University Nanjing 210096 Jiangsu China)

Abstract Clock synchronization is a basic requirement form any network applications. In this paper, a distributed multihost clock synchronization system based on ring topology is designed and implemented. In the proposed system, the Allair& Vega method oriented to one-way delay measurement is adopted to realize the synchronization between multiple hosts, and the dynamic logic ring topology approach and the accumulated synchronous errors on the ring topology are employed to improve the system accuracy. Thus, a high-precision correcting method of relative clock skew is proposed. Test results indicate that the synchronization precision of the proposed method can be controlled in millisecond scale due to the accumulated error revision.

Key words clock synchronization ring topology one-way delay relative clock offset