

# 基于最小往返时差对时算法的电力用户用电信息采集终端时钟同步研究

李新家, 李 平

(江苏方天电力技术有限公司, 江苏 南京 211102)

**摘 要:** 网络时间协议(network time protocol, NTP)是解决现代通信网络时间同步的重要手段之一, 但目前网络结构是通过有线或无线网络进行信息交互, 网络的上下行通信信道并非对称结构, 且通信质量的随机性非常大, 使得往返路径存在很大的差异, 传统NTP对时方法已不能满足当前的网络结构。因此本文提出一种基于NTP的最小往返时差对时算法, 并将其运用到电力采集终端的时钟同步架构中, 其结果证明新时钟同步算法相较于传统时钟同步算法在对时精度和收敛度上都有了较大的提升。

**关键词:** 用电信息采集; 时钟同步; NTP; 最小往返时差

## 0 引言

随着电网的不断发展, 对于电力用户用电信息采集系统的实时数据采集, 事件采集的时效性的要求也越来越高。通信的传输延时<sup>[1-2]</sup>, 采集终端自身故障, 都会使电能表的时钟出现较大误差, 网络时间协议<sup>[3]</sup> (network time protocol, NTP)则是解决这一问题较为简单、经济的方法

NTP 是用于互联网中时间同步的标准互联网协议。NTP 协议机制严格、实用、有效, 适应在各种规模、速度和连接通路情况的网络环境下工作。NTP 采用了 Client/Server 结构, 使用 UTC 作为时间标准, 本身基于 UDP/IP, 使用分层式时间分布模型, 具有相当高的灵活性, 可以适应各种互联网环境。NTP 不仅校正现行时间, 而且持续跟踪时间的变化, 自动进行调节。现今, 在通常的环境下, NTP 提供的时间精确度在毫秒级。但是随着无线网络与有线网络的紧密结合, 由此网络的不对称性不断加大, 使得传统 NTP 对时算法获取的时间偏差将存在较高的误差, 本文提出的最小往返时差 (MRTTD) 算法可以有效地降低此类误差。

## 1 NTP 概述

NTP<sup>[4-5]</sup>协议的精确对时主要是在主从工作模式下实现的。NTP算法首先就要根据服务器和客户端的往返报文来确定两地时钟的差值和报文在网络

中传输的时延, 这里先定义一下, 客户端和时间服务器之间的时间偏差用  $\theta$  表示, 对时过程中的网络传输延迟用表示。

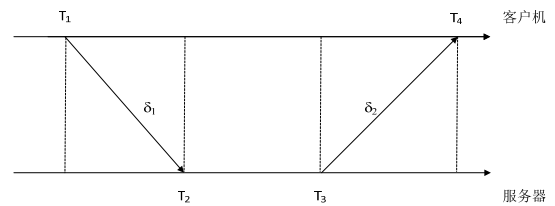


图1 服务器和客户端对时过程

图1中,  $T_1$ 和 $T_4$ 是客户端记录的发送NTP报文和接收NTP报文的时钟读数,  $T_2$ 和 $T_3$ 是服务器端记录的接收和发送NTP报文的时钟读数,  $\theta$ 是服务器和客户端之间时间偏移量的估计值, 从客户端发送报文到服务器端的路径延迟是  $\delta_1$ , 从服务器到客户端的路径延迟是  $\delta_2$ , 可列出以下两个公式:

$$T_2 - T_1 = \theta + \delta_1 \quad (1)$$

$$T_4 - T_3 = \delta_2 - \theta \quad (2)$$

## 2 传统 NTP 对时机制

传统 NTP 时钟同步是假设从客户端到服务器的路径延迟和从服务器到客户端的路径延迟相等,

即  $\delta_1 = \delta_2 = \delta/2$ ，则分别代入 (1) (2) 两式，可得

$$T_2 - T_1 = \theta + \delta/2 \quad (3)$$

$$T_4 - T_3 = \delta/2 - \theta \quad (4)$$

服务器和客户端时钟的时间偏差为：

$$\theta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} \quad (5)$$

客户端与服务器端总的网络传输延迟为：

$$\delta = (T_2 - T_1) - (T_4 - T_3) \quad (6)$$

客户端通过  $\theta$  调整与服务器的时钟偏差。

传统 NTP 的时钟偏差是假设  $\delta_1 = \delta_2$  相等的情况下计算得出，但目前网络结构是通过有线及无线网络进行信息交互，此网络的上下行通信信道并非对称结构，且通信质量的随机性非常大，使得往返路径存在很大的差异，传统 NTP 的时钟偏差并不能解决这一问题，因而提出 NTP-MRTTD 算法将会有效降低这方面的误差。

### 3 NTP-MRTTD 对时机制

最小往返时差 (network time protocol-minimum round trip time difference, NTP-MRTTD) 充分考虑信道的不对称性，通过基础方程式 (1)、(2) 得到

$$\delta_2 - \delta_1 = (T_4 - T_3) - (T_2 - T_1) + 2\theta \quad (7)$$

式 (7) 中表示对时往返路径的传输时差，由于在  $N$  次连续对时期间内服务器与客户端时间偏差的变化可以忽略不计，因此可将式 (7) 中的  $\theta$  作为一个常数，通过  $N$  次连续对时得到  $N$  组不同的  $(\delta_2 - \delta_1)$ 。即

$$\begin{pmatrix} \delta_{21} - \delta_{11} \\ \delta_{22} - \delta_{12} \\ \vdots \\ \delta_{2n} - \delta_{1n} \\ \vdots \\ \delta_{2N} - \delta_{1N} \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ \vdots \\ -1 \\ 1 \end{pmatrix}^T \begin{pmatrix} T_{11} & T_{21} & T_{31} & T_{41} \\ T_{12} & T_{22} & T_{32} & T_{42} \\ \vdots & \vdots & \vdots & \vdots \\ T_{1n} & T_{2n} & T_{3n} & T_{4n} \\ \vdots & \vdots & \vdots & \vdots \\ T_{1N} & T_{2N} & T_{3N} & T_{4N} \end{pmatrix}^T + 2\theta \quad (8)$$

计算出  $N$  组其中  $n = 1, 2, 3, \dots, N$ ，传统 NTP 的  $\delta_{2n} - \delta_{1n} = 0$ ，现今通过找到  $N$  组  $(\delta_{2n} - \delta_{1n})$  最小值就可以计算出  $\theta$ 。

$$\min \begin{pmatrix} \delta_{21} - \delta_{11} \\ \delta_{22} - \delta_{12} \\ \vdots \\ \delta_{2n} - \delta_{1n} \\ \vdots \\ \delta_{2N} - \delta_{1N} \end{pmatrix} = \min \begin{pmatrix} 1 \\ -1 \\ \vdots \\ -1 \\ 1 \end{pmatrix}^T \begin{pmatrix} T_{11} & T_{21} & T_{31} & T_{41} \\ T_{12} & T_{22} & T_{32} & T_{42} \\ \vdots & \vdots & \vdots & \vdots \\ T_{1n} & T_{2n} & T_{3n} & T_{4n} \\ \vdots & \vdots & \vdots & \vdots \\ T_{1N} & T_{2N} & T_{3N} & T_{4N} \end{pmatrix}^T + 2\theta \quad (9)$$

式 (9) 的  $\theta$  为常数，最小值就是取  $\min[(1, -1, -1, 1)(T_{1n}, T_{2n}, T_{3n}, T_{4n})^T]$ ，其中  $n = 1, 2, 3, \dots, N$ ，若得到一组  $(T_{1n}, T_{2n}, T_{3n}, T_{4n})$  直接通过式 (5) 计算出  $\theta$ 。若得到多组数据，在此集合中通过选择  $(\delta_{2n} + \delta_{1n})$  最小的一组即传输总时延最小。

$$\min \begin{pmatrix} \delta_{21} + \delta_{11} \\ \delta_{22} + \delta_{12} \\ \vdots \\ \delta_{2n} + \delta_{1n} \\ \vdots \\ \delta_{2N} + \delta_{1N} \end{pmatrix} = \min \begin{pmatrix} -1 \\ 1 \\ \vdots \\ -1 \\ 1 \end{pmatrix}^T \begin{pmatrix} T_{11} & T_{21} & T_{31} & T_{41} \\ T_{12} & T_{22} & T_{32} & T_{42} \\ \vdots & \vdots & \vdots & \vdots \\ T_{1n} & T_{2n} & T_{3n} & T_{4n} \\ \vdots & \vdots & \vdots & \vdots \\ T_{1N} & T_{2N} & T_{3N} & T_{4N} \end{pmatrix}^T \quad (10)$$

再通过式 (5) 得出  $\theta$ 。

### 4 仿真与分析

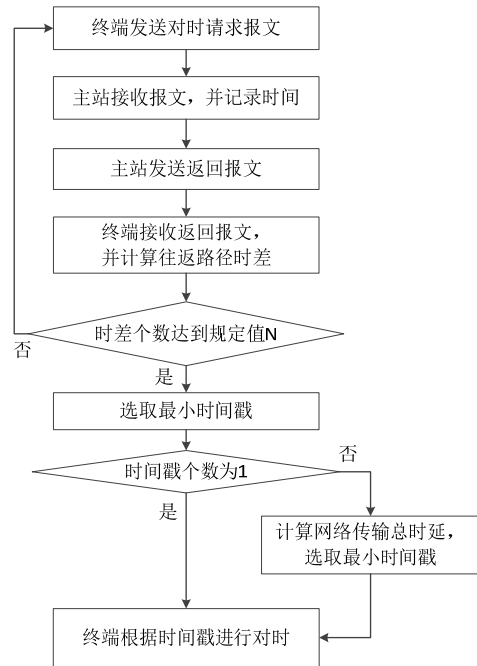


图 2 时钟同步流程

搭建“主站-载波”对时架构，其中主站带有时间服务器，交换机，路由器及防火墙，采集终端通过 GPRS 与主站进行信息的交互交互，流程如图 2 所

示。

(1) 终端根据设置的同步周期和时间定期向主站发送符合NTP协议的对时请求报文，并记录此报文的发送的时间为 $T_1$ 。

(2) 主站接收到采集终端的请求报文，并记录下接收报文的时间 $T_2$ ，向采集终端发送带有时间 $T_2$ 和当前发送时间 $T_3$ 的返回报文；

(3) 采集终端接收到主站的返回报文后，及其接收的时间 $T_4$ ，并根据NTP协议解析出 $T_2$ 和 $T_3$ ，根据公式(7)计算出往返路径的时间差；

(4) 重复前三步 $N$ 次，得到一组往返路径的时间差；

(5) 在这组往返路径时间差中找出最小值，并记录最小值的时间戳( $T_1, T_2, T_3, T_4$ )；

(6) 若当前只有一个时间戳直接根据公式(5)计算出时间偏移；若当前有多组时间戳时需根据公式(10)计算出每个时间戳的网络传输时延，并选取传输时延最小的时间戳；

(7) 采集终端即可根据时间戳计算出的时间偏移对自身时间进行同步调整；

(8) 采集终端定时抄读电能表的时钟，将电表的时钟与采集终端的时钟做比较，如果大于设定的阈值，采集终端生成一个时钟异常的事件，并将该事件上报给主站；

(9) 主站对生成时钟异常事件对应的电表，采用NTP-MRTTD方法，进行时钟同步，同步方式与主站对终端同步方式类似；主站对电表的操作经过采集终端，采集终端采用透明转发的方式，不对时钟同步报文做任何处理。

实验中一组往返路径时差的个数 $N=8$ ，共进行1600次的同步实验，并计算出每次的平均同步误差(Average Error Of Offset, AEOF)。

$$AEOF = \frac{\sum_{i=1}^M (T_i' - T_i'')}{M} \quad (11)$$

式中 $M$ 为对时次数，取值范围1~1600， $T_i'$ 为 $M$ 次对时中第 $i$ 次同步后客户端的时间， $T_i''$ 为 $M$ 次对时中第 $i$ 次同步后服务器的时间。

图3所示NTP-MRTTD对时的精度明显好于传统NTP对时方式，其精度提高了大概一倍，从精度收敛的角度来看NTP-MRTTD较快，NTP-MRTTD大约100次左右就已经收敛了，传统NTP是200次左右才开始收敛。

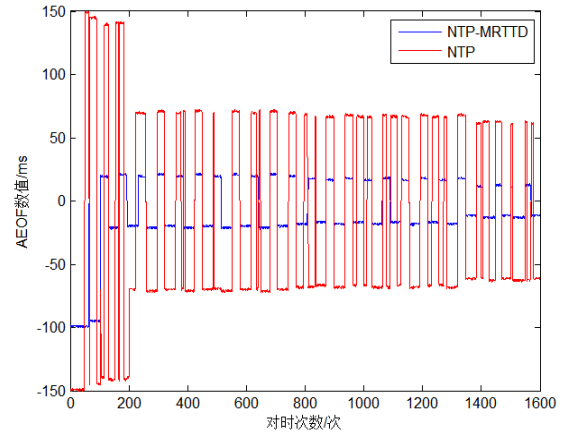


图3 同步误差分布图

## 5 结论

本文结合当前网络的特性，提出了一种基于NTP的MRTTD时钟同步方法，改进了电力用电信息采集系统中对时精度不足的缺陷。实验表明新的算法与传统NTP对时在精度和收敛度上都有极大的提高，在电力用电信息采集系统对时钟要求越来越严格的情况，该方法值得推广和应用。

### 参考文献：

- [1] 任工昌,王党席.基于GPRS和PLC的远程抄表系统可靠性设计与研究[J].机械设计与造, 2009(12):46-48.
- [2] 梁志光. 面向电力系统同步相量测量的时钟同步技术的研究与实现[D].成都:电子科技大学, 2009.
- [3] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems[J].IEEE 3 Park Avenue.2008.6
- [4] Mills,D.L. Internet time synchronization: the network time protocol[J].IEEE Transactions on Communications, 1991, 39(10): 1482-1493.
- [5] Mills,D.L. Adaptive hybrid clock discipline algorithm for the network time protocol networking[J]. IEEE ACM Transactions on Networking, 1998, 6(5): 505-514.

### 作者简介：

李新家(1967-),男,江苏泰州人,高级工程师,从事计算机软件设计与信息化应用;

李平(1985-),男,江苏扬州人,工程师,从事计算机软件设计与信息化应用, E-mail: liping\_ft@js.sgcc.com.cn.