



(12)发明专利

(10)授权公告号 CN 106921456 B

(45)授权公告日 2018.06.19

(21)申请号 201510988867.7

(22)申请日 2015.12.24

(65)同一申请的已公布的文献号  
申请公布号 CN 106921456 A

(43)申请公布日 2017.07.04

(73)专利权人 中国科学院沈阳自动化研究所  
地址 110016 辽宁省沈阳市南塔街114号

(72)发明人 曾鹏 王照伟 李志博 李栋  
王金涛

(74)专利代理机构 沈阳科苑专利商标代理有限公司 21002

代理人 徐丽 周秀梅

(51) Int. Cl.

H04J 3/06(2006.01)

(56)对比文件

EP 2034629 A1, 2009.03.11,  
CN 104507156 A, 2015.04.08,  
CN 103138865 A, 2013.06.05,  
CN 104754722 A, 2015.07.01,

审查员 李泽昆

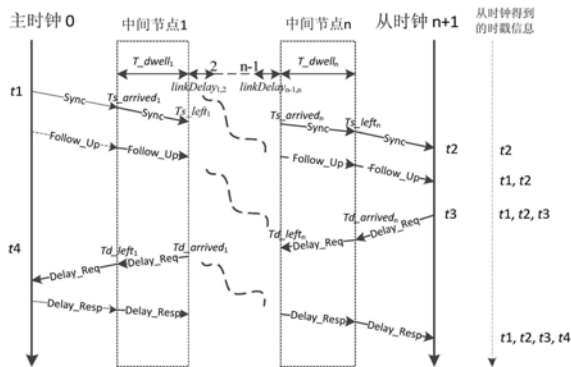
权利要求书2页 说明书5页 附图2页

(54)发明名称

基于PTP协议的多跳无线回程网络时间同步  
误差补偿方法

(57)摘要

本发明涉及无线网络通信技术,具体地说是一种基于PTP协议的多跳无线回程网络时间同步误差补偿方法。基于PTP协议,本发明利用中间节点统计收发PTP同步报文Sync和延时请求报文Delay\_Req的时间戳,并基于线性回归技术检测并补偿同步报文Sync和延时请求报文Delay\_Req在本地的转发时间以及节点间传输的链路时延,最终实现主从节点的无线链路非对称时延修正,完成时间同步误差补偿。本发明不修改现有的PTP协议,利用PTP报文的报头传输补偿时间、报文发送与到达时间以及时间修正值等额外时间信息,减少了报文开销,满足了时间同步误差补偿的实时性以及高精度要求,提高了现有的时间同步精度,实用性强。



1. 一种基于PTP协议的多跳无线回程网络时间同步误差补偿方法,其特征在于,包括以下步骤:

将工业现场的节点搭建成层次化的多跳网络结构;

把根节点作为主节点,叶子节点作为从节点,主从节点之间采用PTP协议进行时间同步,同步方式为two-step;

用于主节点和从节点之间数据转发的中间节点*i*在转发报文时,记录有关PTP报文的时间信息,并基于线性回归技术计算额外的时间信息;

中间节点*i*在发送跟随报文Follow\_Up、延时请求响应报文Delay\_Resp以及同步报文Sync时利用修改后的报头捎带额外的时间信息;

从节点收到PTP报文时根据PTP协议完成同步误差补偿;

所述two-step同步方式是指:利用跟随报文Follow\_Up传输同步报文Sync的时间信息,利用延时请求响应报文Delay\_Resp传输延时请求报文Delay\_Req的时间信息。

2. 根据权利要求1所述的基于PTP协议的多跳无线回程网络时间同步误差补偿方法,其特征在于,所述有关PTP报文的时间信息包括:同步报文Sync和延时请求报文Delay\_Req在中间节点*i*的到达时间Ts\_arrived<sub>*i*</sub>与Td\_arrived<sub>*i*</sub>,离开时间Ts\_left<sub>*i*</sub>与Td\_left<sub>*i*</sub>以及报文ID;从与同步报文Sync的ID一致的跟随报文Follow\_Up中获取同步报文Sync离开节点*i-1*的时间Ts\_left<sub>*i-1*</sub>;从与延时请求报文Delay\_Req的ID一致的延时请求响应报文Delay\_Resp中获取延时请求报文Delay\_Req到达节点*i-1*的时间Td\_arrived<sub>*i-1*</sub>;从同步报文Sync中获取的节点*i-1*时间修正值β<sub>*i-1*</sub>。

3. 根据权利要求1所述的基于PTP协议的多跳无线回程网络时间同步误差补偿方法,其特征在于,所述额外的时间信息包括:本地时间修正值β<sub>*i*</sub>、本地转发时间以及同步报文Sync和延时请求报文Delay\_Req在节点*i*和*i-1*之间传输的链路时延。

4. 根据权利要求1所述的基于PTP协议的多跳无线回程网络时间同步误差补偿方法,其特征在于,所述基于线性回归技术计算额外的时间信息,具体为:

中间节点*i*存储最新的*N*组时间信息(Ts\_arrived<sub>*i*</sub>,Ts\_arrived<sub>*i*</sub>-Ts\_left<sub>*i-1*</sub>),其中(Ts\_arrived<sub>*i,k*</sub>,Ts\_arrived<sub>*i,k*</sub>-Ts\_left<sub>*i-1,k*</sub>)为第*k*组信息。基于最小二乘法的线性回归,中间节点*i*得到节点*i*和*i-1*之间的时间偏移相对于到达时间Ts\_arrived<sub>*i*</sub>的斜率:

$$\alpha_{i-1,i} = \frac{\sum_{k=1}^N (\overline{offset_{i-1,i_k}} - \overline{offset_{i-1,i}}) (Ts\_arrived_{i,k} - \overline{Ts\_arrived_i})}{\sum_{i=1}^N (Ts\_arrived_{i,k} - \overline{Ts\_arrived_i})^2} \quad (3)$$

其中, $\overline{offset_{i-1,i_k}} = Ts\_arrived_{i,k} - Ts\_left_{i-1,k}$ , $\overline{offset_{i-1,i}} = \frac{1}{N} \sum_k \overline{offset_{i-1,i_k}}$ , $\overline{Ts\_arrived_i} = \frac{1}{N} \sum_k Ts\_arrived_{i,k}$ ,则节点*i*的时间修正值为:

$$\beta_i = (1 - \alpha_{i-1,i}) \beta_{i-1} \quad (4)$$

节点*i*和*i-1*之间的时间偏移为:

$$offset_{i-1,i_k} = \alpha_{i-1,i} \cdot (Ts\_arrived_{i,k} - \overline{Ts\_arrived_i}) + \overline{offset_{i-1,i}} \quad (5)$$

同步报文Sync和延时请求报文Delay\_Req的本地转发时间为:

$$T\_dwell_i = (T\_left_i - T\_arrived_i) \beta_i \quad (6)$$

其中,对于延时请求报文Delay\_Req而言, $T_{arrived_i}$ 为 $Td_{arrived_i}$ , $T_{left_i}$ 为 $Td_{left_i}$ ;对于同步报文Sync而言, $T_{arrived_i}$ 为 $Ts_{arrived_i}$ , $T_{left_i}$ 为 $Ts_{left_i}$ 。同步报文Sync在节点i和i-1之间传输的链路时延为:

$$\text{linkDelay}_{i-1,i} = (Ts_{arrived_i} - Ts_{left_{i-1}} - \text{offset}_{i-1,i}) \beta_i \quad (7)$$

延时请求报文Delay\_Req在节点i和i-1之间传输的链路时延为:

$$\text{linkDelay}_{i,i-1} = (Td_{arrived_{i-1}} - Td_{left_i} + \text{offset}_{i-1,i}) \beta_i \quad (8)$$

5. 根据权利要求1所述的基于PTP协议的多跳无线回程网络时间同步误差补偿方法,其特征在于,所述修改后的报头格式为:将报头4字节reserved域用来传输增加的时间信息的PTP通用报头格式。

6. 根据权利要求1所述的基于PTP协议的多跳无线回程网络时间同步误差补偿方法,其特征在于,所述捎带额外的时间信息,具体为:

中间节点i在发送跟随报文Follow\_Up、延时请求响应报文Delay\_Resp时,将与二者ID一致的同步报文Sync和延时请求报文Delay\_Req转发时间以及链路时延分别加到各自报头修正域correctionField中;将与二者ID一致的同步报文Sync离开时间 $Ts_{left_i}$ 和延时请求报文Delay\_Req到达时间 $Td_{arrived_i}$ 分别写入各自报头4字节的reserved域;

中间节点i在发送同步报文Sync时,将本地时间修正值写入报头4字节的reserved域。

7. 根据权利要求6所述的基于PTP协议的多跳无线回程网络时间同步误差补偿方法,其特征在于,所述报头修正域为:在two-step同步中,同步报文Sync和延时请求报文Delay\_Req的该域值为0,跟随报文Follow\_Up和延时请求响应报文Delay\_Resp的该域值分别为同步报文Sync和延时请求报文Delay\_Req在中间节点的转发时延以及链路时延。

8. 根据权利要求1所述的基于PTP协议的多跳无线回程网络时间同步误差补偿方法,其特征在于,所述从节点收到PTP报文时根据PTP协议完成同步误差补偿,具体为:

从节点根据同步报文Sync和延时请求报文Delay\_Req的发送和接受时间、以及与二者ID一致的跟随报文Follow\_Up和延时请求响应报文Delay\_Resp的correctionField域值计算链路平均时延meanPathDelay和时间偏移shift:

$$\text{meanPathDelay} = [(t_2 - t_3) + (t_4 - t_1) - \text{correctionField of Follow\_Up} - \text{correctionField of Delay\_Resp}] / 2 \quad (9)$$

$$\text{shift} = t_2 - t_1 - \text{meanPathDelay} - \text{correctionField of Follow\_Up} \quad (10)$$

完成时间同步误差补偿;

其中, $t_1$ 为主时钟记录发送同步报文Sync的时间, $t_2$ 为从时钟记录收到同步报文Sync的时间, $t_3$ 为从时钟记录发送延时请求报文Delay\_Req的时间, $t_4$ 为主时钟记录收到延时请求报文Delay\_Req的时间。

## 基于PTP协议的多跳无线回程网络时间同步误差补偿方法

### 技术领域

[0001] 本发明涉及无线网络通信领域,具体地说是一种基于PTP协议的多跳无线回程网络时间同步误差补偿方法。

### 背景技术

[0002] 多跳无线回程网络(Multi-hop Wireless Backhaul Networks, MWBNs)旨在提供高速、长距离、非视距以及低成本的无线接入服务,弥补了有线回程网络存在布线困难、成本高、网络覆盖面积受限以及拓扑结构灵活性差等问题的不足。由于工厂环境恶劣、布线困难,同时为了提高工厂生产效率,数字油田、智能电网广域互联、工业机器人协同操作等新型工业应用迫切需要构建管控一体化的无线回程网络,完成区域覆盖,实现测量与控制以及音视频等信息的混合传输。对于工业应用的新要求, MWBNs具有明显的优势。

[0003] 然而设备的控制信息对MWBNs的QoS往往有着严格的需求,如控制信号的可靠性要求100%,时延和抖动要求为ms级,即MWBNs需要满足控制信息传输的高实时和高可靠等要求。时间同步技术作为MWBNs的一个关键技术,在解决信息融合与协同处理、保证信息传输的实时性和可靠性等问题上至关重要。然而节点的本地时间是通过内部晶振的输出脉冲计数来完成,节点间需要交换时间信息来实现同步。因此,受内部晶振差异以及工厂环境下网络通信链路干扰的影响,节点间的时间很难实现精确同步。

[0004] IEEE 1588 v2(Precision Time Protocol, PTP)协议基于现有的网络通信协议,解决了工业有线以太网时间同步的问题,目前的同步精度可以达到50ns。但PTP协议实现高精度时间同步的关键是基于硬件时间戳和对称链路。所谓硬件时间戳,即在近物理层获取网络报文的收发时间戳。同时,为了进一步抵消链路非对称性的影响,PTP协议建立透传时钟的概念,有效保障了上下行通信链路的对称性,进而提高时间同步的精度。然而当前的Wi-Fi芯片多为单芯片结构,即Wi-Fi节点时间戳的选取只能在MAC层以上实现,MAC层排队、PHY层传输等转发时延误差以及时间戳的选取机制会严重影响链路的非对称性。现有方法多采用统计学原理进行链路时延补偿,增加了网络中的报文,同时实时性差。而且,实验发现将PTP协议直接应用到无线网络中的同步精度可达ms级。同时,开源的Wi-Fi芯片在目前的市场上无法获得,而基于FPGA设计Wi-Fi的MAC层所带来的开发成本太大,所以设计基于PTP协议的无线网络节点是不可取的。

### 发明内容

[0005] 针对现有技术中存在的上述不足之处,本发明要解决的技术问题是提供一种基于PTP协议的多跳无线回程网络时间同步误差补偿方法,在当前的PTP以太网设备的基础上,实现PTP设备的无线通信链路的时延补偿,进而实现高精度的时间同步。

[0006] 本发明为实现上述目的所采用的技术方案是:一种基于PTP协议的多跳无线回程网络时间同步误差补偿方法,包括以下步骤:

[0007] 将工业现场的节点搭建成层次化的多跳网络结构;

[0008] 把根节点作为主节点,叶子节点作为从节点,主从节点之间采用PTP协议进行时间同步,同步方式为two-step;

[0009] 用于主节点和从节点之间数据转发的中间节点*i*在转发报文时,记录有关PTP报文的时间信息,并基于线性回归技术计算额外的时间信息;

[0010] 中间节点*i*在发送跟随报文Follow\_Up、延时请求响应报文Delay\_Resp以及同步报文Sync时利用修改后的报头捎带额外的时间信息;

[0011] 从节点收到PTP报文时根据PTP协议完成同步误差补偿。

[0012] 所述two-step同步方式是指:利用跟随报文Follow\_Up传输同步报文Sync的时间信息,利用延时请求响应报文Delay\_Resp传输延时请求报文Delay\_Req的时间信息。

[0013] 所述有关PTP报文的时间信息包括:同步报文Sync和延时请求报文Delay\_Req在中间节点*i*的到达时间Ts\_arrived<sub>*i*</sub>与Td\_arrived<sub>*i*</sub>,离开时间Ts\_left<sub>*i*</sub>与Td\_left<sub>*i*</sub>以及报文ID;从与同步报文Sync的ID一致的跟随报文Follow\_Up中获取同步报文Sync离开节点*i-1*的时间Ts\_left<sub>*i-1*</sub>;从与延时请求报文Delay\_Req的ID一致的延时请求响应报文Delay\_Resp中获取延时请求报文Delay\_Req到达节点*i-1*的时间Td\_arrived<sub>*i-1*</sub>;从同步报文Sync中获取的节点*i-1*时间修正值β<sub>*i-1*</sub>。

[0014] 所述额外的时间信息包括:本地时间修正值β<sub>*i*</sub>、本地转发时间以及同步报文Sync和延时请求报文Delay\_Req在节点*i*和*i-1*之间传输的链路时延。

[0015] 所述基于线性回归技术计算额外的时间信息,具体为:

[0016] 中间节点*i*存储最新的*N*组时间信息(Ts\_arrived<sub>*i*</sub>,Ts\_arrived<sub>*i*</sub>-Ts\_left<sub>*i-1*</sub>),其中(Ts\_arrived<sub>*i,k*</sub>,Ts\_arrived<sub>*i,k*</sub>-Ts\_left<sub>*i-1,k*</sub>)为第*k*组信息。基于最小二乘法的线性回归,中间节点*i*得到节点*i*和*i-1*之间的时间偏移相对于到达时间Ts\_arrived<sub>*i*</sub>的斜率:

$$[0017] \quad \alpha_{i-1,i} = \frac{\sum_{k=1}^N (\widehat{offset}_{i-1,i_k} - \overline{\widehat{offset}_{i-1,i}}) (Ts\_arrived_{i,k} - \overline{Ts\_arrived_i})}{\sum_{i=1}^N (Ts\_arrived_{i,k} - \overline{Ts\_arrived_i})^2} \quad (3)$$

[0018] 其中,  $\widehat{offset}_{i-1,i_k} = Ts\_arrived_{i,k} - Ts\_left_{i-1,k}$ ,  $\overline{\widehat{offset}_{i-1,i}} = \frac{1}{N} \sum_k^N \widehat{offset}_{i-1,i_k}$ ,

$\overline{Ts\_arrived_i} = \frac{1}{N} \sum_k^N Ts\_arrived_{i,k}$ , 则节点*i*的时间修正值为:

$$[0019] \quad \beta_i = (1 - \alpha_{i-1,i}) \beta_{i-1} \quad (4)$$

[0020] 节点*i*和*i-1*之间的时间偏移为:

[0021]

$$offset_{i-1,i_k} = \alpha_{i-1,i} \cdot (Ts\_arrived_{i,k} - \overline{Ts\_arrived_i}) + \overline{\widehat{offset}_{i-1,i}} \quad (5)$$

[0022] 同步报文Sync和延时请求报文Delay\_Req的本地转发时间为:

$$[0023] \quad T\_dwell_i = (T\_left_i - T\_arrived_i) \beta_i \quad (6)$$

[0024] 其中,对于延时请求报文Delay\_Req而言,T\_arrived<sub>*i*</sub>为Td\_arrived<sub>*i*</sub>,T\_left<sub>*i*</sub>为Td\_left<sub>*i*</sub>;对于同步报文Sync而言,T\_arrived<sub>*i*</sub>为Ts\_arrived<sub>*i*</sub>,T\_left<sub>*i*</sub>为Ts\_left<sub>*i*</sub>。同步报文Sync在节点*i*和*i-1*之间传输的链路时延为:

$$[0025] \quad linkDelay_{i-1,i} = (Ts\_arrived_i - Ts\_left_{i-1} - offset_{i-1,i}) \beta_i \quad (7)$$

[0026] 延时请求报文Delay\_Req在节点*i*和*i-1*之间传输的链路时延为:

[0027]  $\text{linkDelay}_{i,i-1} = (\text{Td\_arrived}_{i-1} - \text{Td\_left}_i + \text{offset}_{i-1,i}) \beta_i$  (8)

[0028] 所述修改后的报头格式为:将报头4字节reserved域用来传输增加的时间信息的PTP通用报头格式。

[0029] 所述捎带额外的时间信息,具体为:

[0030] 中间节点*i*在发送跟随报文Follow\_Up、延时请求响应报文Delay\_Resp时,将与二者ID一致的同步报文Sync和延时请求报文Delay\_Req转发时间以及链路时延分别加到各自报头修正域correctionField中;将与二者ID一致的同步报文Sync离开时间Ts\_left*i*和延时请求报文Delay\_Req到达时间Td\_arrived*i*分别写入各自报头4字节的reserved域;

[0031] 中间节点*i*在发送同步报文Sync时,将本地时间修正值写入报头4字节的reserved域。

[0032] 所述报头修正域为:在two-step同步中,同步报文Sync和延时请求报文Delay\_Req的该域值为0,跟随报文Follow\_Up和延时请求响应报文Delay\_Resp的该域值分别为同步报文Sync和延时请求报文Delay\_Req在中间节点的转发时延以及链路时延。

[0033] 所述从节点收到PTP报文时根据PTP协议完成同步误差补偿,具体为:

[0034] 从节点根据同步报文Sync和延时请求报文Delay\_Req的发送和接受时间、以及与二者ID一致的跟随报文Follow\_Up和延时请求响应报文Delay\_Resp的correctionField域值计算链路平均时延meanPathDelay和时间偏移shift:

[0035]  $\text{meanPathDelay} = [(\text{t}_2 - \text{t}_3) + (\text{t}_4 - \text{t}_1) - \text{correctionField of Follow\_Up} - \text{correctionField of Delay\_Resp}] / 2$  (9)

[0036]  $\text{shift} = \text{t}_2 - \text{t}_1 - \text{meanPathDelay} - \text{correctionField of Follow\_Up}$  (10)

[0037] 完成时间同步误差补偿;

[0038] 其中,t1为主时钟记录发送同步报文Sync的时间,t2为从时钟记录收到同步报文Sync的时间,t3为从时钟记录发送延时请求报文Delay\_Req的时间,t4为主时钟记录收到延时请求报文Delay\_Req的时间。

[0039] 本发明提出的基于PTP协议的多跳无线回程网络时间同步误差补偿方法,充分考虑现有设备的局限性以及无线链路的非对称性问题,在不影响现有PTP协议的基础上,减少报文开销,实现链路时延的实时修正,完成时间同步误差补偿。具体表现在:

[0040] 1. 本发明基于现有的PTP协议和设备,降低了开发成本;

[0041] 2. 本发明利用PTP报文报头修正域correctionField和预留域reserved传输额外的时间信息,减少了报文开销;

[0042] 3. 本发明利用中间节点实现PTP报文的转发时间以及链路时延的补偿,满足了链路时延修正的实时性以及同步误差补偿的高精度要求,提高了现有的时间同步精度,实用性强。

## 附图说明

[0043] 图1为本发明的工业无线网络示意图;

[0044] 图2为PTP协议下two-step同步原理;

[0045] 图3为PTP协议下中间节点完成链路时延补偿示意图;

[0046] 图4为本发明采用的修改后的PTP报文报头。

## 具体实施方式

[0047] 下面结合附图及实施例对本发明做进一步的详细说明。

[0048] 本发明方法包括以下步骤：将现场的节点搭建成层次化的多跳网络结构；采用PTP协议下的two-step同步方式；利用中间节点完成PTP同步报文Sync和延时请求报文Delay\_Req的转发时间以及链路时延检测与补偿，实现时间同步误差补偿。

[0049] 如图1所示，本发明搭建的层次化多跳无线网络结构。将网络中的根节点作为时间源，即主节点；叶子节点作为从节点，通过多跳的方式实现与主节点的通信，并作为接入节点负责接入网的信息融合。而中间节点作为转发节点实现上下行链路数据包的转发，并完成报文转发时延以及链路时延补偿。

[0050] 如图2所示，为所述的PTP协议下的two-step同步原理。主时钟周期性发送包含发送时间t1的同步报文Sync；从时钟记录收到同步报文Sync的时间t2，然后发送延时请求报文Delay\_Req，并记录发送时间t3；主时钟记录收到延时请求报文Delay\_Req的时间t4，并发送包含t4的延时请求响应报文Delay\_Resp给从时钟。从时钟根据获得的4个时间值t1、t2、t3、t4来计算其与主时钟之间的链路平均时延meanPathDelay和时间偏移：

$$[0051] \quad \text{meanPathDelay} = [(t2-t1) + (t4-t3)]/2 \quad (1)$$

$$[0052] \quad \text{shift} = [(t2-t1) - (t4-t3)]/2 \quad (2)$$

[0053] 显然，meanPathDelay的计算是建立在上下行链路对称的基础上，即只有上下行链路对称的条件成立，根据式(1)计算得到的链路时延才是精确的链路时延。但在实际的网络环境中，尤其是在多跳无线网络环境中，该条件并不满足，即由(1)和(2)得到的主从节点链路时延和时间偏移存在较大误差，造成时间同步精度严重下降，因此需要对时间同步误差进行实时补偿。

[0054] 网络节点的时间都是通过对本地晶振输出脉冲计数获得，然而在一定时间范围内节点时钟晶振的频率是稳定的。同时，节点时间T和绝对时间t之间是线性关系，即 $T = \alpha \cdot t + \beta$ 。所以，节点间的时间偏移与时间也成线性关系。

[0055] 所述基于PTP协议的多跳无线回程网络时间同步误差补偿方法，通过以下描述的方法实现，如图3所示。

[0056] (1) 中间节点i记录同步报文Sync和延时请求报文Delay\_Req的到达时间Ts\_arrived<sub>i</sub>与Td\_arrived<sub>i</sub>，离开时间Ts\_left<sub>i</sub>与Td\_left<sub>i</sub>以及报文ID；

[0057] (2) 中间节点i根据同步报文Sync在节点i-1的发送时间Ts\_left<sub>i-1</sub>和本地到达时间Ts\_arrived<sub>i</sub>，存储最新的N组时间信息(Ts\_arrived<sub>i</sub>, Ts\_arrived<sub>i</sub>-Ts\_left<sub>i-1</sub>)，其中第k组信息为(Ts\_arrived<sub>i,k</sub>, Ts\_arrived<sub>i,k</sub>-Ts\_left<sub>i-1,k</sub>)，并基于最小二乘法的线性回归技术，得到节点i和i-1之间的时间偏移相对于到达时间Ts\_arrived<sub>i</sub>的斜率：

$$[0058] \quad \alpha_{i-1,i} = \frac{\sum_{k=1}^N (\overline{\text{offset}_{i-1,i,k}} - \overline{\text{offset}_{i-1,i}}) (\overline{Ts\_arrived_{i,k}} - \overline{Ts\_arrived_i})}{\sum_{k=1}^N (\overline{Ts\_arrived_{i,k}} - \overline{Ts\_arrived_i})^2} \quad (3)$$

[0059] 其中， $\overline{\text{offset}_{i-1,i,k}} = Ts\_arrived_{i,k} - Ts\_left_{i-1,k}$ ， $\overline{\text{offset}_{i-1,i}} = \frac{1}{N} \sum_{k=1}^N \overline{\text{offset}_{i-1,i,k}}$ ，

$\overline{Ts\_arrived_i} = \frac{1}{N} \sum_{k=1}^N \overline{Ts\_arrived_{i,k}}$ ，则节点i的时间修正值为：

[0060]  $\beta_i = (1 - \alpha_{i-1,i}) \beta_{i-1}$  (4)

[0061] 节点*i*和*i-1*之间的时间偏移为:

[0062]

$$\text{offset}_{i-1,i,k} = \alpha_{i-1,i} \cdot (\overline{Ts\_arrived_{i,k}} - \overline{Ts\_arrived_i}) + \overline{\text{offset}_{i-1,i}} \quad (5)$$

[0063] 同步报文Sync和延时请求报文Delay\_Req的本地转发时间为:

$$T_{\text{dwell}_i} = (T_{\text{left}_i} - T_{\text{arrived}_i}) \beta_i \quad (6)$$

[0065] 其中,对于延时请求报文Delay\_Req而言, $T_{\text{arrived}_i}$ 为 $T_{\text{d\_arrived}_i}$ , $T_{\text{left}_i}$ 为 $T_{\text{d\_left}_i}$ ;对于同步报文Sync而言, $T_{\text{arrived}_i}$ 为 $T_{\text{s\_arrived}_i}$ , $T_{\text{left}_i}$ 为 $T_{\text{s\_left}_i}$ 。同步报文Sync在节点*i*和*i-1*之间传输的链路时延为:

$$\text{linkDelay}_{i-1,i} = (T_{\text{s\_arrived}_i} - T_{\text{s\_left}_{i-1}} - \text{offset}_{i-1,i}) \beta_i \quad (7)$$

[0067] 延时请求报文Delay\_Req在节点*i*和*i-1*之间传输的链路时延为:

$$\text{linkDelay}_{i,i-1} = (T_{\text{d\_arrived}_{i-1}} - T_{\text{d\_left}_i} + \text{offset}_{i-1,i}) \beta_i \quad (8)$$

[0069] (3) 中间节点*i*在收到跟随报文Follow\_Up和延时请求响应报文Delay\_Resp时,将与二者ID一致的报文转发时间以及链路时延加入到跟随报文Follow\_Up和延时请求响应报文Delay\_Resp的报头修正域correctionField中;在发送跟随报文Follow\_Up、延时请求响应报文Delay\_Resp以及同步报文Sync时采用修改后的报头分别捎带与报文ID一致的同步报文Sync发送时间 $T_{\text{s\_left}_i}$ 、延时请求报文Delay\_Req到达时间 $T_{\text{d\_arrived}_i}$ 以及时间修正值 $\beta_i$ ;

[0070] (4) 从节点根据同步报文Sync和延时请求报文Delay\_Req的发送和接受时间、以及与二者ID一致的跟随报文Follow\_Up和延时请求响应报文Delay\_Resp的correctionField域值计算链路平均时延meanPathDelay和时间偏移shift:

$$\text{meanPathDelay} = [(t_2 - t_3) + (t_4 - t_1) - \text{correctionField of Follow\_Up} - \text{correctionField of Delay\_Resp}] / 2 \quad (9)$$

$$\text{shift} = t_2 - t_1 - \text{meanPathDelay} - \text{correctionField of Follow\_Up} \quad (10)$$

完成时间同步误差补偿。

[0073] 如图4所示,所述报文修正域是指PTP报文的报头correctionField域,在two-step时钟中,同步报文Sync和延时请求报文Delay\_Req的该域值为0,跟随报文Follow\_Up和延时请求响应报文Delay\_Resp的该域值分别为同步报文Sync和延时请求报文Delay\_Req在中间节点的转发时延和链路时延。所述利用修改后的报头传输增加的时间信息,即采用跟随报文Follow\_Up、延时请求响应报文Delay\_Resp以及同步报文Sync的报头4字节reserved域传递与报文ID一致的同步报文Sync发送时间 $T_{\text{s\_left}_i}$ 、延时请求报文Delay\_Req到达时间 $T_{\text{d\_arrived}_i}$ 以及时间修正值 $\beta_i$ 等增加的时间信息。同时,通过检查messageType域检测报文类型,Sync、Delay\_Req、Follow\_Up、Delay\_Resp等报文的该域值分别为0、1、8、9。



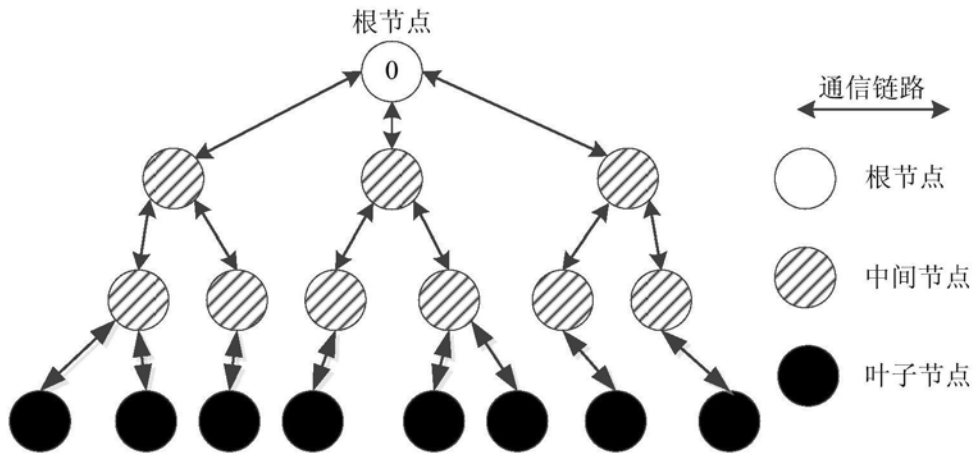


图1

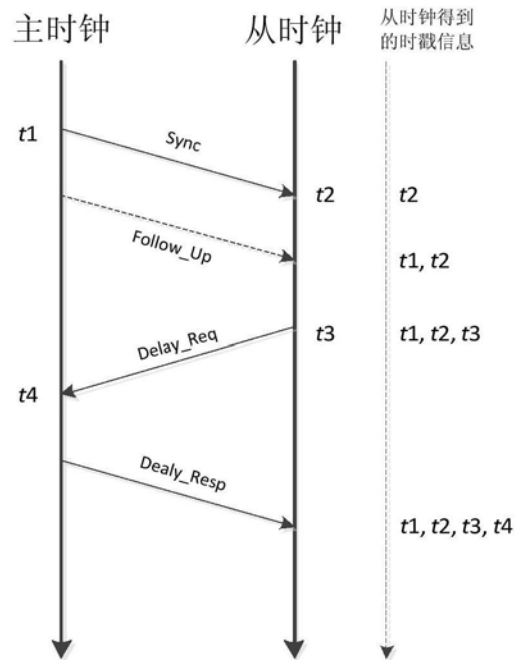


图2

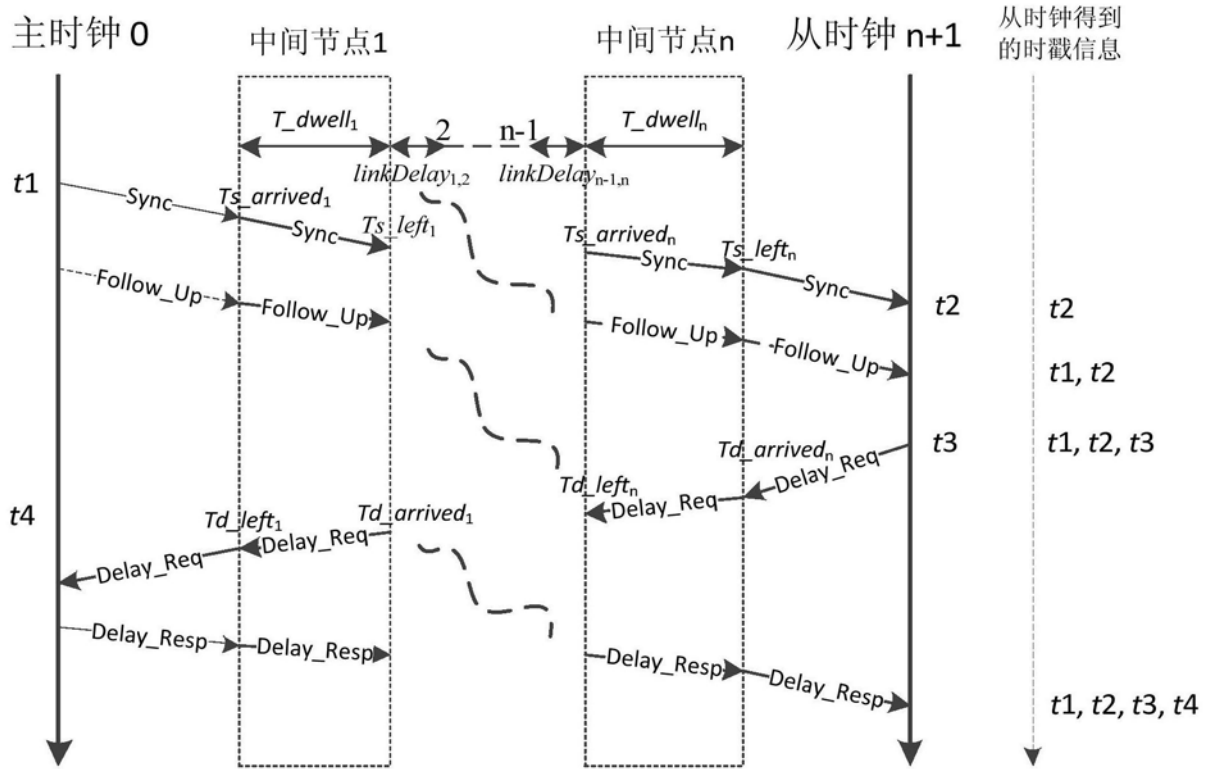


图3

Bits							Octets	Offset
7	6	5	4	3	2	1		
transportSpecific				<b>messageType</b>			1	0
reserved				versionPTP			1	1
messageLength							2	2
domainNumber							1	4
reserved							1	5
flagField							2	6
<b>correctionField</b>							8	8
<b>added time information</b>							4	16
sourcePortIdentity							10	20
sequenceID							2	30
controlField							1	32
logMessageInterval							1	33

图4