

(12) 发明专利

(10) 授权公告号 CN 101449520 B

(45) 授权公告日 2012. 05. 30

(21) 申请号 200780018257. 8

(22) 申请日 2007. 05. 16

(30) 优先权数据

60/802, 023 2006. 05. 19 US

(85) PCT申请进入国家阶段日

2008. 11. 19

(86) PCT申请的申请数据

PCT/US2007/069066 2007. 05. 16

(87) PCT申请的公布数据

W02007/137085 EN 2007. 11. 29

(73) 专利权人 讯腾有限公司

地址 美国加利福尼亚州

(72) 发明人 格雷戈里·路易斯·多德

(74) 专利代理机构 北京东方亿思知识产权代理

有限责任公司 11258

代理人 李晓冬 南霆

(51) Int. Cl.

H04L 12/28(2006. 01)

H04J 3/06(2006. 01)

(56) 对比文件

US 2004/0062278 A1, 2004. 04. 01, 全文.

US 20050125150 A1, 2005. 06. 09, 全文.

CN 1487752 A, 2004. 04. 07, 全文.

审查员 李普昕

权利要求书 4 页 说明书 7 页 附图 6 页

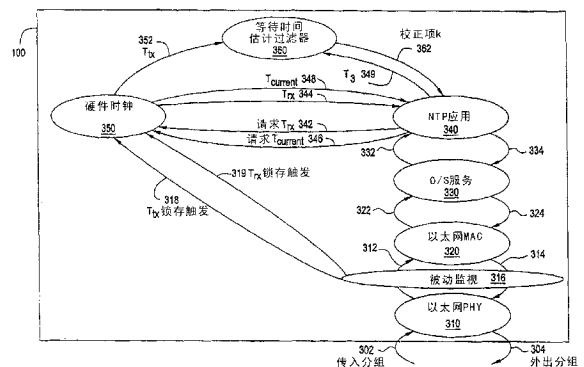
(54) 发明名称

网络时间协议精确时间戳标记服务

(57) 摘要

本发明的实施例提出了用于减小 NTP 服务器所创建的接收和发送时间戳中的不确定性的方法和系统。接收时间戳中的不确定性是通过如下方式来消除的：在传入分组可能通过穿越时间戳标记系统中的各个软件层被延迟之前，将到达时间记录在 NTP 服务器的硬件时钟中。发送时间戳中的不确定性是通过如下方式来消除的：使用对由等待时间估计过滤器计算出的发送等待时间的估计向外出分组提供将来的时间戳。随后，实际离开时间被用于重新计算并更新对发送等待时间的估计。以这种方式，对时间戳标记功能更好控制可以按照保持与当前 NTP 标准之间的相互兼容性的方式被实现在现有 NTP 服务器中。

CN 101449520 B



1. 一种用于建立将被网络时间协议 NTP 服务器插入到外出分组中的发送时间戳的方法,该方法包括:

初始化等待时间估计过滤器 LEF;

创建所述外出分组;

向所述 NTP 服务器中的硬件时钟请求当前时间;

向所述 LEF 请求校正项,其中所述校正项是对与所述外出分组在跨越与外部环境之间的边界之前穿过各个软件层相关联的发送等待时间的估计;

将所述当前时间和所述校正项之和插入到所述外出分组中,作为所述发送时间戳;

将所述发送时间戳提供到所述 LEF;

将带有所述发送时间戳的所述外出分组放置到出站队列中;

监视流过介质无关接口 MII 信道的网络流量以检测所述外出分组;

触发所述硬件时钟来锁存检测到的所述外出分组的离开时间;

将所锁存的离开时间提供到所述 LEF;

执行 LEF 算法的步骤,以通过计算所述发送时间戳和所锁存的离开时间之间的偏移量来更新所述校正项,以在下一外出分组中获得更精确的发送时间戳;以及

存储所更新的校正项并使用所更新的校正项将所述发送时间戳插入到下一外出分组中。

2. 如权利要求 1 所述的方法,其中所述 LEF 或者通过将发送等待时间阵列中的元素设置为 0,或者通过将所述发送等待时间阵列中的元素设置为预定值来初始化。

3. 如权利要求 1 所述的方法,其中所述外出分组被创建为对来自 NTP 客户端的传入分组的回复。

4. 如权利要求 1 所述的方法,其中所述 LEF 算法包括以下步骤:

步骤 1:将发送等待时间阵列中的元素向下移动 1:

$$\mu_{i-1} = \mu_i;$$

步骤 2:更新最近的发送等待时间:

$$\mu_i = T_{tx} - T_3 + k;$$

步骤 3:计算所述校正项的新值:

$$k = \min(\mu_{i-1}, \mu_i),$$

其中  $\mu_{i-1}$  和  $\mu_i$  是所述发送等待时间阵列中的元素,  $i$  表示元素的编号,并且

其中  $T_{tx}$  表示锁存的离开时间,  $T_3$  表示所述发送时间戳,并且  $k$  表示所述校正项。

5. 如权利要求 1 所述的方法,还包括用于建立将被所述 NTP 服务器插入到传入分组中的接收时间戳的步骤,所述步骤包括:

监视流过所述 MII 信道的网络流量,以检测来自 NTP 客户端的传入分组;

触发所述硬件时钟来锁存所检测到的传入分组的到达时间;以及

将所锁存的到达时间插入到所述传入分组中,作为所述接收时间戳。

6. 一种精确时间戳标记服务 PTS 系统,该系统被配置用于将发送时间戳插入到外出分组中,该系统包括:

与以太网 PHY 相关联的物理层;

以太网媒体访问控制器 MAC;

- 连接所述以太网 PHY 和所述以太网 MAC 的介质无关接口 MII 信道；  
 操作系统 O/S 服务；  
 硬件时钟，该时钟被配置用于提供当前时间和锁存所述外出分组的离开时间；  
 等待时间估计过滤器 LEF，其被配置为被初始化并且执行 LEF 算法的步骤，以通过计算所述发送时间戳和由所述硬件时钟锁存的离开时间之间的偏移量来存储和更新校正项，并且在下一外出分组中获得更精确的发送时间戳；以及  
 网络时间协议 NTP 应用，其被配置用于：  
 创建所述外出分组，  
 向硬件时钟请求当前时间，  
 向所述 LEF 请求所述校正项，其中所述校正项是对与所述外出分组在跨越所述物理层与外部环境之间的边界之前穿过各个软件层相关联的发送等待时间的估计，  
 将所述当前时间和所述校正项之和插入到所述外出分组中，作为所述发送时间戳，  
 将所述发送时间戳提供到所述 LEF，  
 将带有所述发送时间戳的所述外出分组放置到出站队列中，被动监视单元，其被配置用于：  
 监视流过所述 MII 信道的网络流量以检测所述外出分组，  
 触发所述硬件时钟来锁存检测到的所述外出分组的离开时间，以及  
 将所锁存的离开时间提供到所述 LEF。
7. 如权利要求 6 所述的系统，其中所述 LEF 或者通过将发送等待时间阵列中的元素设置为 0，或者通过将所述发送等待时间阵列中的元素设置为预定值来初始化。
8. 如权利要求 6 所述的系统，其中所述外出分组被创建为对来自 NTP 客户端的传入分组的回复。
9. 如权利要求 6 所述的系统，其中所述 LEF 算法包括以下步骤：  
 步骤 1：将发送等待时间阵列中的元素向下移动 1：  

$$\mu_{i-1} = \mu_i;$$
 步骤 2：更新最近的发送等待时间：  

$$\mu_i = T_{tx} - T_3 + k;$$
 步骤 3：计算所述校正项的新值：  

$$k = \min(\mu_{i-1}, \mu_i),$$
 其中  $\mu_{i-1}$  和  $\mu_i$  是所述发送等待时间阵列中的元素，i 表示元素的编号，并且其中  $T_{tx}$  表示锁存的离开时间， $T_3$  表示所述发送时间戳，并且 k 表示所述校正项。
10. 如权利要求 6 所述的系统，还包括用于建立将被所述 PTS 系统插入到传入分组中的接收时间戳的步骤，所述步骤包括：  
 监视流过所述 MII 信道的网络流量，以检测来自 NTP 客户端的传入分组；  
 触发所述硬件时钟来锁存所检测到的传入分组的到达时间；以及  
 将所锁存的到达时间插入到所述传入分组中，作为所述接收时间戳。
11. 一种精确网络时间协议 NTP 服务器，被配置用于将发送时间戳插入到外出分组中，所述服务器包括：  
 中央处理单元 CPU；

网络接口 ; 以及

具有硬件时钟的精确时间戳标记服务 PTS 系统, 所述硬件时钟被配置用于提供当前时间和锁存所述外出分组的离开时间, 其中所述 PTS 系统包括 :

与以太网 PHY 相关联的物理层 ;

以太网媒体访问控制器 MAC ;

连接所述以太网 PHY 和所述以太网 MAC 的介质无关接口 MII 信道 ;

操作系统 O/S 服务 ;

等待时间估计过滤器 LEF, 其被配置为被初始化并且执行 LEF 算法的步骤, 以通过计算所述发送时间戳和由所述硬件时钟锁存的离开时间之间的偏移量来更新校正项, 并且在下一外出分组中获得更精确的发送时间戳 ; 以及

网络时间协议 NTP 应用, 其被配置用于 :

创建所述外出分组,

向硬件时钟请求当前时间,

向所述 LEF 请求所述校正项, 其中所述校正项是对与所述外出分组在跨越所述物理层与外部环境之间的边界之前穿过各个软件层相关联的发送等待时间的估计,

将所述当前时间和所述校正项之和插入到所述外出分组中, 作为所述发送时间戳,

将所述发送时间戳提供到所述 LEF,

将带有所述发送时间戳的所述外出分组放置到出站队列中, 被动监视单元, 其被配置用于 :

监视流过所述 MII 信道的网络流量以检测所述外出分组, 触发所述硬件时钟来锁存检测到的所述外出分组的离开时间, 以及

将所锁存的离开时间提供到所述 LEF。

12. 如权利要求 11 所述的服务器, 其中所述 LEF 或者通过将发送等待时间阵列中的元素设置为 0, 或者通过将所述发送等待时间阵列中的元素设置为预定值来初始化。

13. 如权利要求 11 所述的服务器, 其中所述外出分组被创建为对来自 NTP 客户端的传入分组的回复。

14. 如权利要求 11 所述的服务器, 其中所述 LEF 算法包括以下步骤 :

步骤 1 : 将发送等待时间阵列中的元素向下移动 1 :

$$\mu_{i-1} = \mu_i ;$$

步骤 2 : 更新最近的发送等待时间 :

$$\mu_i = T_{tx} - T_3 + k ;$$

步骤 3 : 计算所述校正项的新值 :

$$k = \min(\mu_{i-1}, \mu_i),$$

其中  $\mu_{i-1}$  和  $\mu_i$  是所述发送等待时间阵列中的元素,  $i$  表示元素的编号, 并且

其中  $T_{tx}$  表示锁存的离开时间,  $T_3$  表示所述发送时间戳, 并且  $k$  表示所述校正项。

15. 如权利要求 11 所述的服务器, 还包括用于建立将被所述 PTS 系统插入到传入分组中的接收时间戳的步骤, 所述步骤包括 :

监视流过所述 MII 信道的网络流量, 以检测来自 NTP 客户端的传入分组 ;

触发所述硬件时钟来锁存所检测到的传入分组的到达时间 ; 以及

将所锁存的到达时间插入到所述传入分组中,作为所述接收时间戳。

## 网络时间协议精确时间戳标记服务

### 技术领域

[0001] 本发明的实施例一般地涉及提高数据网络上的发射器和接收器路径之间的时间偏移精确性,更具体而言,涉及网络时间协议精确时间戳标记服务 (precision timestamping service)。

### 背景技术

[0002] 在很多通信应用中,网络中的每个元件具有其自己的内部时钟,该时钟的运行独立于网络中的其他时钟。当网络是同步网络时,例如在很多电信应用中以及在高速广域网中,这些时钟必须与服务器同步。在遗留电信网络中,网络元件使用诸如 T1 和 SONET 之类的“TDM”链路,这些“TDM”链路本身能够在物理层上从服务器向客户端运载定时信息。但是,一般认为下一代网络将基于分组交换基础构架,因此,将使用基于分组的用于传输定时信息的方法。用于同步分组交换数据网络中的时钟的方法之一是网络时间协议 (NTP)。

[0003] 商用 NTP 服务器通常采用高度精确的基于硬件的时钟,所述时钟遵循外部标准。进而,NTP 客户端将手工分组小心地发送到 NTP 服务器并分析服务器的回复,以便确定客户端时钟相对于服务器时钟的偏移量。典型的分组包含四个时间戳。时间戳被设计用于精确计时客户端 / 服务器时间分组交换的发送和接收路径,以便使得可以计算出端点之间的往返延迟和客户端时钟的偏移量。但是,虽然 NTP 服务器处基于硬件的时钟高度精确,但是如果在 NTP 服务器中存在任何与时间戳标记功能相关联的不精确性,NTP 服务器则会在提供给客户端的时间戳中引入误差。随后,这些误差传播到客户端时钟并且损害客户端时钟的准确性。

[0004] NTP 服务器所创建的时间戳中的大多数不精确性是由各个软件层引入的,包括设备驱动器、网络栈和越来越多的非实时操作系统的使用。这在图 1 中被示出,该图概念性地图示出根据现有技术的典型 NTP 系统 100 中的分组时间戳标记流。大多数现有的 NTP 服务器包括诸如 NTP 系统 100 之类的系统。通常,分组中的时间戳被看作反映出涉及物理层接口的到达或离开的时刻。参考图 1,物理层与以太网 PHY110 相关联。在此情况下,到达时间指的是每个传入分组的第一 (或指定) 比特跨越外部环境 (未示出) 和以太网 PHY110 之间的边界的时刻。同样,离开时间指的是每个外出分组的第一 (或指定) 比特跨越以太网 PHY110 和外部环境之间的所述边界的时刻。

[0005] 如图 1 所示,当传入分组 102 到达 NTP 系统 100 时,其穿过如下的若干实体:与以太网 PHY110 相关联的物理层、以太网媒体访问控制器 (MAC) 120、操作系统 (O/S) 服务 130 和 NTP 应用 140。这些实体中的一些是物理的 (例如以太网 MAC120),一些是逻辑的 (例如 O/S 服务 130)。传入分组 102 的路径由箭头 112、122 和 132 示出。如箭头 142 所示,NTP 应用 140 仅在传入分组 102 跨越以太网 PHY110、以太网 MAC120 和 O/S 服务 130 并实际到达 NTP 应用 140 之后,向硬件时钟 150 请求时间戳以便记录到传入分组 102 中 (这种时间戳在这里被称为“接收时间戳”)。在接收到来自 NTP 应用 140 的请求之后,硬件时钟 150 将当前时间值作为接收时间戳发送,如时间戳 144 所示。因此,在 NTP 系统 100 处的实际到达时

间与 NTP 应用 140 实际读取硬件时钟 150 以确认接收时间戳的的时间之间存在一延迟, 该延迟带有极大的随机分量。

[0006] 对于离开 NTP 系统 100 的外出分组 104 会发生类似状况。NTP 应用 140 从硬件时钟 150 读取当前时间值, 并将该值作为时间戳插入到外出分组 104 的适当字段中 (这种时间戳在这里被称为“发送时间戳”)。在插入该发送时间戳之后, NTP 应用 140 在箭头 134、124 和 114 所示的路径上发送外出分组 104。同样, 由于外出分组 104 必须首先穿过 O/S 服务 130、以太网 MAC120 和以太网 PHY110, 因此在从硬件时钟 150 获得发送时间戳的时间与从 NTP 系统 100 的实际离开时间之间存在一延迟, 该延迟带有极大的随机分量。因此, 不精确的接收和发送时间戳从 NTP 系统 100 传播到客户端时钟, 并对客户端时钟处的时间保持和速率保持设置了基线误差。

[0007] 如前所述, 本领域中需要一种改进的 NTP 时间戳标记方法, 该方法能够对 NTP 分组的转移提供更精确且稳定的测量。

### 发明内容

[0008] 本发明的一个实施例提出一种用于建立将被网络时间协议 (NTP) 服务器插入到外出分组中的发送时间戳的方法。该方法包括以下步骤: 向 NTP 服务器中的硬件时钟请求当前时间, 并且向等待时间估计过滤器 (LEF) 请求校正项, 其中校正项是对与外出分组在跨越与外部环境之间的边界之前穿过各个软件层相关联的发送等待时间的估计。该方法还包括以下步骤: 将当前时间和校正项之和插入到外出分组中, 作为发送时间戳; 将带有发送时间戳的外出分组放置到出站队列中; 触发硬件时钟来锁存外出分组的离开时间; 执行 LEF 算法的步骤, 以通过计算发送时间戳和锁存的离开时间之间的偏移量来更新校正项, 以在下一外出分组中获得更精确的发送时间戳; 存储所更新的校正项; 并使用所更新的校正项将发送时间戳插入到下一外出分组中。

[0009] 所公开的系统和方法的一个优点在于对时间戳标记功能的改进控制可以被实现在现有的 NTP 服务器中, 而不会引入与遵循当前 NTP 标准的其他服务器和客户端之间的相互不兼容性的风险。重要的是, 所公开的方法包括以与当前 NTP 标准维持兼容性的方式利用分组时间戳标记引擎和等待时间估计过滤器的步骤。以这种方式, 到达时间和离开时间被硬件检测和寄存, 从而实现高精度度, 并且可以避免与设备驱动器、网络栈、操作系统等相关联的不确定性。结果, 更精确且稳定的 NTP 服务器时间戳被提供到 NTP 客户端。

### 附图说明

[0010] 为了可以详细理解本发明的上述特征, 可以通过参考实施例来对以上简要概括的本发明进行更具体的描述, 其中某些实施例在附图中被示出。但是, 需要注意, 附图仅仅示出本发明的典型实施例, 因此不应该看作限制本发明的范围, 因为本发明可以包含其他同样效果的实施例。

[0011] 图 1 是根据现有技术的典型 NTP 系统中的分组时间戳标记流程的概念性示意图;

[0012] 图 2 示出根据本发明一个实施例包含在 NTP 分组的头部中的时间戳;

[0013] 图 3 是根据本发明一个实施例的精确时间戳标记服务 (PTS) 系统中的分组时间戳标记流程的概念性示意图;

[0014] 图 4 示出根据本发明一个实施例由图 3 的 PTS 系统执行的方法步骤的流程图；

[0015] 图 5 示出根据本发明一个实施例由图 3 的等待时间估计过滤器执行的方法步骤的流程图；以及

[0016] 图 6 是被配置用于实现本发明的一个或多个方面的精确 NTP 服务器的概念性示意图。

### 具体实施方式

[0017] 图 2 示出根据本发明一个实施例包含在 NTP 分组的头部中的时间戳。如图 2 所示，沿时间线 200 的每个时间测量包含四个时间戳，这四个时间戳被定义如下： $T_1$  是代表发起于客户端时钟 210 的分组的发送起始时间的最佳估计的时间戳； $T_2$  是代表终止于精确 NTP 服务器时钟 220 的分组的接收终止时间的最佳估计的时间戳； $T_3$  是代表发起于精确 NTP 服务器时钟 220 的分组的发送起始时间的最佳估计的时间戳；以及  $T_4$  是代表终止于客户端时钟 210 的分组的接收终止时间的最佳估计的时间戳。在时间测量之后，这四个时间戳根据以下原理被用于计算端点之间的往返延迟和客户端时间 210 的偏移量：

[0018] a) 延迟估计： $\sigma = (T_4 - T_1) - (T_3 - T_2)$  (1)

[0019] b) 偏移量估计： $\Theta = \frac{(T_2 - T_1) + (T_3 - T_4)}{2}$  (2)

[0020] c) 误差： $\varepsilon = \phi \cdot (T_4 - T_1) + \alpha$  (3)

[0021] 其中， $\phi(t)$  是描述客户端时钟的倾斜累积率 (skew accumulation rate) 的函数， $\alpha$  是关于精确 NTP 服务器时间测量的不确定性的量度。

[0022] 由于精确 NTP 服务器主要负责向客户端时钟 210 提供  $T_2$  和  $T_3$ ，因此， $T_2$  和  $T_3$  值中的任何不精确性导致延迟和偏移量估计中的不精确性，延迟和偏移量估计中的不精确性进而导致时间测量过程中的不确定性的值  $\alpha$  增大。但是，等式 (3) 表明精确 NTP 服务器也可以通过减小时间测量过程中的不确定性  $\alpha$  来提高时间测量电路的性能。在图 3 至 5 中描述了用于实现来自精确 NTP 服务器的更精确且稳定的时间测量的系统和方法。

[0023] 图 3 是根据本发明一个实施例的精确时间戳标记服务 (PTS) 系统 300 中的分组时间戳标记流程的概念性示意图。如图所示，PTS 系统 300 包括（但不局限于）与以太网 PHY310 相关联的物理层、以太网 MAC320、O/S 服务 330、NTP 应用 340 和硬件时钟 350。如前所述，传入分组 302 的到达时间（记作  $T_{rx}$ ）指的是传入分组 302 的第一（或指定）比特跨越外部环境（未示出）和以太网 PHY310 之间的边界的时间。随后，刚刚穿过所述外部环境和以太网 PHY310 之间的边界的传入分组 302 所进行的第一步是通过接收分组信道 312 继续前进到以太网 MAC320。传入分组 302 的路径由箭头 312、322 和 332 示出。同样如前所述，外出分组 304 的离开时间  $T_{tx}$  指的是外出分组 304 的第一（或指定）比特穿过以太网 PHY310 和外部环境之间的所述边界的时间。将要穿过以太网 PHY310 和外部环境之间的所述边界的外出分组 304 的最后一步是行进穿过发送分组信道 314（外出分组 304 的路径由箭头 334、324 和 314 示出）。结合来看，接收分组信道 312 和发送分组信道 314 这里被称为“介质无关接口 (MII) 信道”。

[0024] 重要的是，PTS 系统 300 还包括被动监视单元 (passive tap unit) 316 和等待时间估计过滤器 (LEF) 360。被动监视单元 316 是一种分组时间戳标记引擎，其被配置用于

监视流过 MII 信道的网络流量。在每个分组穿过时,被动监视单元 316 分析流过 MII 信道的数据,以判断传入分组 302 是否希望去往 NTP 应用 340(即,传入分组 302 是否是 NTP 分组)。如果是这样,被动监视单元 316 则被配置为触发硬件时钟 350 以锁存(即暂时记录)到达时间  $T_{rx}$ ,如箭头 319 所示。类似地,如果被动监视单元 316 检测到由 NTP 应用 340 创建的外出分组 304,被动监视单元 316 则触发硬件时钟 350 锁存离开时间  $T_{tx}$ ,如箭头 318 所示。诸如被动监视单元 316 之类的分组时间戳标记引擎的功能包括监视流过 MII 信道的网络流量和触发硬件时钟 350 处的时间锁存,该分组时间戳标记引擎可以用硬件或软件来实现。硬件实现方式的示例可以参见 2001 年 4 月 17 日递交的题为“Precise Network Time Transfer”的美国专利 No. 6,985,499,该美国专利通过引用被结合于此。本领域技术人员将会意识到,在不同实施例中,被动监视单元 316 和硬件时钟 350 可以被部署在同一模块上/中,或者不同模块上/中。

[0025] LEF360 被配置用于从硬件时钟 350 获取锁存的离开时间  $T_{tx}$ ,并执行图 5 所述的方法步骤来计算校正项  $k$ 。校正项  $k$  是对通过精确 NTP 服务器的发送等待时间的估计,希望该估计能够考虑到 NTP 分组在 NTP 应用 340 和以太网 PHY310 之间行进时所经历的延迟。在每次 NTP 客户端/服务器分组交换之后,LEF360 更新校正项  $k$  的值以维持对下一发送延迟的最佳估计。使用锁存的到达时间  $T_{rx}$  和由 LEF360 计算出的最近校正项  $k$ ,PTS 系统 300 执行图 4 所述的方法步骤,从而产生对时间戳  $T_2$  和  $T_3$  的更精确且稳定的时间测量。

[0026] 图 4 示出根据本发明一个实施例由图 3 的 PTS 系统 300 执行的方法步骤的流程图。虽然这些方法步骤是结合图 3 的系统来描述的,但是本领域技术人员将会理解,以任意顺序执行这些方法步骤的任意系统都处于本发明的范围之内。

[0027] 该方法开始于步骤 402,其中 PTS 系统 300 通过将发送等待时间阵列中的元素设置为 0 来初始化 LEF360。在替换实施例中,发送等待时间阵列中的元素可以被设置为某些其他预定值。在步骤 404 中,被动监视单元 316 判断是否存在作为 NTP 分组的新的传入分组 302。如果不存在新的 NTP 分组,方法则前进到步骤 405,其中 PTS 系统 300 等待直到下一分组到达。方法从步骤 405 返回步骤 404。

[0028] 在步骤 404 中,如果被动监视单元 316 确定新的传入分组 302 是 NTP 分组,方法则前进到步骤 406,其中被动监视单元 316 触发硬件时钟 350 来锁存到达时间  $T_{rx}$ 。当传入分组 302 穿过以太网 MAC320 和 O/S 服务 330 时,方法前进到步骤 408,其中传入分组 302 在 NTP 应用 340 处被接收。一旦 NTP 应用 340 意识到传入分组 302 已经到达,方法就前进到步骤 410,其中 NTP 应用 340 向硬件时钟 350 请求锁存的到达时间  $T_{rx}$ ,如图 3 中的箭头 342 所示。在 NTP 应用 340 获取所请求的锁存的到达时间  $T_{rx}$  之后,如图 3 中的箭头 344 所示,方法前进到步骤 412。在步骤 412 中,NTP 应用 340 将值为  $T_{rx}$  的时间戳  $T_2$  插入到传入分组 302 中。注意,该确定时间戳  $T_2$  的方法非常不同于典型 NTP 系统 100 中的操作方法,在 NTP 系统 100 中,在将时间戳触发 142 发送到硬件时钟 150 之后,NTP 应用 140 接收当前时间值,而不是代表过去传入分组 102 实际到达时间的时间值。

[0029] 在步骤 414 中,NTP 应用 340 创建针对传入分组 302 的回复分组。当回复分组尚未被发送时,无法从硬件时钟 350 获得相应的发送信息。在步骤 416 中,NTP 应用 340 向硬件时钟 350 请求当前时间  $T_{current}$ ,如图 3 中的箭头 346 所示。在 NTP 应用 340 获得所请求的当前时间  $T_{current}$  之后,如图 3 中的箭头 348 所示,方法前进到步骤 418。在步骤 418 中,NTP

应用 340 向 LEF360 请求校正项  $k$ 。图 3 中的箭头 362 示出 LEF360 将校正项  $k$  发送到 NTP 应用 340。在步骤 420 中, NTP 应用 340 将作为对通过精确 NTP 服务器的发送等待时间的估计的校正项  $k$  添加到从硬件时钟 350 获得的  $T_{\text{current}}$  的值, 并将结果作为时间戳  $T_3$  插入到回复分组中。时间戳  $T_3$  也被发送到 LEF360。

[0030] 在步骤 422 中, NTP 应用 340 将带有时间戳  $T_3$  的回复分组放置到出站队列中, 作为外出分组 304。当外出分组 304 穿过 O/S 服务 330、以太网 MAC320 并被被动监视单元 316 检测到时, 方法前进到步骤 424。在步骤 424 中, 被动监视单元 316 触发硬件时钟 350 以锁存实际离开时间  $T_{\text{tx}}$ 。然后方法前进到步骤 426, 其中硬件时钟 350 将实际离开时间  $T_{\text{tx}}$  的值提供到 LEF360, 如图 3 中的箭头 352 所示。进而, 在步骤 428 中, LEF360 执行由图 5 更详细描述的方法步骤, 并计算校正项  $k$  的更新值。然后方法返回上述步骤 404。

[0031] 图 5 示出根据本发明一个实施例由图 3 的 LEF (等待时间估计过滤器) 360 执行的方法步骤的流程图。同样, 虽然这些方法步骤是结合图 3 的系统来描述的, 本领域技术人员将会理解, 以任意顺序执行这些方法步骤的任意系统都处于本发明的范围之内。

[0032] LEF360 被设计用于平滑校正项更新中的抖动 (jitter)。这是通过利用最新条目维护发送等待时间的小阵列以迫使最旧的条目离开阵列来实现的。该阵列可被记作  $\{\mu_i, \mu_{i-1}, \mu_{i-2}, \dots, \mu_{i-N}\}$ , 其中  $N+1$  是阵列中元素的数目。在一个实施例中, 该阵列可以仅包含 2 个元素, 在此情况下,  $N$  等于 1。

[0033] 该方法开始于步骤 502, 其中, 如图 4 的步骤 402 所述, PTS 系统 300 通过将发送等待时间阵列中的元素的值设置为 0 来初始化 LEF360。在替换实施例中, 发送等待时间阵列中的变量可以被设置为某些其他预定值。在步骤 504 中, LEF360 判断是否存在由 NTP 应用 340 提供的新时间戳  $T_3$ 。如果不存在新的时间戳  $T_3$ , 方法则前进到步骤 505, 其中 LEF360 等待一定的时间量。方法从步骤 505 返回步骤 504。

[0034] 在步骤 504 中, 如果 LEF360 确定存在新的时间戳  $T_3$  (如图 3 中的箭头 349 所示), 方法则前进到步骤 506, 其中 LEF360 丢弃旧的时间戳  $T_3$ 。方法随后前进到步骤 508, 其中 LEF360 加载从 NTP 应用 340 获得的新的时间戳  $T_3$ , 以及由硬件时钟 350 提供的实际离开时间  $T_{\text{tx}}$  的值。随后方法前进到步骤 510。

[0035] 在步骤 510 中, LEF360 执行 LEF 算法的步骤以对发送等待时间阵列进行排序, 并选择最小值作为校正项  $k$  的下一值。在一个实施例中, 使用以上定义的变量, 步骤 510 的 LEF 算法如下所述:

[0036] 步骤 1:  $\mu_{i-1} = \mu_i$  (阵列下移 1)

[0037] 步骤 2:  $\mu_i = T_{\text{tx}} - T_3 + k$  (更新发送等待时间)

[0038] 步骤 3:  $k = \min(\mu_{i-1}, \mu_i)$  (计算校正项的新值)

[0039] 作为 LEF 算法步骤的执行结果, 在步骤 512 中, LEF360 产生校正项  $k$  的更新值, 该值代表对通过精确 NTP 服务器的发送等待时间的估计。更新的校正项  $k$  被存储并随后被用于计算回复分组的下一发送时间戳  $T_3$ , 如图 4 所述。本领域技术人员将意识到, 也可采用用于计算校正项  $k$  的最佳估计的其他算法而不会脱离本发明的基本范围。在该实施例中提出的 LEF 算法是一种计算效率很高的操作。从步骤 512, 方法返回上述步骤 502。

[0040] 图 6 是被配置用于实现本发明的一个或多个方面的精确 NTP 服务器 600 的概念性示意图。如图所示, 该精确 NTP 服务器 600 包括 (但不局限于) 中央处理单元 (CPU) 610、

网络接口 620 和 PTS 系统 300, 该 PTS 系统 300 进一步包括硬件时钟 350。网络接口 620 与 PTS 系统 300 之间传输分组, PTS 系统 300 进而与 CPU610 通信并执行这里描述的方法步骤来建立接收和发送时间戳  $T_2$  和  $T_3$ 。

[0041] 本发明使得精确 NTP 服务器能够进行:

[0042] - 监视流过 MII 信道的网络流量, 计算通过精确 NTP 服务器的发送等待时间, 基于由硬件时钟锁存的到达时间将接收时间戳  $T_2$  插入到传入 NTP 分组中, 基于对外出 NTP 分组预期经历的发送等待时间的估计将发送时间戳  $T_3$  插入到外出 NTP 分组中, 并且在每次 NTP 客户端 / 服务器分组交换之后通过以下步骤来更新对发送等待时间的估计:

[0043] ○在来自精确 NTP 客户端的传入分组穿过外部环境和精确时间戳标记服务 (PTS) 系统的物理层之间的边界之后立即检测该传入分组;

[0044] ○在硬件时钟处触发时间锁存, 以暂时记录传入分组的到达时间;

[0045] ○从硬件时钟获取锁存的到达时间, 并将等于锁存的到达时间的接收时间戳  $T_2$  插入到传入分组中;

[0046] ○在外出分组实际跨越 PTS 系统的物理层和外部环境之间的边界之前, 向硬件时钟请求当前时间和向等待时间估计过滤器请求对发送等待时间的最近估计, 并将等于当前时间和对发送等待时间的最近估计之和的发送时间戳  $T_3$  插入到外出分组中;

[0047] ○在外出分组即将跨越物理层和外部环境之间的边界之前检测外出分组;

[0048] ○在硬件时钟处触发时间锁存, 以暂时记录外出分组的离开时间;

[0049] ○通过计算软件计算的发送时间戳值和实际测量的发送时间戳之间的偏移量并过滤该差值来更新对发送等待时间的估计, 以选择发送等待时间的最小值作为对发送等待时间的下一估计;

[0050] - 与现有技术方法相比, 对 NTP 分组进入和离开精确 NTP 服务器的转移提供了更加精确且稳定的测量。

[0051] 所公开的系统和方法的一个优点在于对时间戳标记功能的更好控制可以被实现在现有的 NTP 服务器中, 而不存在引入与遵循当前 NTP 标准的其他服务器和客户端之间的相互不兼容性的风险。所公开的精确时间戳标记服务 (PTS) 系统包括以硬件实现的分组时间戳标记引擎, 该引擎被配置用于识别 NTP 分组。所公开的 PTS 系统还包括等待时间估计过滤器, 其被配置用于计算对通过 NTP 服务器中的各个软件层的发送等待时间的估计。重要的是, 所公开的方法包括以与当前 NTP 标准维持兼容性的方式利用分组时间戳标记引擎和等待时间估计过滤器的步骤。以这种方式, 到达时间和离开时间被硬件检测和寄存, 从而实现高精度度, 并且可以避免与设备驱动器、网络栈、操作系统等相关联的不确定性。结果, 向 NTP 客户端提供了更精确且稳定的 NTP 服务器时间戳。

[0052] 本发明的一个实施例被实现为用于计算机系统的程序产品。该程序产品的程序定义实施例 (包括这里描述的方法) 的功能, 并且可被包含在多种计算机可读存储介质之上。示例性的计算机可读存储介质包括 (但不局限于): (i) 可永久存储信息的不可写存储介质 (例如计算机中的只读存储设备, 比如可由 CD-ROM 驱动器读取的 CD-ROM 盘); (ii) 存储可更改的信息的可写存储介质 (例如磁盘驱动器或硬盘驱动器中的软盘)。这些计算机可读存储介质在承载指示本发明的功能的计算机可读指令时是本发明的实施例。

[0053] 其它介质包括通过其向计算机传递信息的通信介质, 例如计算机或电话网络, 包

括无线通信网络。后一实施例具体包括向 / 从因特网和其他网络传送信息。这些通信介质在承载指示本发明的功能的计算机可读指令时是本发明的实施例。

[0054] 虽然以上描述针对本发明的实施例,但是在不脱离本发明的基本范围的情况下,可以设计出本发明的其他和更多实施例。

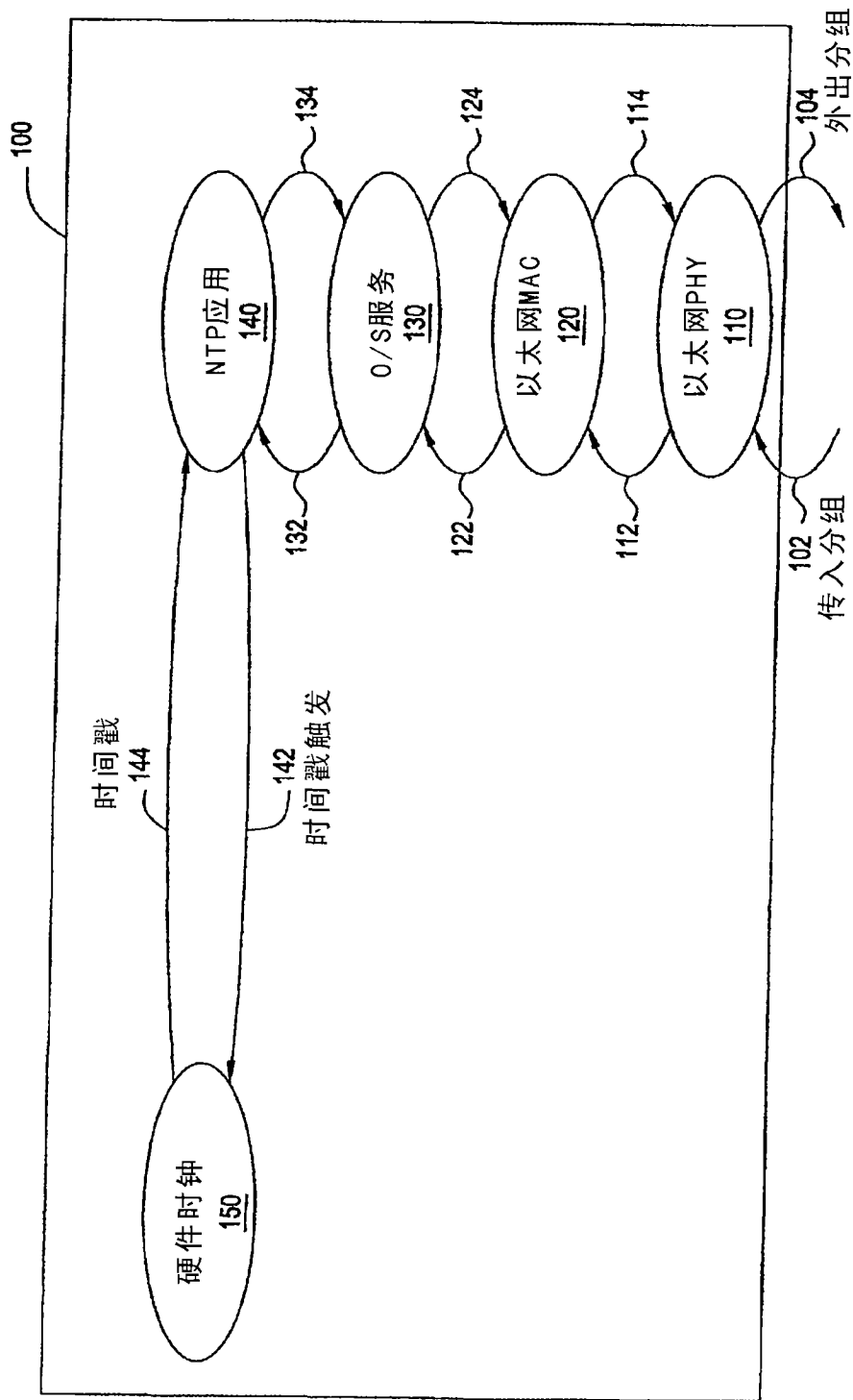


图1  
(现有技术)

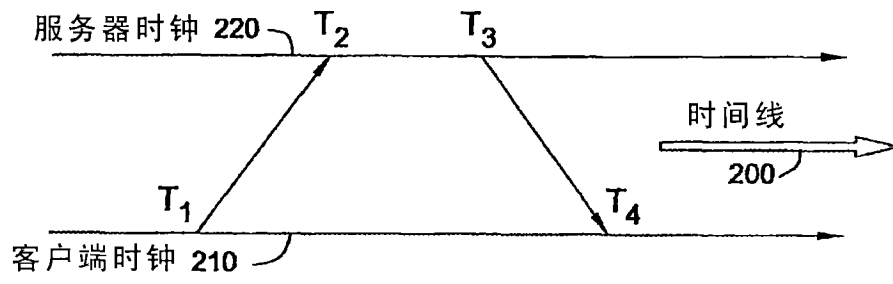


图 2

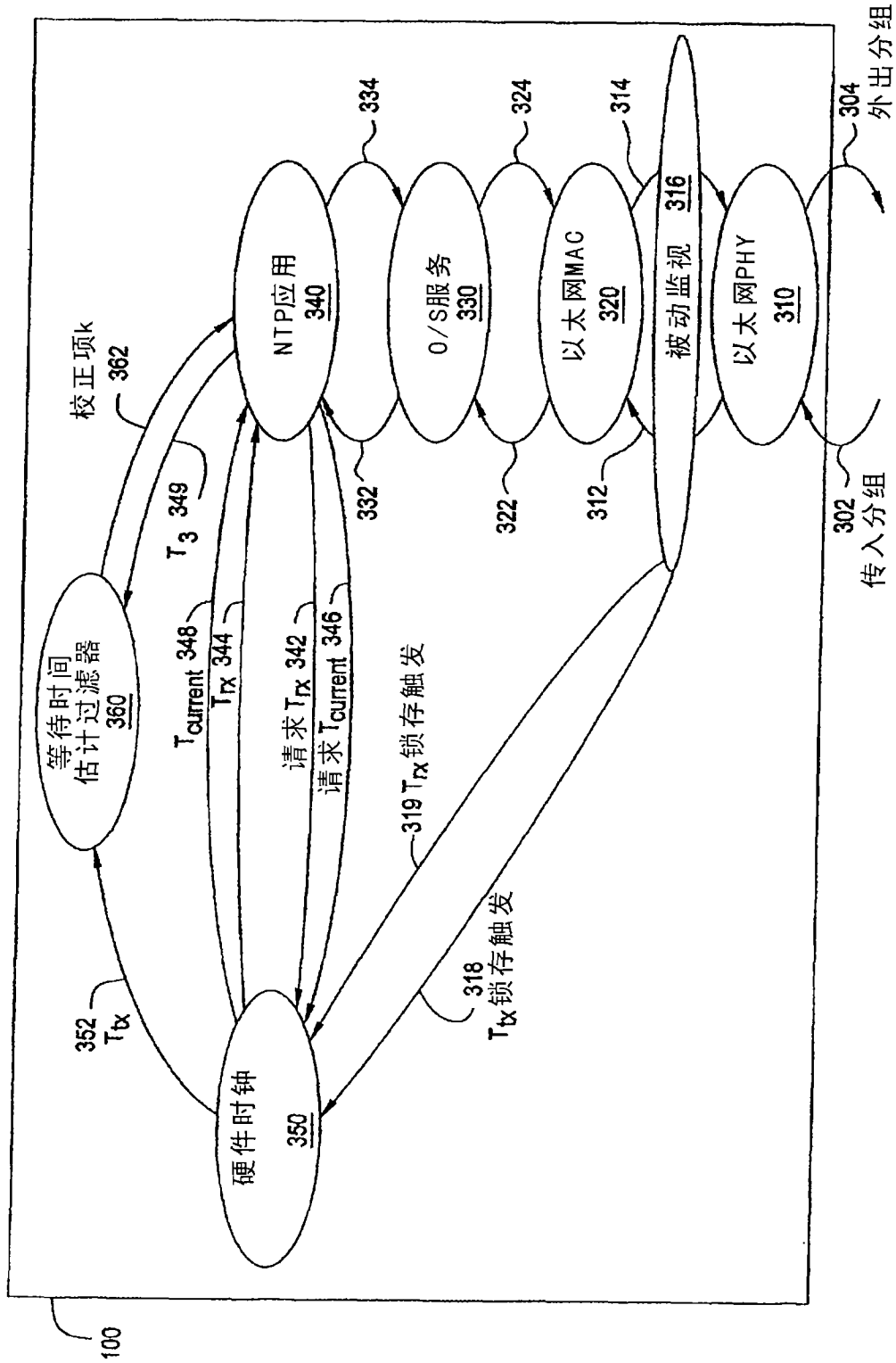


图3

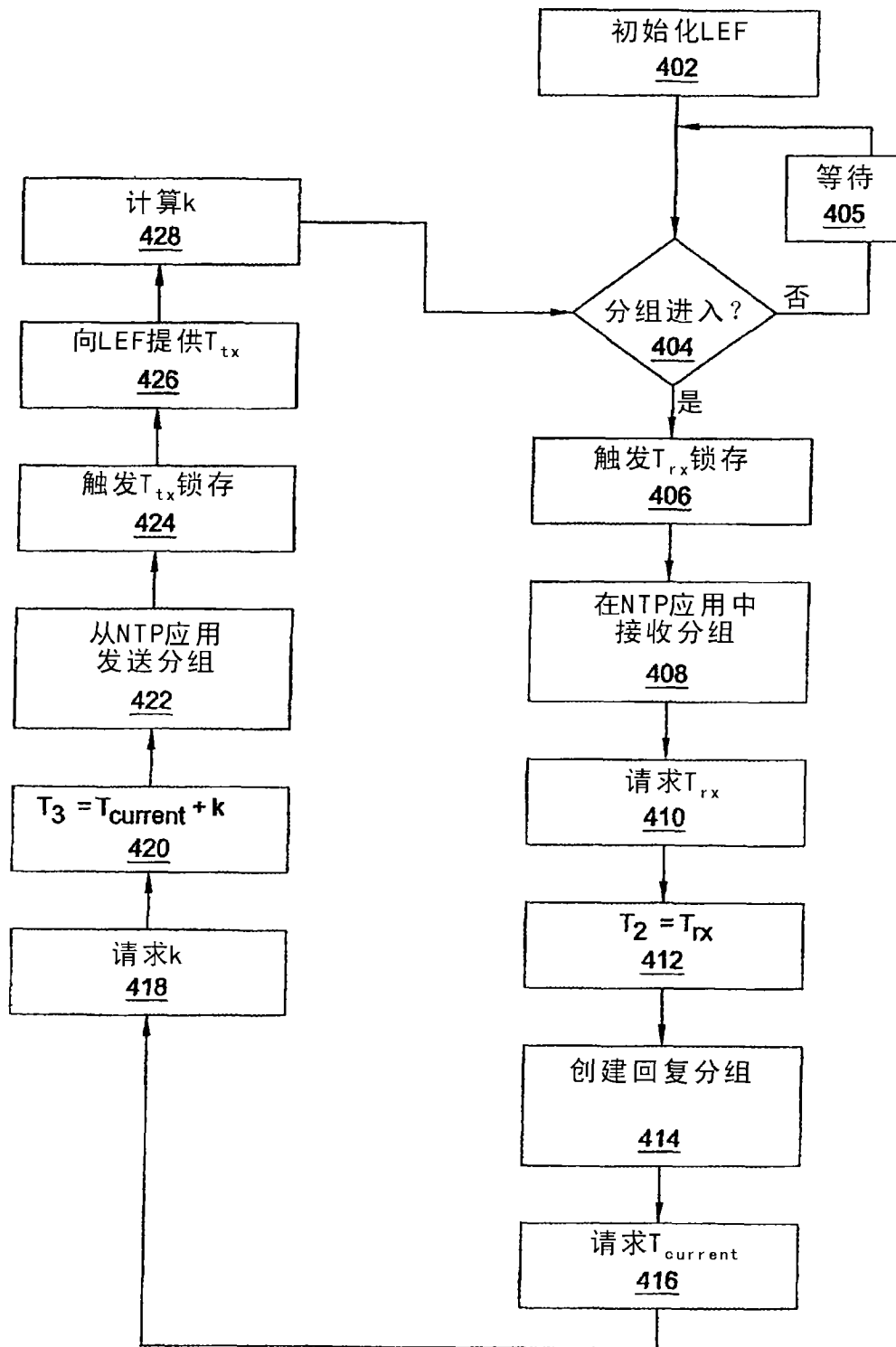


图 4

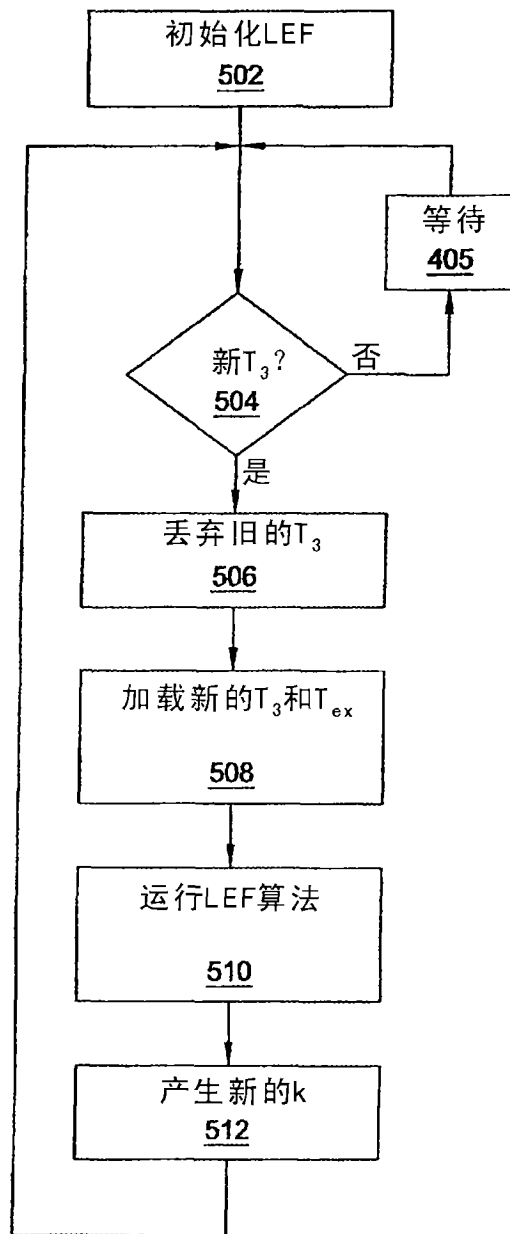


图 5

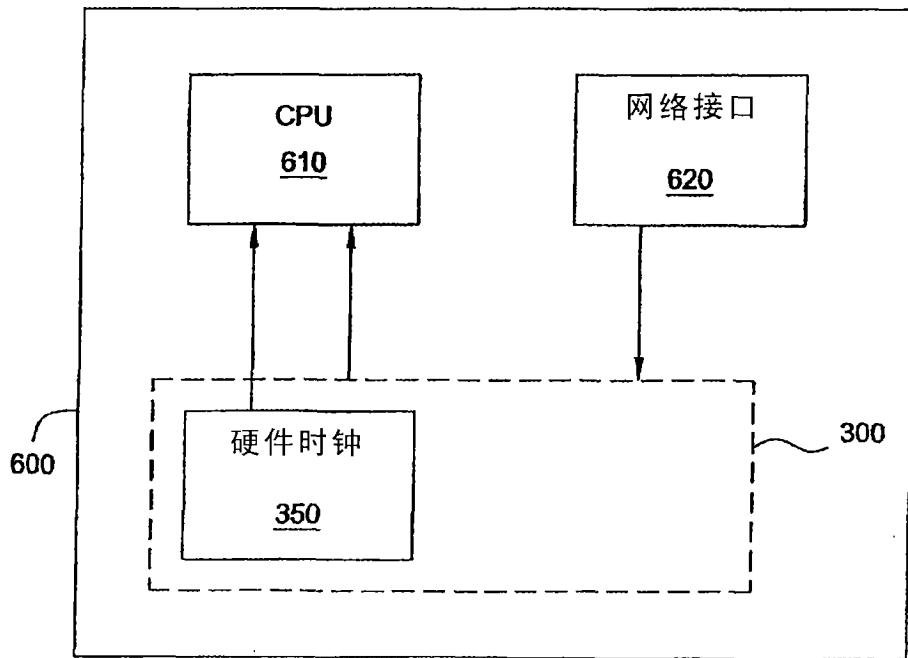


图 6