目 录

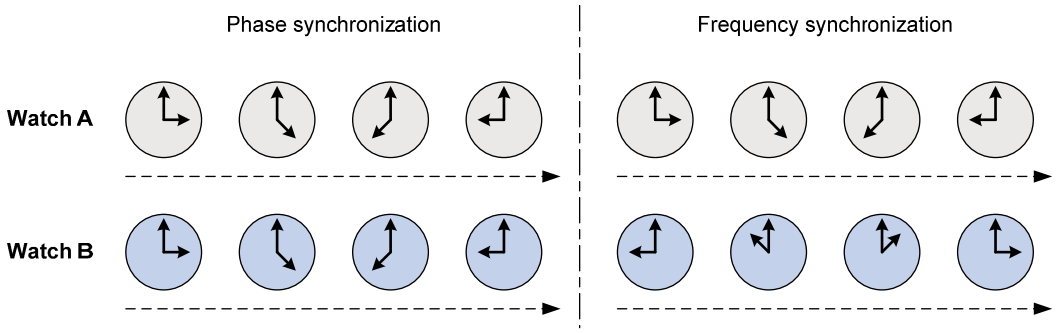
PTP

PTP 简介

在通信网络中，许多业务的正常运行都要求网络时钟同步，即整个网络各设备之间的时间或频率差保持在合理的误差水平内。网络时钟同步包括以下两个概念：

- 时间同步：也叫相位同步（Phase synchronization），是指信号之间的频率和相位都保持一致，即信号之间的相位差恒为零。
- 频率同步（Frequency synchronization）：也叫时钟同步，是指信号之间在频率或相位上保持某种严格的特定关系，信号在其对应的有效瞬间以同一平均速率出现，以保证通信网络中的所有设备都以相同的速率运行，即信号之间保持恒定的相位差。

图 1 时间同步和频率同步示意图



如 图 1所示，有两个表Watch A与Watch B，如果这两个表的时间每时每刻都保持一致，这个状态就是时间同步；如果这两个表的时间不一致，但保持一个恒定的差值（如图中的Watch B总比Watch A晚 6 个小时），这个状态就是频率同步。

PTP（Precision Time Protocol，精确时间协议）是一种时间同步的协议，其本身只是用于设备之间的高精度时间同步，但也可被借用于设备之间的频率同步。相比现有的各种时间同步机制，PTP具备以下优势：

- 相比 NTP（Network Time Protocol，网络时间协议），PTP 能够满足更高精度的时间同步要求：NTP 一般只能达到亚秒级的时间同步精度，而 PTP 则可达到亚微秒级。
- 相比 GPS（Global Positioning System，全球定位系统），PTP 具备更低的建设和维护成本，并且由于可以摆脱对 GPS 的依赖，在国家安全方面也具备特殊的意义。

说明

H3C 的设备目前只支持 PTP 的时间同步功能，而不支持频率同步功能。本文只介绍 PTP 时间同步的相关概念和原理。

PTP 基本概念

1. PTP 域

我们将应用了 PTP 协议的网络称为 PTP 域。PTP 域内有且只有一个同步时钟，域内的所有设备都与该时钟保持同步。

2. PTP 端口

我们将设备上运行了PTP协议的端口称为PTP端口。如 图 2所示，PTP端口的角色可分为以下三种：

- 主端口（Master Port）：发布同步时间的端口，可存在于 BC 或 OC 上。
- 从端口（Slave Port）：接收同步时间的端口，可存在于 BC 或 OC 上。
- 被动端口（Passive Port）：既不接收同步时间、也不对外发布同步时间的端口，只存在于 BC 上。

3. 时钟节点

PTP 域中的节点称为时钟节点，PTP 协议定义了以下三种类型的基本时钟节点：

(1) OC（Ordinary Clock，普通时钟）：该时钟节点在同一个 PTP 域内只有一个 PTP 端口参与时间同步，并通过该端口从上游时钟节点同步时间。此外，当时钟节点作为时钟源时，可以只通过一个 PTP 端口向下游时钟节点发布时间，我们也称其为 OC。

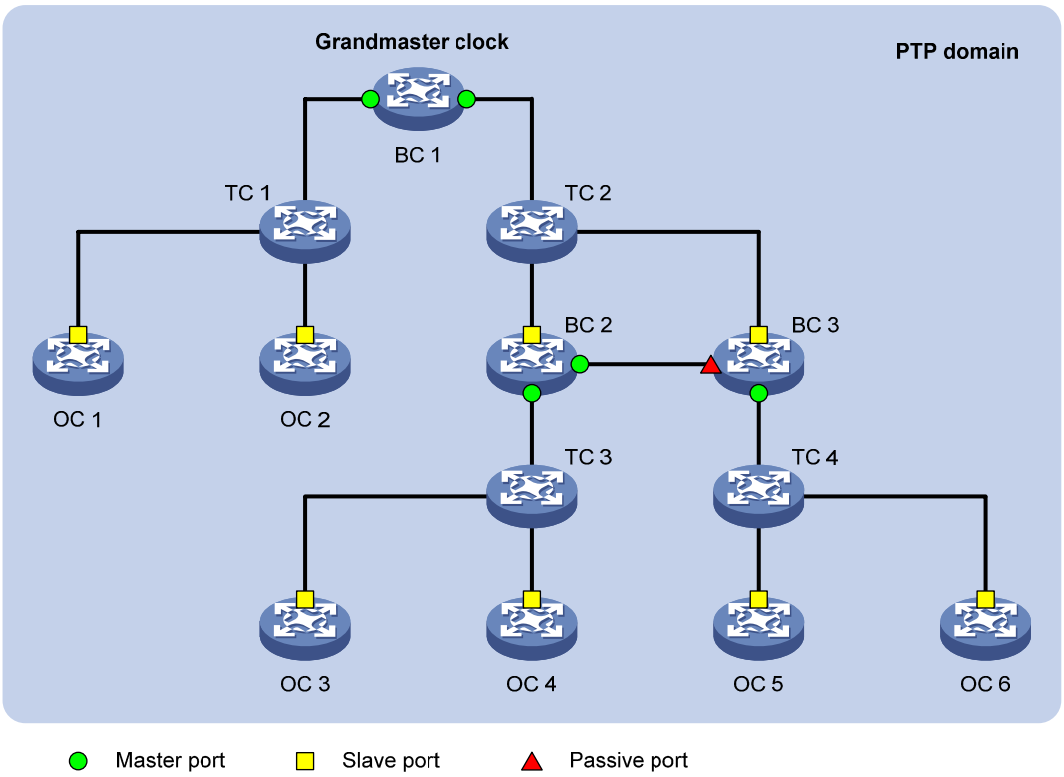
(2) BC（Boundary Clock，边界时钟）：该时钟节点在同一个PTP域内拥有多个PTP端口参与时间同步。它通过其中一个端口从上游时钟节点同步时间，并通过其余端口向下游时钟节点发布时间。此外，当时钟节点作为时钟源时，可以通过多个PTP端口向下游时钟节点发布时间的，我们也称其为BC，如 图 2中的BC 1。

(3) TC（Transparent clock，透明时钟）：与 BC/OC 相比，BC/OC 需要与其它时钟节点保持时间同步，而 TC 则不与其它时钟节点保持时间同步。TC 有多个 PTP 端口，但它只在这些端口间转发 PTP 协议报文并对其进行转发延时校正，而不会通过任何一个端口同步时间。TC 包括以下两种类型：

- E2ETC（End-to-End Transparent Clock，端到端透明时钟）：直接转发网络中非 P2P（Peer-to-Peer，点到点）类型的协议报文，并参与计算整条链路的延时。
- P2PTC（Peer-to-Peer Transparent Clock，点到点透明时钟）：只直接转发 Sync 报文、Follow_Up 报文和 Announce 报文，而终结其它 PTP 协议报文，并参与计算整条链路上每一段链路的延时。

如 图 2所示，是上述三种基本时钟节点在PTP域中的位置。

图 2 基本时钟节点示意图



除了上述三种基本时钟节点以外，还有一些混合时钟节点，譬如融合了 TC 和 OC 各自特点的 TC+OC：它在同一个 PTP 域内拥有多个 PTP 端口，其中一个端口为 OC 类型，其它端口则为 TC 类型。一方面，它通过 TC 类型的端口转发 PTP 协议报文并对其进行转发延时校正；另一方面，它通过 OC 类型的端口进行时间的同步。与 TC 的分类类似，TC+OC 也包括两种类型：E2ETC+OC 和 P2PTC+OC。

4. 主从关系

主从关系（Master-Slave）是相对而言的，对于相互同步的一对时钟节点来说，存在如下主从关系：

- 发布同步时间的节点称为主节点，而接收同步时间的节点则称为从节点。
- 主节点上的时钟称为主时钟，而从节点上的时钟则称为从时钟。
- 发布同步时间的端口称为主端口，而接收同步时间的端口则称为从端口。

5. 最优时钟

如 图 2所示，PTP域中所有的时钟节点都按一定层次组织在一起，整个域的参考时间就是最优时钟（Grandmaster Clock，GM），即最高层次的时钟。通过各时钟节点间PTP协议报文的交互，最优时钟的时间最终将被同步到整个PTP域中，因此也称其为时钟源。

最优时钟可以通过手工配置静态指定，也可以通过 BMC（Best Master Clock，最佳主时钟）协议动态选举，动态选举的过程如下：

(1) 各时钟节点之间通过交互的 Announce 报文中所携带的最优时钟优先级、时间等级、时间精度等信息，最终选出一个节点作为 PTP 域的最优时钟，与此同时，各节点之间的主从关系以及

各节点上的主从端口也确定了下来。通过这个过程，整个 PTP 域中建立起了一棵无环路、全连通，并以最优时钟为根的生成树。

(2) 此后，主节点会定期发送 Announce 报文给从节点，如果在一段时间内，从节点没有收到主节点发来的 Announce 报文，便认为该主节点失效，于是重新进行最优时钟的选择。

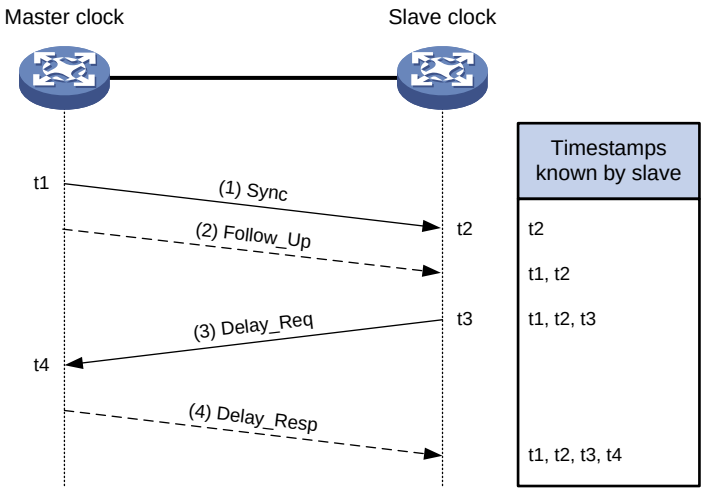
PTP 同步原理

PTP 同步的基本原理如下：主、从时钟之间交互同步报文并记录报文的收发时间，通过计算报文往返的时间差来计算主、从时钟之间的往返总延时，如果网络是对称的（即两个方向的传输延时相同），则往返总延时的一半就是单向延时，这个单向延时便是主、从时钟之间的时钟偏差，从时钟按照该偏差来调整本地时间，就可以实现其与主时钟的同步。

PTP 协议定义了两种传播延时测量机制：请求应答（Requset_Response）机制和端延时（Peer Delay）机制，且这两种机制都以网络对称为前提。

1. 请求应答机制

图 3 请求应答机制实现过程



请求应答方式用于端到端的延时测量。如 图 3所示，其实现过程如下：

(1) 主时钟向从时钟发送 Sync 报文，并记录发送时间 t1；从时钟收到该报文后，记录接收时间 t2。

(2) 主时钟发送 Sync 报文之后，紧接着发送一个携带有 t1 的 Follow_Up 报文。

(3) 从时钟向主时钟发送 Delay_Req 报文，用于发起反向传输延时的计算，并记录发送时间 t3；主时钟收到该报文后，记录接收时间 t4。

(4) 主时钟收到 Delay_Req 报文之后，回复一个携带有 t4 的 Delay_Resp 报文。

此时，从时钟便拥有了 t1～t4 这四个时间戳，由此可计算出主、从时钟间的往返总延时为[(t2 – t1) + (t4 – t3)]，由于网络是对称的，所以主、从时钟间的单向延时为[(t2 – t1) + (t4 – t3)] / 2。因此，从时钟相对于主时钟的时钟偏差为：Offset = (t2 – t1) - [(t2 – t1) + (t4 – t3)] / 2 = [(t2 – t1) - (t4 – t3)] / 2。

此外，根据是否需要发送 Follow_Up 报文，请求应答机制又分为单步模式和双步模式两种：

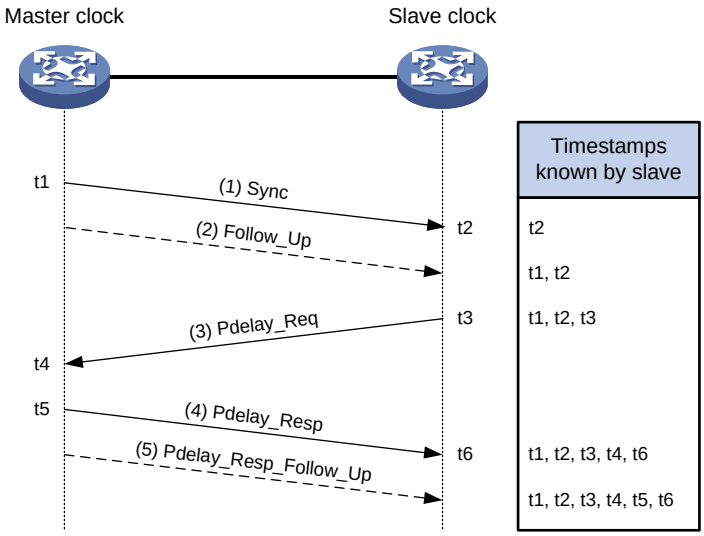
- 在单步模式下，Sync 报文的发送时间戳 t1 由 Sync 报文自己携带，不发送 Follow_Up 报文。
- 在双步模式下，Sync 报文的发送时间戳 t1 由 Follow_Up 报文携带。

说明

图 3以双步模式为例来说明请求应答机制的实现过程。

2. 端延时机制

图 4 端延时机制实现过程



与请求应答机制相比，端延时机制不仅对转发延时进行扣除，还对上游链路的延时进行扣除。如 图 4所示，其实现过程如下：

(1) 主时钟向从时钟发送 Sync 报文，并记录发送时间 t1；从时钟收到该报文后，记录接收时间 t2。

(2) 主时钟发送 Sync 报文之后，紧接着发送一个携带有 t1 的 Follow_Up 报文。

(3) 从时钟向主时钟发送 Pdelay_Req 报文，用于发起反向传输延时的计算，并记录发送时间 t3；主时钟收到该报文后，记录接收时间 t4。

(4) 主时钟收到 Pdelay_Req 报文之后，回复一个携带有 t4 的 Pdelay_Resp 报文，并记录发送时间 t5；从时钟收到该报文后，记录接收时间 t6。

(5) 主时钟回复 Pdelay_Resp 报文之后，紧接着发送一个携带有 t5 的 Pdelay_Resp_Follow_Up 报文。

此时，从时钟便拥有了 t1～t6 这六个时间戳，由此可计算出主、从时钟间的往返总延时为[(t4 – t3) + (t6 – t5)]，由于网络是对称的，所以主、从时钟间的单向延时为[(t4 – t3) + (t6 – t5)] / 2。因此，从时钟相对于主时钟的时钟偏差为：Offset = (t2 – t1) - [(t4 – t3) + (t6 – t5)] / 2。

此外，根据是否需要发送 Follow_Up 报文，端延时机制也分为单步模式和双步模式两种：

- 在单步模式下，Sync 报文的发送时间戳 t1 由 Sync 报文自己携带，不发送 Follow_Up 报文；而 t5 和 t4 的差值由 Pdelay_Resp 报文携带，不发送 Pdelay_Resp_Follow_Up 报文。
- 在双步模式下，Sync 报文的发送时间戳 t1 由 Follow_Up 报文携带，而 t4 和 t5 则分别由 Pdelay_Resp 报文和 Pdelay_Resp_Follow_Up 报文携带。


说明

图 4以双步模式为例来说明端延时机制的实现过程。