

智能网联系统导论



时间同步技术

裴欣，副研究员
中央主楼809房间
peixin@tsinghua.edu.cn



2020/11/25

清华大学自动化系系统工程研究所

大纲

- 1 时间同步的必要性与挑战
- 2 时钟偏差与同步问题
- 3 时间同步方法概述
- 4 时间同步协议

1

时间同步的必要性与挑战

1. 时间同步的必要性与挑战

物联网系统是典型的分布式系统，每个节点都有自己的时钟和对时间的定义；但网络整体工作时，应该需要一个共同的时间标准。

时间同步问题简介

- ◆ 时钟组成：振荡器（确定时间间隔）、计数器（记录经过的时间间隔个数）
- ◆ 计时方式：日晷->水钟->浑天仪->摆钟->石英钟（晶振）->原子钟
- ◆ 时间标准：UT（世界时：UT0, UT1, UT2），TAI（国际原子时），UTC（协调世界时），GPS
- ◆ 时间类型：
 - 物理时间：用来表示人类社会使用的绝对时间。
 - 逻辑时间：表达事件发生的顺序关系，是一个相对概念。
 - 全局时间：分布式系统通常需要一个表示整个系统的全局时间，全局时间根据需要可以是物理时间或逻辑时间。
- ◆ 时间同步概念：各个节点定期或不定期与其他节点交换本地时钟信息，并在协议或算法控制下调整本地时钟，实现全局时间一致的过程。

1. 时间同步的必要性与挑战

WSN中时间同步的必要性

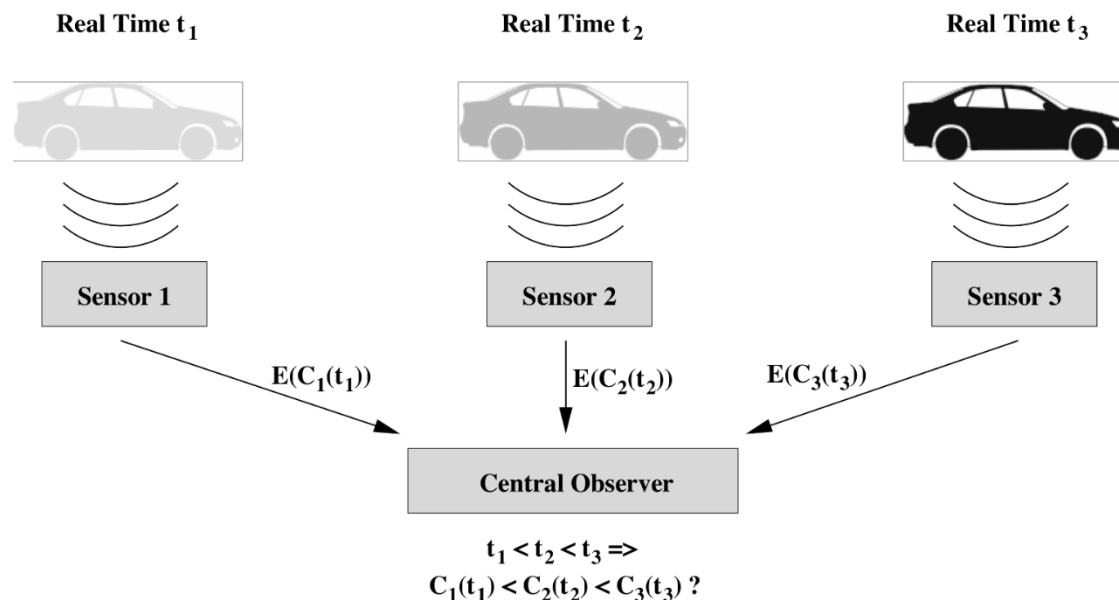
- ◆ 传感器节点之间受能量、存储、带宽限制，以及不同的晶体振荡频率，存在时间偏差
- ◆ 传感器的物理世界活动和事件具有强时间相关性
 - 逻辑顺序、运动速率、运动方向
 - 数据融合：观测相同或相关事件的多传感器数据集合，消除冗余信息，挖掘价值信息

Sequence: If real-time ordering of events is $t_1 < t_2 < t_3$, then sensor times should reflect this ordering:

$$C_1(t_1) < C_2(t_2) < C_3(t_3)$$

Speed: Time difference between sensor time stamps should correspond to real-time differences:

$$\Delta = C_2(t_2) - C_1(t_1) = t_2 - t_1$$



1. 时间同步的必要性与挑战

WSN中时间同步的必要性示例

- ◆ WSN中每个节点的时钟都有各自的时钟漂移。在MICA mote平台中，时钟漂移达到40微秒/秒，时钟频率是7.37MHz
- ◆ 时钟漂移达到40微秒/秒，究竟有多糟？ How bad is 40 microsec/sec?
 - $40 \times 60 = 2,400$ microsec/min
 - $2,400 \times 60 = 144,000$ microsec/hr
 - $144,000 \times 24 = 3,456,000$ microsec/day = **3.456 sec./day**

1. 时间同步的必要性与挑战

WSN中时间同步的必要性

- ◆ 对于分布式系统中的各种应用程序和算法而言，时间同步是必需的
 - 通信协议 communication protocols (at-most-once message delivery)
 - 安全 security (limit use of keys, detect replay attacks)
 - 数据一致性 data consistency (caches, replicated data)
 - 并发控制 concurrency control (atomicity and mutual exclusion)
 - 介质接入控制协议 medium access control (accurate timing of channel access)
 - 基于休眠/唤醒机制的能量管理 sleeping/active based energy management
 - 定位 localization (based on techniques such as time-of-flight measurements)

1. 时间同步的必要性与挑战

WSN中时间同步问题面临挑战

- ◆ 传统的针对有线网络设计的时间同步协议并不可用，如：NTP（Network Time Protocol）
 - 体积、计算能力和存储空间存在限制
 - 传输方式不同：无线而非有线
 - 目标不同：局部最优而非全局最优
- ◆ WSN环境还存在一些额外的影响因素
 - **环境影响**
 - 传感器通常放置在恶劣环境中，湿度、温度、压力都会有波动
 - **能量约束**
 - 由电池提供有限的能量供应
 - 时间同步协议必须是能量高效的
 - **无线媒质和移动性**
 - 吞吐量变化、差错率变化、射频干扰、非对称链路
 - 拓扑变化、密度变化，节点失效或死亡
 - **其它方面的约束**
 - 受限的处理器速度或内存容量
 - 同步硬件的尺寸、价格

1. 时间同步的必要性与挑战

WSN中时间同步问题面临挑战

扩展性

在WSN中，由于网络规模和节点数量的不同，时间同步机制应能适应不同的网络场景

稳定性

网络拓扑发生变换时，同步机制应能保持连续性和同步精度

鲁棒性

在受到环境、信道通信质量、节点失效等干扰情况下，保持正常工作

收敛性

时间同步机制应能在较短的时间内达成节点时间的同步

低能耗

WSN能量受限，同步机制应能在保证运作机能的前提下，尽量降低能耗

2

时钟偏差与同步问题

2. 时钟偏差与同步问题

WSN节点通用时钟工作原理

- ◆ 典型的时钟组成：一个稳定的石英振荡器 + 一个计数器 Typical clocks consist of quartz-stabilized oscillator and a counter
- ◆ 时钟计时工作原理或过程：
 - 计数器随着每次石英晶体的振荡递减，直到0，重置初始值，并产生一个中断 interrupt
 - 每个中断（时钟周期）都触发一个软件时钟加1 Each interrupt (clock tick) increments software clock (another counter)
 - 软件时钟可通过应用程序API读取 Software clock can be read by applications using API
 - 软件时钟为传感器节点提供了一个本地时间：t时刻的本地时钟读数C(t)

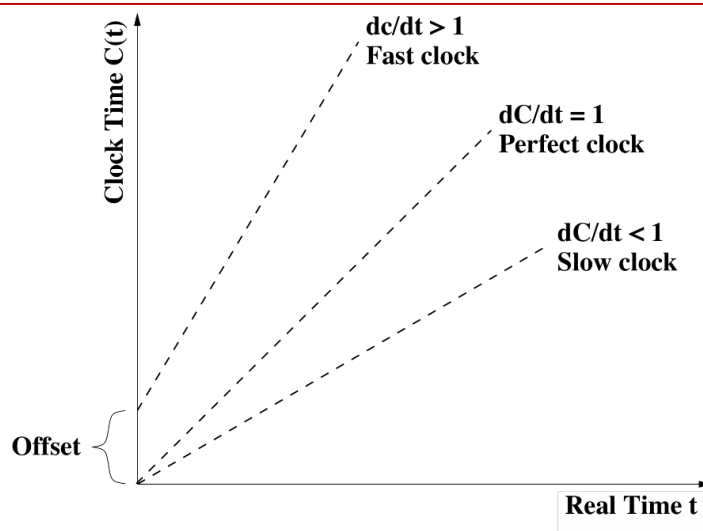
2. 时钟偏差与同步问题

时钟相关概念

- ◆ 时钟偏移：两节点本地时间之间计数差 Clock offset: difference between the local times of two nodes
- ◆ 时钟漂移率 clock drift rate
 - 时钟漂移率 dC/dt ，理想情况下恒定为1；实际情况下，受温度、湿度、电压、石英晶体年龄等因素的影响，会产生漂移 Clock rate dC/dt depends on temperature, humidity, supply voltage, age of quartz, etc., resulting in drift rate (dC/dt)
 - 时钟的相对漂移率： $dC/dt - 1$
 - 使得时钟变快或变慢
 - 石英晶体典型的最大漂移率 ρ 为：1ppm ~ 100ppm，由厂商给出 Maximum drift rate ρ given by manufacturer (typical 1ppm to 100ppm)

最大漂移率需满足：

$$1 - \rho \leq \frac{dC}{dt} \leq 1 + \rho$$



- 时钟时间 $C(t)$ 必须是分段连续的，即是时间的严格单调递增函数
- 时钟的调整包括时间偏移和漂移的调整

2. 时钟偏差与同步问题

时钟相关概念

◆ 时钟漂移率clock drift rate

- 漂移率会导致两个时钟在同步后再次产生时钟偏移
 - 两个同步后的时钟，再次产生的时钟的漂移值最大为 $2\rho_{\max}$ Two synchronized identical clocks can drift from each other at rate of at most $2\rho_{\max}$
 - 若想限制相对时钟偏移到 δ 秒，则同步操作的间隔 τ_{sync} 需要满足一定的条件

同步操作的间隔 $\tau_{\text{sync}} \leq \frac{\delta}{2\rho_{\max}}$

2. 时钟偏差与同步问题

时间同步：通过时钟的调整，使得多个节点的本地时间读数一致. **Time Synchronization**

外部同步：所有节点的时钟与一个外部时间源（或参考时钟）同步

- ◆ 此参考时钟必须是一个精确的实时标准，如：世界协调时间 UTC reference clock is accurate real-time standard (e.g., UTC)

内部同步：在没有外部参考时钟支持下，所有节点的时钟相互同步

- ◆ 其目标是：获取网络内部所有时钟的一致性 goal is to obtain consistent view of time across all nodes in network
 - 网络内的时钟可能与外部实时标准时间不同 network-wide time may differ from external real-time standards

2. 时钟偏差与同步问题

时间同步：通过时钟的调整，使得多个节点的本地时间读数一致. **Time Synchronization**

时间同步相关概念

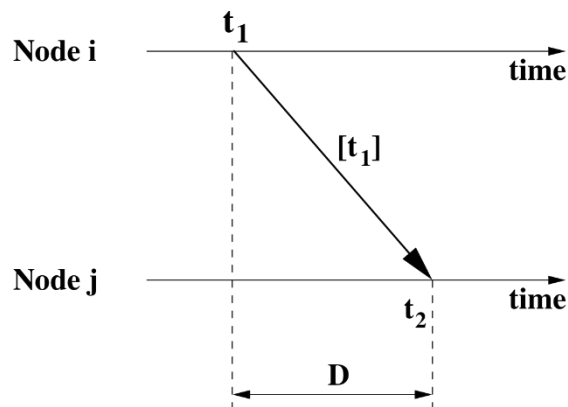
- ◆ 时钟准确度：当节点与外部参考时钟同步时，时钟相对参考时钟的最大偏移 **Accuracy: maximum offset of a clock with respect to reference clock**
- ◆ 时钟精度：当网络中的节点内同步时，任意两时钟之间的最大偏移 **Precision: maximum offset between any two clocks**
 - 如果两个节点外部同步的准确度为 Δ ，则它们内部时钟同步的精度为 2Δ **If two nodes synchronized externally with accuracy of Δ , also synchronized internally with precision 2Δ**

时间同步方法概述

3. 时间同步方法概述

时间同步通常是基于传感器节点之间的某种形式的信息交换，即：依赖于同步消息

- ◆ **成对同步**：两节点同步，至少需要一个同步消息 Pairwise synchronization: two nodes synchronize using at least one message
- ◆ **网络同步**：在整个网络中重复成对同步过程 Network-wide synchronization: repeat pairwise synchronization throughout network
- ◆ **单向消息交换** One-way message exchange
 - 单个消息：包含时间戳 single message containing a time stamp
 - 时钟差可以计算得到： $(t_2 - t_1) = D + \delta$ (D =propagation delay, 传播时延，一般可以忽略不计)



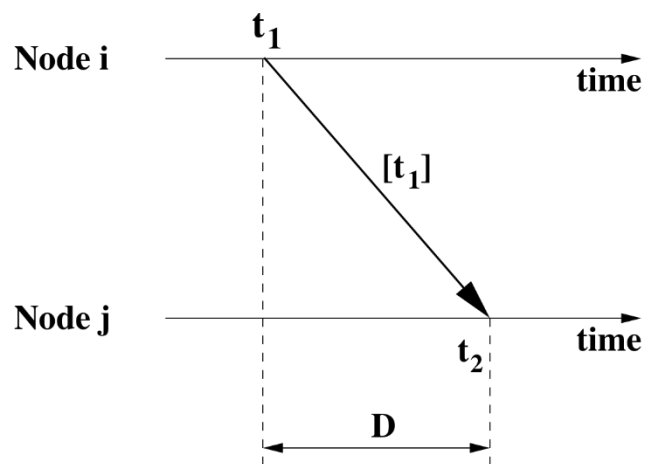
$$t_2 = t_1 + D + \delta \quad \leftarrow \text{时钟偏移}$$

3. 时间同步方法概述

时间同步通常是基于传感器节点之间的某种形式的信息交换，即：依赖于同步消息

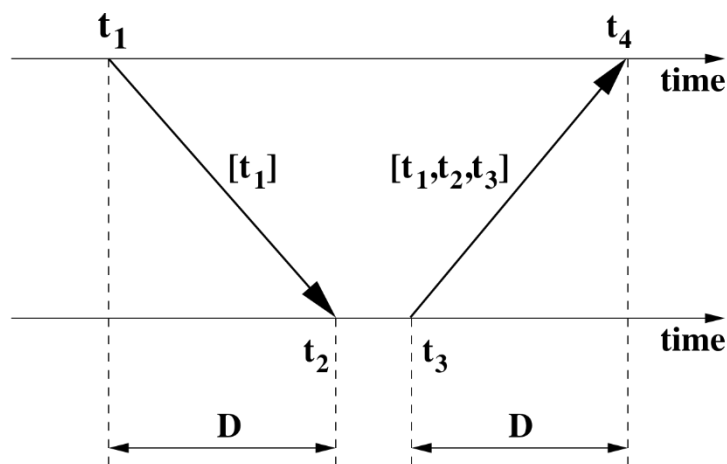
◆ 双向消息交换 Two-way message exchange

- 两个同步消息
 - 在单向同步消息交换的基础上，接收节点再回应一个消息，包含3个时间戳
- 前提假设：双向传播时延是一致的；在两次测量过程中，时钟漂移不变



$$t_2 = t_1 + D + \delta$$

单向消息交换



$$t_2 = t_1 + D + \delta$$

$$t_4 = t_3 + D - \delta \quad \leftarrow \text{时钟偏移}$$

双向消息交换

$$D = \frac{(t_2 - t_1) + (t_4 - t_3)}{2}$$

$$\text{offset} = \frac{(t_2 - t_1) - (t_4 - t_3)}{2}$$

两种方式下，谁是参考节点，谁是待同步节点？

3. 时间同步方法概述

时间同步通常是基于传感器节点之间的某种形式的信息交换，即：依赖于同步消息

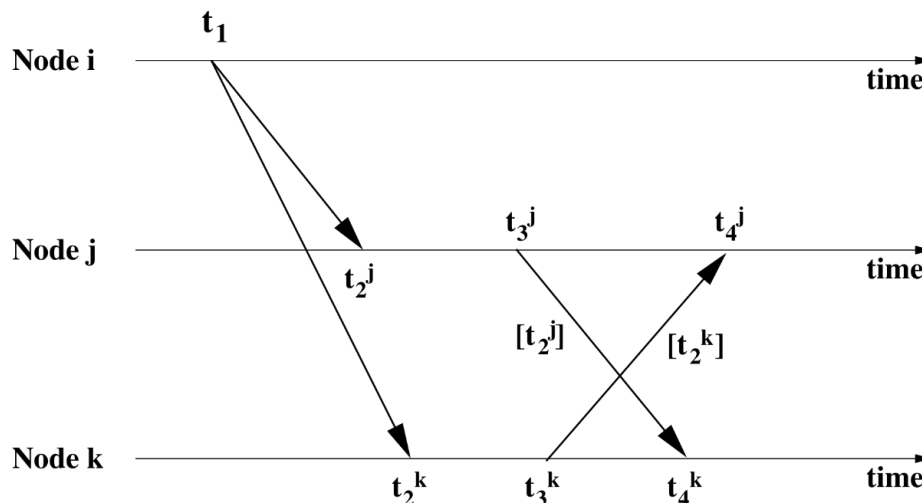
◆ 单向或双向消息交换属于发送端-接收端同步方法

◆ 接收端-接收端同步 Receiver-Receiver Synchronization

- 接收端-接收端同步：广播消息的多个接收端之间交换它们各自的接收时间，以此计算它们之间偏移

Receiver-receiver: multiple receivers of broadcast messages exchange their message arrival times to compute offsets among them

- 广播消息并不包含时间戳，而是利用广播消息到达接收节点的时间不同来使节点相互同步
- 如：2个接收，使用3个消息即可完成两个接收端的同步

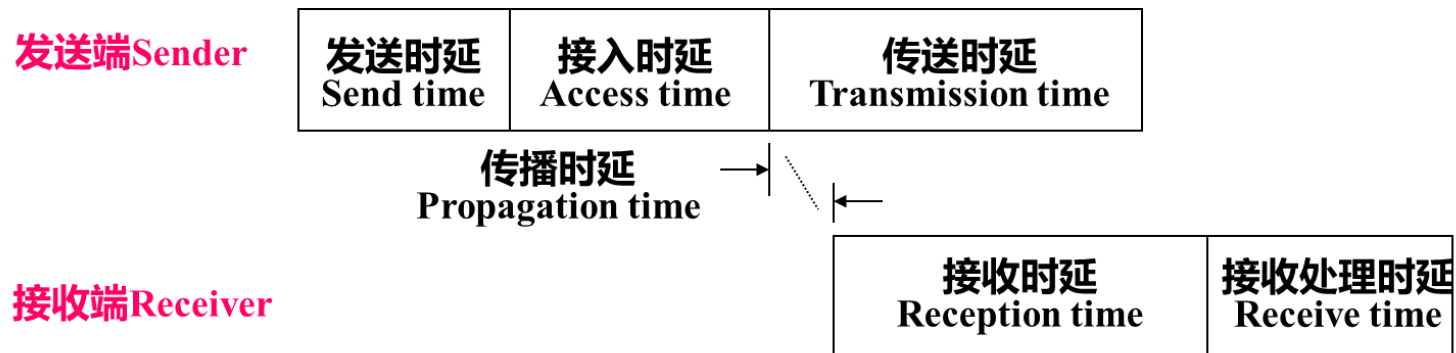


3. 时间同步方法概述

时间同步通常是基于传感器节点之间的某种形式的信息交换，即：依赖于同步消息

◆ 通信时延的不确定性对同步消息的影响

- 通信时延的不通定性对时间同步所能达到的精度有很大影响
- 同步消息的时延包含以下几个方面：



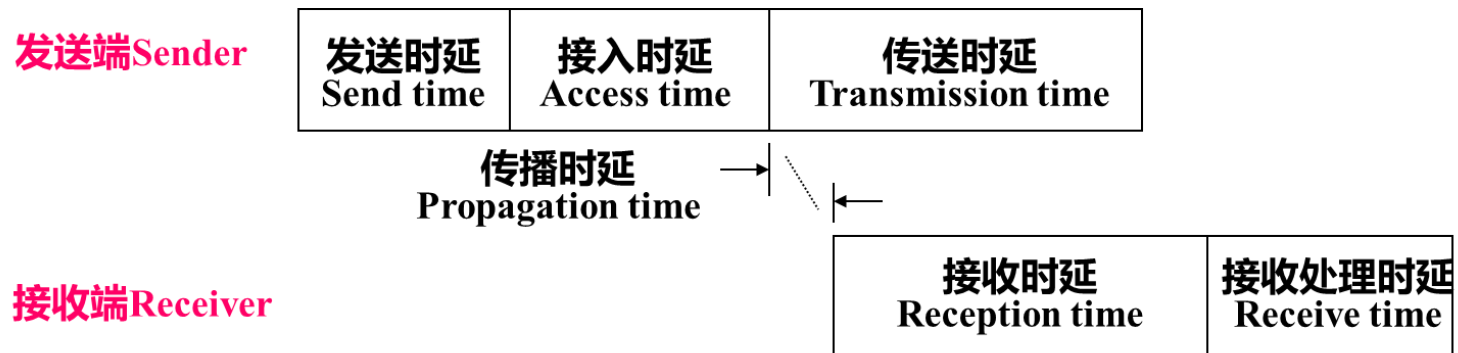
发送时延：发送节点生成同步消息和将消息发送到网络接口的时间，包括：操作系统活动、网络协议以及网络设备驱动时延。

3. 时间同步方法概述

时间同步通常是基于传感器节点之间的某种形式的信息交换，即：依赖于同步消息

◆ 通信时延的不确定性对同步消息的影响

- 通信时延的不通定性对时间同步所能达到的精度有很大影响
- 同步消息的时延包含以下几个方面：



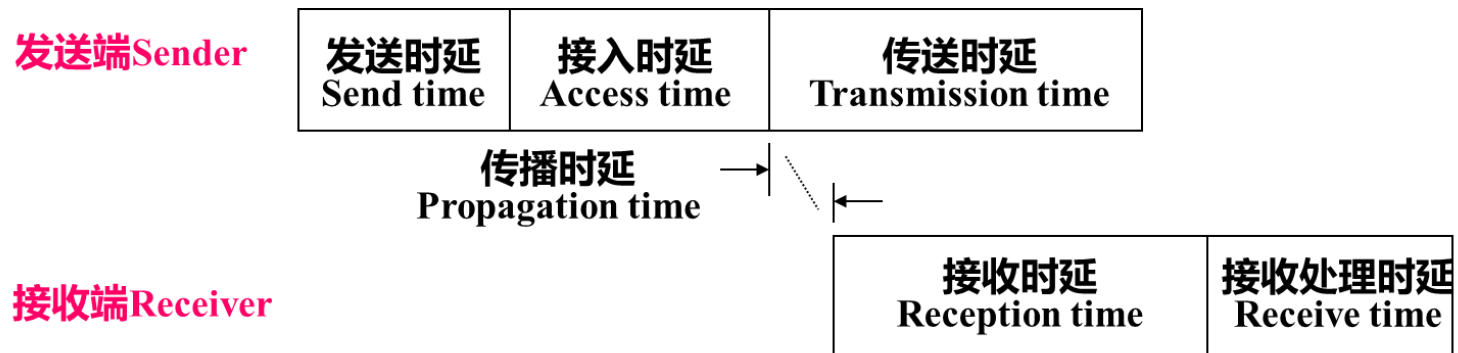
接入时延：发送节点访问物理信道的时延，主要取决于MAC协议，等待信道空闲或冲突引起的时延。

3. 时间同步方法概述

时间同步通常是基于传感器节点之间的某种形式的信息交换，即：依赖于同步消息

◆ 通信时延的不确定性对同步消息的影响

- 通信时延的不通定性对时间同步所能达到的精度有很大影响
- 同步消息的时延包含以下几个方面：



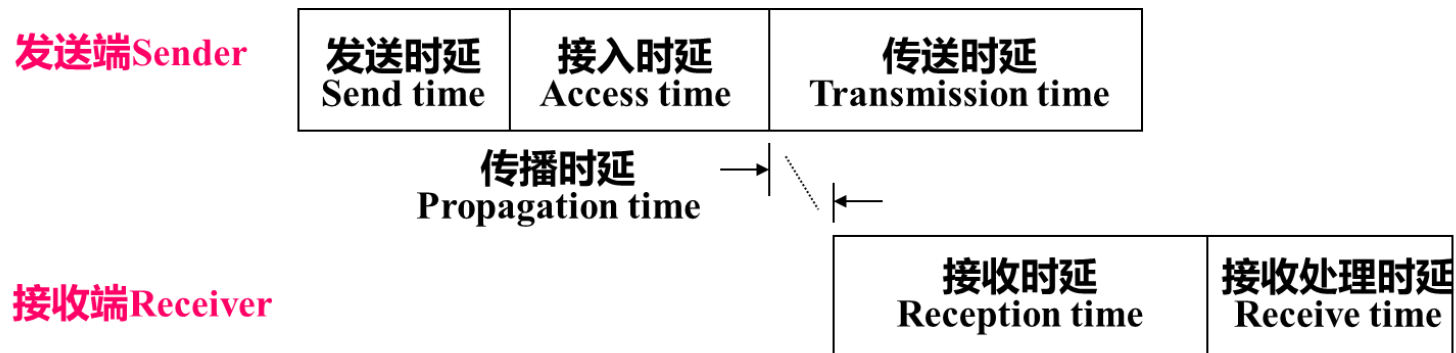
传播时延：实际传播信息的时延，无线信道通常可以忽略或设为常数。

3. 时间同步方法概述

时间同步通常是基于传感器节点之间的某种形式的信息交换，即：依赖于同步消息

◆ 通信时延的不确定性对同步消息的影响

- 通信时延的不通定性对时间同步所能达到的精度有很大影响
- 同步消息的时延包含以下几个方面：



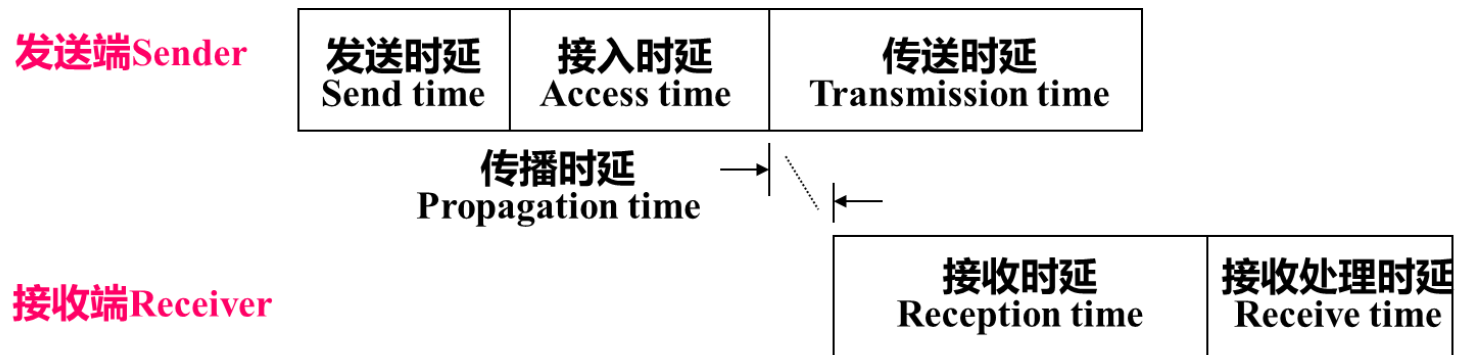
传送/接收时延：传输信息的时长，由消息长度与传输速率决定。

3. 时间同步方法概述

时间同步通常是基于传感器节点之间的某种形式的信息交换，即：依赖于同步消息

◆ 通信时延的不确定性对同步消息的影响

- 通信时延的不通定性对时间同步所能达到的精度有很大影响
- 同步消息的时延包含以下几个方面：



接收处理时延：接收端从介质层接收消息、处理消息以及将到达消息告知主机的时间，采用中断方式告知，通常小于发送时延。

3. 时间同步方法概述

时间同步通常是基于传感器节点之间的某种形式的信息交换，即：依赖于同步消息

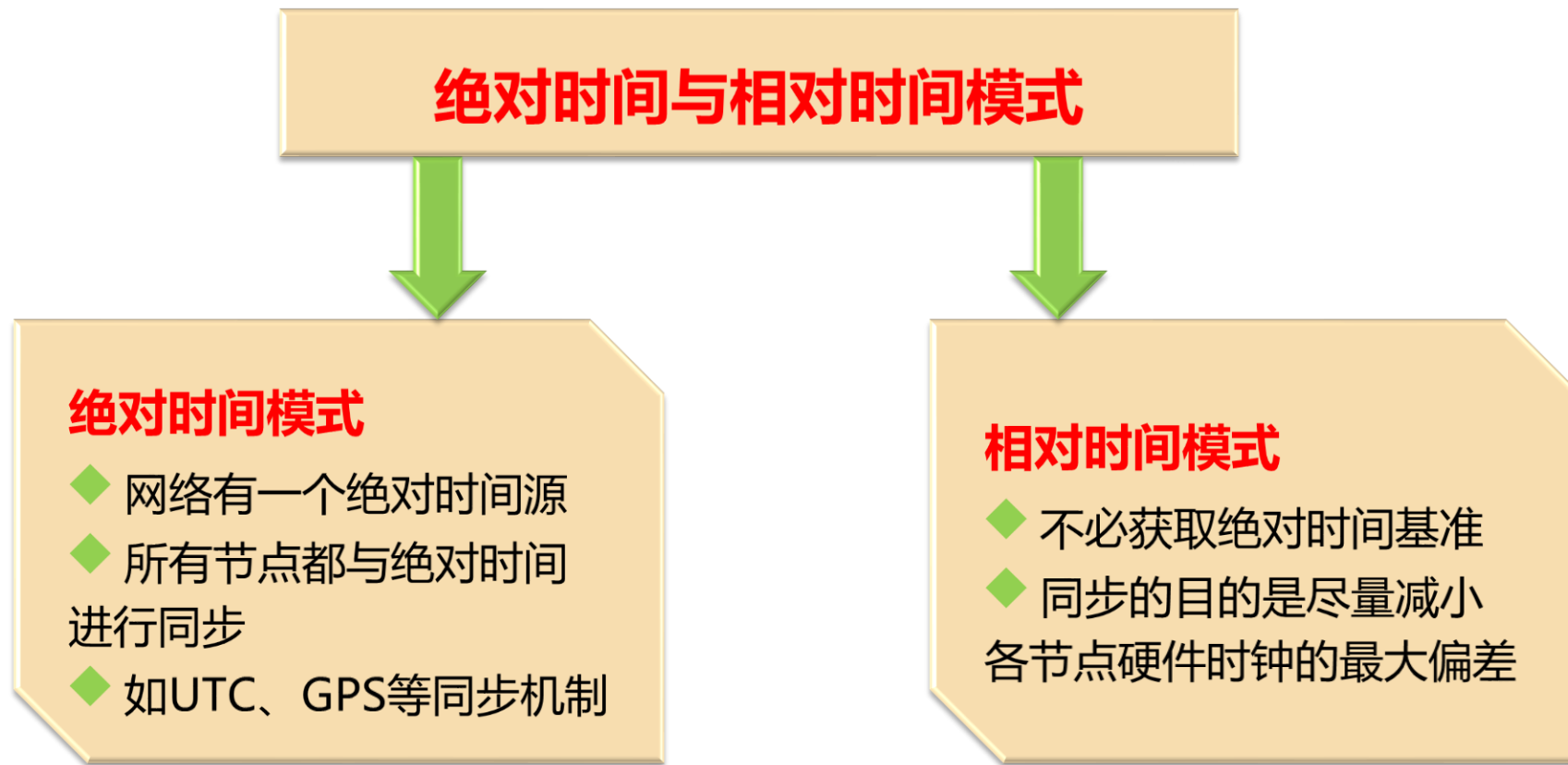
◆ 通信时延的不确定性对同步消息的影响

- 通信时延的不通定性对时间同步所能达到的精度有很大影响
- 同步消息的时延包含以下几个方面：

时间	典型值	特性
发送时延 / 接收处理时延 Send time / Receive time	0~100ms	不确定，依赖处理器负载、操作系统系统调用开销
接入时延 Access time	10~500ms	不确定，依赖信道负载。
传送时延 / 接收时延 Transmission time / Reception time	10~20ms	确定，依赖报文长度和发送速率。
传播时延 Propagation time	<1 μ s（距离<300米）	确定，依赖收发方物理距离和传播媒质特性。

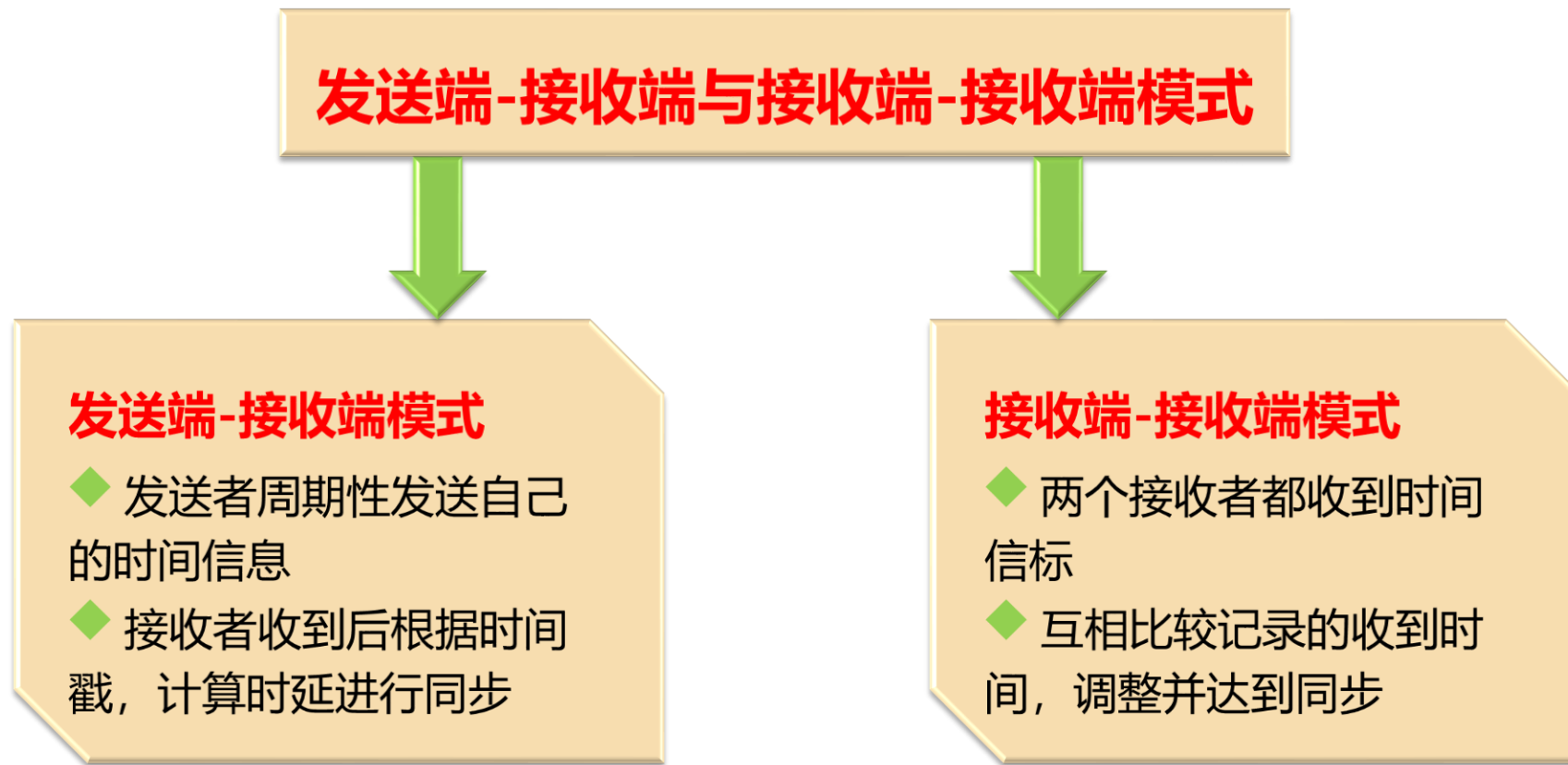
3. 时间同步方法概述

无线传感器网络的时间同步协议按照同步机制的不同可以分为以下几种类型:



3. 时间同步方法概述

无线传感器网络的时间同步协议按照同步机制的不同可以分为以下几种类型:



3. 时间同步方法概述

评价一个时间同步协议的可以参考以下性能参数

最大误差

指节点间最大相对误差，或者与外部标准时间的最大误差；网络规模越大，最大误差越大

同步时间

节点间进行时间同步所需要的时间，也指周期进行同步的时间间隔

同步范围

时间同步过程所包含的节点数量或区域范围，全网范围或部分区域

效率

达到同步精度与所耗费的代价的比值，代价一般指时间和能量

硬件代价

指为了完成某些协议的同步操作所需要的特殊硬件，会增加节点的成本和复杂性

3. 时间同步方法概述

典型时间同步协议

- ◆ 传输层（有线网络）同步协议：NTP协议
- ◆ 发送端-接收端双向消息同步协议：LTS协议、TPSN协议
- ◆ 发送端-接收端单向消息同步协议：DMTS协议、FTSP协议
- ◆ 接收端-接收端同步协议：RBS协议
- ◆ 绝对时间同步协议：GPS同步
- ◆ 协同互同步方法：基于脉冲耦合振荡器模型的同步协议

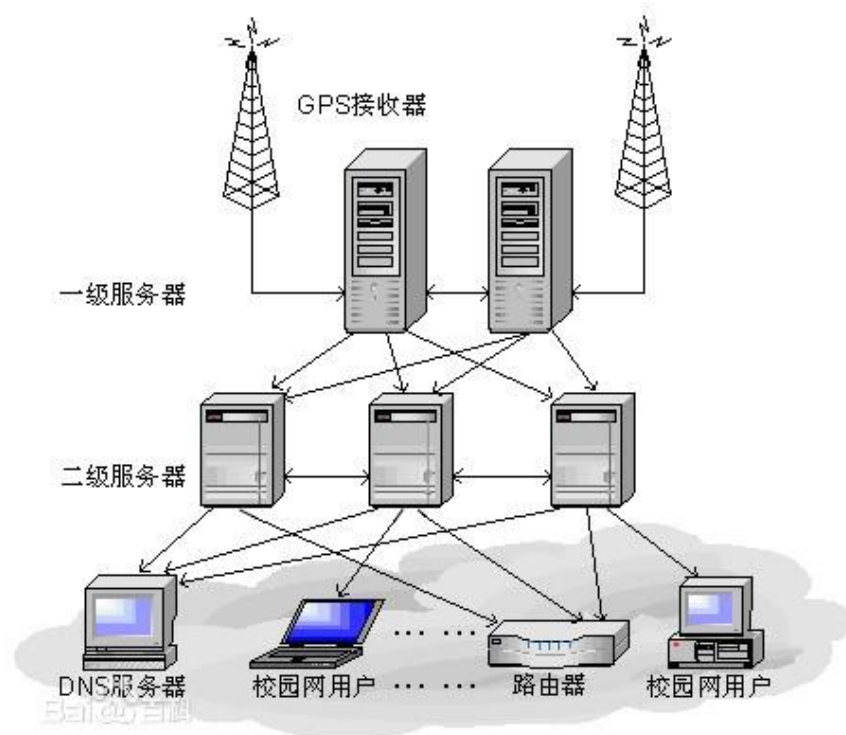
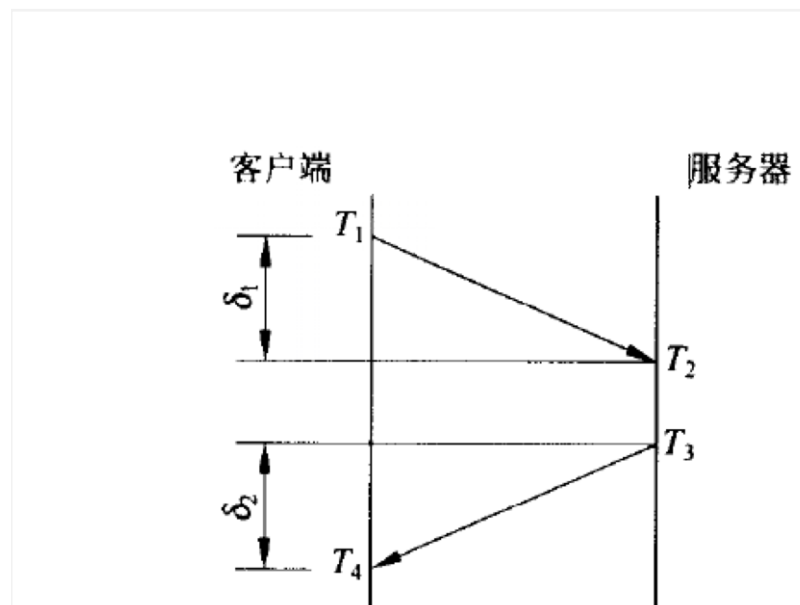
4

时间同步协议

4. 时间同步协议

NTP协议 (Network Time Protocol)

- ◆ NTP协议即网络时间协议，是一个集中式协议，同步精度达毫秒级。
- ◆ NTP基于节点间的双向握手来预测节点间时延并计算相对的时间偏差：需同步的终端发送时间请求，服务器回应包含时间信息的应答消息。
- ◆ 协议假定两个主机的传播时延是一致的。



4. 时间同步协议

NTP协议 (Network Time Protocol)

◆ NTP协议特点:

- NTP协议通过频繁交换消息来校准时钟频率偏差带来的误差, 能耗较高
- NTP基准服务器间的时间同步需要基础设施的协助

◆ NTP协议不适合WSN

- WSN节点能量有限, 无法满足NTP频繁通信需求
- WSN时间同步目的实现局部最优, 而NTP实现全局同步, 属于集中式协议

! 但可以借鉴其双向握手机制通过双向消息交互实现时延的计算

4. 时间同步协议

LTS同步协议: 一种基于树的轻量级同步协议. **Lightweight time synchronization**

特点: 低开销, 基于双向消息交换

LTS同步协议

- 发送端-接收端同步协议: 双向消息交换
- 可用于集中式或分布式多跳同步算法

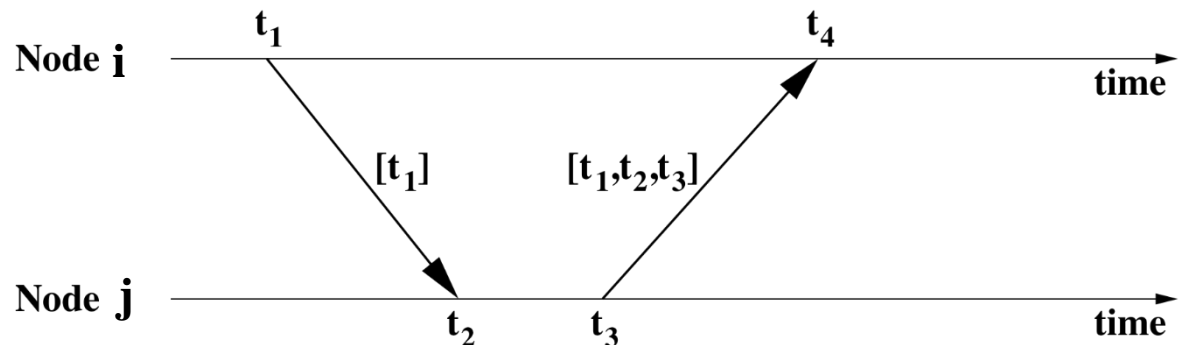
复习: 双向消息交换

1. message from i to j, containing time stamp t_1 (i's clock)
2. message from j to i, containing t_1 (i's clock) and t_2, t_3 (j's clock)
3. assuming message delay D



时钟偏移

$$offset = \frac{(t_2 - t_1) - (t_4 - t_3)}{2}$$



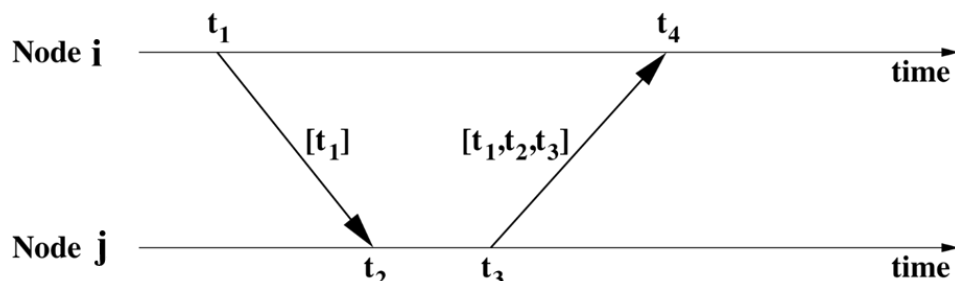
4. 时间同步协议

LTS同步协议: 一种基于树的轻量级同步协议. **Lightweight time synchronization**

特点: 低开销, 基于双向消息交换

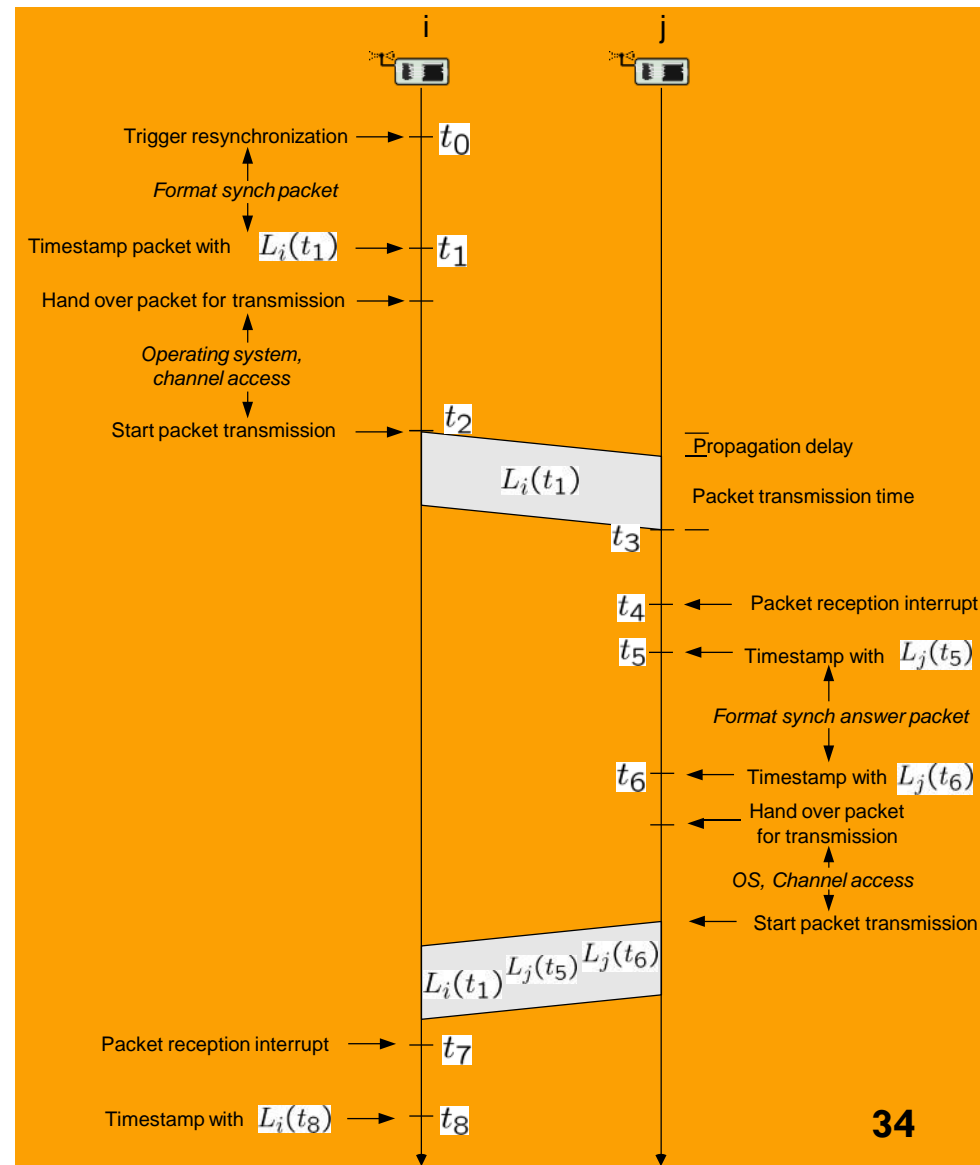
LTS同步协议

- 发送端-接收端同步协议: 双向消息交换
- 可用于集中式或分布式多跳同步算法



详细传输示例分析:

1. 节点i想要估计真实的时钟偏移 $O = \Delta(t_1) = L_i(t_1) - L_j(t_1)$, L 时本地时钟
2. 算法实质上是估计的 $\Delta(t_5)$, 并假定 $\Delta(t_5) = \Delta(t_1)$
3. 传播时延 τ , 节点i和节点j之间的传输时延 t_p



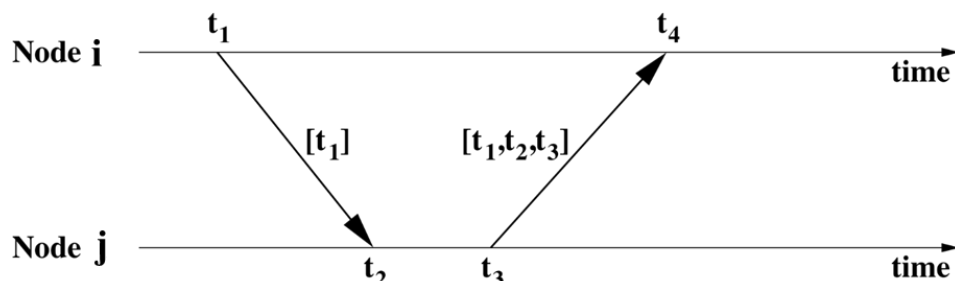
4. 时间同步协议

LTS同步协议: 一种基于树的轻量级同步协议. **Lightweight time synchronization**

特点: 低开销, 基于双向消息交换

LTS同步协议

- 发送端-接收端同步协议: 双向消息交换
- 可用于集中式或分布式多跳同步算法

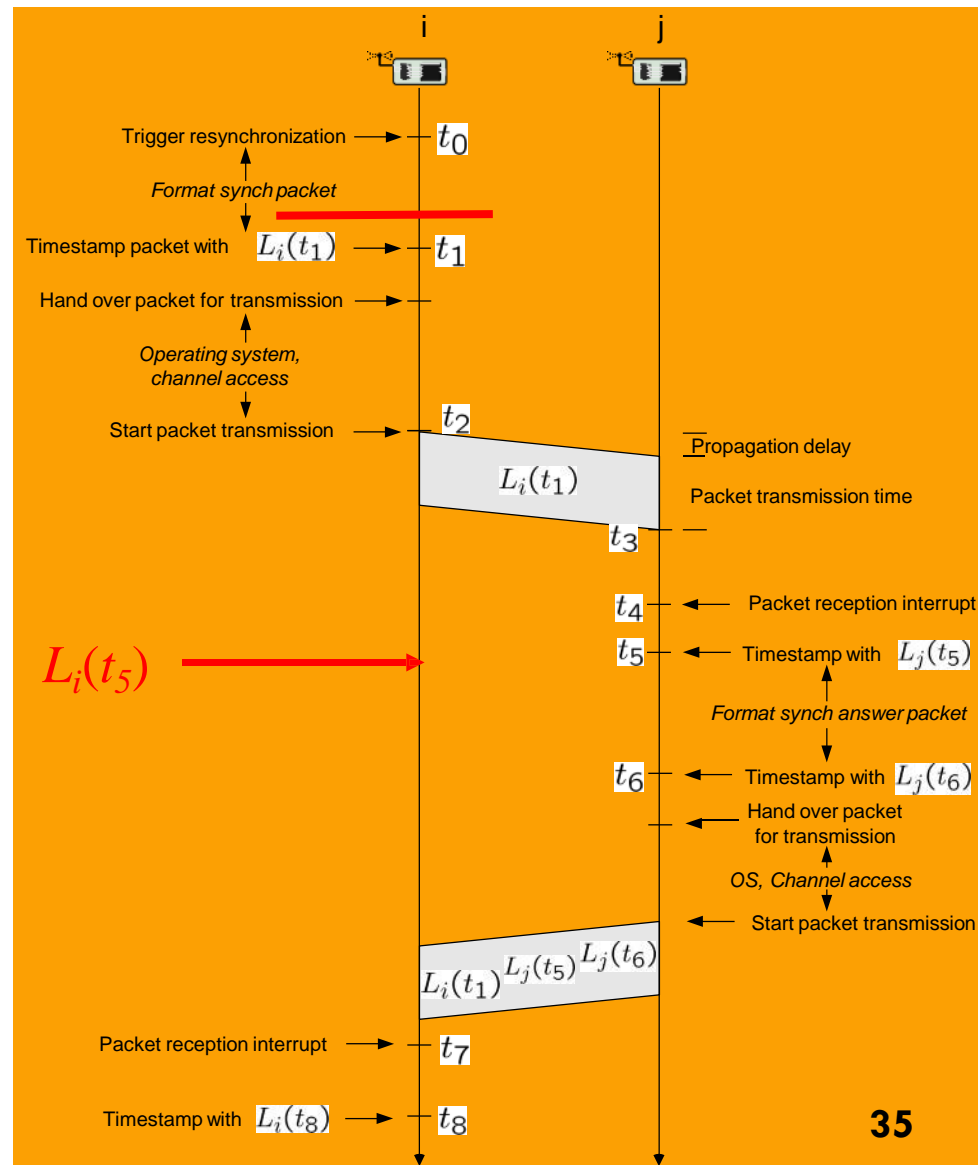


详细传输示例分析:

1. 节点i想要估计真实的时钟偏移 $O = \Delta(t_1) = L_i(t_1) - L_j(t_1)$, L 时本地时钟
2. 算法实质上是估计的 $\Delta(t_5)$, 并假定 $\Delta(t_5) = \Delta(t_1)$
3. 传播时延 τ , 节点i和节点j之间的传输时延 t_p
4. 时延不确定性在区间内: $[L_i(t_1) + \tau + t_p, L_i(t_8) - \tau - t_p - (L_j(t_6) - L_j(t_5))]$

$$t_5 \geq t_1 + \tau + t_p$$

$$t_5 \leq t_8 - \tau - t_p$$



4. 时间同步协议

LTS同步协议: 一种基于树的轻量级同步协议. Lightweight time synchronization

特点: 低开销, 基于双向消息交换

LTS同步协议

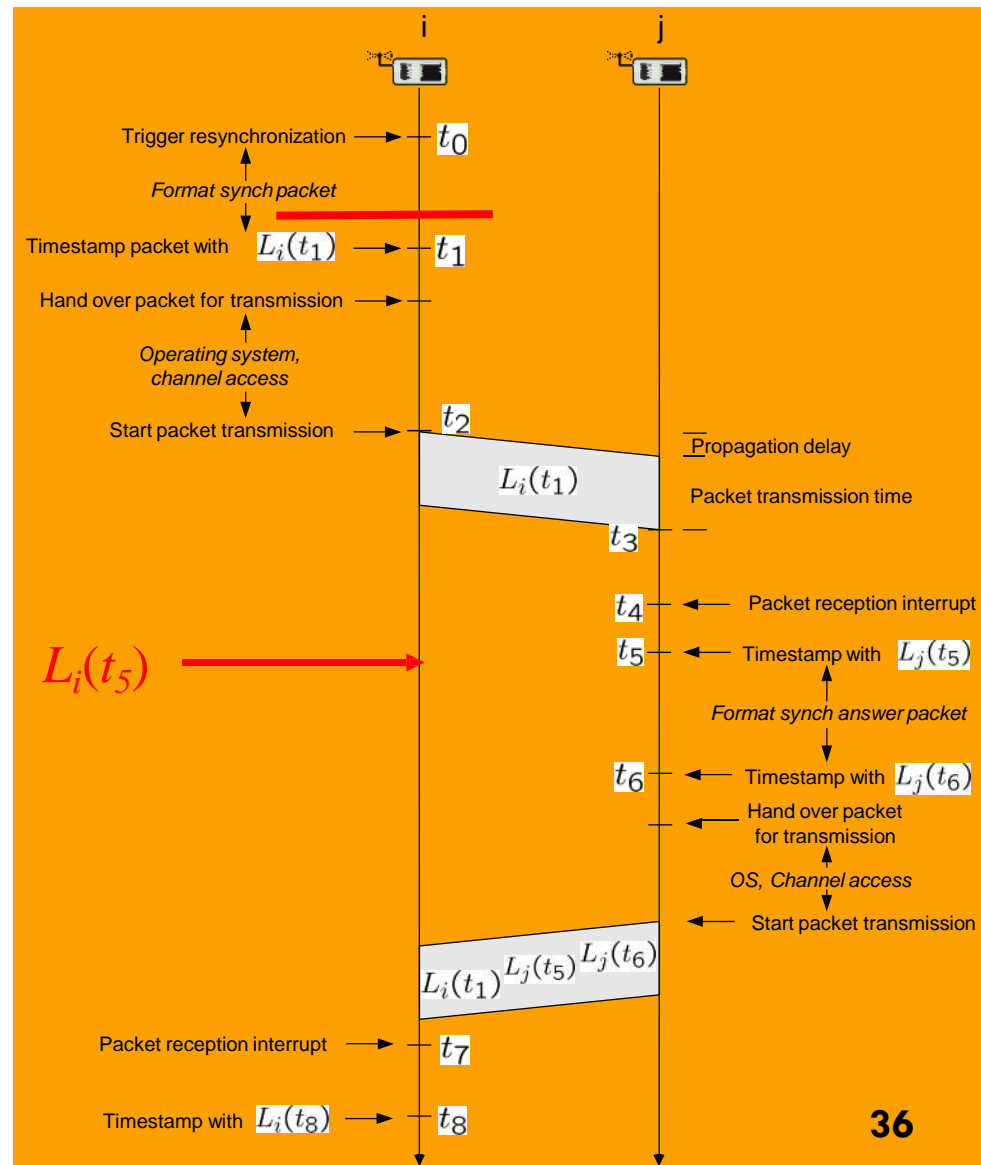
- 发送端-接收端同步协议: 双向消息交换
- 可用于集中式或分布式多跳同步算法

详细传输示例分析:

1. 节点i想要估计真实的时钟偏移 $O = \Delta(t_1) = L_i(t_1) - L_j(t_1)$, L 时本地时钟
2. 算法实质上是估计的 $\Delta(t_5)$, 并假定 $\Delta(t_5) = \Delta(t_1)$
3. 传播时延 τ , 节点i和节点j之间的传输时延 t_p
4. 时延不确定性在区间内: $[L_i(t_1) + \tau + t_p, L_i(t_8) - \tau - t_p - (L_j(t_6) - L_j(t_5))]$
5. 在以上假设下, 节点i可以估计 $L_i(t_5)$, 并继而计算出时钟偏移 O :

$$L_i(t_5) = \frac{L_i(t_1) + \tau + t_p + L_i(t_8) - \tau - t_p - (L_j(t_6) - L_j(t_5))}{2}$$

$$O = \Delta(t_5) = L_i(t_5) - L_j(t_5) = \frac{L_i(t_8) + L_i(t_1) - L_j(t_6) - L_j(t_5)}{2}$$



4. 时间同步协议

LTS同步协议: 一种基于树的轻量级同步协议. Lightweight time synchronization

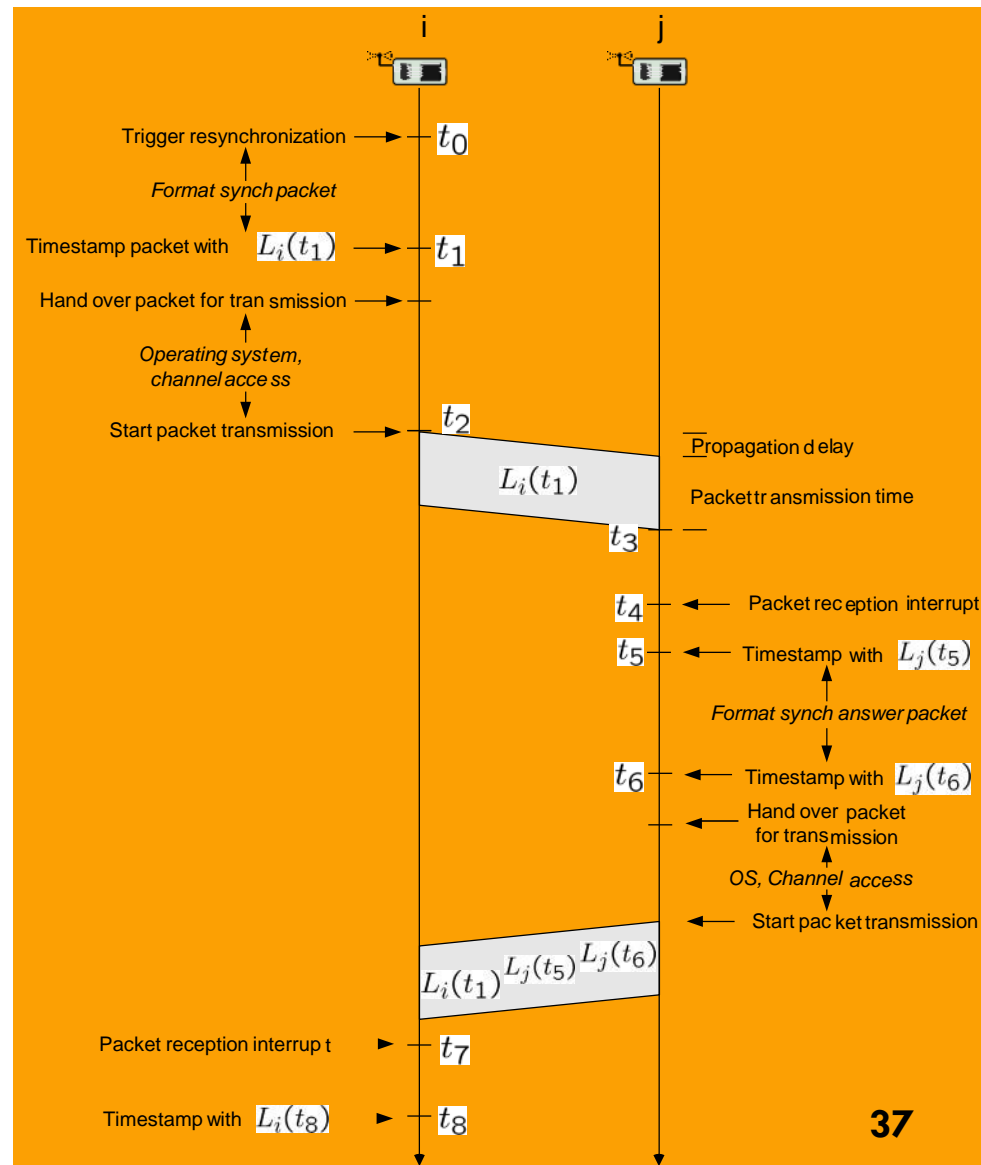
特点: 低开销, 基于双向消息交换

LTS同步协议

- 发送端-接收端同步协议: 双向消息交换
- 可用于集中式或分布式多跳同步算法

详细传输示例分析:

1. 节点i想要估计真实的时钟偏移 $O = \Delta(t_1) = L_i(t_1) - L_j(t_1)$, L 为本地时钟
2. 算法实质上是估计的 $\Delta(t_5)$, 并假定 $\Delta(t_5) = \Delta(t_1)$
3. 传播时延 τ , 节点i和节点j之间的传输时延 t_p
4. 时延不确定性在区间内: $[L_i(t_1) + \tau + t_p, L_i(t_8) - \tau - t_p - (L_j(t_6) - L_j(t_5))]$
5. 在以上假设下, 节点i可以估计 $L_i(t_5)$, 并继而计算出时钟偏移 O
6. 不确定性来源 (影响了精度):
 1. 介质接入
 2. 接收数据包后的中断时延
 3. 数据包中断和时间戳操作之间的时延
 4. 操作系统和协议带来的时延



4. 时间同步协议

LTS同步协议: 一种基于树的轻量级同步协议. Lightweight time synchronization

特点: 低开销, 基于双向消息交换

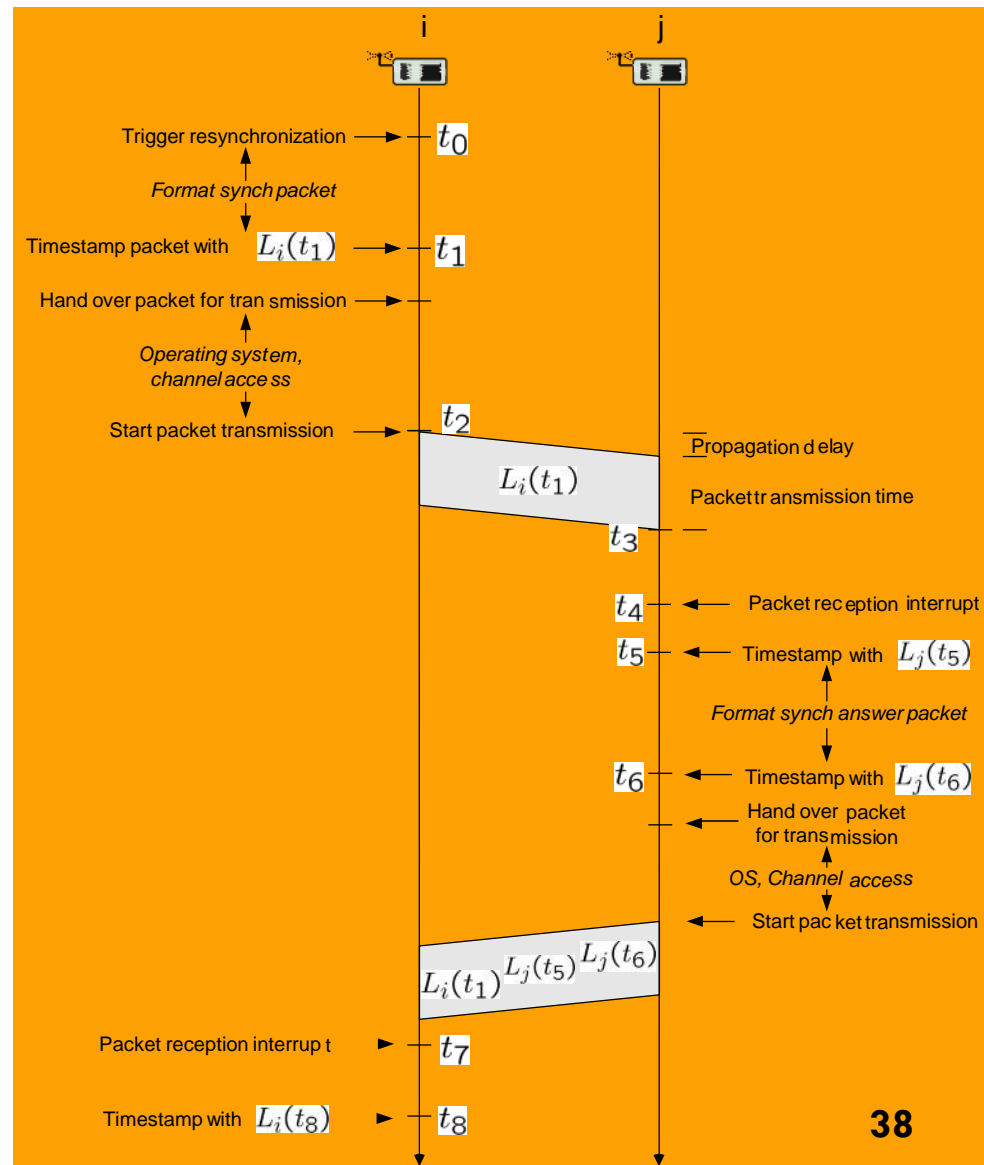
LTS同步协议

- 发送端-接收端同步协议: 双向消息交换
- 可用于集中式或分布式多跳同步算法

详细传输示例分析:

不确定性来源 (影响了精度):

1. 介质接入
2. 接收数据包后的中断时延
3. 数据包中断和时间戳操作之间的时延
4. 操作系统和协议带来的时延
5. 改进方法:
 - 在发送节点i上, 在MAC时延后, 第一个bit传输前立即给分组包打时间戳, 以消除MAC时延、操作系统时延和协议栈时延
 - 在接收节点j上, 尽早给接收包打上时间戳, 如: 在接收中断过程中, 从而, 仅仅只剩下中断时延



4. 时间同步协议

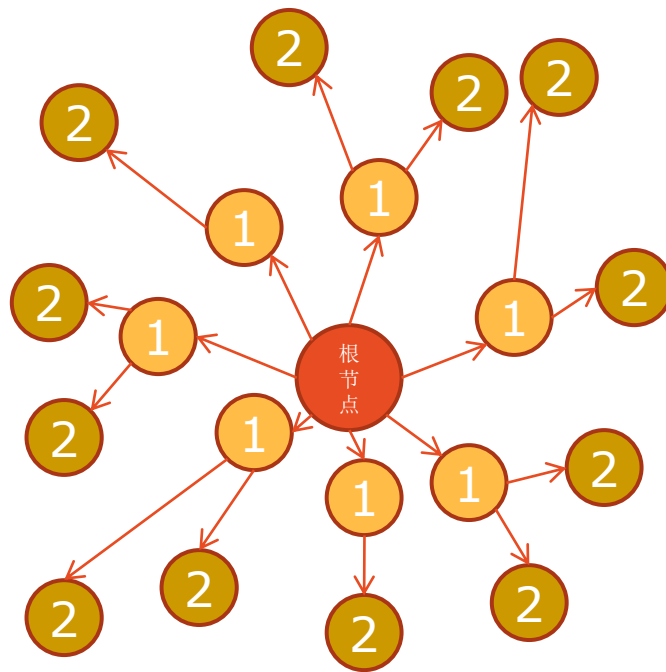
LTS同步协议: 一种基于树的轻量级同步协议. **Lightweight time synchronization**

特点: 低开销, 基于双向消息交换

LTS同步协议

◆ 集中式多跳LTS

- 参考节点作为包含所有节点的生成树的根
- 使用广度优先搜索构建树
- 一旦树建立完成, 参考节点首先发起同步, 继而与叶子节点通过两两成对同步完成网络级同步过程
 - 成对同步的误差会累积: 要尽量使树的深度较小
 - 成对同步的开销: 3个消息
 - 网络级同步的开销: $3n - 3$ 个消息, n 为边的数量



4. 时间同步协议

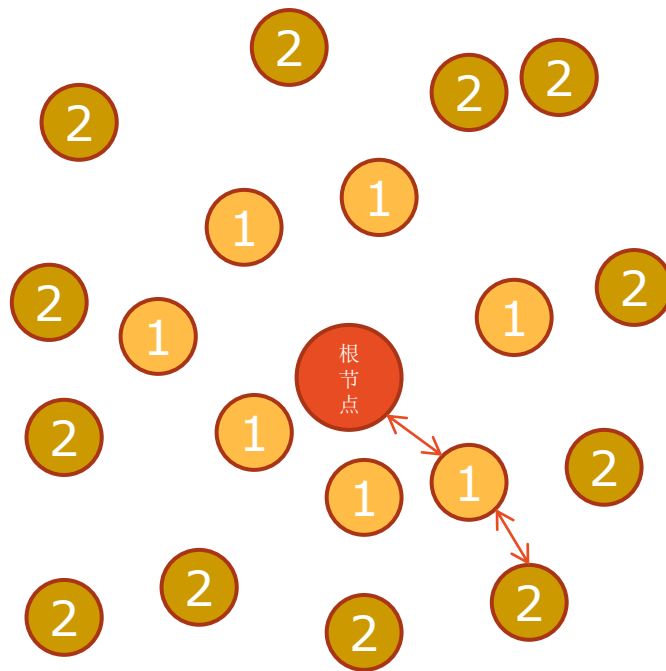
LTS同步协议: 一种基于树的轻量级同步协议. **Lightweight time synchronization**

特点: 低开销, 基于双向消息交换

LTS同步协议

◆ 分布式多跳LTS

- 基于事件的机制, 即每个节点只在发送数据时才进行同步;
- 同步精度随距离增大而降低;
- 每个节点都可以发起同步, 节点会询问邻居节点潜在的同步请求, 即: 如果邻居有同步要求, 节点会与其同步, 而不是与参考节点同步
- 同步周期由同步需求节点自身确定, 需考虑的因素包括: 它到参考节点的距离、时钟漂移、上次同步的时间等等



4. 时间同步协议

TPSN同步协议: 一种基于树结构的同步协议. **Timing-sync Protocol for Sensor Networks**

特点: 分层拓扑, 基于双向消息交换

TPSN同步协议

- ◆ 与LTS类似, 同属发送端-接收端同步协议, 并基于双向消息交换设计整个过程 sender-receiver technique
- ◆ 使用树结构组织分层整个网络
- ◆ 协议步骤分两个阶段:
 - 发现阶段 discovery phase
 - 同步阶段 synchronization phase

4. 时间同步协议

TPSN同步协议: 一种基于树结构的同步协议. **Timing-sync Protocol for Sensor Networks**

特点: 分层拓扑, 基于双向消息交换

TPSN同步协议

TPSN同步协议: 发现阶段 **Discovery phase**

- ◆ 目标: 创建网络的分层拓扑结构, 并为每个节点分配一个级别 **establish hierarchical topology**
 - 根节点为级别0
- ◆ 总体步骤:
 1. 根节点启动发现过程: 通过广播一个**level_discovery**消息; 该消息包含了级别、发送者身份
 - 每个节点都拥有一个唯一的ID
 2. 接收节点利用收到的**level_discovery**消息, 决定自己的级别 (级别加1), 并广播**level_discovery**消息包含该节点自身的级别和身份
 3. 重复步骤2, 直到所有节点都确定了自己的级别
 - 若某节点已确定了自己的级别, 当再次收到**level_discovery**消息时, 将直接丢弃
 - 若某节点没有被分配到级别, 它将向邻居节点发送**level_request**消息, 邻居将回复它们的级别; 该节点在所有收到的回复消息中, 选择最小级别值, 加1, 决定自身的级别
 - 若某节点 (级别*i*) 发现没有*i-1*级邻节点, 它也会发送**level_request**消息, 重新插入到树结构中
 - 若根节点失效, 级别1的节点, 将执行领导选举算法, 选出根节点, 再重新开始新的发现过程 **leader election algorithm**

4. 时间同步协议

TPSN同步协议: 一种基于树结构的同步协议. **Timing-sync Protocol for Sensor Networks**

特点: 分层拓扑, 基于双向消息交换

TPSN同步协议

TPSN同步协议: 同步阶段 **Synchronization phase**

◆ 目标: 沿分层树结构边缘, 使用成对同步机制, 达到网络级同步 **pairwise synchronization along the edges of hierarchical structure**

◆ 成对同步步骤:

每个级别*i*的节点都与级别*i-1*的节点进行时钟同步, 整个同步过程类似于LTS协议 (*j*是*k*的下级节点, 即, 待同步节点)

- 节点*j*在时刻*t*₁发出一个同步消息, 包含级别和时间戳
- 节点*k*在时刻*t*₂收到消息, 将在*t*₃时刻回应ACK, 包含*t*₁, *t*₂, *t*₃和级别
- 节点*j*在时刻*t*₄收到ACK
- 节点*j*计算时钟偏移和传播时延

时钟偏移和传播时延计算:

$$D = \frac{(t_2 - t_1) + (t_4 - t_3)}{2}$$

$$offset = \frac{(t_2 - t_1) - (t_4 - t_3)}{2}$$

4. 时间同步协议

TPSN同步协议: 一种基于树结构的同步协议. **Timing-sync Protocol for Sensor Networks**

特点: 分层拓扑, 基于双向消息交换

TPSN同步协议

TPSN同步协议: 同步阶段 **Synchronization phase**

- ◆ 目标: 沿分层树结构边缘, 使用成对同步机制, 达到网络级同步 **pairwise synchronization along the edges of hierarchical structure**
- ◆ 总体步骤:
 - 同步阶段是从根节点开始的: 根节点发送 **time_sync** 分组包
 - 随机等待一个时间间隔后 (以降低冲突), 级别1的节点开始初始化双向消息交换机制, 即: 与根节点进行双向消息交换
 - 当级别1节点收到根节点的确认消息时, 会计算自身时钟偏移, 并调整自身时钟
 - 重复以双向消息交换机制, 扩散至整个网络

4. 时间同步协议

TPSN同步协议: 一种基于树结构的同步协议. Timing-sync Protocol for Sensor Networks

特点: 分层拓扑, 基于双向消息交换

TPSN同步协议: 同步阶段 Synchronization phase

◆ 同步误差来源:

- 分层结构的深度
- 端到端时延

◆ 改进:

- 在MAC层发送时打时间戳减少时延

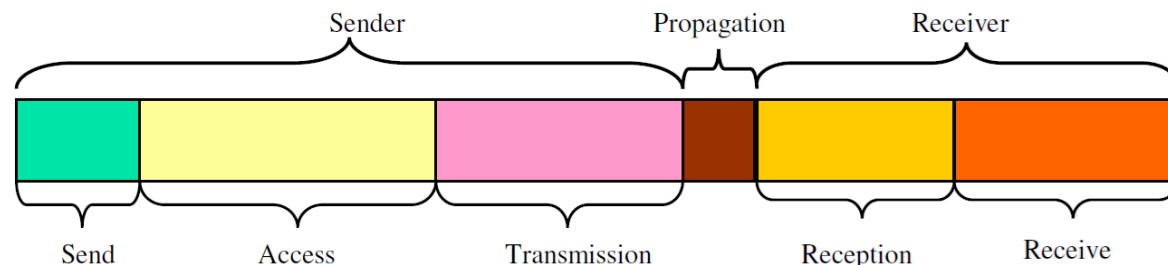
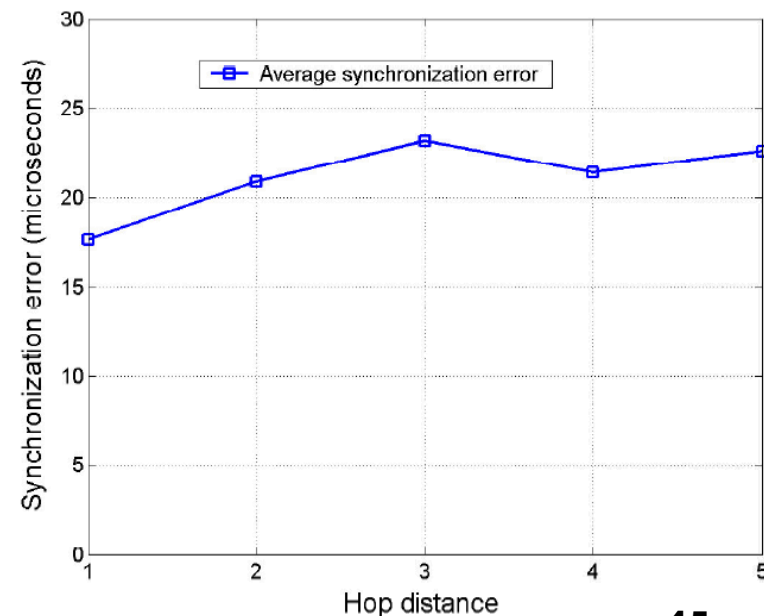


Figure 2: Decomposition of packet delay over a wireless link

Table 2: Statistics of synchronization error over multihop (only magnitude)

	1 hop distance	2 hop distance	3 hop distance	4 hop distance	5 hop distance
Average error (in μs)	17.61	20.91	23.23	21.436	22.66
Worst case error (in μs)	45.2	51.6	66.8	64	73.6
Best case error (in μs)	0	0	2.8	0	0
Percentage of time error is less than or equal to average error	62	57	63	54	64



4. 时间同步协议

TPSN同步协议: 一种基于树结构的同步协议. **Timing-sync Protocol for Sensor Networks**

特点: 分层拓扑, 基于双向消息交换

TPSN同步协议:

◆ 优点:

- 类似于传统的NTP时间同步协议, 目的是提供传感器网络全网范围内节点间的时间同步。
- 如果需要长时间的全网节点时间同步, 可周期性执行TPSN时间同步协议进行重同步。
- 在MAC层消息开始发送到无线信道时才添加时间信标, 消除了访问时间带来的误差
- 利用双向交换信息计算消息的平均延迟, **精度相对较高**

◆ 缺点:

- 全网同步, 开销较大, 能耗较大
- 节点移动时, 需初始化层次发现步骤, 扩展性差
- 节点失效(尤其是靠近根节点的节点失效)会导致同步错误, 并在网络扩散
- 多跳网络累积误差较大

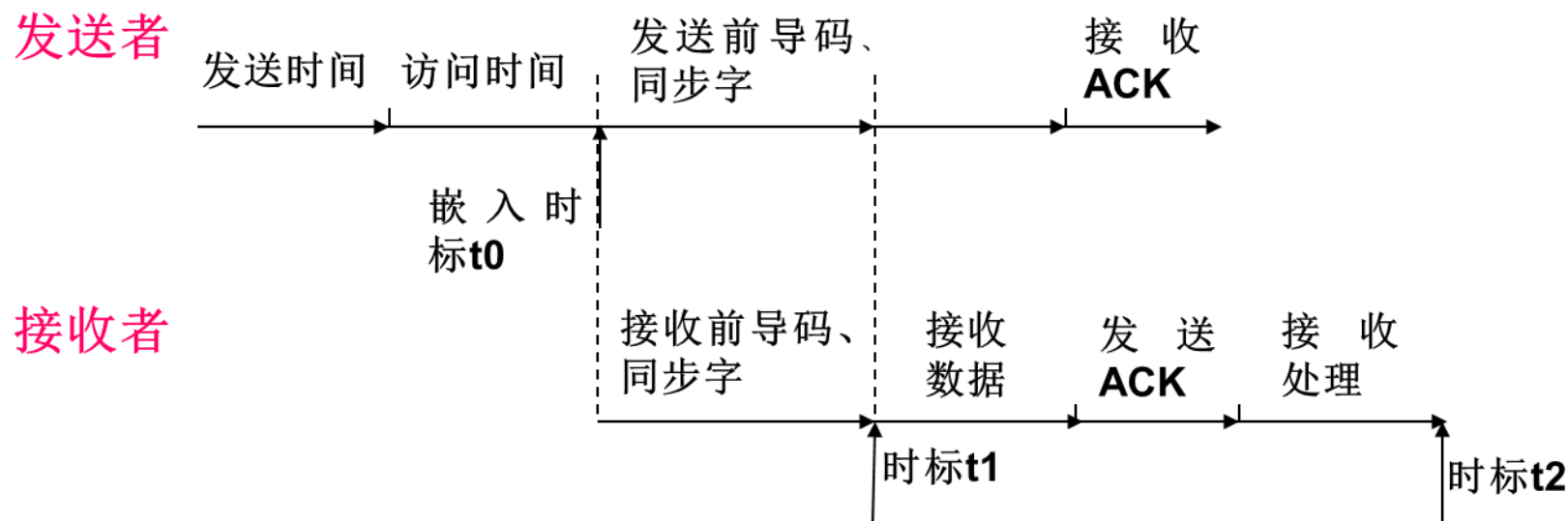
4. 时间同步协议

DMTS延迟测量同步协议 **Delay Measurement Time Synchronization**

特点：发送端-接收端，单向消息同步协议。

同步机制

- ◆ 简易思路：设定接收端时间 $t = \text{发送端时间 } t_0$ ，实现同步，但未考虑传输延迟
- ◆ 改进思路：接收端时间 $t = \text{发送端时间 } t_0 + \text{传输延迟}$
 - 前导码和同步字的发送时间 $= nT$ (n 为bit位数, T 为传输单个bit的时间)
 - 接收处理延迟 $= t_2 - t_1$
 - 忽略无线信号的传播延迟
 - 接收端时钟 $t = t_0 + nT + (t_2 - t_1)$



4. 时间同步协议

DMTS延迟测量同步协议 **Delay Measurement Time Synchronization**

特点：发送端-接收端，单向消息同步协议。

◆ DMTS同步协议优缺点分析：

- DMTS是一种灵活的、轻量的和能量高效的能够实现全部网络节点时间同步的机制。（DMTS机制在多跳网络中采用层次型分级结构实现全网内所有节点的时间同步）
- 与双向消息机制相比，DMTS机制的计算开销小，需要传输的消息条数少。
- 能够与外部世界标准时间同步，但未考虑传播时间、编解码时间，同步精度相对较低。
- DMTS在实现复杂度、能量高效与同步精度之间进行了折中，能够应用在对时间同步要求不是非常高的网络中。

◆ 进一步改进：FTSP洪泛时间同步协议（**Flooding Time Synchronization Protocol**）：

- 发送端-接收端，单向消息同步。
- 通过MAC层时间戳技术，在单个报文中的不同位置打入多个时间戳。
- 根据单个报文中的多个时间戳，可对中断等待时间进行补偿，通过线性回归补偿时钟漂移。
- 进一步通过洪泛方式，实现全网节点的时间同步。

4. 时间同步协议

RBS参考广播同步协议 Reference Broadcast Synchronization

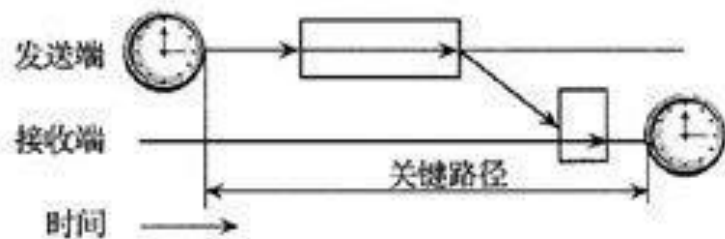
特点：接收端-接收端同步协议。

◆ 同步机制：

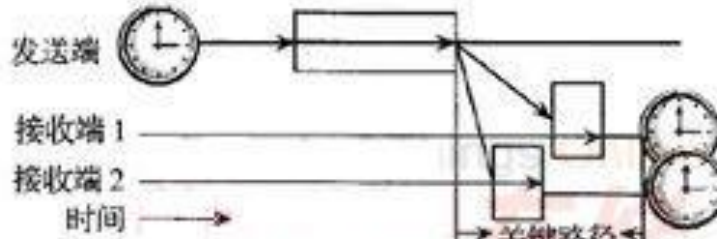
- 发送节点广播的同步消息不含有时间戳，接收节点记录接收消息的时间并进行相互比较(多次通信)，估计相对的时间偏差并进行同步。
- 对于高频信号，可忽略传播时间带来的误差；对于声信号，则传播时间必须考虑在内。

◆ 同步过程：

- 发送者发出“信标分组”数据包，广播域内节点“同时”接收（忽略传播时间差）
- 接收节点分别记录接收时间
- 接收节点交换各自记录的接收时间，计算出差值
- 存在差值的节点根据差值更改本地时间，实现接收节点之间的同步



(a) 传统时钟同步系统



(b) RBS 时钟同步系统

4. 时间同步协议

RBS参考广播同步协议 **Reference Broadcast Synchronization**

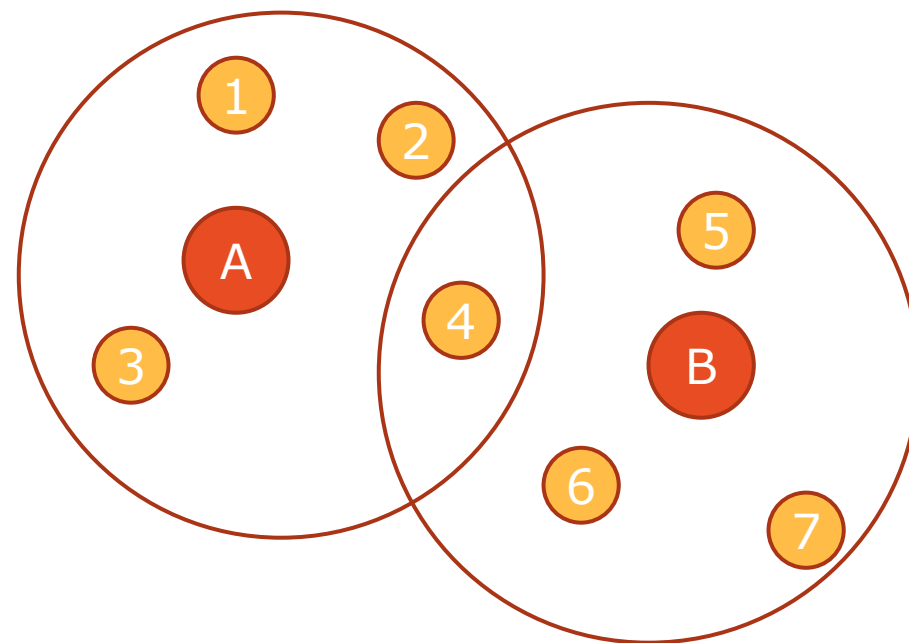
特点：接收端-接收端同步协议。

◆ 多跳同步机制：

- A发送信标分组后，完成与节点1、2、3、4的时间同步
- B发送信标分组后，完成与节点4、5、6、7的时间同步
- 节点4位于两个广播域的交集内，所以通过节点4可以同步两个广播域内的节点

◆ 多跳同步举例：

- 发生在节点1和节点7附近的两个事件，分别记为E1和E7
- 假设节点A和节点B分别在 P_a 和 P_b 时间点发送信标分组
- 节点1在收到节点A发送的分组后2秒观察到事件E1
- 节点7在观察到事件E7后4秒收到节点B发送的分组
- 其他节点从节点4知道节点A发送分组比节点B晚10秒
- $P_a = P_b + 10$
- $E1 = E7 + 4 + 10 + 2 = E7 + 16$



4. 时间同步协议

RBS参考广播同步协议 Reference Broadcast Synchronization

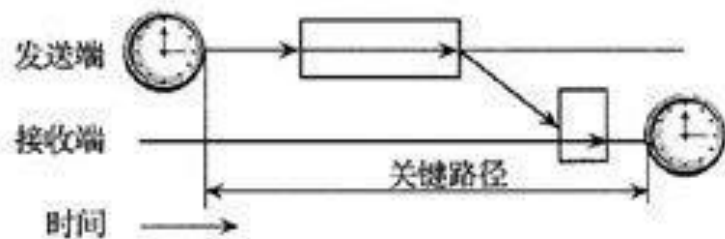
特点：接收端-接收端同步协议。

◆ 优点：

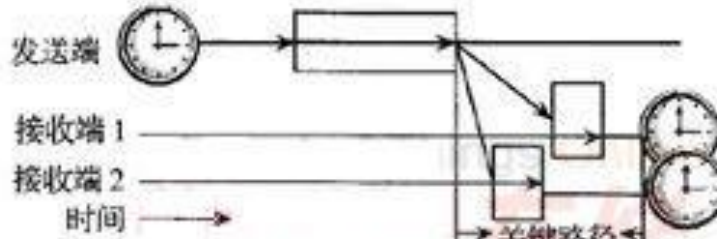
- RBS机制利用信道的广播特性来同步接收节点时间，去除了时间同步误差中所有发送节点引入的部分，减少了同步误差，提高了效率
- 轻量级，自适应
- 精度优于DMTS

◆ 缺点：

- 同步精度受“接收端接收时间差”（节点间存在硬件和软件差异）影响较大
- 用于多跳网络的RBS机制需要依赖有效的分簇方法，保证簇之间有共同节点以便进行簇间时间同步
- 节点间通信量较大，增加了能量消耗，节点间交换信息导致传输碰撞概率增大



(a) 传统时钟同步系统



(b) RBS 时钟同步系统

4. 时间同步协议

绝对时间同步协议：GPS或北斗授时

特点：高精度（100ns），高功耗，硬件要求.

- 从根本上解决了人类在地球上的导航与定位问题。
- 每颗卫星上配备有高精度的铷、铯原子钟，并不断发射其时间信息
 - 接收站位置已知，接收一颗卫星信号即可同步
 - 接收站位置未知，需同时接收至少4颗卫星的时间信息，采用伪距测量定位方法可计算出时间和位置信息
- 缺点（室内、功耗、安全性、分布式）

4. 时间同步协议

协同同步方法 **Mutual Synchronization**

特点：不依赖同步消息的同步方法. 同步性vs同时性

◆ 基于同步信息交换的同步机制面临着挑战

- 单跳同步协议较为成熟，误差较小（几到几十微秒）
- 多跳同步还依赖网络拓扑结构的建立，多跳同步造成同步误差的累积，带来同步精度问题、可扩展性问题
- 依赖网络传输必然受到传输时间不确定性的影响

◆ 自同步或互同步：

- 自然现象：惠更斯同步钟摆，同步萤火虫，心肌细胞的同步跳动

#奇妙实验：不知不觉就... - @智造圈 的视频 - 视频 - 微博 (weibo.com)

#到康加里国家公园邂逅同步萤火虫-资讯-搜索最新资讯-爱奇艺 (iqiyi.com)

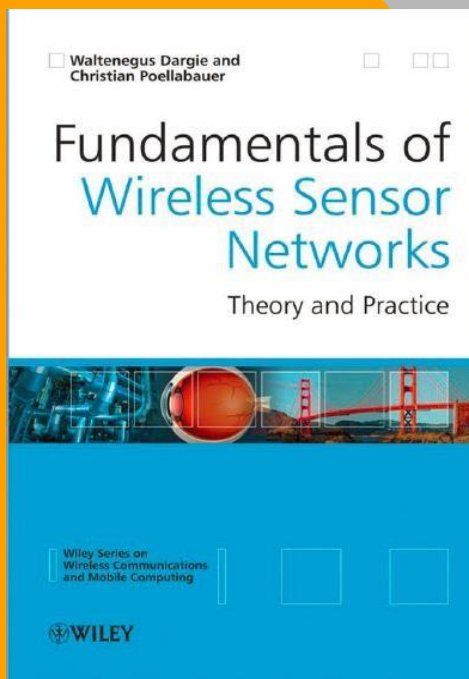
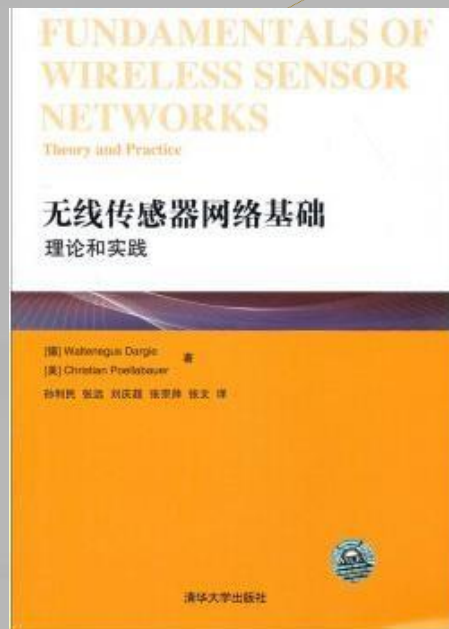
- 基于脉冲耦合振荡器模型的时间同步协议 (**Pulse-Coupled Oscillators**)
 - 要求节点完全相似
 - 直接在物理层用硬件实现
 - 有待进一步研究

3. 时间同步方法概述

常见时间同步算法之间比较:

	TPSN	DMTS	RBS
偏移 / 漂移	偏移	偏移	两者
连续 / 按需	连续	连续	按需
全网 / 子网	全网	全网	子网
广播	是	是	是
单 / 双向通信	双	单	单
复杂性	一般	低	一般
同步精度	较高	低	较高
收敛时间	一般	长	一般
鲁棒性	差	较好	较好

小结



时间同步
技术

1 时间同步的必要性与挑战

2 时钟偏差与同步问题

时钟组成、时间偏移、时间漂移、同步概念

3 时间同步方法概述

同步消息：发送端-接收端，单向/双向消息

接收端-接收端

通信时延组成

4 时间同步协议

- ✓ 传输层（有线网络）同步协议：NTP
- ✓ 发送端-接收端双向消息同步协议：LTS、TPSN
- ✓ 发送端-接收端单向消息同步协议：DMTS、FTSP
- ✓ 接收端-接收端同步协议：RBS
- ✓ 协同互同步方法：基于脉冲耦合振荡器模型的同步协议