

基于重叠网的无线传感器网络时钟同步模型^①

Wireless Sensor Network Synchronization Model Based on Overlay Networks

艾林 李思敏 唐智灵 杨锐雄 (桂林电子科技大学 信息与通信学院 广西 桂林 541004)

摘要: 试图在 WSN 网络通信中,在节点不可靠,消息的延迟或者发送失败的恶劣情况下,解决网络时钟同步问题。文章首先通过一个例子说明了 WSN 之间的数据融合的必要性,由此引出在 WSN 间通信中存在的时钟同步问题。通过简单介绍因特网中两种时钟同步方式,提出了一种新的点对点通信的时钟同步模型。详细讲述了该模型的数学模型和算法,并给出仿真结果。结果显示,萤火虫模型在 WSN 层叠网的同步上有较好的效果。

关键词: 重叠网络 无线传感器网络 时钟同步 数据融合 点对点通信

无线传感器网络(Wireless Sensor Networks, WSN)是由部署在监测区域内大量的低成本微型传感器节点组成,节点间通过无线、多跳通信方式形成一个自组织网络系统。其在医疗、军事、监控、测量等方面都有成功的应用^[1]。然而,随着 WSN 研究和应用的进一步深入发展,各种无线传感器网络平台的数量也在不断增加,但却很少看到一个通用的平台。由于通信平台之间的差异,使得无线传感器网络之间的数据不能方便有效的传递,因此有关无线传感器网络之间的数据融合的研究成为一个新兴的课题。

1 一般的WSN结构和特点

如图 1 所示:传感器节点负责将所监测的数据以广播的形式,通过其他传感器节点传输,经过多跳路由,到达汇聚节点(sink),最后通过卫星或者互联网到达管理节点,终端用户 USER 通过管理节点对传感器网络进行管理,发布监测任务及收集监测数据。通过无线传感器网络可以实现数据采集、数据融合、任务的协同控制等。与目前常见的无线网络(移动通信网、Ad Hoc 网络、无线局域网、蓝牙)相比,WSN 具有:硬件资源有限、电源容量有限、无中心节点、自组织网络、多跳路由、动态拓扑、节点数量多、分布密集等特征。

2 无线传感器网络间时钟同步

2.1 问题假设

假设如图 2 所示情况:一个区域附近有四座小型水库 ABCD,可以用于灌溉,发电和城市供水水,他们各自分别有自己的传感器网络,实时监控各自水库的水位和发电量。4 个水库通过水位的高低以及外界的用水和发电电需求来调节各个水库的用水量,提高能源的综合利用率。这需要我们对各个水库的水位,发电量以及灌溉用水需求,用电需求等数据信息的相互快速的传递,并且对发生的紧急情况,在无人职守的情况下自动作出控制反应,调节各个水库的用水量,达到提高能源利用率的目的。

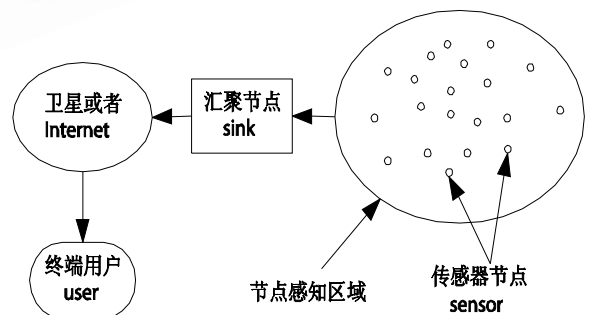


图 1 WSN 的基本结构

① 基金项目:广西科学研究与技术开发计划(桂科基 0731019)

收稿时间:2009-01-04

有幸的是文献[2]借助分布式网络结构的概念，提出使用 P2P 的通信方式，使得两个不同的 WSN 网络所提供的数据能够融合并且在 Internet 上进行传播。其方案是使用 JXTA 协议作为 P2P 平台，实现不同的传感器网络间的数据互联。在 JXTA 协议中，传输数据传输成功的解决了 WSN 网络数据融合的连通性、异构性、互操作性、保密性等问题。

但是，在 WSN 网络间，由于其分布式系统的自组织性，其并没有标准的时间统一系统或共同的时间基准，而信息的传播同样依赖于时钟同步。做为无线传感器网络之间的信息传递，时钟同步算法方面我们除了考虑其分布式网络的传输特性以外，还要考虑其无中心节点、自组织网络、多跳路由、动态拓扑等特点。所以建立统一的时间服务系统或时间服务器对于实时系统尤为重要。

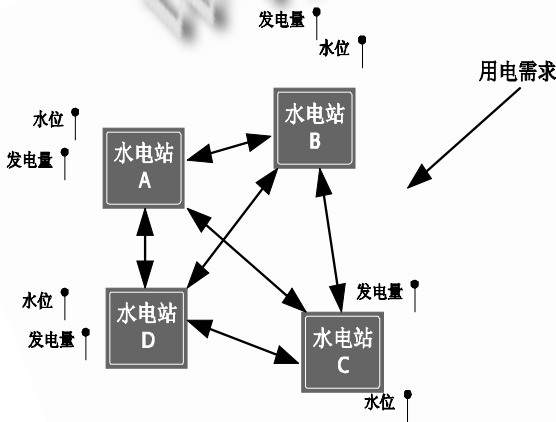


图 2 问题假设

2.2 早期的 P2P 时钟同步协议介绍

早期在 Internet 中，传统的时间同步协议通常都是使用 NTP(Network Time Protocol)协议，它使用 UTC(Universal Time Coordinated)作为时间标准，本身基于 UDP/IP，使用层次式时间分布模型。在局域网，它一般可以提供 0.1ms 的同步准确度；在广域网上同步精度一般为几十毫秒。NTP 系统的网络结构包括主时间服务器、从时间服务器及客户与节点间的传输路径。但这些都无法适应无线传感器网络的同步要求。

随着时钟同步技术的发展，出现了如 STP 协议和

Almost Peer-to-Peer (AP2P)^[3]协议等分布式系统时钟同步协议。STP 协议是在 NTP 协议的基础上出的一种为系统提供精确时间源的协议，能够在无序的网络环境下提供精确的时间服务，消除时间的不一致性对协议测试系统所造成的影响。AP2P 是在 P2P 同步的原理上发展起来的，它并不同于一般的 P2P 同步协议。常见的 P2P 同步协议是依靠读取相邻节点的时间轴来校正时间，而是在所有节点中选出一个节点作为基准时钟，而且不要求网络中所有的节点都承认该节点为基准节点，这样能大大的降低运算的复杂度，完全避免了传统通信方式中的非线性通信开销。

3 萤火虫算法描述

在 2.1 的例子中我们同样也需要各个传感器网络之间进行协调同步，根据无线传感器的网络特点，我们提出了一个基于心跳机制的同步算法：萤火虫算法。其算法模型来自于文献[4]。

3.1 算法原理

假设 sink(i) 为各个传感器网络的中心节点，不同的 sink 有各自不同的周期，我们用 δ_i 在这个模型里表示 sink(i) 的实际周期 ($\Delta_l < \delta_i < \Delta_u$)，其中 Δ_l 和 Δ_u 分别为所有节点周期的最小和最大值，为全局常量。另外，还有一个固有周期 Δ ($\Delta_l < \Delta < \Delta_u$)，每个 Δ 周期节点向外发送一次同步信息 Flash，用于与其他节点进行同步信息交换。模型用频率可以表示为： $\Omega_l = 1/\Delta_u$, $\Omega_u = 1/\Delta_l$, $\Omega = 1/\Delta$ 和 $\omega_i = 1/\delta_i$ 。

这个模型的算法特点在于 processflash() 修改的是 ω 值，如果 flash 的到来延迟了 ($\Phi < 1/2$)，节点就向最低频率 Ω_l 逼近(意味着周期延长)于此同时相位保持不变。这样能够为下一个 flash 的出现提供充足的时间，使得它更有可能在此期间接收到来自同一个节点的下一个 flash。同理，如果 flash 到来得太早 ($\Phi > 1/2$)，那么频率就 Ω_u 向增加。

根据文献[4]，我们将 processflash() 算法的数学表示为：

$$\omega' = \omega + \varepsilon(\Omega - \omega) + g^+(\Phi)(\Omega_l - \omega) + g^-(\Phi)(\Omega_u - \omega) \quad (1)$$

(1) 式中 ω' 代表新的频率， ω 代表当前频率，此时相位 Φ 为即时相位， ε 函数能够控制频率向标准频

率 Ω 靠近。函数 g^+ 和 g^- 的定义如下:

$$g^+(\Phi) = \max\left(\frac{\sin 2\pi\Phi}{2\pi}, 0\right) \quad (2)$$

$$g^-(\Phi) = -\min\left(\frac{\sin 2\pi\Phi}{2\pi}, 0\right) \quad (3)$$

函数 $g^+(\Phi)$ 在 $\Phi < 1/2$ 时有效, 否则为 0, 函数 $g^-(\Phi)$ 在 $\Phi > 1/2$ 时起作用, 否则为 0。通过这个方法(1)式就可以根据接收到的 flash 提前或者延迟来调整自己的频率 ω 。这个模型的优点在于其建立在较少的前提假设情况下(最重要的是没有规定同样的周期长度), 这样比较近我们真实 P2P 网络。

3.2 算法流程

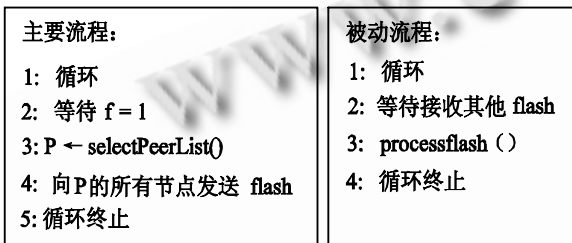


图 3 算法流程

如图 3 所示: 算法流程分为主要和被动循环两段, 其中“主要流程”是随时执行的, “被动流程”只有接收到外部 flash 才执行

算法流程描述: 每个节点 sink 可以看作一个振荡器, 它的特性由各自的相位 ϕ 和周期 Δ 所决定, 其中 ϕ 的值域为 0 到 1, 它的值是关于时间 t 的锯齿波函数 $\phi(t)$, 由此可得: $\partial\phi/\partial t = 1/\Delta$ 。在主动流程中: ϕ 在单位时间 Δ 内由 0 到 1 线性增加, 当 $\phi = 1$ 时, 节点发送一个同步时钟信息 flash, 相当与一个 ping 信息, 之后让 $\phi = 0$ 。所有节点的周期 Δ 初始值可以不同。

在被动流程中: 当一个节点接收到一个 flash 信息的时候, 函数 processflash() 被执行, 这个函数是时钟同步算法的核心, 它负责修正节点内的 ϕ , 同样也改变了 Δ , 这样它能够提前或者延迟相位, 调节周期函数 selectPeerList() 依赖于底层的重叠网, 这个重叠网的协议能够返回一个相邻节点的列表。在仿真中, 我们将假设这个重叠网的每个节点的相邻节点数是动

态的, 是一个较小的随机常数。

4 试验结果

4.1 试验环境

在这个模型中, 每个节点都运行了我们编写的心跳同步协议。我们使用 peersim 仿真软件, 节点协议采用基于 NEWSCAST 节点采样协议。这个协议具有系统消耗低、极强的健壮性和极少的条件假设。协议不仅在正常的运行中提供高质量的节点列表, 而且在巨大扰动甚至灾难环境下(70%以上节点失效), 通过从本地有效 view 中快速的移除失效节点的 view 来提供高质量的节点样本地址。

仿真层, 心跳机制的 flash 消息我们用简单的 UDP ping 代替, 默认的周期 $\Delta = 1s$, 这么设定的原因在于我们想检验在周期和报文长度相当这样恶劣的环境下产生的结果。我们用 f 表示节点的扇出, 它决定了报文在每个周期的发送数量。

除了 Δ 和 f , 协议中还有一个自由变量 ϵ 为短调节点单位, 在试验中我们让 $\epsilon = 0.01$, 已经能够产生很好的结果了。我们所有的实验在传输层中模拟了一些随机延迟 1—200ms, 另外还引入了一些不可预料的延迟。

初始设定: 在刚开始的时候, 我们在整个网络中建立了约 $2^{10} - 2^{14}$ 个节点。这些节点在投三秒中随机发送自己的第一个 flash, 并且在 $0.85s(\Delta_l)$ 到 $1.15s(\Delta_u)$ 之间随机设定各自的时钟周期。这样, 每个节点在开始的时候都是不同步的, 而且自有周期都存在巨大的差异。

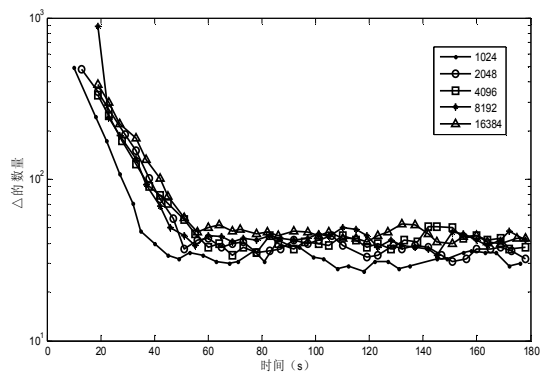


图 4 仿真结果分析

4.2 仿真结果

仿真结果如图 4 所示,该图描述了 5 个独立的仿真结果,每个仿真的节点数量分别从 210 到 214 个不相同。图中纵坐标表示所有节点中输出 flash 的周期的数量,横坐标表示经历的时间。开始的时候节点各自的周期都不尽相同(大约都在 500 个以上),在前 40 秒的数量急剧下降,约经过 60 秒后,其数量基本维持在 30 个左右。

5 结论

本文围绕在重叠网中心跳同步的问题展开论述,通过点对点的网络呈现了一个特殊的环境:节点能通过路由服务直接相互访问。其挑战性在于动态的重叠网络中不可靠的传输信道对同步造成的影响。针对这种情况,我们提出了萤火虫 FLASH 同步算法。本文已经证明,在不同的节点个数和灾难环境下,节点时钟通过同步,经

过约 40s,其周期个数下降到原来的 1%~10%。

参考文献

- 1 Heidemann J, Govindan R. An Overview of Embedded Sensor Networks. Technical Report, USC/Information Sciences Institute, 2004,(15):594.
- 2 Isomura M, Riedel T, Decker C, Beigl M, Horiuchi H. Sharing sensor networks. Proceedings of the 26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'06), 2006,55(26): 452 - 455.
- 3 Sobeih A, Hack M, Liu Z, Zhang L. Almost Peer-to-Peer Clock Synchronization. IEEE, 2007,27(1):42 - 44.
- 4 Ermentrout B. An adaptive model for synchrony in the firefly *pteroptyx malaccae*. Journal of Mathematical