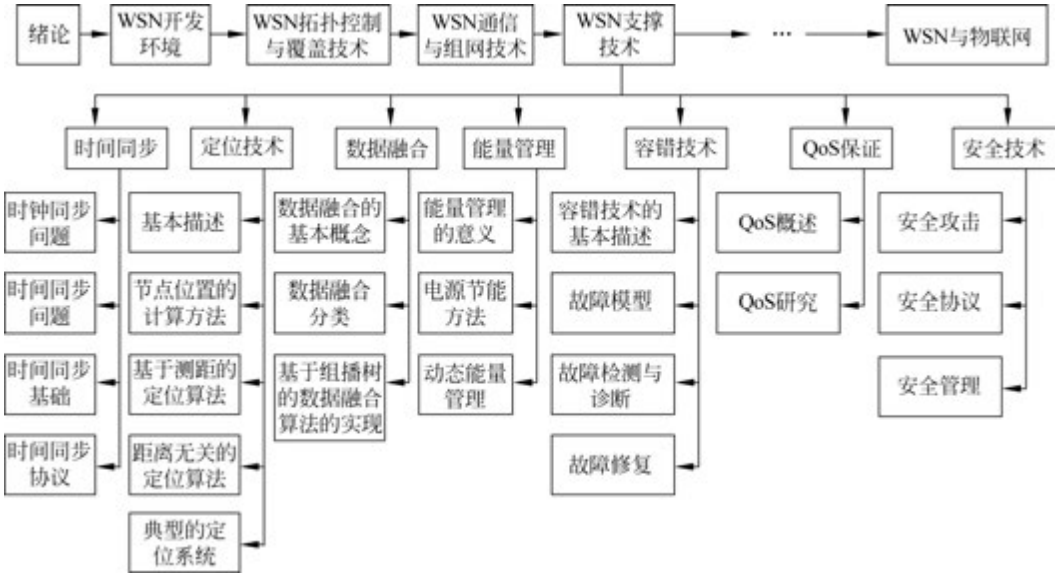
第5章 WSN支撑技术

学习目标

- 掌握时钟同步的问题。
- 掌握时间同步问题。
- 掌握时间同步基础。
- 掌握时间同步协议。
- 了解节点定位的基本概念。
- 了解基于测距的定位算法。
- 了解距离无关的定位算法。
- 了解典型的定位系统。
- 掌握数据融合的基本概念。
- 掌握数据融合的分类。
- 了解常用的数据融合算法。
- 了解 WSN 数据融合算法的实现。
- 了解能量管理的意义。
- 了解传感器网络的电源节能方法。
- 了解动态能量管理。
- 了解故障模型。
- 了解故障检测与诊断。
- 了解故障修复。
- 了解 QoS。
- 了解安全技术。

学习导航



5.1 时间同步

在分布式系统中，每个节点都有自己的时钟和对于时间的定义。然而，为了确定物理世界中事件之间的因果关系，为了消除传感器的冗余数据，为了能在整体上促进传感器网络的工作，传感器节点之间需要遵循一个共同的时标。传感器网络中的每个节点都独立运作，并且依赖于其自身的时钟，所以不同的传感器节点的时钟读数也不同。除了这些随机差异（相位偏移），不同传感器时钟之间的间隙也会由于振荡器漂移率的变化而进一步增大。为了确保感测到的时间可以以有意义的方式进行比较，时间（或时钟）必须同步。有线网络的时间同步技术已经得到广泛的关注，但这些技术并不适用于无线传感器，原因是无线感知环境会带来一些特殊的问题。这些挑战包括 WSN 可能的大规模性、自主配置需求以及健壮性，潜在的传感器的移动性以及对节能的需求。

5.1.1 时钟同步问题

基于硬件振荡器的计算机时钟是所有计算设备的重要组成部分。典型的时钟由一个稳定的石英振荡器和一个计数器组成，这个计数器随着每次石英晶体的振荡递减。当计数器的值为 0 时，它将复位为初始值，并产生一个中断。而每个中断（或者时钟周期）都将触发一个软件时钟（另一个计数器）。应用程序可以通过一个适当的 API 读取并使用软件时钟。因此，软件时钟为每个传感器节点提供了一个本地时间，其中 $C(t)$ 表示在某一个实时时间 t 时的时钟读数。时间分辨率是软件时钟的两个增量（计数）之间的间距。

对于两个节点的本地时间，时钟偏移量表示时钟之间的时间差。同步是指调整一个或两个时钟，从而使它们的读数匹配。时钟率则表示一个时钟推移的频率，而时钟偏差则表示两个时钟频率之间的差别。理想时钟的时钟率的值恒为 $dC/dt=1$，但实际上很多参数影响

了实际的时钟率,如环境的温度和湿度、电源电压以及石英的年龄。偏移率的偏差结果表明两个时钟的相对漂移速率,即 $dC/dt-1$。一个时钟的最大漂移率用 ρ 表示,石英钟的典型值为(1～100)ppm(1ppm=10^{-6})。这个数值由振荡器的制造厂商给出,且满足

$$1-\rho \leqslant \frac{dC}{dt} \leqslant 1+\rho \tag{5-1}$$

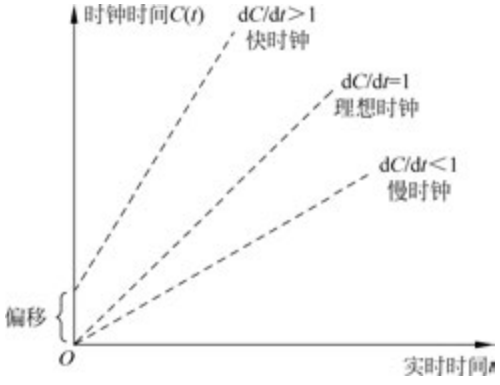


图 5-1 本地时间 $C(t)$与实时时间 t 的关系

图 5-1 显示了漂移率(Drift Rate)如何影响时钟的读数,它使得时钟要么准确无误,要么变快或变慢。漂移率导致传感器时钟读数即使在同步以后也不一致,因此有必要定期执行同步过程。假设时钟完全相同,那么任意两个被同步以后的时钟之间最大的漂移为 $2\rho_{max}$。为了把相对偏移限制到 δs,同步操作之间的间隔 τ_{sync} 必须满足

$$\tau_{sync} \leqslant \frac{\delta}{2\rho_{max}} \tag{5-2}$$

$C(t)$必须是分段连续的,即它必须是一个时间的严格单调函数。因此,时钟的调整是一个渐进过程,如可以使用线性补偿函数改变本地时间斜率。单纯地让时钟向前或向后跳转可能会带来很严重的后果。例如,设定一个计时器在某个特定时间产生一个中断,然而执行同步会漏掉一些时间,可能使得这个特定的时间永远不会到来。

同步有两种,一种是外部的,另一种是内部的。外部同步是指所有节点的时钟都与一个外部时间源(或者参考时钟)同步。外部参考时钟是一个类似于世界协调时(Universal Time Coordinated,UTC)的精确的实时标准。内部同步是指在没有外部参考时钟支持的情况下,所有节点的时钟之间相互同步。内部同步的目的是尽管时间可能与外部参考时间不同,但是网络中所有节点的时间都一样。外部时间同步既保证了网络中的所有时钟一致,又保证了与外部时间源一致。当节点与外部参考时钟同步时,时钟精度表示时钟相对于参考时钟的最大偏移。当网络中的节点内同步时,精度表示网络中任意两时钟之间的最大偏移。需要注意的是,如果两个节点外部时钟同步的精度为 Δ,则它们内部时钟同步的精度为 2Δ。

5.1.2 时间同步问题

1. 时间同步的必要性

WSN 中的传感器检测物理世界中的对象,并将活动和事件报告给感兴趣的观察者。例如,一些近身检测传感器,当有活动物体(如车辆)经过时,会触发一个事件,磁、电容以及声学传感器都属于这一类。在传感器密集分布的网络中,多个传感器将进行同样的活动,并发生同样的事件。这些事件之间精确的时间相关性对于解决下面的问题至关重要:检测到了多少移动的物体?物体向哪个方向移动?物体以怎样的速度移动?因此,观察者能否为事件建立正确的逻辑顺序非常重要。如图 5-2 所示,实时时间的顺序是 $t_1<t_2<t_3$,那么传感器标注的时间顺序必须是 $C_1(t_1)<C_2(t_2)<C_3(t_3)$。为了精确地确定物体移动的速度,传感器时间标注的时间差必须与实时时间的时间差对应,即 $\Delta=C_2(t_2)-C_1(t_1)=t_2-t_1$。

对 WSN 的数据融合而言，这是非常重要的，因为数据融合所关注的是观测相同或相关事件的多传感器的数据集合。而数据融合更深一层的目标是消除冗余的传感器信息，缩短重要事件的响应时间以及降低对资源的需求（如能源消耗）。

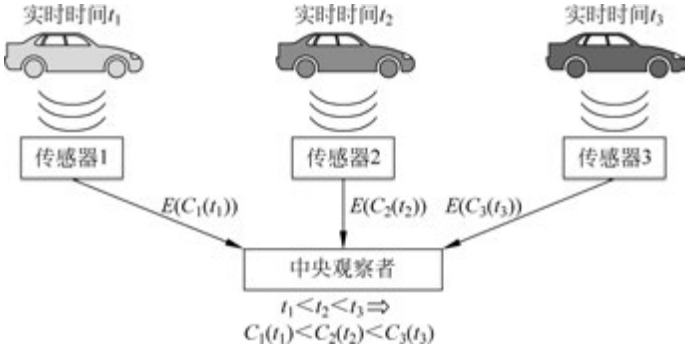


图 5-2 使用多传感器对移动物体的速度和方向的检测

对于分布式系统中各种应用程序以及算法而言，时间同步也是必需的。这些程序和算法包括通信协议（如最多一次消息传递）、安全（如在基于 Kerberos 的身份验证系统中限制使用一些特别的关键字，并协助检测重放消息）、数据一致性（缓存一致性和数据复制一致性）以及并发控制（原子性和相互排斥）等。

MAC 协议（如时分多址）允许多个设备共享一个通信访问介质。将时间分为时隙，再把时隙分配给无线设备，并且每个时隙仅属于一个无线装置。基于 TDMA 方法的优点是介质接入可以预测（只允许每个节点在一个或多个反复出现的时隙发送数据），而且该算法是能量高效的（当节点在一个时隙中既不是发送方也不是接收方时，就进入省电休眠模式）。然而，为了实施 TDMA，各节点必须拥有统一的时间视角，也就是说，它们需要知道每个时隙确切的开始和结束时刻。

在能量使用方面，许多 WSN 都依赖于休眠/唤醒机制，这个机制允许一个网络选择性地关闭一些传感器节点或让一些节点进入低功耗休眠状态。这个机制中，传感器之间的时间协调是非常重要的，因为节点需要知道它们应该何时进入休眠状态，何时被唤醒，从而确保相邻节点之间的唤醒状态相同，保证节点间能够通信。

最后，在 WSN 中，需要准确地定位传感器节点或所监测的对象。许多定位技术都依赖测量技术估计节点间的距离。而要检测无线电或声音信号的传播时间，同步技术是必不可少的。

2. 时间同步面临的挑战

传统的时间同步协议都是为有线网络设计的，它们并没有把低成本低功耗的传感器节点和无线介质等因素考虑在内。与有线环境相似，WSN 也面临着时钟噪声攻击（Clock Glitch），以及由于温度和湿度变化所导致的时钟漂移等问题。尽管如此，传感器网络的时间同步协议必须将一系列额外的问题和限制条件考虑在内。

1）环境影响

时钟的漂移率随环境的温度、压力和湿度的波动而有所不同。有线计算机一般工作在相当稳定的环境（如 A/C 控制的群集室或办公室）中。与之不同，无线传感器常常部署在室外以及一些恶劣的环境中，在这样的条件下，这些环境属性很容易波动。在受控制的环境下，振荡器的频率变化最多为 3ppm，而 1ppm 的误差相当于每 12 天有 1s 的错误，该误差由

室内温度的变化导致。然而,对于工作在户外的低成本传感器节点而言,变化幅度可能更大。

2) 能量限制

无线传感器通常用能量有限的电源来驱动,也就是说,其电源是一次性电池或充电电池(如通过太阳能电池板充电)。更换电池会极大地增加成本,特别是当网络规模较大和部署节点比较困难时。因此,为了保证电池的寿命,时间同步协议不能消耗过多的能量。由于节点间的通信是时间同步的基础,一个高能效的同步协议就应该使节点间达到同步的通信量最小。

3) 无线介质和移动性

众所周知,无线通信介质是不可预知的。雨水、雾、风和温度的改变都会带来环境特性的变化,从而导致通信介质性能的波动。这些波动限制了网络的吞吐量,提高了误码率,并产生无线电干扰等问题。无线连接之间的非对称性使节点之间的信息交换产生更多的问题,也就是说,节点 A 可以接收节点 B 的消息,但是节点 A 的消息过弱而使得节点 B 无法准确解析。通常,A 到 B 路径的特征(如延时)可能与 B 到 A 路径的特征有显著的不同,从而产生非对称通信延迟。此外,无线网络中的通信干扰受网络的密度、无线设备的通信和干扰范围以及这些设备的活动水平等因素影响。许多无线传感器是移动的(如安装在车辆上或由人携带),这会导致拓扑结构和链路质量变化显著且快速。最后,传感器节点可能不再工作或能量耗尽,因此即使网络的拓扑结构或密度发生变化时,时间同步也要能继续工作。由于面临这些挑战,时间同步协议必须具有鲁棒性和可重构性。

4) 其他约束

除了电源能量的限制,低功耗、低成本的传感器节点在处理速度和存储空间上也受到限制,这更要求时间同步协议必须低消耗、轻量化。小尺寸、低成本的传感器设备不允许采用大尺寸、昂贵的硬件实现同步(如 GPS)。因此,设计时间同步协议时,应该基于资源有限的环境,并尽可能少增加或不增加总体开支。WSN 通常是大规模部署的,同步协议应该适应节点数目或密度的增长。最后,不同的传感器应用对时钟的精度和准度有不同的要求。例如,在目标追踪的应用中,时间同步能保证事件和消息的正确排序(在没有外部参考时钟的条件下)就足够了,但是对精度的要求是微秒级别。另外,传感器网络检测公共场所在一天的某段时间内的人流量时,需要外同步,其时间精度达到秒级就可以了。

5.1.3 时间同步基础

时间同步通常基于传感器节点之间某种形式的信息交换。如果介质像在无线系统中那样支持广播模式,那么收发较少的消息就可以使多个设备同时完成时间同步。

1. 同步消息

大多数现有的时间同步协议基于两两同步的模式。在这种模式中,两个节点之间进行同步至少需要一个同步消息(Synchronization Messages)。要在整个网络范围内实现时间同步,可以在多个节点对之间不断重复该过程,直到每个节点都根据参考时钟将自己的时钟调整好。

1) 单向消息交换

最简单的两两时间同步是在两个节点之间同步时只用一个消息,也就是一个节点发送一个时间戳给另一个节点。如图 5-3(a)所示,t_1 时刻,节点 i 向节点 j 发送一个时间同步消

息，将时间 t_1 作为时间戳嵌入其中。收到消息时，从本地时钟中取得一个本地时间戳 t_2。以 t_1、t_2 两个时间戳的差作为节点 i 和 j 之间的时钟偏移的一个量度。准确地说，这两个时间的差值可以表示为

$$t_2 - t_1 = D + \delta \tag{5-3}$$

其中，D 表示未知的传播延时。在无线介质中，传播延时是非常小的(几微秒)，通常可以忽略或默认为某个特定值。节点 j 可以用这种方法计算出频偏，从而调整自己的时钟，实现与节点 i 的同步。

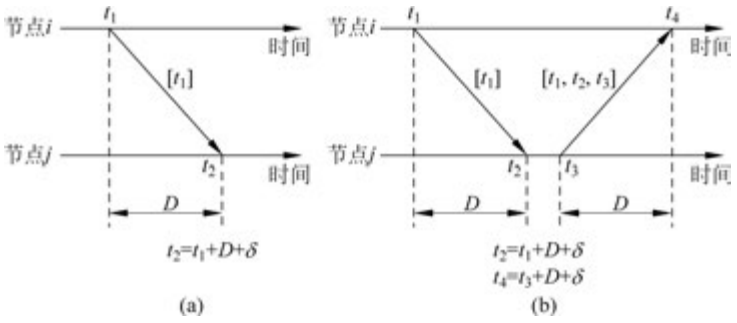


图 5-3 两两时间同步示例

2) 双向消息交换

另一种更加准确的方式是采用两个同步消息，如图 5-3(b)所示。在 t_3 时刻，节点 j 给节点 i 一个包含时间戳 t_1、t_2、t_3 的回复消息。在 t_4 时刻，节点 i 接收到第二个消息时，在假定传播延时为固定值时，两个节点都能确定频偏。然而，节点 i 能更准确地确定传播延时 D 和频偏 offset。

$$D = \frac{(t_2 - t_1) + (t_4 - t_3)}{2} \tag{5-4}$$

$$\text{offset} = \frac{(t_2 - t_1) + (t_4 - t_3)}{2} \tag{5-5}$$

需要注意的是，这里假设传播延时在两个方向上都是相同的，并且在时间的量度尺度上，时钟漂移并不改变(因为时间单位的跨度很短，所以认为频偏是一致的)。尽管只有节点 i 有足够的信息确定频偏，但是它可以在第三个消息中与节点 j 共享。

3) 接收端-接收端同步

另一个实现方法是采用接收端-接收端同步准则的协议，这种模式是根据同一消息到达不同节点的时差实现同步的。与大多数传统的接收端-发送端同步模式不同，在广播环境中，这些节点几乎在相同的时间接收到消息，接收节点可以通过交换各自的接收时间计算彼此的频偏(即接收时间的差异可以反映它们的频偏)。这种协议的一个例子如图 5-4 所示，如果有两个接收器，只要 3 个消息即可使两者同步。注意，广播消息不包含时间戳，而是利用广播消息到达接收节点的时间不同使节点相互同步。

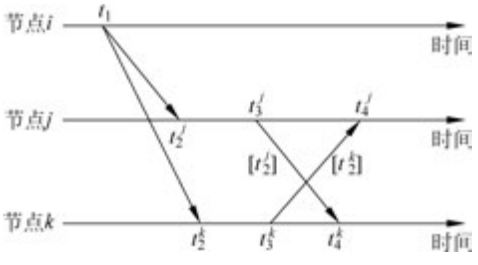


图 5-4 接收端-接收端同步策略

2. 通信延时的不确定性

通信延时的不确定性对于时间同步所能达到的精度有很大的影响。如图 5-5 所示,通常同步消息的延时包含以下 4 部分。

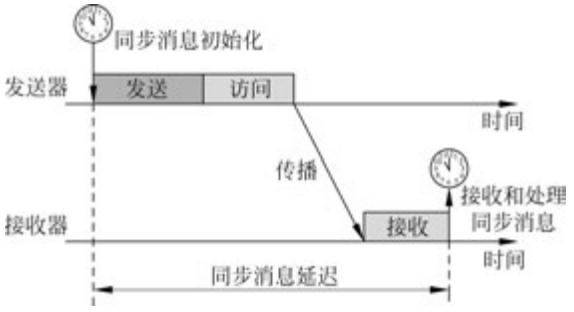


图 5-5 同步信息的端到端延时

(1) 发送延时: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(系统调用接口、内容切换)、网络协议栈以及网络设备驱动器等引起的延时。

(2) 访问延时:发送节点访问物理信道的延时,主要取决于 MAC 协议。基于竞争的协议,如 IEEE 802.11 的 CSMA/CA,必须等待信道空闲才能进行访问。当同时有多个设备访问信道时,冲突会引起更长的延时(如 MAC 协议的指数补偿机制)。更容易预测到的延时是,在基于 TDMA 的协议中,设备在发送消息前,必须在一个周期内等待属于它的那个时隙到来。

(3) 传播延时:消息从发送端到接收端真正的延时。当节点共享物理信道时,传播延时是非常小的,在分析关键路径时通常可以忽略。

(4) 接收延时:接收设备从介质层中接收消息、处理消息以及将到达消息告知主机所需的时间。一般通过中断方式告知主机,用这种方式可以读取中断发生的本地时间(即消息到达时间)。因此,接收时间一般比发送时间短一些。

为了减小其中一些组件的数量和种类,许多 WSN 的同步方案采用了底层的技术。例如,MAC 层时间戳可以分别缩短接收和发送的延时。

5.1.4 时间同步协议

目前,已经开发了许多 WSN 的时间同步协议,这些协议多数是在前面介绍的消息交换思想的基础上加以改进得到的。

1. 基于全球时间源的参考广播

全球定位系统(Global Positioning System,GPS)连续广播从 1980 年 1 月 6 日 0 时起开始测量的 UTC(世界标准时间)。然而,与 UTC 不同的是,GPS 不受到闰秒的影响,因此比 UTC 时间快若干整数秒(2009 年是 15s)。甚至廉价的 GPS 接收器都可以接收到精度为 200ns 的 GPS 时间。时间信息也可以通过路基的无线电基站来传播。例如,美国国家标准及技术研究所用无线电基站 WWV、WWVH 和 WWVB 持续广播基于原子时钟的时间。然而,这些方案有许多限制,从而影响了在 WSN 中的应用。例如,GPS 信号不是任何地方都可以接收到的(如水下、室内、茂密的森林中),并且对电源的要求也相对较高,这对低成本的传感器节点来说是不可行的,并且添加 GPS 对小小的节点来说太大也太贵了。可是,许多

传感器网络是既包含能量有限的传感器设备，也包含功率较大的设备的层次化系统，功率较大的设备通常作为网关或簇头，这些大功率设备可以支持 GPS 或无线接收器，可以作为主时钟源。网络内其他所有节点可以利用它，使用本节介绍的发射端-接收端模式进行时间同步。

2. 基于树的轻量级同步

基于树的轻量级同步(Lightweight Time Synchronization，LTS)协议的主要目的是用尽可能小的开销提供特定的精度(而不是最大精度)。LTS 协议能够用于多种集中式或分布式的多跳同步算法中。为了理解这种方案，将先讨论对于一对节点的同步信息交换。图 5-6 用图形化的方式描述了这种方案。

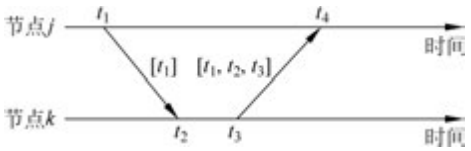


图 5-6 LTS 的两两同步

首先，节点 j 发送一个同步消息给节点 k，同步消息的时间戳包含了传输时间 t_1。节点 k 在时刻 t_2 收到消息，回复一个包含了时间戳 t_3 和之前记录的时间 t_1 和 t_2 的消息。这个消息在 t_4 时刻被节点 j 收到。注意，t_1 和 t_4 是基于节点 j 的时钟，而时间 t_2 和 t_3 是基于节点 k 的时钟。假设传输延时为 D(更进一步，认为 D 在两个方向是一样的)，两个时钟之间未知的时钟偏移为 offset，节点 k 的时间 $t_2 = t_1 + D + \text{offset}$。同样，$t_4 = t_3 + D - \text{offset}$。所以，offset 可以计算为

$$\text{offset} = \frac{t_2 - t_4 - t_1 + t_3}{2} \tag{5-6}$$

集中式的多跳 LTS 算法基于单一的参考节点，这个节点是网络内所有节点最大生成树的根节点。为了使同步准确性最高，树的深度必须最小。鉴于节点对之间两两同步会使产生的错误不断累加，因此误差会随着跳数增加在树节点增加。在 LTS 中，同步算法每执行一次，树的生成算法(如广度优先算法)就执行一次，树一旦生成，参考节点就与它的子节点进行两两同步完成同步过程。一旦完成，每个子节点就重复这个过程直到所有节点都完成同步。两两同步有 3 个消息的固定开销，因此，如果一棵树拥有 n 条边，那么开销为 $3n-3$。

分布式的多跳 LTS 算法不需要建立最大生成树，同步的责任从参考节点转移到传感器节点自己。这种模式假定无论何时某个节点需要同步，总存在一个或几个节点能与它通信。这种分布式的方案允许节点自己确定再同步周期。也就是说，节点根据合适的准确度、到最近参考节点的距离、它们自己的时钟漂移 ρ 以及它们上次同步的时间决定它们的再同步周期。最后，为了消除潜在的低效性，分布式 LTS 尽量满足邻节点的要求。为此，在挂起一个同步要求时，一个节点可以询问它的邻居，如果邻居有同步要求，这个节点与自己的一跳邻节点同步，而不是与参考节点同步。

3. 传感器网络的时间同步协议

传感器网络的时间同步协议(Time-synchronization Protocol for Sensor Networks，TPSN)是另一种传统的使用树结构组织网络的发射端-接收端同步方式。TPSN 同步有两个阶段：级别探测阶段(在网络部署时执行)和同步阶段。

1) 级别探测阶段

级别探测阶段的目标是创建网络的分层拓扑结构，每个节点被分配了一个级别，根节点(如一个配备了 GPS，可以通向外部世界的网关)驻留在级别 0。根节点通过发出一个 level_

discovery 消息开始这个过程,这个消息包含了级别信息和发射者独有的身份信息。

与根节点相邻的每个节点利用这个消息确定自己的级别(即级别 1),同时再次广播包含自己的级别和身份信息的 level_discovery 消息。重复这一进程,直到网络中每个节点都确定了自己的级别。若一个节点已经确立了自己在层次结构中的级别,当再次收到级别发现消息时,就将其直接丢弃。当然,也会发生某些节点没有分配到一个级别的情况。例如,当 MAC 层发生冲突时,节点就无法收到 level_discovery 消息,或者某个节点加入网络时级别探测过程已经结束。在这种情况下,节点可以向邻节点发送一个 level_request 信号,这些节点会回复它们所分配的级别。然后,这个节点将自己的级别设置为一个比收到的最小级别值大一级别的值。节点故障可以用相同的方法解决。当一个 i 级节点发现没有 $i-1$ 级邻节点(在接下来描述的同步阶段的通信步骤中)时,它也会发出一个 level_request 信号重新插入层次结构中。如果根节点失效,1 级节点不会发出 level_request 信号,而是会执行领导选举算法,然后开始新的级别探测,重新开始 TPSN 过程。

2) 同步阶段

在同步阶段,TPSN 沿着在前一阶段建立起的分层结构的边缘使用双向同步机制,也就是每个 i 级节点会与处于 $i-1$ 级的节点进行时钟同步。TPSN 的双向同步机制与 LTS 采取的方式相似。节点 j 在 t_1 时刻发出一个同步脉冲信号,这个脉冲信号包含节点级别和时间戳。节点 k 在 t_2 时刻收到这个信号,然后在 t_3 时刻发出确认响应(包含时间戳 t_1、t_2、t_3 以及节点 k 的级别),最终,节点 j 在 t_4 时刻收到这个数据包。与 LTS 一样,TPSN 假设传播延时 D 和时钟偏移在短时间内不会发生改变。t_1 与 t_4 通过节点 j 的时钟计量,t_2 和 t_3 通过计算后的时钟计量,这几个时间点有以下关系:$t_2 = t_1 + D + \text{offset}$;$t_4 = t_3 + D - \text{offset}$。基于这些参数,节点 j 可以计算延时 D 和偏移量。

$$D = \frac{(t_2 - t_1) + (t_4 - t_3)}{2} \tag{5-7}$$

$$\text{offset} = \frac{(t_2 - t_1) - (t_4 - t_3)}{2} \tag{5-8}$$

同步阶段是从根节点发出一个时间同步数据包(time_sync)开始的。等待随机时间之后(为了减少在介质存取时的冲突),1 级节点开始与根节点进行双向信息交换。当一个 1 级节点收到了根节点的确认信息时,会计算自身的时钟偏移量从而调整自己的时钟。2 级节点会监听与它相邻的 1 级节点发出的同步脉冲信号,在经过一定的补偿时间之后,与 1 级节点开始双向同步。为了给 1 级节点足够的时间接收和确认自己的同步脉冲,补偿时间是必要的。不断在所有层次中执行该过程,直到所有节点都与根节点进行了同步。

与 LTS 相似,TPSN 的同步误差取决于分层结构的层次深度和双向同步时端到端的信息传输延时。TPSN 依靠在 MAC 层的数据包的时间戳使延时最小化,减小误差。

4. 洪泛时间同步协议

洪泛时间同步协议(Flooding Time Synchronization Protocol,FTSP)的目的是将整个网络的同步误差控制在微秒级;可伸缩性达到数百个节点;在网络拓扑结构变化时保持健壮性,包括连接故障和节点故障引起的网络变化。与其他方案不同的是,FPSP 在消除了大部分同步误差来源的同时,使用一个单独的信号建立起发射节点与接收节点之间的同步。为此,FTSP 扩展了前面所描述的延时分析,并将端到端延时分解成如图 5-7 所示的几部分。

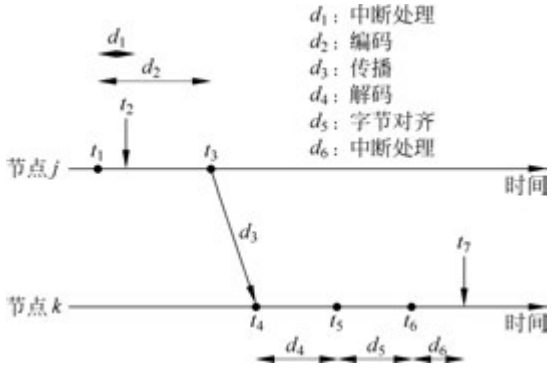


图 5-7 同步信息中的端到端延时

在延时分析中，传感器节点的无线通信模块会在 t_1 时刻通过中断告知 CPU，自己已经准备好接收将要被发出的消息的下一部分。经过中断处理时间 d_1 之后，CPU 在 t_2 时刻生成一个时间戳。无线通信模块用来编码和将信息转变成电磁波所需要的时间称为编码时间 d_2（在 t_1 和 t_3 之间）。传播延时（在节点 j 时钟上的时间 t_3 和节点 k 时钟上的时间 t_4 之间）之后会有解码时间 d_4（在 t_4 和 t_5 之间）。解码时间是无线通信模块用来将电磁波形式的信息重新解码成二进制码的时间。字节对齐时间 d_5 是由节点 j 和 k 之间不同的字节对齐（位偏移）引起的延时，也就是说，接收模块必须确定已知同步字节的偏移量，相应地调整接收的信息。最后，节点 k 上的无线通信模块在 t_6 时刻发出一个中断信号，允许 CPU 在 t_7 时刻获得最终的时间戳。

这些延时对整个端到端延时变化的影响是很大的。例如，传播延时 d_3 通常非常小（小于 1μs），而且是可以确定的。同样，编码时间 d_2 和解码时间 d_4 也是可以确定的，一般都在比较低的百微秒级别。字节对齐延时 d_5 取决于位偏移，一般长达到几百微秒。中断处理时间 d_1 和 d_6 是不确定的，通常会占用几微秒。

1）FTSP 的时间戳

在 FTSP 中，发送器通过一个无线信号与一个或几个接收器进行同步，广播消息中包含发送器的时间戳（估计传输给定字节数的消息所需要的全部时间）。当消息到达后，接收器提取其中的时间戳，然后使用自己的本地时钟对到达信息标记时间。全局-本地时间对提供了一个同步点。因为发送器的时间戳必须嵌入当前要传输的消息中，因此必须在包含时间戳的字节发送之前进行时间标记操作。在 FTSP 中，同步信息从一段前导码开始，这些字节后面是一些 SYNC 字节、数据字段和一个用于错误检测的循环冗余校验码（Cyclic Redundancy Check，CRC），如图 5-8 所示。这段前导码用于将接收器的无线通信模块和载波频率同步；SYNC 字节用来计算位偏移，这些字节是正确重组信息的必要部分。FTSP 在发送器和接收器上使用多重时间戳减少由中断处理、编码/解码时产生抖动的时间。在发送或接收时，这些时间戳被记录在每个 SYNC 字节之后的字节中。时间戳通过减去正常字节传输时间的整数倍进行归一化（如在 Mica2 平台上大约为 417μs）。由中断处理时间引起的抖动可以使用归一化的时间戳中最小的那个来消除。此外，对正确的归一化时间戳求平均，可以减少编码和解码时产生的抖动。只有最终的（误差校正）时间戳被添加到了信息数据之中。在接收方，时间戳必须通过字节对齐时间（可以通过传输速度和位偏移量判定）进

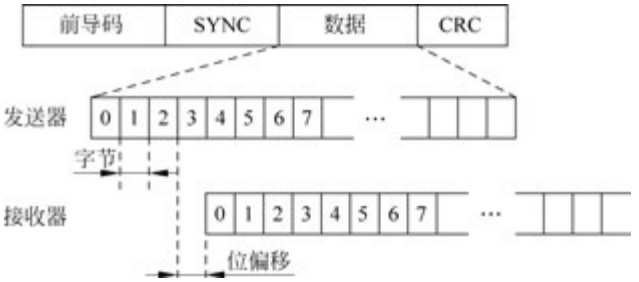


图 5-8 同步信息的格式以及发送器和接收器之间的位偏移

行进一步校正。

2) 多跳同步

与 TPSN 相似,FTSP 通过选举产生的同步根节点同步网络,在这个网络中,根节点的选举基于每个节点独有的 ID(如将级别最低的节点选作根节点)。根节点保留全局时间,网络中的其他所有节点都会将自己的时钟与根节点的全局时间同步。同步过程由根节点发出的包含时间戳的广播信息触发。所有在根节点的通信范围之内的节点可以直接从广播信息中建立同步点。其他节点从靠近根节点且已经同步过的节点发出的信号中收集同步点。

与 TPSN 相似,FTSP 依赖于根节点选举算法保证网络中只有一个同步根节点。每个广播信息包含根节点特有的 ID(根 ID)和一组序列号(除了已经讨论过的时间戳以外)。无论什么时候,若一个节点在一定的时间内没有接收到同步信息,它将声明自己成为一个新的根节点。当节点收到了一个来自 ID 级别比自己低的根节点发出的同步信息时,它会放弃自己的根状态。新的节点加入网络时,若它的 ID 级别比根节点 ID 级别还低,它不会立刻声明自己成为根节点,而是等待一段时间收集同步信息,并根据当前的全局时间调整自己的时钟。这些技术确保可以处理网络拓扑结构改变的问题,包括节点移动而导致网络拓扑结构发生变化的情况。

5. 参考广播同步

参考广播同步(Reference Broadcast Synchronization,RBS)协议依靠在一系列接收器之间广播消息实现同步。在无线传播介质中,广播消息几乎同时到达多个接收器。消息延迟的不确定性主要由传播延时和接收器接收以及处理广播消息时所需的时间决定。RBS 的强大之处在于它可以消除由发送器带来的不确定性同步误差。所有同步方法都是以某种形式的消息交换进行的,因此这些消息中的不确定性延迟都会限制可以获得的时间同步方式的粒度(Granularity)。图 5-9 对传统同步协议和 RBS 协议的关键路径进行了比较。利用无线传播介质的性质,对于两个接收器,广播信号的发送延时和访问延时是相同的,也就是说,两个接收器实际的信息到达时间只会因为传播路径的变化和接收延时的不同而不同。因此,RBS 协议的关键路径比传统同步协议的关键路径要短很多。

例如,当有两个接收器时,接收到信标时每个接收器都会记录一次(使用它们的本地时钟)。接下来,两个接收器会交换它们记录的信息,从而可以计算偏移量(如本地信息到达时间的差异)。当有两个以上的接收器时,所有接收器对之间的最大相位误差称为群差值(Group Dispersion)。当接收器数目增加时,很可能至少存在一个接收器不能很好地同步,这会导致更大的群差值。另外,增大参考广播消息的数量会减小群差值。因为接收节点在

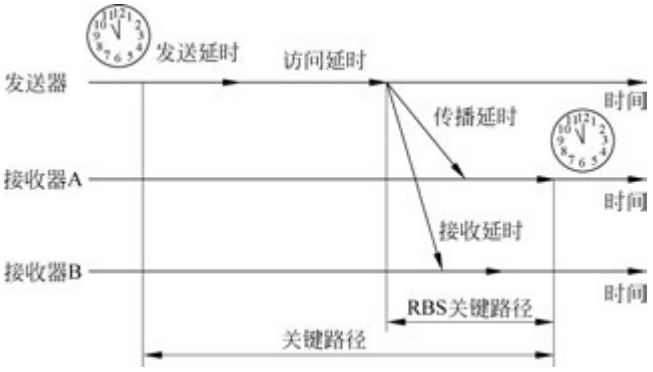


图 5-9 同步信息交换过程中的关键路径分析

接收时间上会有很多变化，使用多个参考广播消息可以提高同步的精确性。换言之，接收器 j 可以计算自己相对于另一个接收器 i 的偏移量，因为接收器 i 和 j 接收 m 个数据包的平均相位偏移为

$$\text{offset}[i,j]=\frac{1}{m}\sum_{k=1}^{m}(T_{j,k}-T_{i,k}) \tag{5-9}$$

通过建立多个包含各自广播域的参考信标，RBS 可以拓展到多跳情况。这些域可以重叠，重叠域中的节点可以作为桥节点，从而允许多个域之间的同步。例如，节点 A 和节点 B 在参考节点 C 的范围内，节点 C 和节点 D 在参考节点 E 的范围内，那么节点 C 就是两个广播域之间的桥节点。

传感器节点之间同步需要大量的消息交换，这使得 RBS 成为一个成本很高的同步技术。因此，在 RBS 的基础上衍生出了后同步计划。在后同步中，除非发生兴趣事件，否则节点间不进行同步。如果同步过程在这个兴趣事件发生后很快开始，那么传感器节点可以仅在感兴趣时才调整它们的时钟，不必因收发不必要的同步消息而浪费能源。

6. 时间扩散同步协议

时间扩散同步协议(Time Diffusion Synchronization Protocol，TDP)允许 WSN 达到一个平衡时间，也就是各节点都协商出一个全网的统一时间，并且保持它们的时钟相对该平衡时间有一点小的偏离。网络中的各节点从两种角色中选择一个，然后动态地加入树状结构。两种角色是指主节点和分散领导节点。TDP 的时间扩散程序负责将时间消息从主节点扩散到其邻节点，其中一些邻节点会成为分散领导节点，并负责将主节点的消息传播至更远的节点。TDP 在两个阶段的操作中是有区别的。在激活阶段，每 τs 选举一次主节点(基于选举/再选举程序)，这样可以平衡网络的工作量并使网络能协商出平衡时间。紧跟每个激活状态后的是非激活状态，非激活状态没有时间同步。每个 τs 的间隔被进一步分为 δs 的间隔，每个间隔从选举分散领导节点开始。选举程序会除掉叶节点和那些时钟偏离邻节点超过某一临界值的节点。该操作通过邻节点间交换信息，比较它们的读数来实现。而且，选举程序会考虑传感器节点的能量状态以保证主节点和分散的领导节点的正常工作。

图 5-10 描述了时间扩散同步的概念。一开始，被选举出的主节点向其邻节点发送时间消息。所有分散的领导节点都可以接收到这个消息(在图中，节点 C 和节点 D 是节点 A 的领导节点)，并回复一个 ACK 确认消息。主节点据此确定到每个邻节点 j 的双向传输延时

Δ_j、双向延时的标准差以及到所有邻节点的单向延时的估计值(如果用 Δ 表示所有双向传输的延时,那么单向延时就是 $\Delta/2$)。主节点用另一个含有时间戳的消息将标准差发送给每个相邻的分散领导节点。分散领导节点通过这个时间戳、单向估计延时以及标准差调整自己的时钟,然后再与邻节点重复整个扩散过程。重复该过程 n 次,n 为从主节点到最后一跳的距离(图 5-10,$n=2$)。注意,节点从多个主节点接收到时间消息时,就使用它们的标准差作为加权系数确定它们对需调整的时钟的贡献大小。

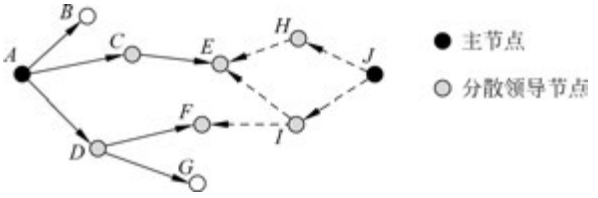


图 5-10 TDP 同步示例(两个主节点都是 $n=2$)

7. Mini-Sync 和 Tiny-Sync 同步

Mini-Sync 和 Tiny-Sync 是两个紧密相关的协议,它提供了低带宽、低存储空间、低处理需求的成对同步协议,可以用来作为整个传感器网络同步的基础模块。同一个传感器网络中两个节点的时钟关系可以表示为

$$C_1(t)=a_{12}C_2(t)+b_{12} \tag{5-10}$$

其中,a_{12} 和 b_{12} 分别表示节点 1 和节点 2 的相对频偏和相对时钟差。为了确定它们的关系,节点可以使用双向通信方案。例如,节点 1 在 t_0 时刻向节点 2 发送一个时间戳探测消息,节点 2 在 t_1 时刻立即回复一个时间戳消息。节点 1 记录第二条消息的到达时间 t_2,便可得到一个三元组的时间戳(t_0,t_1,t_2),这个标签也叫作数据点。因为 t_0 在 t_1 之前,t_1 在 t_2 之前,以下不等式应当成立:

$$t_0 < a_{12}t_1+b_{12} \tag{5-11}$$

$$t_2 > a_{12}t_1+b_{12} \tag{5-12}$$

多次重复这个过程,便会产生一系列数据点,a_{12} 和 b_{12} 允许在新的约束条件下取值,从而提高了算法精度。

这两种协议的基础是并非所有数据点都是有用的。每个数据点会产生两种约束下的相对频偏和相对补偿。Tiny-Sync 协议只包含了这些约束条件中的 4 个,一旦获得一个新的数据点,现有的 4 个约束条件需要与新产生的两个约束条件进行比较,只保留 4 个能精确估计频偏与时钟差的约束条件。这种协议的一个缺点是,如果结合其他尚未发生的数据点,可能会做出更好的估计,但是有可能会丢掉这些约束。所以,只有在已经确认某个数据点没用时,Mini-Sync 协议才将其丢弃,它比 Tiny-Sync 协议需要消耗更多的计算和存储成本,但优点是可以提高算法精度。

5.2 定位技术

5.2.1 基本描述

1. 节点定位的基本概念

节点定位是无线传感器网络系统布设完成后面临的首要问题。节点定位是指根据有限

的位置已知的节点确定无线传感器网络中其他节点的位置，在无线传感器网络的节点之间建立起位置关联关系的定位机制。

在无线传感器网络中，需要定位的节点称为未知节点（Unknown Node），即不知道自身位置的节点，在一些资料中也称为盲节点（Blind Node）。而已知位置，并协助未知节点定位的节点称为锚节点（Anchor Node），部分资料中也称为参考节点（Reference Node）、信标节点（Beacon Node）。为了描述方便，统称为锚节点，并遵从 CC2530 数据手册，称位置已知的节点为参考节点。锚节点在网络节点中所占的比例很小，可以通过 GPS 或根据预先指定等手段获取自身的精确位置。每个节点通信半径以内的其他节点，称为邻居节点（Neighbor Node）。如图 5-11 所示，通过锚节点向网络广播信标信息（Beacon），或未知节点通过与邻近的锚节点或已经知道位置信息的邻居节点之间通信，未知节点获得与其他节点的距离或跳数信息，然后根据一定的定位算法得到自身的位置信息。

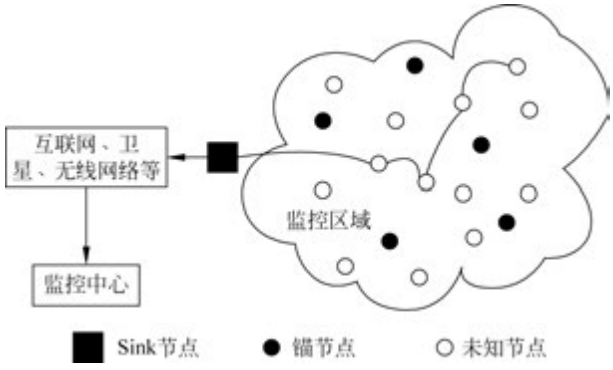


图 5-11 无线传感器网络结构

2. 定位算法的分类

近年来，国内外提出了很多关于传感器网络节点的定位算法，每种算法都有各自的特点。无线传感器网络定位算法到现在为止也没有一个通用的分类标准，通过查阅国内外的相关书籍和资料，从测量技术、定位形式、定位效果、实现成本等方面考虑，节点定位算法分类如下。

1）基于测距的定位算法和距离无关的定位算法

根据是否需要测量实际节点间的距离将定位算法分为基于测距（Range-Based）的定位算法和与距离无关（Range-Free）的定位算法。基于测距的定位算法需要测量相邻节点间的距离或方位，并利用实际测得的距离计算未知节点的位置。与距离无关的定位算法则不需要测量距离和角度信息，利用网络的连通性等信息，估计节点的位置。

2）基于锚节点定位算法和无锚节点辅助的定位算法

根据定位过程中是否使用锚节点将定位算法分为基于锚节点（Anchor-Based）的定位算法和无锚节点辅助（Anchor-Free）的定位算法。基于锚节点的定位算法以锚节点作为定位中的参考点，各节点定位后产生整体的绝对坐标系统。无锚节点辅助的定位算法的定位过程中无须锚节点参与辅助，只需知道节点之间的相对位置，然后各节点先以自身作为参考点，将邻居节点纳入自己的坐标系统，相邻的坐标系统依次合并转换，最后得到整体的相对坐标系统。

3) 集中式计算定位算法与分布式计算定位算法

集中式计算定位算法需要把信息传输到某个中心节点(如服务器),在中心节点完成节点位置的计算。集中式计算定位算法从全局角度统筹规划,计算量和存储量几乎没有限制,可以获得相对精确的位置估算。它的缺点是中心节点位置较近的节点因通信开销大而过早消耗完能量,会导致整个网络与中心节点信息交流的中断,无法实时定位。

分布式计算定位算法也称为并发式算法,依赖节点间的信息交换和协调,由节点自行计算自身位置。在定位精度上,分布式计算定位算法低于集中式计算定位算法,但分布式计算定位算法不会出现部分节点因通信开销大而过早耗完能量。

4) 紧密耦合定位算法与松散耦合定位算法

紧密耦合定位算法是指锚节点不仅被仔细地部署在固定的位置,并且通过有线介质连接到中心控制器。它的特点是适用于室内环境,具有较高的精确性和实时性,时间同步和锚节点间的协调问题容易解决。但这种部署策略限制了系统的可扩展性,代价较大,无法应用于布线工作不可行的室外环境。

松散耦合定位算法的节点采用无中心控制器的分布式无线协调方式。无线中心控制器的协调在精确性上低于紧密耦合定位算法,但是系统部署灵活,只需要节点间的协调和信息交换实现定位。

3. 定位算法的性能分析

由于无线传感器网络与具体的应用有关,根据具体应用的需要,定位的条件和要求会有所不同,所以不同的定位方法和技术在不同的应用场合下所表现出的性能指标存在着巨大的差异。在无线传感器网络中,节点定位算法的性能直接影响其可用性,如何评价一个定位算法的可用性是设计无线传感器网络时需要考虑的问题。

评价一个定位算法,可以从定位精度、规模、锚节点密度、节点密度、覆盖率、容错性和自适应性、功耗和成本等方面考虑,这些性能指标是相互关联的,必须根据应用的具体需求作出权衡,以选择和设计合适的定位方案。下面具体分析各性能指标。

(1) 定位精度。定位精度是衡量定位算法的重要指标之一,使用误差值与节点无线射程的比值 R 来表示。

(2) 规模。不同的定位算法适用于不同的应用场合,定位算法可以定位的目标规模也不尽相同。因此,可以从定位规模方面评价一个定位算法——在一定数量的基础设施或在一段时间内,以定位算法定位的目标数量为评价指标。

(3) 锚节点密度。锚节点密度是使用无线传感器网络中锚节点数与节点总数的比值来表示的。锚节点的位置已知,通常依赖人工部署或携带 GPS 设备实现,用于辅助定位其他节点。锚节点密度也是评价定位系统和算法性能的重要指标之一。

(4) 节点密度。节点密度通常以网络的平均连通度来表示,节点密度的增大会导致网络部署费用的增加,同时会因为节点间的通信冲突问题带来有限带宽的阻塞。

(5) 覆盖率。覆盖率是无线传感器网络定位算法的另一个重要指标。覆盖率用能够实现定位的未知节点与未知节点总数的比值来表示。在保证一定精度的前提下,覆盖率越高越好。

(6) 容错性和自适应性。定位系统和算法在实际的应用中,由于环境、功耗和其他原因的影响,存在严重的多径传播、非视距、传播衰减或节点失效等干扰,因此定位系统和算法的

软硬件必须具有很强的容错性和自适应性，能克服环境和功耗等方面的干扰，尤其是在少量节点发生异常时，能够通过自动调整或重构纠正错误，以适应环境的变化，从而减小定位误差。

(7) 功耗。由于传感器节点一般用电池供电，电池能量有限且不能进行补充，因此功耗是无线传感器网络设计和功能实现最重要的影响因素之一。由于不同的定位算法的通信开销、计算量、存储开销、时间复杂性等一系列关键性指标不同，它们的功耗差别很大。其中通信开销是节点最主要的功耗，因此可以简单地用通信报文量近似衡量节点定位算法的功耗。

(8) 成本。定位系统或算法的成本可从时间、空间和资金成本等方面来评价。时间成本包括定位系统的安装时间、配置时间、定位所需时间等。空间成本包括定位系统或算法所需的基础设施和网络节点的数量、硬件尺寸等。资金成本则包括实现一种定位系统或算法的基础设施、节点设备的总费用。

5.2.2 节点位置的计算方法

未知节点获得与其他邻居节点的距离或相对角度信息，并满足节点定位的计算条件，就可以使用相关定位计算的基本方法计算出自身的位置。定位计算的基本方法包括三边测量法、三角测量法、极大似然估计法、最小最大法。

1. 三边测量法

当未知节点到至少 3 个节点的估计距离已知时，则可使用三边测量法(Trilateration)。三边测量法以 3 个节点圆中心的圆的交点作为未知节点的估计位置。

如图 5-12 所示，已知 A、B、C 三个节点的坐标分别为(x_1,y_1)、(x_2,y_2)、(x_3,y_3)，它们到未知节点 D 的距离分别为 d_1、d_2、d_3，假设节点 D 的坐标为(x,y)，那么存在

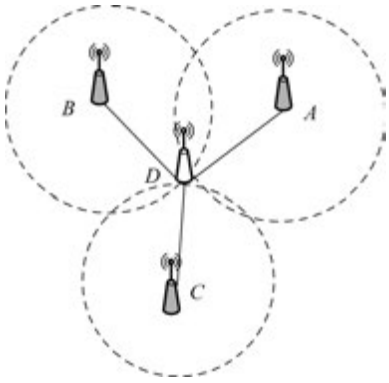


图 5-12 三边测量法

$$\begin{cases}\sqrt{(x-x_1)^2+(y-y_1)^2}=d_1\\ \sqrt{(x-x_2)^2+(y-y_2)^2}=d_2\\ \sqrt{(x-x_3)^2+(y-y_3)^2}=d_3\end{cases} \tag{5-13}$$

由式(5-13)得到节点 D 的坐标为

$$\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}2(x_1-x_3)2(y_1-y_3)\\2(x_2-x_3)2(y_2-y_3)\end{bmatrix}^{-1}\begin{bmatrix}x_1^2-x_3^2+y_1^2-y_3^2+d_3^2-d_1^2\\x_2^2-x_3^2+y_2^2-y_3^3+d_3^2-d_2^2\end{bmatrix} \tag{5-14}$$

虽然这种方法简单，但由于无线传感器网络节点的硬件和能耗限制，通常节点间测距误差较大，因此经常出现 3 个圆无法交于一点的情况。如果这 3 个圆不能交于一点，方程就会得出不正确解，这时就需要使用极大似然估计法实现节点定位。

2. 三角测量法

三角测量法(Triangulation)根据三角形的几何关系进行位置估算。三角测量法首先进行“点在三角形中”的测试，即任意选取 3 个锚节点组成三角形，以测试未知节点是否落在该三角形内。根据测试结果，如果在三角形内部，就可以采用以下方法计算节点的位置。

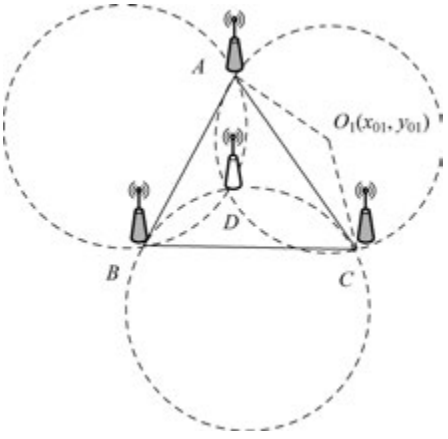


图 5-13 三角测量法

如图 5-13 所示,已知 A、B、C 3 个节点的坐标分别为(x_1,y_1)、(x_2,y_2)、(x_3,y_3),节点 D 到 A、B、C 的角度分别为$\angle ADB$、$\angle ADC$、$\angle BDC$,假设节点 D 的坐标为(x,y)。对于节点 A、C 和 $\angle ADC$,确定圆心为 $O_1(x_{01},y_{01})$半径为 r_1 的圆,$\alpha=\angle AO_1C$,则

$$\begin{cases}\sqrt{(x_{01}-x_1)^2+(y_{01}-y_1)^2}=r_1\\ \sqrt{(x_{01}-x_2)^2+(y_{01}-y_2)^2}=r_1\\ (x_1-x_3)^2+(y_1-y_3)^2=2r_1^2-2r_1^2\cos\alpha\end{cases} \tag{5-15}$$

由式(5-15)能够计算得到圆心 $O_1(x_{01},y_{01})$的坐标和半径 r_1。同理,A、B、$\angle ADB$ 和 B、C、$\angle BDC$ 分别确定相应的圆心 $O_2(x_{02},y_{02})$和 $O_3(x_{03},y_{03})$。最后使用三边测量法,由(x,y)、$O_1(x_{01},y_{01})$、$O_2(x_{02},y_{02})$、$O_3(x_{03},y_{03})$确定节点 D 的坐标。

3. 极大似然估计法

极大似然估计法(Multilateration)寻找一个使测距距离与估计距离之间存在最小差异的点,并以该点作为未知节点的位置。其基本思想为假设一个节点可以获得足够多的信息形成一个由多个方程式组成并拥有唯一解的超限制条件或限制条件完整的系统,那么就可以同时定位跨越多跳的一组节点。如图 5-14 所示,节点 $A_1,A_2,\cdots,A_n$ 的位置坐标已知,D 为未知节点,其估计位置通过最小化测量值间的误差和残余项来获得,具体实现过程如下。

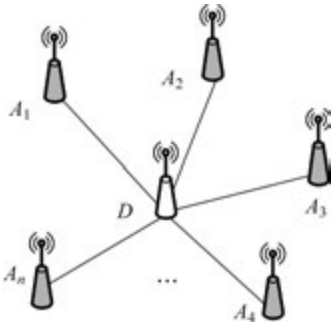


图 5-14 极大似然估计法

已知 n 个节点的坐标分别为 $A_1(x_1,y_1)$,$A_2(x_2,y_2)$,$\cdots$,$A_n(x_n,y_n)$,它们到点 D 的距离分别为 $d_1,d_2,\cdots,d_n$,假设未知节点 D 的坐标是(x,y),则存在以下关系式。

$$\begin{cases}(x_1-x)^2+(y_1-y)^2=d_1^2\\ \vdots\\ (x_n-x)^2+(y_n-y)^2=d_n^2\end{cases} \tag{5-16}$$

将式(5-16)视为线性方程组 $\boldsymbol{AX}=\boldsymbol{b}$,解之可得

$$\boldsymbol{A}=\begin{pmatrix}2(x_1-x_n) & 2(y_1-y_n)\\ \vdots & \vdots\\ 2(x_{n-1}-x_n) & 2(y_{n-1}-y_n)\end{pmatrix},\quad \boldsymbol{X}=\begin{pmatrix}x\\ y\end{pmatrix} \tag{5-17}$$

$$\boldsymbol{b}=\begin{pmatrix}x_1^2-x_n^2+y_1^2-y_n^2+d_n^2-d_1^2\\ \vdots\\ x_{n-1}^2-x_n^2+y_{n-1}^2-y_n^2+d_n^2-d_{n-1}^2\end{pmatrix} \tag{5-18}$$

由标准的最小均方差估计方法可以得到节点 D 的坐标为$\hat{\boldsymbol{X}}=(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$。极大似然估

计法的缺点在于需要进行较多的浮点运算，对于计算能力有限的传感器节点，其计算开销带来的功耗仍不容忽视。

4. 最小最大法

最小最大法(Min-Max)的基本思想是依据未知节点到各锚节点的距离测量值及锚节点的坐标构造若干边界框，即以参考节点为圆心，未知节点到该锚节点的距离测量值为半径所构成圆的外接矩形，计算外接矩形的质心作为未知节点的估计坐标。

如图 5-15 所示，已计算得到未知节点 D 到锚节点的估计距离 d_i，锚节点 A、B、C 的坐标为 $(x_i, y_i)(i=1,2,3)$，加上或减去测距值 d_i，得到锚节点的限制框，即

$$[x_i - d_i, y_i - d_i] \times [x_i + d_i, y_i + d_i] \tag{5-19}$$

这些限制框的交集为 $[\max(x_i - d_i), \max(y_i - d_i)] \times [\min(x_i + d_i), \min(y_i + d_i)]$。3 个锚节点共同形成交叉矩形，取矩形的质心为所求节点的估计位置，最小最大法在不需要进行大量计算的情况下能得到较好的效果。

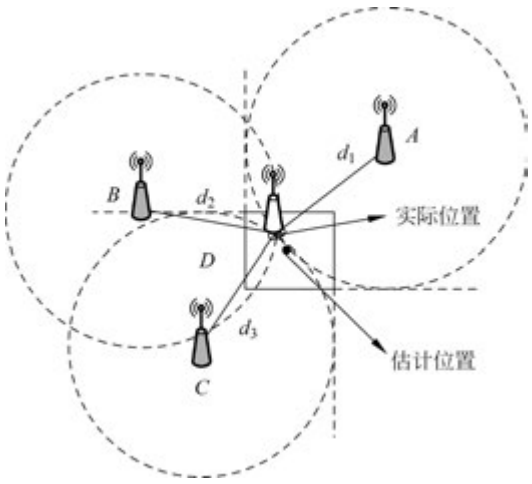


图 5-15 最小最大法

5.2.3 基于测距的定位算法

基于测距的(Range-Based)定位机制通过测量相邻节点间的距离或角度信息，然后再使用三边测量法、三角测量法或最大似然估计法计算节点位置。常用的测距技术有 RSSI、TOA、TDOA 和 AOA。测距定位过程分为以下 3 个阶段。

(1) 测距阶段：未知节点测量到邻近锚节点的距离或角度。

(2) 定位阶段：计算出未知节点与 3 个或 3 个以上锚节点的距离或角度后，利用三边测量法、三角测量法或极大似然估计法计算未知节点的坐标。

(3) 校正阶段：对计算得到的节点的坐标进行循环求精，减小误差，提高定位算法的精度。

基于测距的定位算法需要额外的硬件支持，势必会增加系统的成本，而且基于测距的定位机制还使用各种算法减小测距误差对定位的影响，包括多次测量、循环定位求精等，这些都要产生大量计算和通信开销。基于测距技术的定位机制虽然在定位精度上有可取之处，但并不适用于低功耗、低成本的应用领域。

1. 基于 RSSI 的定位机制

基于 RSSI 的定位机制即基于接收信号强度指示的定位,已知发送节点的发送信号强度,通过测量接收信号强度,计算信号的传播损耗,根据理论或经验信号传播衰减模型将传播损耗转换为距离。得到锚节点与未知节点之间的距离信息后,采用三边测量法或最大似然估计法可计算出未知节点的位置。

根据信号传播理论,由无线信号发射功率和接收功率之间的关系式可计算得到距离 d 的估计值。已知发射功率,在接收节点测量接收功率,计算传播损耗,使用理论或经验的信号传播模型将传播损耗转换为距离。在自由空间中,距发射机 d 处的天线接收到的信号强度为

$$p_r(d)=\frac{p_t G_t G_r \lambda^2}{(4\pi)^2 d^2 l} \tag{5-20}$$

其中,$p_r(d)$为在距离 d 处的接收功率;p_t 为发射机功率;G_t、G_r 分别为发射天线和接收天线的增益;l 为与传播无关的系统损耗因子;λ 为波长。由公式可知,在自由空间中,接收机功率随发射机与接收机距离的二次方衰减。通过测量接收信号的强度,再利用式(5-20)就能计算出收发节点间的估计距离。

实际应用中的情况要复杂得多,尤其是在分布密集的无线传感器网络中。反射、多径传播、非视距(Non-Line-of-Sight,NLOS)、天线增益等问题都会对相同距离产生显著不同的传播损耗。实际环境中通常采用以下经验公式。

$$p_r(d)[\mathrm{dBm}]=p_0(d_0)[\mathrm{dBm}]-10n\lg\left(\frac{d}{d_0}\right)+X_\sigma \tag{5-21}$$

其中,$p_0(d_0)$为距发射机 d_0 的参考信号强度;n 为信号传播路径衰落系数,与特定环境相关;X_σ 为由遮蔽效应引起的正态分布的随机变量。

目前很多通信控制芯片(如 TI 的 CC2530 无线芯片)通常会提供测量 RSSI 的方法,可以在锚节点广播自身坐标的同时即完成 RSSI 的测量,因此 RSSI 是一种低功率、廉价的测距技术,RADAR、SpotON 等许多定位系统中使用了该技术。它的主要误差来源是环境影响所造成的信号传播模型的建模复杂性,反射、多径传播、非视距(NLOS)、天线增益等问题都会对相同距离产生显著不同的传播损耗。因此,这种方法是一种粗糙的测距技术,它有可能产生±50%的测距误差,一般只能适用于对误差要求不高的场合。

2. 基于 TOA 的定位机制

基于 TOA(Time of Arrival)的定位机制,即基于到达时间的定位机制,已知信号的传播速度,通过测量信号传播时间测量距离。

到达时间(TOA)测距法通过测量信号传输时间估算两节点之间的距离,可以通过以下两种方式完成节点间距离的测量。

(1) 测量信号单向传播时间。发送节点记录信号的发送时间并同步告知接收节点,接收节点记录信号的接收时间,通过这种方法测量到信号的传播时间,并由信号的传播速度计算得到节点间的距离。这种方法需要发送节点和接收节点的本地时间精确同步。

(2) 测量信号往返时间差。接收节点在收到信号后直接返回,发送节点测量收发的时间差,由于仅使用发送节点的时钟,因此不需满足节点间时间同步的要求。这种方法的误差来源于接收节点的处理延时,可以通过预先校准等方法获得比较准确的估计。

在基于 TOA 的定位中，需要未知节点与锚节点时间同步，否则会产生较大的定位误差。若无线电从锚节点到未知节点的传播时间为 t，无线电传播速度为 c，则锚节点到未知节点的距离为 tc。如果采用基于 TOA 定位的机制，无线电同步消息告知接收节点超声波发送时间 T_1，接收节点在接收到超声波信号后，记录信号到达时间 T_2，则超声波信号的传播时间 $t=T_2-T_1$。超声波速度乘以传播时间即为节点间距离。节点在计算出与多个邻近锚节点的距离后，可以利用三边测量法或极大似然估计法计算出自身位置。

全球定位系统(GPS)使用 TOA 技术实现定位。GPS 使用了昂贵、高能耗的电子设备精确同步卫星时钟，因此其具有较高的定位精度。在无线传感器网络中，节点在硬件尺寸、价格和功耗方面的要求限制了 TOA 技术在无线传感器网络中的应用。

3. 基于 TDOA 的定位机制

基于 TDOA(Time Difference of Arrival)的定位机制，即基于到达时间差的定位机制，通过计算两种不同无线信号到达未知节点的时间差，再根据两种信号传播速度计算得到未知节点与锚节点之间的距离。例如，Cricket 定位系统中，如图 5-16 所示，发射节点同时发射无线射频信号和超声波信号，接收节点记录到达时间 T_1 和 T_2，已知无线射频信号和超声波传播的速度分别为 c_1 和 c_2，那么两节点间的距离为

$$(T_2-T_1)\times\frac{c_1c_2}{c_1-c_2} \tag{5-22}$$

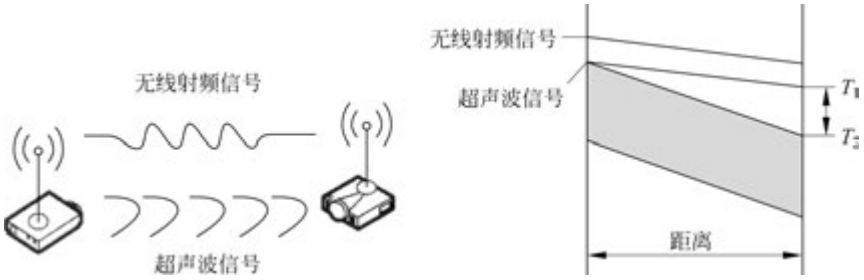


图 5-16 TDOA 定位原理

在节点同步要求上，基于 TOA 的定位机制需要锚节点与未知节点时间同步，基于 TDOA 的定位机制无须同步锚节点与未知节点的时间，只需在网络节点部署完成后，实现锚节点之间的时间同步，由于锚节点在整个无线传感器网络中所占比例很小，因此实现 TDOA 的时间同步比 TOA 代价要小。基于 TDOA 的定位机制定位精度高，易于实现，在无线传感器网络定位方案中得到较多应用。

4. 基于 AOA 的定位机制

基于 AOA(Angle of Arrival)的定位机制，即基于到达角的定位机制，接收节点通过天线阵列或多个接收器结合得到发射节点发送信号的方向，计算接收节点和发射节点之间的相对方位或角度，再通过一定的算法得到节点的估计位置。

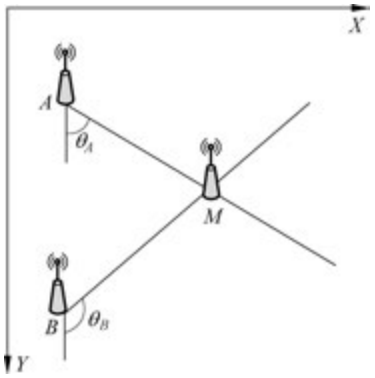


图 5-17 AOA 定位原理

如图 5-17 所示，在二维平面中，未知节点 M 得到与锚节点 A 和 B 所构成的角度之后，就可以利用式(5-23)

确定自身位置,计算出未知节点的估计位置。

$$\tan(\theta_i)=\frac{x-x_i}{x-y_i} \tag{5-23}$$

AOA 定位需要天线阵列或多个接收器实现定位,硬件系统设备复杂,并不适用于对成本敏感的大规模无线传感器网络。且 AOA 定位需要两节点之间存在视线传输(Line-of-Sight,LOS),即使是在 LOS 传输为主的情况下,无线传输的多径效应依然会干扰定位。

5.2.4 距离无关的定位算法

基于测距的定位算法得到结果的精确度要比无须测距的算法高,但是由于基于测距的定位算法必须有测距的过程,势必会增加硬件或增大通信量和计算量。在许多情况下,由于节点尺寸、消耗和能量的限制,基于测距的定位在许多大规模传感器网络中是不实用的。而且通常情况下,当定位误差小于传感器节点无线通信半径的 40%时,定位误差对路由性能和目标追踪精确度的影响不会很大,因此无须测量节点间的绝对距离或方位的定位技术在实际的应用中表现出巨大的应用前景,距离无关的定位技术应运而生。

典型的距离无关的定位算法有质心定位算法、凸规划定位算法、APS 定位算法、Amorphous 定位算法、APIT 算法、SeRLoc 算法等。

1. 质心定位算法

质心定位算法是南加州大学 Nirupama Bulusu 等学者提出的一种仅基于网络连通性的室外定位算法。未知节点以所有在其通信范围内的锚节点的几何质心作为自己的估计位置。

质心定位算法首先确定包含未知节点的区域,计算这个区域的质心,并将其作为未知节点的位置。在质心定位算法中,锚节点周期性地向临近节点广播信标分组,信标分组中包含锚节点的标识号和位置信息。当未知节点接收到来自不同锚节点的信标分组数量超过某一门限或接收一定时间后,就确定自身位置为这些锚节点所组成的多边形的质心。

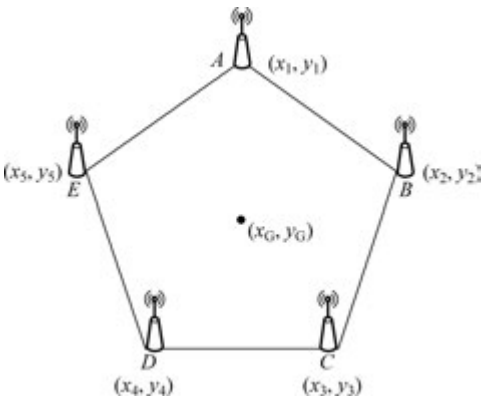


图 5-18 质心定位算法

多边形的几何中心称为质心,设多边形顶点为 $A_1(x_1,y_1)$,$A_2(x_2,y_2)$,…,$A_n(x_n,y_n)$,则其坐标的平均值就是多边形质心的坐标,如式(5-24)所示。

$$(X_G,Y_G)=\left(\frac{\sum_{i=1}^{n}X_i}{n},\frac{\sum_{i=1}^{n}Y_i}{n}\right) \tag{5-24}$$

例如,如图 5-18 所示,多边形 $ABCDE$ 的顶点坐标分别为 $A(x_1,y_1)$、$B(x_2,y_2)$、$C(x_3,y_3)$、$D(x_4,y_4)$、$E(x_5,y_5)$,其质心坐标为

$$(x,y)=\left(\frac{x_1+x_2+x_3+x_4+x_5}{5},\frac{y_1+y_2+y_3+y_4+y_5}{5}\right)$$

质心定位算法的最大优点是非常简单,计算量小,易于实现;完全基于网络的连通性,无需锚节点和未知节点之间的协调。但是,估计的精度和锚节点的密度有关,密度越大,定

位精度越高。

2. 凸规划定位算法

凸规划(Convex Optimization)定位算法是加州大学 Doherty 等提出的一种完全基于网络连通性诱导约束的定位算法。

1) 凸规划定位算法的基本思想

凸规划定位算法的思想是把整个网络模型化为一个凸集,将节点间点到点的通信连接视为节点位置的几何约束,通过产生一系列相邻的约束条件,蕴含着节点的位置信息,将这些约束条件组合起来,就可以得到此未知节点可能存在的区域。从而将节点定位问题转化为凸约束优化问题,然后使用线性矩阵不等式、半定规划或线性规划方法得到一个全局优化的定位解决方案,确定节点位置。如图 5-19 所示,根据未知节点 D 与锚节点 A、B、C 之间的通信连接和节点的无线通信半径,计算得到 3 个圆相交的部分,为未知节点可能存在的区域,使用相应的计算方法得到包含 3 个圆公共部分的矩形区域(图中的阴影部分),然后计算该矩形区域的质心作为未知节点的位置估计。

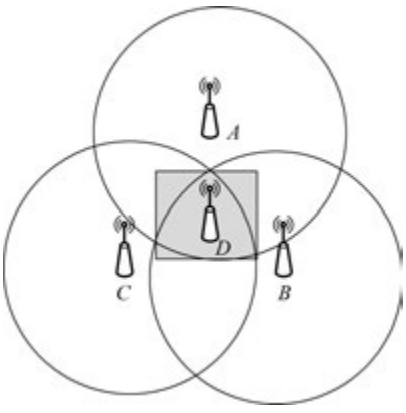


图 5-19 凸规划定位原理

2) 凸规划定位算法的理论基础

凸规划定位算法中,每个节点所面对的几何约束都被表示为线性矩阵不等式(Linear Matrix Inequality,LMI),而线性矩阵不等式可以利用半定规划(Semi Definite Program,SDP)的方法进行简单求解,最后求解的结果就是得到每个节点的约束区域。其中 SDP 问题可以表示为

$$\text{minimum } \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x},\text{约束于}\begin{cases}F(x)=\boldsymbol{F}_0+x_1\boldsymbol{F}_1+\cdots+x_n\boldsymbol{F}_n=0\\ \boldsymbol{F}_i=\boldsymbol{F}_i^{\mathrm{T}}\\ \boldsymbol{AX}=\boldsymbol{b}\end{cases} \tag{5-25}$$

$\boldsymbol{x}$ 为所求问题的决策向量。在二维空间,每个节点位置表示为$(\boldsymbol{x},\boldsymbol{y})$。在位置估计时,所有节点的坐标可以表示为一个向量 $\boldsymbol{x}=[x_1 \quad y_1\cdots x_m \quad y_m \quad x_{m+1} \quad y_{m+1}\cdots x_n \quad y_n]^{\mathrm{T}}$,前 m 个坐标为锚节点坐标,锚节点位置坐标已知无须计算,因此剩下的 $n-m$ 个未知节点的坐标由算法计算得到。

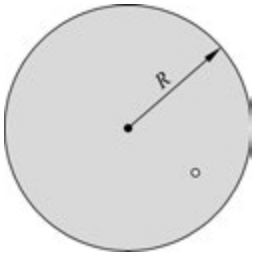


图 5-20 径向约束示意图

3) 凸约束模型

在无线传感器网络中,无线传感器节点的无线通信模型可看作如图 5-20 所示的模型,此时凸约束模型为径向约束(Radial Constraints),LMI 可表示为

$$\|\boldsymbol{a}-\boldsymbol{b}\|_2 \leqslant R \Rightarrow \begin{bmatrix}\boldsymbol{I}_2 R & \boldsymbol{a}-\boldsymbol{b}\\ (\boldsymbol{a}-\boldsymbol{b})^{\mathrm{T}} & R\end{bmatrix} \geqslant 0 \tag{5-26}$$

其中,$\boldsymbol{I}_2$ 为 2×2 单位矩阵;R 为节点通信半径;$\boldsymbol{a}$ 和 $\boldsymbol{b}$ 分别为两节点坐标。在二维空间,用矩阵的 Schur 定理将非线性矩阵

不等式转化为线性矩阵不等式。

另外,并不是所有约束都能用线性矩阵不等式来表示的,只有当约束区域是凸区域时才能用线性矩阵不等式表示。例如,圆、圆弧、梯形等都是凸区域。半定规划的方法是集中式的,对于大规模无线传感器网络,其约束条件个数 n 可能非常大,收集如此多的约束条件需要极大的通信开销,而且这些约束条件需要汇集到一个中心节点进行统一计算,中心节点周围的节点通信开销较大,可能过早地消耗完能量。

凸规划是一种集中式定位算法,在锚节点比例为 10%的条件下,定位精度约为 100%。为了高效工作,锚节点需要被部署在网络的边缘,否则外围节点的位置估算会向网络中心偏移。

5.2.5 典型的定位系统

到目前为止,国内外研究机构和大学已经开发出很多成熟的无线传感器网络定位系统,很多定位系统已经走出实验室,应用于工农业中。现有的定位系统既有偏重紧密耦合型和基于基础设施的定位系统,又有偏重松散耦合型和无需基础设施的定位系统。无需基础设施的定位系统已经成为 WSN 领域的研究热点。在设计定位系统时,可以参考借鉴一些典型的定位系统的设计原理和设计方法。下面对典型室内定位系统 Active Badge、Active Office 和 Cricket 等加以归纳总结。

1. Active Badge 定位系统

Active Badge 定位系统是最早的室内定位系统之一,是符号定位的便携式定位系统。该定位系统在建筑物内部署红外传感器节点,通过以太网将节点组成一个传感器网络,用于定位建筑物内部的人或物体。

Active Badge 定位系统如图 5-21 所示,传感器节点(Sensor)的位置已知,配备有红外信号接收器,在整个系统中充当锚节点的角色。需要定位的人或物体携带一个称为 Badge 的设备,作为定位系统中的未知节点。Badge 每隔 15s 发射一个持续时间约 0.1s 的全局唯一

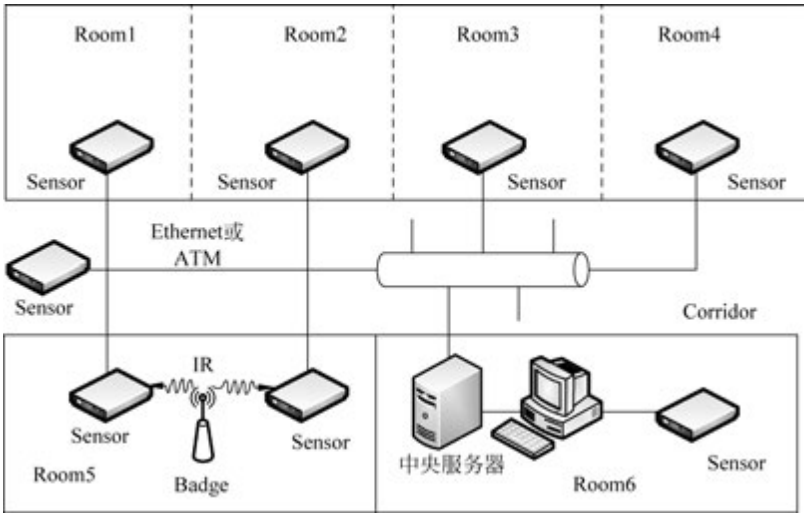


图 5-21 Active Badge 定位系统

的红外(IR)信号。Badge在建筑物内部活动,周期性主动地向周围发送全局唯一的身份标识信息,发送的周期性信号被附近传感器节点捕捉到并存储,通过网络送往中央服务器。中央服务器不断收集和更新Badge的位置信息,所以可以从中央服务器上查询出某个Badge的当前所在位置。因为红外光不能穿过墙壁,所以每个房间(Room)就成为Badge所能分辨的最小单位,而且每个房间中至少有一个红外接收器。

Active Badge定位系统的定位精度以房间为单位。因Badge设备依靠电池供电,它的信号发射频率是决定系统性能的重要因素,每15s发射一次红外信号,系统平均能耗会非常小。理论上,Badge的电池可维持一年左右。红外信号的持续时间仅为0.1s,当两个Badge被放置在同一位置时,两个信号冲突的概率为2/150,则在同一个位置可安全地检测多个目标。并且,为了减小信号冲突的概率,在Badge晶振上进行了特殊的设计,设计10%的摆动率,即使两个Badge在某一时间同步,在几个红外信号周期后也将失去同步。

2. Active Office定位系统

Active Office定位系统是Andy Ward等开发的一种室内定位系统。与Active Badge定位系统使用红外定位不同,Active Office定位系统使用了超声波定位。

Active Office定位系统中,需要定位的移动设备由微处理器、418MHz无线收发器、Xilinx FPGA和5个半球形排列的超声波换能器组成,如图5-22(a)所示。每个移动设备都分配有全局唯一的16位地址。接收器都有一个超声波探测器,用以接收移动设备发出的超声波,如图5-22(b)所示。接收器以阵列方式(间距为1.2m)安装在固定的位置,如房间的天花板,接收器均配备有网络接口,以菊花链(Daisy-Chain)的形式连接到中心控制PC。

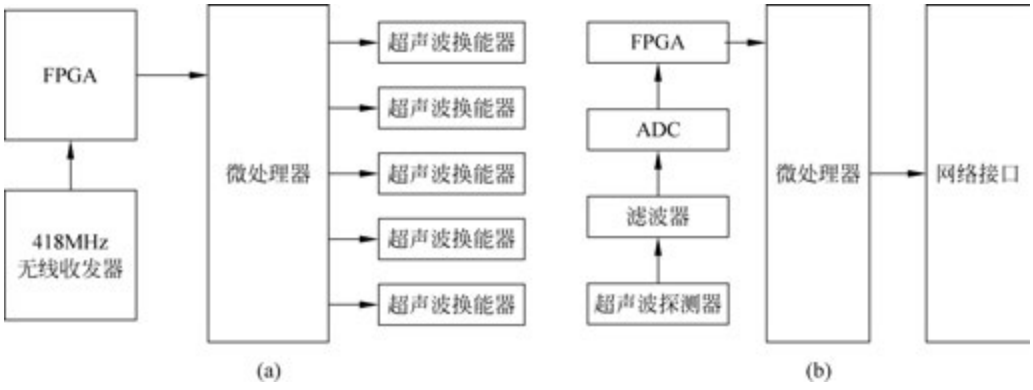


图5-22 Active Office节点组成框图

Active Office定位系统的工作流程如图5-23所示。定位开始时,中央控制器发送一个无线消息,该消息包含目标移动设备的16位地址,无线消息被所有移动设备接收,由FPGA解码判断消息地址是否与自身地址相符。需要定位的目标设备收到无线消息以后,发出一个短时的超声波脉冲。中心控制器发送无线消息的同时通过网络向接收器发送复位信号,接收器监控超声波探测器是否接收到信号,监听时间为20ms。当超声波脉冲信号被接收器阵列接收到时,接收器计算接收到超声波信号的峰值时间,并计算超声波脉冲与中央控制器复位信号的到达时间差。中央控制器通过网络收集接收器记录的时间差,根据这些时间差

可以计算出目标设备到每个超声波接收器之间的距离,在中央控制器上通过求解多边问题(Multilateration)计算设备位置,从而得到移动设备的位置。尽管移动设备只发送了一个短时超声波脉冲,但是由于环境的影响,可能在接收器端接收到多个超声波信号,因此接收器捕获超声波信号的峰值。无线消息的发送周期为200ms,移动设备在接收无线消息后,进入能量节省状态,在195ms以后恢复工作,等待下一次无线消息,因此设备在大多数时间处于休眠状态,从而降低了设备的能耗。

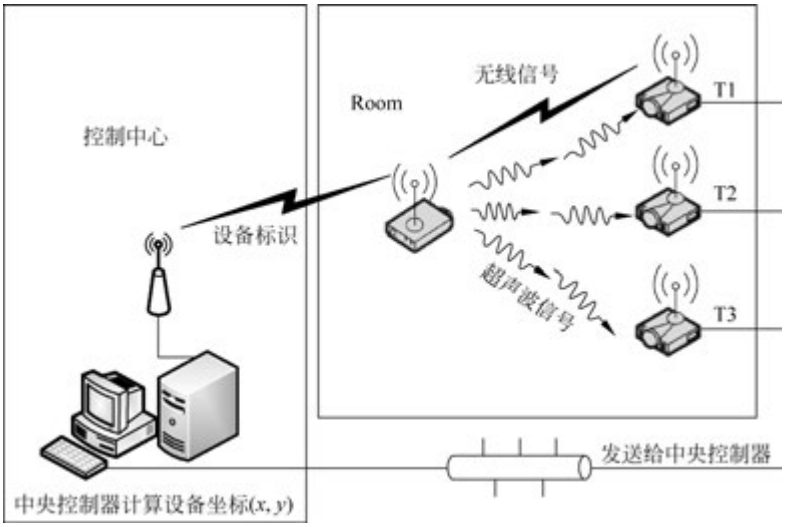


图 5-23 Active Office 定位系统的工作流程

Active Office定位系统可以通过统计的方法去除那些过分偏离正常的数据,从而提高距离估计的准确性。实验测试结果表明,95%的位置估计值与真实位置误差在14cm以内。

3. Cricket 定位系统

Cricket定位系统是麻省理工学院为了弥补紧密耦合定位系统的不足而开发的最早的松散耦合室内定位系统,主要用于定位移动节点在建筑物内部的位置。

如图5-24所示,在Cricket定位系统中,锚节点部署在建筑物内部,安装在固定位置,如天花板上。锚节点周期地同时发射无线射频信号和超声波信号,射频信号中包含该锚节点的位置和全局唯一的标识信息,超声波信号用于辅助未知节点计算和锚节点的距离。未知节点(Cricket定位系统中为Listener)是需要定位的人或物体,当接收到锚节点的射频信号时,Listener打开超声波接收器,接收到超声波信号后使用TDOA技术测量其与锚节点的距离。Listener从锚节点处获得信息,通过这些信息推断自身所处的位置,并为应用程序提供API。在实验测试中,当Listener能够获得3个以上锚节点距离时,使用三边测量法实现物理定位,否则就以房间为单位提供符号定位。

与Active Badge、Active Office定位系统不同的是,Cricket定位系统不需要外部计算机系统或控制中心辅助计算节点位置,节点自身就能完成位置的估计。

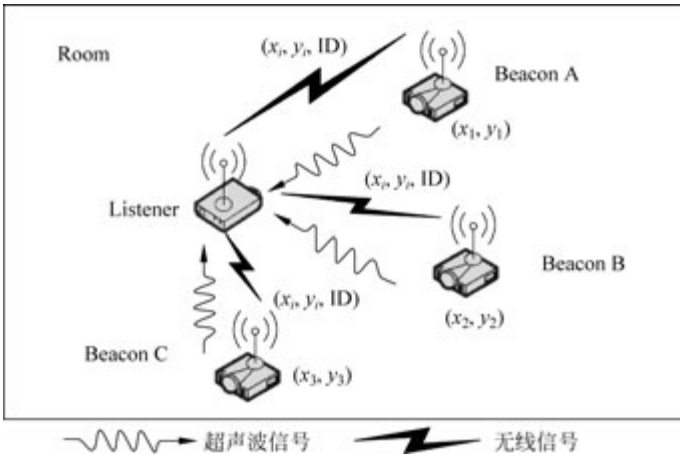


图 5-24 Cricket 定位系统

5.3 数据融合

5.3.1 数据融合的基本概念

数据融合可以充分利用多个传感器资源，通过合理支配和使用这些传感器及其观测信息，把多个传感器在空间或时间上的冗余或互补信息依据某种准则进行组合，以获得被测对象的一致性解释或描述。

数据融合利用计算机技术对按时序获得的多传感器观测信息在一定的准则下进行多级别、多方面、多层次的信息检测、相关估计和综合，以获得目标的状态和特征估计，产生比单一传感器更精确、完整、可靠的信息以及更优越的性能，而这种信息是任何单一传感器所无法获得的。

数据融合功能模型如图 5-25 所示。

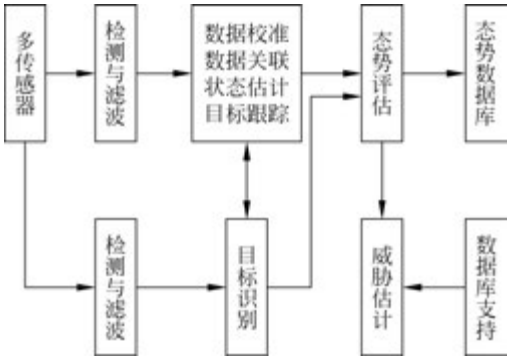


图 5-25 数据融合功能模型

数据融合技术可以带来的好处有以下 6 方面。

(1) 提高信息的可信度。利用多传感器能够更加准确地获得环境与目标的某一特征或一组相关特征，整个系统所获得的综合信息比任何单一传感器所获得的信息具有更高的精度和可靠性。

(2) 扩展系统的空间、时间覆盖能力。多个传感器在空间上交叠,时间上轮流工作,扩展了整个系统的时空覆盖范围。

(3) 降低系统的信息模糊程度。由于采用多传感器信息进行检测、判断、推理等运算,降低了事件的不确定性。

(4) 改善系统的检测能力。多个传感器可以从不同的角度得出结论,提高了系统发现问题的概率。

(5) 提高系统的可靠性。多传感器相互配合,系统就具有冗余度,某个传感器失效不会影响整个系统,降低了系统的故障率。

(6) 提高系统决策正确性。多传感器工作提高了事件的可信度,决策级融合所得结论也更可靠。

5.3.2 数据融合分类

根据处理融合信息方法的不同,数据融合可分为集中式、分布式和混合式 3 种。

(1) 集中式。各传感器的数据都发送到融合中心进行融合处理。这种方法实时性能好,数据处理精度高,可以实现时间和空间的融合。但融合中心的负荷大,可靠性低,数据传输量大,对融合中心的数据处理能力要求高。

(2) 分布式。各传感器对自己测量的数据单独进行处理,然后将处理结果发送到融合中心,由融合中心对各传感器的局部结果进行融合处理。与集中式相比,分布式处理对通信带宽要求低,计算速度快,可靠性和延续性好,系统生命力强。但分布式数据融合精度没有集中式高,每个传感器自己作出决策的过程中增大了融合处理的不确定性。

(3) 混合式。集中式与分布式的组合,可以均衡上述两种方式的优缺点,但系统结构也会变得复杂。

根据融合处理的数据种类,数据融合可以分为时间融合、空间融合和时空融合。

(1) 时间融合。对同一传感器对目标在不同时间的测量值进行融合。

(2) 空间融合。对不同传感器在同一时刻的测量值进行融合。

(3) 时空融合。对不同传感器在一段时间内的测量值不断地进行融合。

根据信息的抽象程度,数据融合可分为数据级融合、特征级融合和决策级融合 3 种。

(1) 数据级融合。如图 5-26 所示,直接在采集到的原始数据层上进行融合,在传感器采集的原始数据未经处理之前就对数据进行分析和综合,这是最低层次的数据融合。这种融合的主要优点是能保留尽可能多的原始现场数据,提供更多其他融合层次不能提供的细节信息。由于这种融合是在信息的最底层进行的,传感器的原始信息存在不确定性、不完全性和不稳定性,这就要求在进行数据融合时有较高的纠错能力。要求各传感器信息之间具有精确到一个数据的校准精度,故要求各传感器信息来自拥有同样校准精度的传感器。

数据级融合通常用于多源图像复合、图像分析和理解、同类(同质)雷达波形的直接合成、多传感器遥感信息融合等。

(2) 特征级融合。如图 5-27 所示,在中间层次进行数据融合,先对来自传感器的原始数据提取特征信息(通常来讲,提取的特征信息应是像素信息的充分表示量或充分统计量),然后按特征信息对多传感器数据进行分类、汇集和综合。特征级融合的优点在于实现了可观的信息压缩,有利于实时处理,并且由于所提取的特征信息直接与决策分析有关,因而融

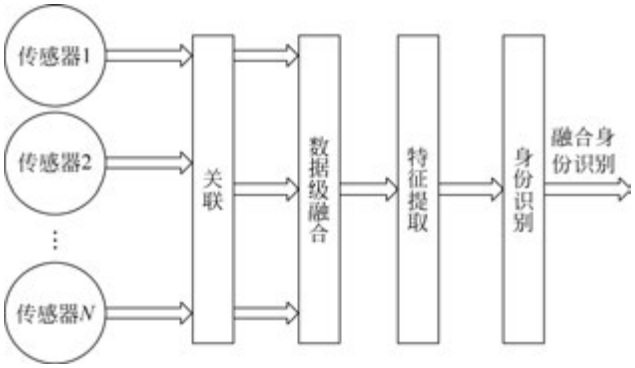


图 5-26 数据级融合过程

合结果能最大限度地给出决策分析所需要的特征信息。特征级融合分为两大类：目标状态数据融合和目标特性融合。目标状态数据融合主要用于多传感器目标跟踪领域。目标特性融合多用于多传感器的目标识别领域。

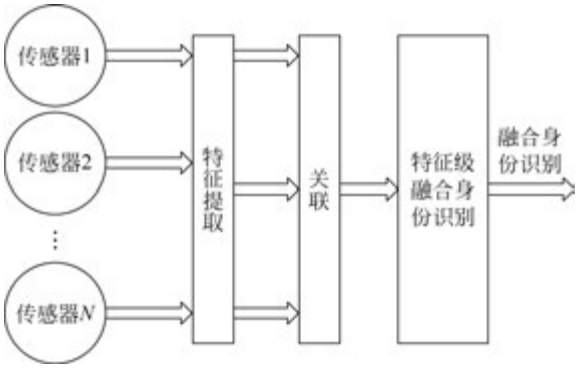


图 5-27 特征级融合过程

(3) 决策级融合。如图 5-28 所示，在最高级层进行数据融合。决策级融合是一种高层次融合，融合之前，每种传感器的信号处理装置已完成决策或分类任务。信息融合只是根据一定的准则和决策的可信度作出最优决策，以便具有良好的实时性和容错性，使在一种或几种传感器失效时也能工作。决策级融合的结果是为决策提供依据，因此，决策级融合通常是从具体的决策问题出发，充分利用特征级融合所提取的对象的各类特征信息，采用特定的融合技术来实现。决策级融合是直接针对具体决策目标的，融合结果对决策的水平有直接影响。

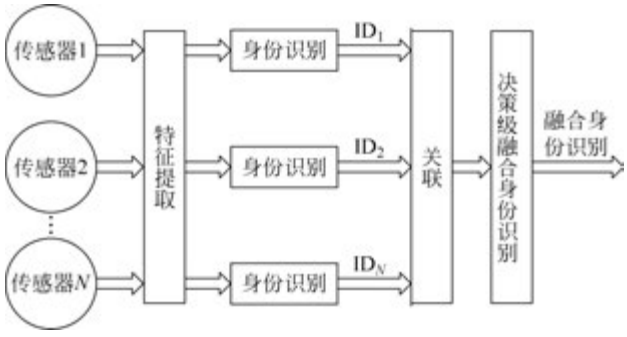


图 5-28 决策级融合过程

决策级融合的主要优点如下。

① 具有很高的灵活性；

② 系统对信息传输的带宽要求较低；

③ 能有效反映环境或目标各个侧面的不同类型信息；

④ 当一个或几个传感器出现错误时，通过适当融合，系统还能获得正确的结果，所以具有容错性；

⑤ 通信量小，抗干扰能力强；

⑥ 对传感器的依赖性小，传感器可以是同质的，也可以是异质的；

⑦ 融合中心处理代价低。

但是，决策级融合需要对传感器采集的原始信息进行预处理并获得各自的判决结果，因此预处理代价很高。

5.3.3 基于组播树的数据融合算法的实现

WSN 数据融合算法是一种准确度和实时性都较好的基于组播树的数据融合算法，该算法分 3 个阶段。第一阶段为组播树的构造阶段，建立一棵数据融合树；第二阶段为兴趣散布阶段，实现对数据的查询和路由；第三阶段为数据融合阶段，实现数据的融合。

1. 组播树的构造

定义 5-1 给定图 $G=(V,E)$，其 V 是图 G 的节点集，E 是图 G 的边集；边的费用函数 $C:E\rightarrow R$；组播节点集 D 为 V 的子集。从图 G 中找出覆盖 D 中所有节点的最小生成树 $T_s(V_T,E_T)$，使得树的费用最小，该费用可表示为

$$\mathrm{Cost}(T_s)=\min_{T(V_T,E_T)}\sum_{E\in T(V_T,E_T)}C(E) \tag{5-27}$$

该最小生成树称为 Steiner 树。在传感器网络中，$C(E)$可以粗略取值为 1，因为在无线链路中，所有节点以相同的功率发送数据，只要节点在其通信半径内均可以收到。当然，如果考虑到节点可以自适应地调整发射功率，则 $C(E)$应该是两个直接相邻节点的距离。

定义 5-2 在生成的 Steiner 树中，称属于 D 的节点为组播节点，同时任何一个多播节点都可以是源节点；属于 V_T 但不属于 D 的节点称为非组播节点或 Steiner 节点，D 称为组播节点集或多播组。

当 $D=V$ 时，求 Steiner 树简化为求图的最小生成树问题；当 $D=2$ 时，求 Steiner 树退化为求两点之间的最短路径问题；除此之外，求 Steiner 树问题是一个 NP 完全问题。

定义 5-3 给定一组 $D\in V$，指示函数 $I_D:V\rightarrow(0,1)$的定义：如果 $u\in D$，$I_D(u)=0$；如果 $u\notin D$，$I_D(u)=1$。指示函数用于区分节点 D 是否属于组播节点组。

传统的组播树是单个节点向多个目的节点分发数据的过程，而在传感器网络中是多源节点向一个 Sink 节点发送数据的过程，这是一个反向组播树的构造过程。但基于传感器网络的特点，有效的组播树构造算法应该具有时间复杂度低且分布式的特点，这样对传感器网络而言，才具有实用性。目前典型的求解 Steiner 树的算法包括 KMB 算法、SPH 算法等。

组播树构造阶段的主要任务就是构造一棵组播树，为下一阶段的数据融合创造条件。组播树的构造可参考 DDMC(Destination Driven Multi-Cast)算法。$\pi[v]$表示节点 v 的父节点，$d[v]$表示节点的标注值，$w(u,v)$表示链路的费用。DDMC 算法的伪代码如下。

```
for each v∈V do
    D[v] = 0;
    π[v] = null;
S = 0;
Q = V;
while Q≠0 do
    U = pop-min(Q);
    S = S⋃{u};
for each V∈adj[u] do
    if d[v]>I_D(u) + w(u,v);
    if v∉S
        d[v] = I_D(u) + w(u,V);
        π[v] = u;
u 记录其子节点 v;
```

2. 兴趣散布阶段

兴趣散布阶段的主要任务是簇首节点向整个网络散布其感兴趣的数据类型。Sink 节点可以简单地通过泛洪(Flooding)的方式进行散布,当网络规模比较小,而且 Sink 节点的兴趣变化比较缓慢时,由于实现简单,具有较好的性能。但当网络规模很大,或兴趣变化比较频繁时,泛洪法并不合适,这里采用一种基于分簇的优化定向扩散路由进行散布。

基于分簇的优化定向扩散(Cluster Based Optimizing Directed Diffusion,CBODD)路由是利用分簇思想提高定向扩散能源有效性的无线传感器网络路由协议。通过分簇简化网络拓扑以及减少泛洪中的冗余消息达到提高网络能源有效性的目的,使之能适用于大规模无线传感器网络。

1) 分簇算法

这里采用一种基于被动分簇算法的分簇机制,通过定向扩散协议本身的泛洪传播建立网络拓扑。因此,不需要额外的控制信息(通过泛洪传播)建立或维护网络拓扑,仅需在原有的包结构中加入一个与簇相关的信息结构实现路由的建立与维护。

该被动分簇算法是指仅当网络中有数据通信需求时才在网络中建立分簇网络拓扑,网络拓扑的建立与维护在本地完成。因此,不需要单独的控制命令,这样就达到了降低能耗的目的。

2) 路由的建立与维护

CBODD 路由的建立是由网络中的 Sink 节点发起的,当节点分布完成后,网络处于初始状态,如果之后仍然没有数据传输要求,则网络保持当前的初始状态,因此形成分簇的网络拓扑是由 Sink 节点发起的,分簇的网络拓扑形成以后,只有当网络拓扑受到损害或无法维持下去时,网络才会重新回到初始状态,等待 Sink 节点的下一次传输要求重新建立网络拓扑并形成由源节点到 Sink 节点的路由。

在整个 CBODD 路由协议的建立过程中,主要分为 3 个阶段:扩散阶段、建立阶段、路径加强阶段。这与原定向扩散协议相同,但是这 3 个阶段是基于分簇完成的。CBODD 路由建立的完整过程如图 5-29 所示。

(1) 扩散阶段。当 Sink 节点广播兴趣时,若网络处于初始状态,则网络拓扑随兴趣在网络中泛洪传播而建立起来,同时建立起从源节点(S)到 Sink 节点的网络梯度;当网络成簇时,则兴趣仅沿簇首扩散,之后建立从源节点到 Sink 节点的网络梯度。

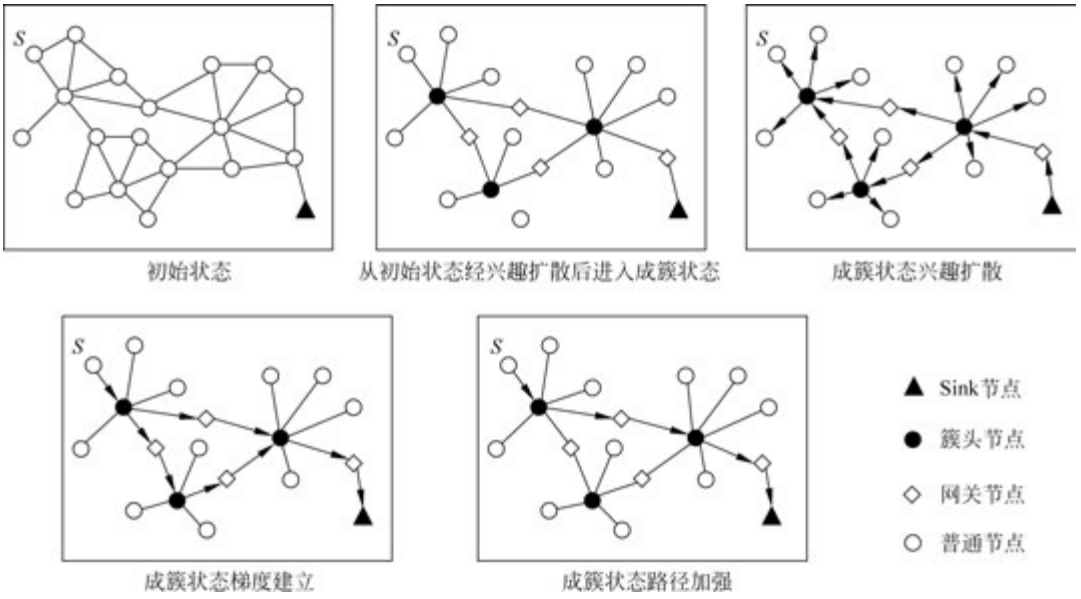


图 5-29 CBODD 路由建立的完整过程

(2) 建立阶段。当源节点采集到与兴趣匹配的数据时,源节点把数据发送给自己的簇首节点,簇首节点通过网关节点向多个相邻的簇首节点发送数据,Sink 节点可能收到多个路径的相同数据。中间的簇首节点收到其他簇首节点转发的数据后,首先查询兴趣列表的表项。如果没有匹配的兴趣表项就丢弃数据。如果存在响应的兴趣表项,则检查与这个兴趣对应的数据缓冲池(Data-Cache),数据缓冲池用来保存最近转发的数据。若在数据缓冲池中有与接收到的数据匹配的副本,说明已经转发过这个数据,为避免出现传输环路而丢弃这个数据;否则,检查该兴趣表项中的邻居节点信息。对于转发的数据,数据缓冲池保留一个副本,并记录转发时间。

(3) 路径加强阶段。节点在收到从 Sink 节点发来的数据后,启动建立到源节点的加强路径,后续数据将沿着加强路径以较高的速率进行传输。

3) CBODD 的实现机制

利用 NS2 提供的网络路由应用编程接口(Network Routing Application Programming Interface)实现基于定向扩散的 CBODD 协议。在网络应用编程接口中提供了一种过滤器 API(Filter API),利用这个 API 可以将设计的模块与定向扩散内核(Directed Diffusion Core)连接起来。在 CBODD 协议的实现过程中,对 NS2 中的定向扩散协议作以下修改。

(1) 原定向扩散协议包结构中增加与簇相关的信息。

(2) 增加一个优先级比梯度过滤器(Gradient Filter)高的分簇过滤器(Clustering Filter),分簇过滤器的主要功能是实现网络拓扑的建立与维护。如图 5-30 所示,描述了加入分簇过滤器的定向扩散中的消息流。分簇过滤器在本地应用(Local Application)与梯度过滤器之间,当有消息输入(来自网络 1 或本地应用 3)时,消息先于梯度过滤器进入分簇过滤器,经过分簇过滤器的包过滤处理,然后采取以下方式进行转发消息包。

① 丢弃;

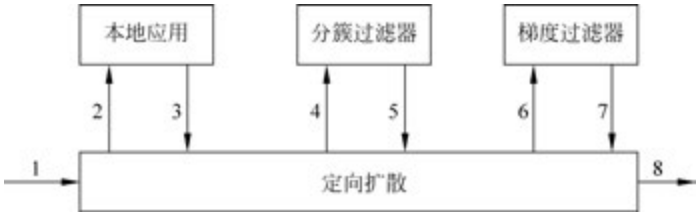


图 5-30 加入分簇过滤器的定向扩散中的消息流

② 转发给本地应用(来自网络的消息);

③ 转发给梯度过滤器,经梯度过滤器再处理;

④ 直接转发到网络上。

因此,实现 CBODD 协议实际上主要是如何设计分簇过滤器。

在 NS2 中利用网络应用编程接口设计实现一个过滤器分为两步。

(1) 创建定向扩散路由类,如

```
dr = NR::createNR();
fcb = newClusterFilter;
```

其中,NR::createNR()是 NR(Network Routing)类的一个方法,用于创建一个 NR 类,并返回一个指向它的指针。ClusterFilter 是本文定义的一个分簇过滤器的类,定义如下。

```
//定义分簇过滤器的优先级
#define LOGGING_APP_PRORITY 16
//定义分簇过滤器类
class ClusterFilter:public FilterCallbackf{
    public:
    void recv(Message * msg,handle h);
};
```

(2) 建立分簇过滤器。

```
filterHandle = setupFilter();
```

其中,setupFilter()函数用于创建一个与所有兴趣消息相匹配的属性,并将分簇过滤器加入定向扩散路由,用来过滤进入定向扩散路由中的数据包。函数过程描述如下。

```
//分簇过滤器建立过程函数
Handle setupFilter(){
    NRAttrVec  attrs;
    Handle h:
    //这个属性用于匹配所有的兴趣消息
    attrs.push_back(NRClassAttr.make(NRAttribute::EQ,
    NRAttribute::INTEREST_CLASS)):
    //利用 Filter API 的加入过滤器函数建立分簇过滤器
    h = ((DiffusionRouting * )dr)->addFilter(&attrs,
    LOGGING_APP_PRORITY,fcb):
    ClearAttrs(&attrs);
    //返回分簇过滤器的一个句柄
    return;
}
```

3. 数据融合阶段

当组播树构造后,每个源节点将数据转发给其父节点,父节点根据其子节点数目做以下

动作：如果父节点仅有一个子节点，父节点将数据转发给其父节点；如果父节点有多个子节点，父节点将等待一段合适的时间，直到其所有子节点都将数据转发给其为止，然后该父节点进行数据融合再进行转发，而其他的非组播节点不断进行数据融合，直到将数据融合到Sink节点为止，其融合采取一种改进的一致性融合算法。

1）一致性融合算法

基于数理统计方法的一致性算法的基本原理如下。设有 n 个传感器从不同位置各自独立地对某一目标参数进行测量，假设第 i 个传感器的测量值为 x_i。由于各种因素的影响，测量值具有随机性，一般 x_i 服从正态分布 $N(u_i,\sigma_i)$，σ_i 表示第 i 个传感器的测量精度，其测量模型可表示为

$$p(x_i)=\frac{1}{\sqrt{2\pi\sigma_i}}\mathrm{e}^{\frac{(x-\mu_i)^2}{2\sigma_i^2}},\quad i=1,2,\cdots,N \tag{5-28}$$

采用 d_{ij} 和 d_{ji} 作为传感器 i 和传感器 j 之间测量数据的相互支持性，称为置信距离。d_{ij} 和 d_{ji} 越小，表示两个传感器的测量值 x_i 和 x_j 越接近；反之则表示两个传感器的测量值相差越大。因此，d_{ij} 和 d_{ji} 也称为传感器 i 和传感器 j 的融合度，表示为

$$\begin{cases} d_{ij}=2\left|\int_{x_i}^{\sigma_j}p_i(x\mid x_i)\mathrm{d}x\right|=2A \\ d_{ji}=2\left|\int_{x_j}^{\sigma_i}p_j(x\mid x_j)\mathrm{d}x\right|=2B \end{cases} \tag{5-29}$$

其中，$p_i(x|x_i)$和 $p_j(x|x_j)$为条件概率。

$$p_i(x\mid x_i)=\frac{1}{\sqrt{2\pi\sigma_i}}\mathrm{e}^{\frac{(x-\mu_i)^2}{2\sigma_i^2}}$$

$$p_j(x\mid x_j)=\frac{1}{\sqrt{2\pi\sigma_j}}\mathrm{e}^{\frac{(x-\mu_j)^2}{2\sigma_j^2}}$$

利用正态分布的性质可以推出置信距离矩阵 $\boldsymbol{D}$ 为

$$\boldsymbol{D}=(d_{ij})_{N\times N}=\begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1N} \\ d_{21} & d_{22} & \cdots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & d_{NN} \end{bmatrix} \tag{5-30}$$

根据多个传感器的置信距离矩阵 $\boldsymbol{D}$ 可以确定任意传感器的测量值与另一传感器测量值的相互关系。

根据经验或多次测试的结果，给定一个阈值 ε_{ij}，令

$$r_{ij}=\begin{cases}1, & d_{ij}>\varepsilon_{ij}\\ 0, & d_{ij}<\varepsilon_{ij}\end{cases}$$

若第 i 个传感器被第 j 个传感器支持的程度 r_{ij} 大于给定的阈值 ε_{ij}，则表示第 i 个传感器的测量数据是有效的；反之，则表示第 i 个传感器的测量数据是无效的。

由传感器之间的置信距离矩阵 $\boldsymbol{D}$ 可以得到传感器之间的关系矩阵 $\boldsymbol{R}$ 为

$$\boldsymbol{R}=(r_{ij})_{N\times N}=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1N} \\ r_{21} & r_{22} & \cdots & r_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ r_{N1} & r_{N2} & \cdots & r_{NN} \end{bmatrix} \tag{5-31}$$

在关系矩阵 $\boldsymbol{R}$ 中，对于第 i 个传感器和第 j 个传感器共存在 3 种关系：

(1) $r_{ij}=r_{ji}=0$，表示传感器 i 与传感器 j 相互独立；

(2) $r_{ij}=1, r_{ji}=0$，表示传感器 i 对传感器 j 弱支持；

(3) $r_{ij}=0, r_{ji}=1$，表示传感器 i 与传感器 j 相互强支持。

由关系矩阵 $\boldsymbol{R}$ 可以计算出各传感器的测量数据被其他传感器测量数据支持的程度，被支持程度越高的传感器数据在融合中的重要程度越大，根据各传感器数据的重要程度可对其进行融合处理，以获得对被测对象较精确的估计值。

该算法的不足之处如下。

(1) 当 $d_{ij}\neq d_{ji}$，传感器 i 与传感器 $j(i\neq j)$ 的测量精度不同时，置信距离 d_{ij} 和 d_{ji} 是不相同的，这与通常的距离对称性的要求不一致。

(2) 当 $\sigma_i\neq\sigma_j$ 时，$|x_i-x_j|/\sigma_i$ 的统计意义不明确，只有在 $\sigma_i=\sigma_j$ 时才有明确的统计意义，表示标准正态分布的取值。

(3) 在确定关系矩阵 $\boldsymbol{R}$ 时，阈值 ε_{ij} 是根据经验确定的，具有很大的主观性，阈值选取不当可能会对结果产生很大的影响。

2) 改进的一致性融合算法

对于结构监测系统，其主要监测内容有荷载(包括风、地震、温度作用)、几何监测和结构的静动力反应等，具体包括网格线形监测、构件静态应力和应变监测和振动监测等，主要使用的传感器有位移计、倾角仪、应变仪、测力计和加速度计等。对于大型复杂结构，一些主要承重和受力构件以及关键部位和构件是需要重点监测的对象，因此需要多种不同类型的传感器进行协同损伤监测。由于每类传感器进行损伤监测的方法和标准不同，其测量精度也可能不同，若直接采用一致性融合算法，会使置信距离不同，将导致最终分析得到的损伤程度结果不同。

置信距离和关系矩阵是一致性数据融合方法的关键。定义一种新的置信距离，如式(5-32)所示。

$$d_{ij}=d_{ji}=P_{\mathrm{r}}\left(|Z|\leqslant\frac{|x_i-x_j|}{\sqrt{\sigma_i^2+\sigma_j^2}}\right) \tag{5-32}$$

其中，x_i、x_j 为第 i、第 j 个传感器的测量值；σ_i^2、σ_j^2 为第 i、第 j 个传感器的测量方差；Z 为服从正态分布的随机变量。

比较两种方法的置信距离公式可知，对相同的测量值 x_i、x_j，后者定义的置信距离要小。这样定义的置信距离 d_{ij} 克服了当传感器测量精度不同时的不一致性。分析算法可知，d_{ij} 越接近阈值 ε_{ij}，此距离所涉及的传感器 i 与传感器 j 支持与否越模糊，只有 d_{ij} 远离阈值 ε_{ij}，才能清楚说明其支持程度。因此，定义 $d_{ij}=\varepsilon_{ij}$ 时，其支持程度为 50%。d_{ij} 越小，支持程度越高；d_{ij} 越大，支持程度越低。由 d_{ij} 给出传感器 i 与传感器 j 支持程度的量度，令

$$r_{ij}=1-d_{ij}$$

这样可以克服人为定义阈值带来的主观误差,将各传感器测定数据的相互支持程度模糊化,能够有效地减小由于各种扰动因素造成的融合结果的变化。计算出所有传感器的关系矩阵 $\boldsymbol{R}$,在关系矩阵 $\boldsymbol{R}$ 中 r_{ij} 仅表示两个传感器的测量值 x_i 和 x_j 之间的相互支持程度,并不能反映传感器 i 测量值 x_i 被系统中所有传感器测量值的综合支持程度。实际上,传感器 i 的测量值 x_i 的真实程度是由 $r_{i1},r_{i2},\cdots,r_{iN}$ 综合体现的。设 $\beta_i(0\leqslant\beta_i\leqslant1)$表示测量值被所有传感器测量值的综合支持程度,$\beta_i$ 越大,表明 x_i 被其他测量值支持的程度越高,即第 i 个传感器测量值的真实性也越高。根据信息分享原理:最优融合估计的信息量之和可等效分解成若干信息量之和,或者说一个信息可被若干子系统所分享,且各信息所具有的权系数应满足

$$\sum_{i=1}^{N}\beta_i=1 \tag{5-33}$$

根据概率源合并理论,存在一组非负数 $y_1,y_2,\cdots,y_n$,使得

$$\beta_i=y_1r_{i1}+y_2r_{i2}+\cdots+y_Nr_{iN},\quad i=1,2,\cdots,N$$

写成矩阵形式,则有

$$\boldsymbol{\beta}=\boldsymbol{R}\cdot\boldsymbol{Y} \tag{5-34}$$

其中,$\boldsymbol{\beta}=(\beta_1,\beta_2,\cdots,\beta_N)^{\mathrm{T}}$;$\boldsymbol{Y}=(y_1,y_2,\cdots,y_N)^{\mathrm{T}}$。

关系矩阵 $\boldsymbol{R}$ 是一个非负矩阵,由 Perronwe Frobenius 定理可知,$\boldsymbol{R}$ 存在最大模特征值 $\lambda(\lambda>0)$使该特征值对应的特征向量 $\boldsymbol{Y}$ 为正,并且满足

$$\boldsymbol{RY}=\lambda\boldsymbol{Y} \tag{5-35}$$

展开可得

$$\lambda_{y_k}=y_1r_{i1}+y_2r_{i2}+\cdots+y_Nr_{iN},\quad k=1,2,\cdots,N$$

令

$$a_k=\frac{\lambda y_k}{\sum_{i=1}^{N}\lambda y_i}=\frac{y_k}{\sum_{i=1}^{N}y_i} \tag{5-36}$$

其中,a_k 为第 k 个传感器的综合支持程度,最终的数据融合值 x 表示为

$$x=\sum_{i=1}^{N}a_kx_k$$

4. 算法验证及性能分析

采用数值算例的方式对改进的一致性融合算法进行验证。

方差 σ_i^2 定义为

$$\sigma_i^2=\frac{1}{N}\sum_{i=1}^{N}(x_i-s_i)^2 \tag{5-37}$$

用 10 个传感器测量某系统的特性参数,这些数据存在大量冗余测量值。x_i 和方差 σ_i^2 分别如下,用于验证提出的数据融合算法的有效性。

$$x_i=1.00,0.99,0.98,0.97,0.50,0.65,1.01,1.02,1.02,1.50$$

$$\sigma_i^2=0.05,0.07,0.10,0.20,0.30,0.25,0.10,0.10,0.20,0.30$$

利用改进算法得到最终的融合数据值为

$$x=\sum_{i=1}^{N}a_k x_k=0.97853$$

利用现有的一致性算法中定义的置信距离和关系矩阵计算此算例，为简化计算，假设所有传感器的阈值相同，分别取不同的值(0.25、0.50、0.75、1.00)进行对比计算，结果如表 5-1 所示。

表 5-1 一致性算法的数据融合结果

阈值	0.25	0.50	0.75	1.00
融合结果	1.000	0.979	0.952	0.965

从计算结果可以看出，不同的阈值对应的数据融合结果是不同的，这是由于人为给定的阈值大小直接影响参加融合的最大传感器组，从而影响各传感器测量数据的有效性，导致了融合结果的不同。在改进算法中，将各传感器之间的支持程度模糊化，有效避免了这种人为给定阈值带来的主观误差。

5.4 能量管理

5.4.1 能量管理的意义

传感器节点采用电池供电，工作环境通常比较恶劣，一次部署终生使用，所以更换电池就比较困难。如何节省电源、最大化网络生命周期和低功耗设计是传感器网络的关键技术之一。

传感器节点中消耗能量的模块有传感器模块、处理器模块和通信模块。随着集成电路工艺的进步，处理器和传感器模块的功耗都很低。无线通信模块可以处于发送、接收、空闲或睡眠状态，空闲状态就是侦听无线信道上的信息，但不发送或接收。睡眠状态就是无线通信模块处于不工作状态。

网络协议控制了传感器网络各节点之间的通信机制，决定无线通信模块的工作过程。传感器网络协议栈的核心部分是网络层协议和数据链路层协议。网络层主要是路由协议，选择采集信息和控制消息的传输路径，就是决定哪些节点形成转发路径，路径上的所有节点都要消耗一定的能量转发数据。数据链路层的关键是 MAC 协议，控制相邻节点之间无线信道的使用方式，决定无线收发模块的工作模式(发送、接收、空闲或睡眠)。因此，路由协议和 MAC 协议是影响传感器网络能量消耗的重要因素。

无线传感器网络的能量管理(Energy Management，EM)主要体现在传感器节点电源管理(Power Management，PM)和有效的节能通信协议设计上。在一个典型的传感器节点结构中，与电源单元发生关联的模块有很多，除了供电模块以外，其余模块都存在电源能量消耗。从传感器网络的协议体系结构来看，它的能量管理机制是一个覆盖从物理层到应用层的跨层协议设计问题。

传感器节点通常由 4 部分组成：传感器单元、无线传输单元、处理器单元和电源管理单元，如图 5-31 所示。其中，传感器单元能耗与应用特征相关，采样周期越短，采样精度越高，则传感器单元的能耗越大。可以通过在应用允许的范围内适当地延长采样周期，以及降低

采样精度的方法降低传感器单元的能耗。事实上,由于传感器单元的能耗要比处理器单元和无线传输单元低得多,几乎可以忽略,因此通常只讨论处理器单元和无线传输单元的能耗问题。

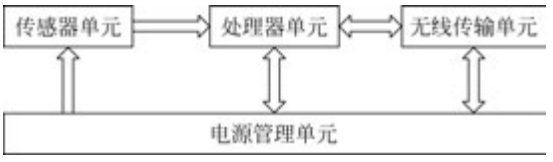


图 5-31 传感器节点的单元模块构成

(1) 处理器单元能耗。处理器单元包括微处理器和存储器,用于数据存储与预处理。节点的处理器单元能耗与节点的硬件设计、计算模式紧密相关。目前对能量管理的设计都是在应用低能耗器件的基础上,在操作系统中使用能量感知方式进一步减少能耗,延长节点的工作寿命。

(2) 无线传输单元能耗。无线传输单元用于节点间的数据通信,它是节点中能耗最大的部件。因此,无线传输单元节能是设计的重点。传感器网络的通信能耗与无线收发器以及各协议层紧密相关,它的管理体现在无线收发器的设计和网络协议设计的每个环节。

5.4.2 电源节能方法

目前采用的节能策略主要有休眠机制、数据融合等,它们被应用在计算单元和通信单元的各环节。

1. 休眠机制

休眠机制的主要思想是当节点周围没有感兴趣的事件发生时,计算与通信单元处于空闲状态,把这些组件关掉或调到更低能耗的状态,即休眠状态。该机制对于延长传感器节点的生存周期非常重要。但休眠状态与工作状态的转换需要消耗一定的能量,并且产生延时,所以状态转换策略对于休眠机制比较重要。如果状态转换策略不合适,不仅无法节能,反而会导致能耗的增加。

通过休眠实现节能的策略主要体现在以下几方面。

1) 硬件支持

目前很多处理器(如 StrongARM 和 MSP430 等)芯片,都支持对工作电压和工作频率的调节,为处理器单元的休眠提供了有力的支持。

从大量的实践中可知,传感器节点的绝大部分能量消耗在无线通信模块上,而且无线通信模块在空闲状态和接收状态的能耗接近。

表 5-2 无线收发器各状态的能耗

无线收发器状态	能耗/mW
发送	14.880
接收	12.500
空闲	12.360
睡眠	0.0160

现有的无线收发器也支持休眠,而且可以通过唤醒装置唤醒休眠中的节点,从而实现在全负载周期运行时的低能耗。无线收发器有 4 种操作模式:发送、接收、空闲和休眠。表 5-2 给出了一种无线收发器的能耗情况,除了休眠状态外,其他 3 种状态的能耗都很大,空闲状态的能耗接近于接收状态,所以如果传感器节点不再收发数据,最好把无线收发器关掉或进入休

眠状态以降低能耗。

无线收发器的能耗与其工作状态相关。在低发射功率的短距离无线通信中，数据收/发能耗基本相同。收发器电路中的混频器、频率合成器、压控振荡器、锁相环和能量放大器是主要的能耗部件。收发器启动时，由于锁相环的锁存时间较长，导致启动时间一般需要几百微秒，因此收发器的启动能耗是节能操作中必须考虑的因素。若采用无数据收发时关闭收发器的节能方法，则必须考虑收发器启动能耗与持续工作能耗之间的关系。

2) 采用休眠机制的网络协议

通常无线传感器网络的 MAC 协议都采用休眠机制，如 S-MAC 协议。在 S-MAC 协议中，数据发送时，如果节点既不是发送者，也不是接收者，就转入休眠状态，在醒来后有数据发送就竞争无线信道，无数据发送就侦听其是否为下一个数据接收者。S-MAC 协议通过建立周期性的侦听和休眠机制，减少侦听时间，从而实现节能。

3) 专门的节点功率管理机制

(1) 动态电源管理(Dynamic Power Management，DPM)。当节点周围没有感兴趣的事件发生时，部分模块处于空闲状态，应该把这些组件关闭或调到更低能耗的状态(即休眠状态)，从而节省能量。

这种事件驱动能量管理对于延长传感器节点的生存期十分必要。在动态电源管理中，由于状态转换需要消耗一定的能量，并且带有时延，所以状态转换策略非常重要。如果状态转换过程的策略不合适，不仅无法节能，反而会导致能耗的增加。需要指出的是，如果节点进入完全休眠的状态，则可能会引起事件的丢失，所以节点进入完全休眠状态的时机和时间长度必须合理控制。

(2) 动态电压调节(Dynamic Voltage Scaling，DVS)。对于大多数传感器节点，计算负荷的大小是随时间变化的，因而并不需要节点的微处理器在所有时刻都保持峰值性能。根据 CMOS 电路设计的理论，微处理器执行单条指令所消耗的能量 Eop 与工作电压 V 的二次方成正比，即 $Eop \propto V^2$。

动态电压调节技术就是利用了这一特点，动态改变微处理器的工作电压和频率，使其刚好满足当时的运行需求，从而在性能和功耗之间取得平衡。很多微处理器，如 StrongARM 和 Crusoe，都支持电压频率的动态调节。

动态电压调节要解决的核心问题是实现微处理器计算负荷与工作电压及频率之间的匹配。如果计算负载较高，而工作电压和频率较低，则计算时间将会延长，甚至会影响某些实时性任务的执行。但由于传感器网络的任务往往具有随机性，因而在动态电压调节过程中必须对计算负载进行预测。

2. 数据融合

相对于计算所消耗的能量，无线通信所消耗的能量要更多。例如，研究表明，传感器节点使用无线方式将 1 比特数据进行 100m 距离的传输，所消耗的能量可供执行 3000 条指令。通常传感器节点采集的原始数据的数据量非常大，同一区域内的节点所采集的信息具有很大的冗余性。通过本地计算和融合，原始数据可以在多跳数据传输过程中进行处理，仅发送有用信息，有效地减少通信量。

数据融合的节能效果主要体现在路由协议的实现上。路由过程的中间节点并不是简单地转发所收到的数据，由于同一区域内的节点发送的数据具有很大的冗余性，中间节点需要

对这些数据进行数据融合,将经过本地融合处理后的数据路由到 Sink 节点,只转发有用的信息。数据融合有效地降低了整个网络的数据流量。

LEACH 路由协议就具有这种功能,它是一种自组织的在节点之间随机分布能量负载的分层路由协议,工作原理如下:相邻的节点形成簇并选举簇首,簇内节点将数据发送给簇首,由簇首融合数据并把数据发给用户。其中,簇首完成簇内数据的融合工作,负责收集簇中各节点的信息,融合产生有用的信息,并对数据包进行压缩,然后才发送给用户,这样就可以大大地减少数据流量,从而实现节能的目的。

5.4.3 动态能量管理

无线传感器网络利用大量具有感知、处理和无线通信功能的智能微传感器节点在特定测量区域完成复杂任务。无线传感器节点通常采用电池供电,因而能量有限。为使布置后的传感器节点寿命最大化,电路、结构体系、算法和协议等方面必须根据能量有效性进行设计。一旦设计了系统,额外的能量节省可通过采用 DPM 技术获得。另外,节点需要具有适度的可扩展能量特性,若应用需要,用户可根据传感精度延长任务时限。空闲能量管理的基本思想是在不需要时关闭设备,在必要时将其唤醒。对多种睡眠状态执行正确的转换策略对有效的空闲能量管理十分重要。DVS 对于降低处理器能耗是一种十分有效的有功能量管理技术。基于微处理器的系统多数表现出时变计算负荷的特征。在活跃性降低阶段,简单地降低工作频率可使能耗线性降低,但不会影响任务的总体能耗。降低工作电压意味着更大的电路延迟,会降低最佳性能。由于最佳性能并不是时刻需要的,因而可实现显著的能量节省。

机动目标跟踪是 WSN 的典型应用,其中测量目标的状态信息对节点睡眠状态调整十分重要。由于无线传感器节点具有分布计算能力,为提高 WSN 能效性,可以利用目标先验信息进行动态能量优化并设计分布式优化算法提高优化性能。这里提出目标预测动态能量优化方法,采用粒子滤波(Particle Filter,PF)算法预测目标状态,研究节点动态唤醒机制和测量过程的自适应优化,运用分布式遗传模拟退火算法(Distributed Genetic Algorithm and Simulated Anneal,DGASA)优化 WSN 能耗,通过多节点分布式优化增强寻优能力,另外还采用了中转节点的路由方式。最后,通过仿真实验验证目标预测动态能量优化方法在 WSN 目标跟踪应用中的有效性。

1. 空闲能量管理

空闲模式的有效 DPM 需要多种具有能量差异的状态和各状态间转换的最优操作系统(Operation System,OS)策略。

1) 多种关闭状态

具有多种能量模式的设备有很多,如 StrongARM SA-1100 处理器有 3 种能量模式:运行、空闲和睡眠。每种模式对应于较低水平的耗能情况。运行模式是处理器的一般工作模式,所有能量供应激活,所有时钟均运行,且所有资源均工作。空闲模式允许软件暂停未使用的 CPU,而继续侦听中断服务请求。CPU 时钟停止,并保存所有处理器的相关指令。中断产生时,处理器返回运行模式,并继续从暂停点开始工作。睡眠状态节省的能量最多,提供的功能最少,大部分电路的能量供应被切断,睡眠状态机守候预排程序的唤醒事件,这与蓝牙无线装置中的 4 种不同的能耗模式(激活、保持、嗅探和暂停)类似。

多数能量感知的设备支持多种断电模式(提供不同级别的能耗和功能)。具有多个此类设备的嵌入式系统按照设备能量状态的各种组合,拥有一系列能量状态。实际上,称为高级设置和能量管理接口(Advanced Configuration and Power Management Interface,ACPI)的开放式接口规范受到 Intel、Microsoft 和 Toshiba 的共同支持,这些规范制定了 OS 与具有多种能量状态的设备连接并提供动态能量管理的标准。ACPI 支持系统资源的有限状态模型,并指定了软/硬件的控制接口。ACPI 控制整个系统的能耗和各设备的能量状态。遵守 ACPI 规范的系统具有 5 个全局状态:S_0(工作状态),以及 $S_1 \sim S_4$。$S_1 \sim S_4$ 对应于 4 种不同程度的睡眠状态。类似地,遵守 ACPI 规范的设备有 4 种状态:D_0(工作状态),以及 $D_1 \sim D_3$。睡眠状态根据能耗、进入睡眠需要的管理花费和唤醒时间来区分。

2) 传感器节点的构成

图 5-32 所示为基本传感器节点的构成。各节点由嵌入式传感器、A/D 转换器、带有存储器的处理器(此情形下为 StrongARM SA-11x0 处理器),以及射频中路电路组成。每部分通过基本设备驱动受 OS 控制。OS 的一个重要功能是能量管理(PM)。OS 根据事件统计情况决定设备的开启和关闭。传感器网络由分布在矩形区域 R 的 η 类传感器节点组成,区域尺寸为 $W \times L$,各节点可见度半径为 ρ。

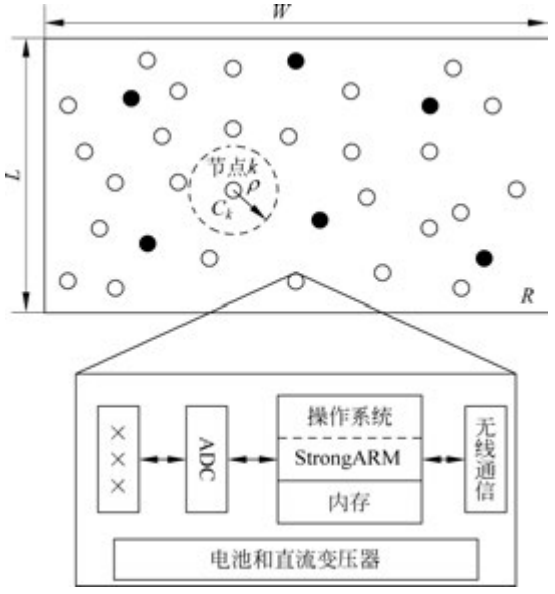


图 5-32 基本传感器节点的构成

对于传感器节点,表 5-3 列举了与 5 种不同的有用睡眠状态相关的各部分能量模式。各节点睡眠模式对应于越来越深的睡眠状态,因而其特征描述为渐增的延时和渐减的能耗。需要根据传感器节点的工作条件选择这些睡眠状态,例如,在激活状态中关闭存储器,或关闭其他任何部分是没有意义的。

(1) 状态 S_1 是节点的完全激活状态,节点可传感、处理、发送和接收数据。

(2) 状态 S_2 中,节点处于传感和接收模式,而处理器处于待命状态。

(3) 状态 S_2 与状态 S_1 类似,不同点在于处理器断电,当传感器或无线电接收到数据时会被唤醒。

(4) 状态 S_3 是仅传感的模式,其中除了传感前端外均关闭。

(5) 状态 S_4 表示设备的全关闭的状态。

表 5-3 传感器节点有用睡眠状态

状态	StrongARM	存储器	传感器,ADC	无线电
S_0	激活	激活	开	发送,接收
S_1	空闲	睡眠	开	接收
S_2	睡眠	睡眠	开	接收
S_3	睡眠	睡眠	开	关
S_4	睡眠	睡眠	关	关

能量管理是根据观测事件进行状态转换的策略,目的是使能量有效性最大。可见,能量唤醒传感器模型与 ACPI 标准的系统能量模型类似。睡眠状态通过消耗的能量、进入睡眠的管理花费,以及唤醒时间来区分。睡眠状态越深,则能耗越少,唤醒时间越长。

3) 睡眠状态转换策略

假设传感器节点在 t_0 时刻探测到一个事件,在 t_1 时刻结束处理,下一事件在 $t_2=t_1+t_i$ 时刻发生。在 t_1 时刻,节点决定从激活状态 S_0 转换到睡眠状态 S_k,如图 5-33 所示。各状态 S_k 的能耗为 P_k,而且转换到此状态和恢复时间分别为 $\tau_{d,k}$ 和 $\tau_{u,k}$。假设节点睡眠状态中,对于任意 $i>j$,$P_j>P_i$,$\tau_{d,i}>\tau_{d,j}$ 且 $\tau_{u,i}>\tau_{u,j}$。睡眠模式间的能耗可采用状态间线性变化的模型描述。例如,当节点从状态 S_0 转换到状态 S_k 时,无线电、存储器和处理器这些单个部件逐步断电,状态间能耗产生阶梯变化。线性变化在解析上比较容易求解,并能合理地近似此过程。

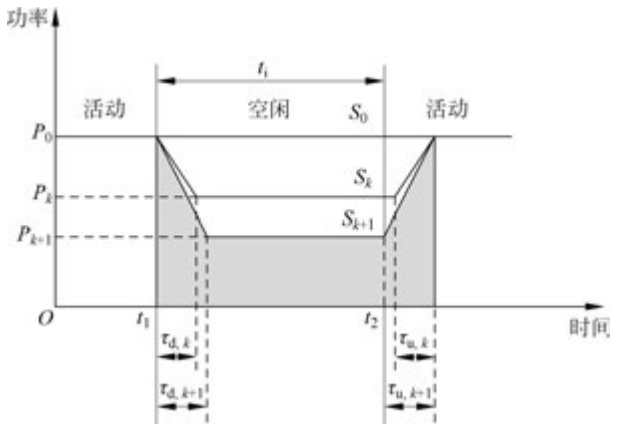


图 5-33 传感器节点睡眠状态转换策略示意

现在获得一组与状态$\{S_k\}$相应的睡眠时间阈值$\{T_{th,k}\}$。若空闲时间 $t_i<T_{th,k}$,由于存在转换能量管理花费,从状态 S_0 转换到睡眠状态 S_k 将造成网络能量损失。假设在转换阶段无须完成其他工作(如当处理器醒来时,转换时间包括 PLL 锁定、时钟稳定和处理器相关指令恢复的时间)。图 5-33 中,图线下方区域表示状态转换节省的能量,可计算为

$$E_{save,k}=(P_0-P_k)t_i-\left(\frac{P_0-P_k}{2}\right)\tau_{d,k}-\left(\frac{P_0+P_k}{2}\right)\tau_{u,k} \tag{5-38}$$

仅当 $E_{save,k}>0$ 时这种转换是合理的。于是,可得到能量增益阈值

$$T_{\mathrm{th},k}=\frac{1}{2}\left[\tau_{\mathrm{d},k}+\left(\frac{P_0+P_k}{P_0-P_k}\right)\tau_{\mathrm{u},k}\right] \tag{5-39}$$

这意味着转换的延时花费越大，能量增益阈值越高，而且 P_0 与 P_k 间的区别越大，阈值越小。

表 5-4 列出了图 5-33 所描述传感器节点的能耗，说明了现有组件在不同能量模式下相应的能量增益阈值。由此可见，阈值处于毫秒级。OS 的关闭策略以事件执行间隔统计和能量增益阈值为基础，可视为一个优化问题。若事件采用泊松过程模型，t_i 时刻至少发生一个事件的概率为

$$P_{\mathrm{E}}(t)=1-\mathrm{e}^{\lambda t_i} \tag{5-40}$$

此时，采用简单算法更新每单位时间的平均事件数 λ，计算阈值内的事件发生概率，并根据有效的最小概率阈值选择最深的睡眠状态。

表 5-4 睡眠状态能量、延时和阈值

状态	P_k/mW	τ_k/ms	$T_{\mathrm{th},k}$/ms	状态	P_k/mW	τ_k/ms	$T_{\mathrm{th},k}$/ms
S_0	1040	—	—	S_3	200	20	25
S_1	400	5	5	S_4	10	50	50
S_2	270	15	15				

2. 有功能量管理

对于具有能量约束的传感器节点，OS 能对有功能耗进行管理。将工作频率和电压降低到正适合传感应用的等级，性能不会有显著下降，但可以降低能耗。

DVS 是一种对降低 CPU 能量十分有效的技术。一些传感器系统具有时变的计算负荷。在活性较低阶段，简单地降低工作频率会造成能耗的线性降低，但不会影响每个任务的总体能耗，如图 5-34(a)所示(阴影区域表示能量)。降低工作频率意味着工作电压同样会降低。因为转换能耗与频率线性成比例，并与供电电压的二次方成比例，可获得二次能量降低，如图 5-34(b)所示。由于最佳性能不是时刻需要的，因此能显著降低系统能耗，这意味着处理器的工作电压和频率可根据瞬时处理需要进行动态调整。

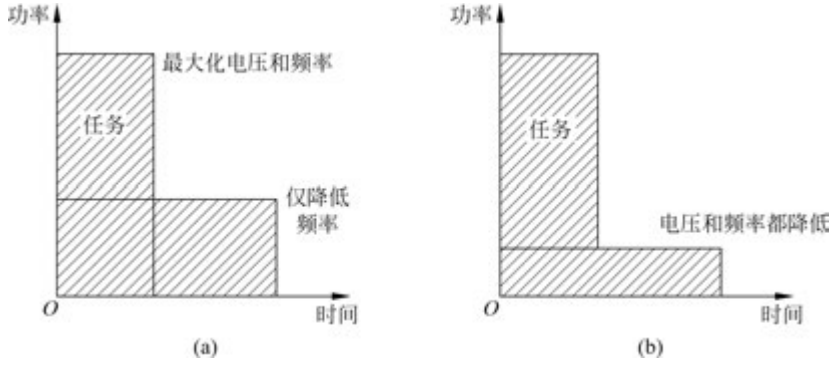


图 5-34 动态电压和频率缩放

3. 系统实现

无线通信模块在一块尺寸类似的电路板上由 2.4GHz 双功率无线电构成，范围为 10m 和 100m。16 位总线接口连接器使无线通信模块能连接在处理器电路板上。另外，连接器

支持不同传感器电路板(如振动传感器)的接入。处理器电路板具有一个 RS-232 和一个 USB 连接器,用于远程调试和与 PC 相连。传感器节点包括固定振动传感器(扬声器、运算放大器与 A/D 电路),此传感器采用同步串行端口(Synchronous Serial Port,SSP)与 StrongARM 处理器通信。运算放大器增益是可编程的,受处理器控制。传感器电路还结合了封装探测机制,当信号能量超过编程确定的阈值时,可绕过 A/D 电路唤醒处理器。这样可显著降低传感模式的能耗,并支持事件驱动算法。

1) DVS 电路

图 5-35 描述了一个基本的核心能量供应调节方式。MAX1717 降压控制器用于动态调节核心供电电压,采用 5 位数模转换器(DAC),输入范围是 0.925～2V。转换器工作原理:可变工作循环脉冲宽度调制(Pulse Width Modulate,PWM)信号交替开启功率管 VF_1 和 VF_2。功率管在工作周期 D 输出一个方波。LC 低通滤波器使等于 $DV_{battery}$ 的直流输出通过,而使交流分量衰减到可接受的范围内。工作周期 D 是可控的,采用 DAC 引脚(D_0～D_4),可产生 30 个电压级别。双线远程传感方案补偿了地线与输出电压线上的电压降。StrongARM 根据是否需要 DVS 设置其使能引脚,将其作为电压调节器。调节器的反馈信号告知处理器输出核心电压是否稳定,这对于能量缩放中的无误差工作是必需的。

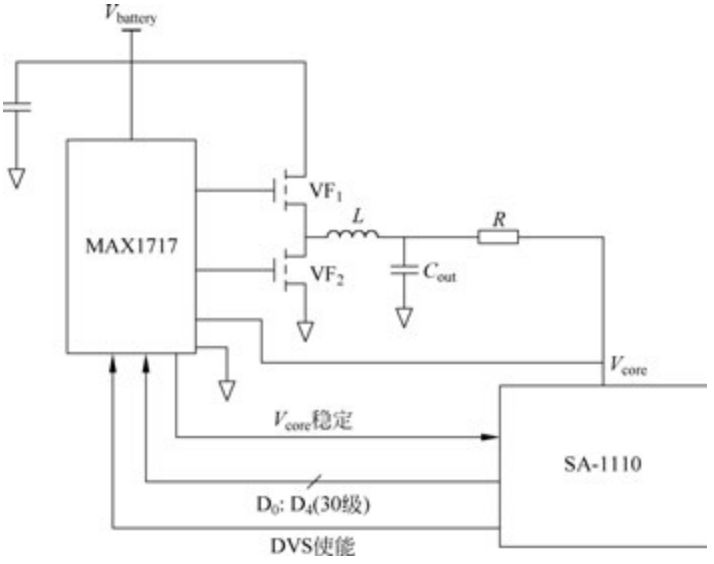


图 5-35 DVS 电路

处理器时钟频率调节包括了 SA-1110 的核心时钟设置(Core Clock Configuration,CCF)状态更新。核心时钟通过标准晶振时钟的倍频获得,采用基于 CCF 寄存器设置的锁相环(PLL)。核心时钟(Core Clock,CCLK)可用快速 CCLK 或存储器时钟(Memory Clock,MCLK)驱动,其中 MCLK 运行频率为 CCLK 的一半。核心时钟除了在存储丢失时等待填充完成外,一般采用 CCLK 方式。通过适当设置控制寄存器可取消核心时钟在 CCLK 和 MCLK 间转换的能力。

电压和频率更新中的操作序列取决于操作是否提高处理器时钟频率,如图 5-36 所示。当时钟频率提高时,将核心供电电压 V_{core} 提高到特定频率所需的最小值是最有必要的。最优电压-频率对存储在查询表中。一旦核心电压稳定,可开始更新频率。第一步包括重新校

准存储计时器，可通过在 MSC 控制寄存器中设置适当值实现。在 CCLK 频率提高前设置时钟，使之不能在 CCLK 和 MCLK 间转换，防止核心时钟的意外转换。通过设置 CCF 寄存器，完成 CCLK 频率的改变。完成了这些步骤后，恢复核心时钟在 CCLK 和 MCLK 间转换的能力。

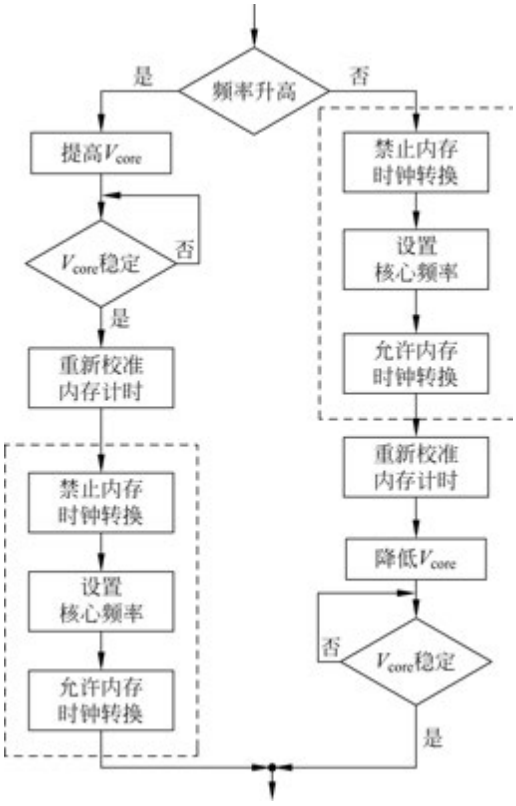


图 5-36 DVS 工作过程

降低频率时，操作序列略有不同。首先，更新核心时钟频率（按照前面提到的 3 个基本步骤）。在降低核心电压前需要重新校准存储计时器，因为一旦核心时钟频率降低，若不调整存储计时器，存储器读-写将造成误差（如在读电压-频率查询表时）。接下来，核心电压降低，当其稳定时开始进行一般操作。为了确保正确操作，所完成的所有电压频率更新采用原子方式。例如，当频率更新而存储器未重新校准时，若发生一个中断，可能产生执行错误。

2）空闲能量管理硬件实现

经过特别设计的传感器节点拥有一系列与上述类似的睡眠状态。另外，节点硬件支持事件驱动算法。DVS 的硬件结构如图 5-37 所示。StrongARM 的 GPIO 引脚与外围设备相连，用于生成和接收各种信号。SA-1110 包含 28 个 GPIO 引脚，各引脚设置为输入或输出功能。另外，GPIO 引脚经过特别设置可用于探测上升沿或下降沿。4 个 GPIO 引脚专门用于实现系统能量供应控制。当不需要进行测量时可选择关闭所有模拟能量供应，或者仅关闭低通滤波器（Low Pass Filter，LPF）的能量供应，而封装能量传感电路用于向处理器发出触发信号。此时，处理器激活 LPF，并开始采用 SSP 从 ADC 读入数据。信号探测阈值同样可用其他 GPIO 引脚编程，对无线通信模块可采用类似的能量供应控制。不需要无线通信

时,处理器可将其关闭。

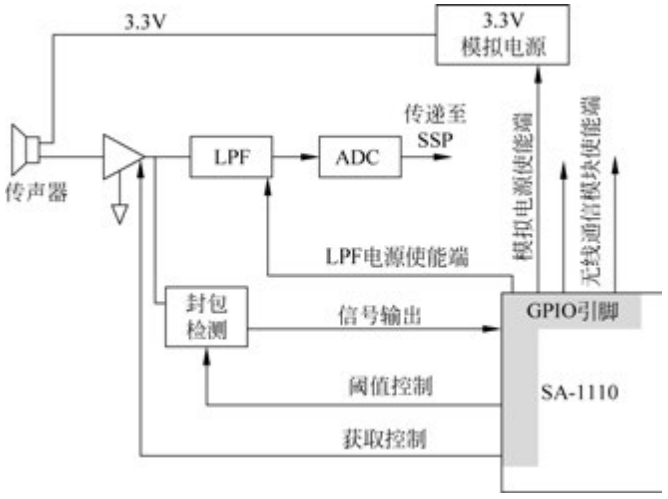


图 5-37 DVS的硬件结构

3) 处理器能量模式

SA-1110包含能量管理逻辑电路,控制3种不同模式的转换:运行、空闲和睡眠。各模式对应于较低的能耗水平。

运行模式是SA-1110的一般工作模式。所有单片能量供应开启,所有时钟开启,而且所有单片资源可用。处理器通常经过上电或重置在运行模式下启动。

空闲模式允许未使用时停止CPU,同时继续监视中断请求。CPU时钟停止时,因为SA-1110是全静态设计,所有状态信息均被保存。当重新开始一般操作时,CPU精确地从停止处开始执行。在空闲模式下,所有单片资源(实时时钟、OS计时器、中断控制器、GPIO、能量管理器、DMA与LCD控制器等)都开启。PLL也保持锁定,于是处理器可快速地进入和退出空闲模式。

睡眠模式为处理器节省最多能量,同时提供最少的功能。SA-1110从运行/空闲模式转换到睡眠模式时顺序关闭单片资源,对处理器采用内部中断,并取消能量使能(PWR_EN)引脚的作用,从而给外部系统指示,说明能量供应可关闭。32.768kHz晶振停止,睡眠状态机守候预排程序唤醒事件的发生。进入睡眠模式有两种方式:软件控制或能量供应错误。设置能量管理控制寄存器(Power Manager Control Register,PMCR)中的强制睡眠位,可进入睡眠模式。睡眠时通过软件设置标志位,然后采用硬件清空此标志位。于是,当处理器醒来时,此标志位已清空。整体睡眠的关闭序列耗时约90ms。

表5-5列出了各种模式下的能耗,包括SA-1110处理器的两种不同频率和相应电压规格。空闲模式能量约降低了75%,而睡眠模式几乎节省了所有能量。

表 5-5 SA-1110处理器能耗

频率/MHz	供电电压/V	能耗/mW		睡眠/μA
		正常	空闲	
133	1.55	<240	<75	<50
206	1.75	<400	<100	<50

4. 动态能量管理实验

将传感器节点在完全激活状态(所有模块开启)的能耗作为 SA-1110 工作频率的函数,采用 DVS 和仅使用频率调节(采用固定核心电压 1.65V)的能耗。系统能量供应为 4.0V。在激活模式下,能量管理主要由 DVS 实现。当在最高工作电压和工作频率下运行时,系统的能耗约为 1W。采用 DVS 有功能量管理的最大系统能量节省约为 53%。实际节省依赖于负荷需要。

采用 DVS 时,当处理速率变化程度最小时,由于能量负荷模型具有凸性,因而存在最低能耗。虽然平均负荷可能是固定的,但电池寿命从 DVS 获得的改善会随着负荷波动的增加而降低。

表 5-6 列出了各种传感器工作模式的能耗。无线电模块的能量需求受到处理器控制(在 3.3V 下约为 70mA)。DVS 可将系统能耗降低 53%。关闭各部分(模拟能量供应、无线电模块,以及处理器)可得到 44%的额外能量节省,也就是说,空闲能量管理的系统级能量节省约为 97%。

表 5-6 各种传感器工作模式的能耗

状 态	系统模式	成分模式			功率/mW
		处理器	无线电	模拟	
激活状态	激活	最高频率	开	开	975.6
	弱激活	最低频率	开	开	457.2
	空闲	空闲	开	开	443.0
睡眠状态	接收	空闲	开	关	403.0
	传感	空闲	关	关	103.0
	睡眠	睡眠	关	关	28.0

实际能量节省明显依赖于处理速率和事件统计。为了评价激活模式下获得的能量节省,需要估计系统负荷变化的程度。若平均负荷需求为 50%,变化缓慢,则预计能量节省约为 30%。

另外,空闲模式能量节省十分显著。假设操作的工作周期为 1%,则估计能量节省约为 96%。这意味着传感器节点电池寿命的提高因子会超过 27(也就是说,若无能量管理时节点能维持一天,则现在几乎能维持一个月)。工作周期为 10%时,电池寿命提高因子约为 10。这里的重点在于系统是能量可扩展的,即系统可根据计算负荷和传感需要调整能耗。

5.5 容错技术

在超大规模集成电路、分布式系统中,容错技术已经得到了深入研究。无线传感器网络的出现,对容错技术提出了新的挑战。无线传感器网络不仅自身容易发生故障,而且还受到外界环境的影响,因此更需要有效的容错设计技术满足其可靠性要求。

5.5.1 容错技术的基本描述

容错技术经过长期的发展,已经形成了一个专门的领域。这个领域中有一些基本的概

念,如失效、故障、差错等。另外,由于无线传感器网络自身的特点,导致无线传感器网络容错设计与传统容错设计也有所不同。

早期计算机、通信、存储设备的可靠性曾被认为是计算机系统的关键问题之一。自20世纪70年代以来,容错在超大规模集成电路、分布式系统、数据库和互联网等领域都得到了充分的重视。容错的内容包括部件可信性、容错体系结构、软件可信性、可信性验证与评估等许多方面。容错技术是由部件级向系统级方向发展的,1964年IBM推出大型主机System 360时,存储器的差错校正码技术立了汗马功劳;从1975年开始,商业化的容错机制推向市场;到20世纪90年代,软件容错的问题被提出,进而发展到网络容错。

容错领域有几个基本概念:失效(Failure)、故障(Fault)、差错(Error)。失效是指某个设备中止了它完成所要求功能的能力。故障是指一个设备、元件或组件的一种物理状态,在此状态下它们不能按照所要求的方式工作。差错是指一个不正确的步骤、过程或结果。故障只有在某些条件下才能在其输出端产生差错,这些差错由于在系统内部,不易观测到。只有这种差错积累到一定程度或在某种系统环境下,才能使系统失效。所以,失效是面向用户的,而故障和差错是面向制造和维修的。

无线传感器网络容错是指网络中某个节点或节点的某些部件发生故障时,网络仍然能够完成指定的任务。容错的要求在不同的应用中有所不同。例如,一个办公室有6个人,分别标记为A、B、C、D、E、F。在办公室门口布设一个无线传感器网络,要求能识别出这6个人。传感器节点能感知每个人的高度和声音两种传感器,高度由一串光感应器测得,声音由每个人经过门口时对着麦克风发声从而实现识别。图5-38(a)表示了这6个人的身高和声音特征;图5-38(b)将这些特征映射到一个二维图上。由于所有人的特征被映射在图5-38(b)的不同方格中,所以无故障时系统能够区分出任意两个人。从图5-38(b)可以看出,有一个高度或声音传感器失效时,系统仍然能够区分出大部分的人,但是也有一些不能识别的情况,如B和E的高度相同,声音是区分他们的唯一方式,所以在区分B和E时,网络不能容忍声音传感器V_s发生故障。如果办公室只有5个人(没有B或E),那么系统能够容忍任何一个感应器故障。

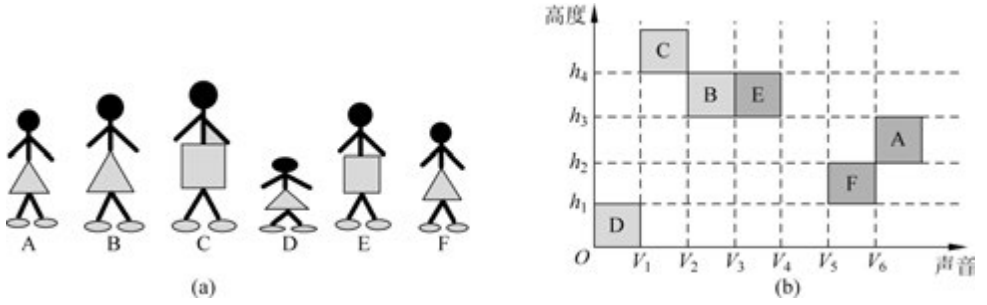


图5-38 多模型传感器网络容错实例

可见,容错能力通常建立在信息的冗余上。

无线传感器网络的出现给容错设计技术带来了新的挑战,因为无线传感器网络需要考虑以下情况。

(1) 技术和实现因素。与集成电路封装得很好不同,传感器节点通常需要直接暴露在环境中,更容易受到物理、化学、生物等外力的破坏,所以本身可靠性要差很多。而且,成百

上千的传感器节点组成一个分布式网络，在受到成本和能量限制的同时，需要完成一系列的任务，如感知、执行、通信、信号处理、数值计算等，这本身就是一个挑战。

(2) 无线传感器网络的应用模式。无线传感器网络通常运行在无人干预的模式，它们需要具有更强的容错能力，确保某些部件或节点发生故障时，网络仍然能够完成预设的任务。

(3) 无线传感器网络是一个新兴的研究和工程领域，处理特定问题的最优方法还不明确。无线传感器网络的技术和预期应用还在快速地变化着，所以在特定传感器网络中容错处理还难以预见。

5.5.2 故障模型

无线传感器网络容错设计需要考虑 3 方面：故障模型、故障检测与诊断、修复机制。

从整体上考虑，无线传感器网络中的故障可以分为 3 个层面，即部件级、节点级和网络级，表 5-7 给出了部件级和节点级故障的详细描述。由于网络、节点、部件间的包含关系，所以高层故障本质也是由低层故障造成的。

表 5-7 故障模型的层次

故障级别	故障表征	故障检测	修复机制
部件	故障节点能够正常通信，但是测量数据是错误的	检测出错误的测量数据	舍弃或校正出错的测量数据
节点	故障节点不能与其他节点进行通信	通过询问或重新路由等方法检测故障节点	通过移动冗余节点弥补形成的连接和覆盖问题

传感器节点由计算、通信、存储、能量供应、传感等功能部件构成，每个部件都可能发生故障。由于以往的容错研究主要关注处理器、SRAM 和 DRAM、非易失性的存储器和磁盘以及通信等部件，且它们的可靠性已经很高了，而传感部件受到低成本、能量有限、外界环境的限制，其可靠性是传感器网络应用的重要挑战。传感部件由感应器和模数转换器(ADC)组成，由于感应器暴露在外界环境中，所以很容易发生故障。

发生故障的传感器可能完全不工作了，或者它仍能给出测量值，只是这个测量值是错误的。下面来描述这种错误的测量值，设某个节点所在地的真实值为 $\gamma(t)$，记测量误差服从正态分布 $N(0,\sigma^2)$。传感器发生故障时，测量值可以形式化为 $f(t)=\beta_0(t)+\beta_1\gamma(t)+\varepsilon(t)$，其中 $\beta_0(t)$是偏移值，β_1 是缩放倍数，$\varepsilon(t)$是测量噪声，由此可以得到下面几种故障模型。

(1) 固定故障。固定故障是指感应器的读数一直为某个固定的值。这个值通常不在正常的感知范围内。发生固定故障的感应器不能提供任何感知环境的信息。固定故障可以形式化为 $f(t)=\beta_0(t)$。

(2) 偏移故障。偏移故障是指在真实值的基础上附加一个常量，可以形式化为 $f(t)=\beta_0(t)+\gamma(t)+\varepsilon(t)$。

(3) 倍数故障。倍数故障是指真实值被放大或缩小某个倍数，可以形式化为 $f(t)=\beta_1\gamma(t)+\varepsilon(t)$。如果没有对测量值的先验知识，仅从结果不能分辨出偏移故障和倍数故障。

(4) 方差下降故障。这类故障通常是由于使用时间过长，感应器老化后变得越来越不

精确而产生的。设测量方差为 σ_m^2,故障方差为 σ_f^2,当 $\sigma_f^2 > \sigma_m^2$ 时,则误差演变为故障,包含故障的 $f(t)=\gamma(t)+\varepsilon(t)$。

由于能量耗尽或通信部件发生故障,节点不能与其邻居通信,这时节点被判断为出现故障,即使节点的其他部件仍然正常。网络级的故障是指在某个区域内的节点都出现了故障,造成部分网络停止工作。

5.5.3 故障检测与诊断

故障检测的目标是检测网络中的异常行为。故障检测分为部件故障检测和节点故障检测。部件故障检测主要研究传感部件故障;节点故障检测主要是定位发生故障的节点。

1. 部件故障检测

传感部件是很容易发生故障的部件,并且传感部件发生故障后,节点也随即发生故障,所以本节重点介绍传感部件的故障检测。一种简单的传感器故障模型是传感器要么起作用,要么不能正常工作。对这种故障模型,故障检测过程非常直接,通常只需要观察传感器的输出。另一种故障模型是针对有连续的或多级数字输出的传感器。针对这类传感器的故障模型就更加复杂了。下面将分为基于空间相关性的故障检测和基于贝叶斯信任网络的故障检测介绍这类故障的检测。对于检测的效果,通常用两个指标来衡量,即识别率和误报率。识别率是指发生故障的部件被检测为有故障的概率;误报率是指把正常的部件判断为发生了故障的概率。

1) 基于空间相关性的故障检测

无线传感器网络相邻节点的同类传感器所测量的值通常很相近,称这种特性为空间相关性。一个节点通过周围邻居的同类传感器检测自己的传感器是否发生了故障。根据故障检测时是否需要节点地理位置信息,可以分为以下两类。

(1) 需要地理位置信息。

某节点的传感器测量到的结果与周围节点测量到的结果都不相同时,这个节点的传感部件很可能发生了故障。如图 5-39 所示,节点 1～节点 8 都感应到事件的发生,而节点 n 没有感应到事件的发生,则认为节点 n 的传感部件发生了故障。如果节点 n 及节点 1～节点 7 都感应到事件,则可以判断节点 8 的传感部件发生了故障。

在地理位置信息已知时,利用 3 个可信节点就可以实现感应器故障检测。如图 5-40 所示,节点 $n_j(j=1,2,3)$为无故障的可信节点,$O_j(j=1,2,3)$为节点所处位置,以 $O_j(j=1,2,3)$为圆心的圆表示节点 $n_j(j=1,2,3)$的感知范围。为每个圆作两条切线,这两条切线分别垂直于这个圆心与另外两个圆心连线,记两条切线的交点为 $X_j(j=1,2,3)$。设节点 n 位于由 $X_j(j=1,2,3)$组成的三角区域内,而节点 n 对事件的判断与 n_1、n_2、n_3 不同,即如果节点 $n_j(j=1,2,3)$都感应到事件发生而节点 n 没有,或者节点 $n_j(j=1,2,3)$没有感应到事件发生而节点 n 感应到了,则认为节点 n 发生了故障。

(2) 不需要地理位置信息。

无线传感器网络中的正常节点都能侦听到邻居发送的消息。节点可以依据侦听到的邻居数据判断自己的测量值是否正确,判断策略可以分为多数投票策略、均值策略和中值策略。

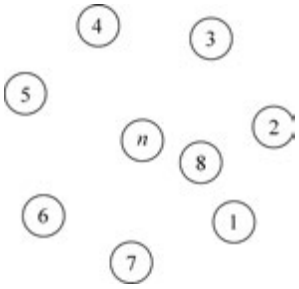


图 5-39 除了节点 n 或节点 8，其他节点都感应到事件发生

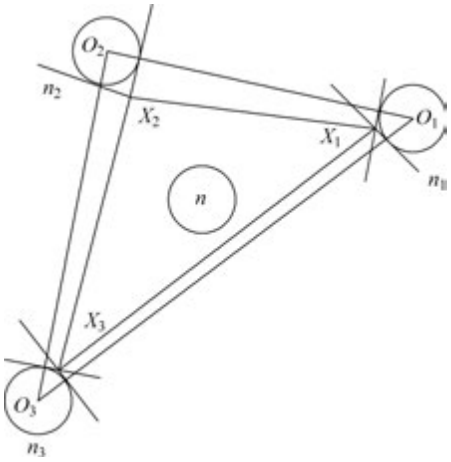


图 5-40 节点 n 在 3 个可信节点的三角区域内

多数投票策略是通过与邻居节点测量值进行比较，得到与自己的测量值相同或差距在允许范围内的邻居测量值个数，如果个数超过邻居数目的一半，则判定自己的测量值为正确的，否则就是错误的。多数投票策略的故障识别率比较高，但是在空间相关性不是很大的应用中，这种方法会存在较大的误报率。如果邻居节点分布较分散或离自己较远，即使邻居测量值与自己的测量值都正确，但是它们之间的差距也可能超出允许的范围。

均值策略是计算邻居测量值的平均值，然后比较这个均值和自己的测量值，如果它们的差距在允许的范围内，则认为自己的测量值为正确的。由于邻居测量值可能存在错误，这些错误值偏离正确值较大时会使得均值可信度降低，这样就可能导致误判。

中值策略很大程度上避免了错误的邻居测量值对测量精度的影响。中值策略比较邻居测量值的中值与自己的测量值，这样即使有很多邻居测量值错误，仍然能正确地判断出自己的测量值是否正确。

设节点 n_i 有 N 个邻居，邻居测量值为 $x_j(j=1,2,\cdots,N)$。判断 n_i 的测量值 x_i 是否正确的 3 种策略的详细步骤如表 5-8 所示。

表 5-8 3 种故障检测策略

多数投票策略
a. 得到节点 n_i 的邻居测量值 $x_j(j=1,2,\cdots,N)$
b. 比较 x_i 与 $x_j(j=1,2,\cdots,N)$，得到与 x_i 相同或在允许差距范围内的 x_j 的个数 k_i
c. 如果 $k_i \geqslant 0.5N$，则 x_i 正确；否则 x_i 错误
均值策略
a. 得到节点 n_i 的邻居测量值 $x_j(j=1,2,\cdots,N)$
b. 计算邻居读数的均值 $\tilde{x}_i = \dfrac{\sum_{j=1}^{N} x_j}{N}$
c. 自身测量值与上述均值比较，$f(x_i,\tilde{x}_i)=\begin{cases}1, & \left\lvert\dfrac{x_i-\tilde{x}_i}{\tilde{x}_i}\right\rvert>\xi \\ 0, & \text{其他}\end{cases}$
d. 若 $f(x_i,\tilde{x}_i)$ 值为 1，则认为这次测量值有误

续表

中值策略
a. 得到节点 n_i 的邻居测量值 $x_j(j=1,2,\cdots,N)$
b. 计算邻居测量值的中值 $\tilde{x}_i=\mathrm{MED}\left\{x_j \middle\| \begin{matrix} N \\ j=1 \end{matrix}\right\}$，MED 为求中值的运算
c. 自身测量值与上述中值比较，$f(x_i,\tilde{x}_i)=\begin{cases}1, & \left\|\dfrac{x_i-\tilde{x}_i}{\tilde{x}_i}\right\|>\xi \\ 0, & \text{其他}\end{cases}$
d. 若 $f(x_i,\tilde{x}_i)$ 值为 1，则认为这次测量值有误

判断策略比较如表 5-9 所示。

表 5-9 判断策略比较

策 略	识别率	误报率	时间复杂度
多数投票	较高	低	$O(n)$
均值	较高	较低	$O(n)$
中值	高	低	$O(n\mathrm{lb}n)$

使用上述策略，会出现错误的判断。如图 5-41 所示，A_0 和 B_0 同时有两个正常邻居和两个故障邻居，以多数投票策略为例，它们被同时判断为正常，这样必然有一个是判断错误；C_0 的邻居中有 3 个与自己不同，所以被判断为出现故障，而事实上 C_0 是一个正常的节点。

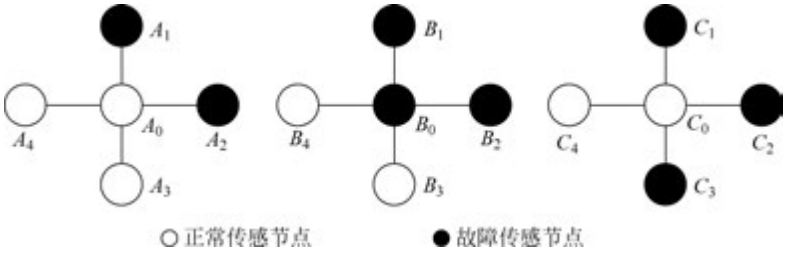


图 5-41 决策判断实例

考虑到这种情况，在对检测精度要求很高的应用中，可以通过加权的方法提高识别率，同时降低误报率。首先给每个传感器一个初值作为它的可信度；其次，每当节点的测量值被判断为错误时，就将它的可信度减 1。在以后的判断中，将可信度作为传感器测量值的权重，这将提高检测的精度。

2) 基于贝叶斯信任网络的故障检测

不同部件间的潜在关系可以用来检测传感部件是否发生故障。这些部件可以属于同一个节点，也可以属于不同节点。这类故障检测方法主要是利用了部件间的信任关系，可以通过贝叶斯信任网络(Bayesian Belief Networks，BBN)进行表述。下面先介绍贝叶斯信任网络的基本含义。

贝叶斯信任网络包含一个有向图和与之对应的概率表集合。有向图中的顶点表示变量，边表示变量之间的影响关系。贝叶斯信任网络的关键特征是能够模型化并推理出不确定因素。模型化节点间的可靠关系是通过节点概率表实现的。

应用贝叶斯信任网络分为构造、学习、推理 3 个阶段。在构造阶段需要得到所有变量的联合概率分布。例如，有 5 个变量时，其联合概率分布为 $P(A,B,C,D,E)=P(A)P(B\mid$

$A)P(C \mid B,A)P(D|C,B,A)P(E \mid D,C,B,A)$。在实际中不是所有变量都有依赖关系。假设 B 与 A、C 都独立，那么联合概率分布可以写成 $P(A,B,C,D,E)=P(A)P(B)P(C|A)P(D|B)P(E|D,B)$。学习阶段的任务是通过训练得到各变量间的条件概率。如果网络结构未知，学习阶段还需要推断出潜在的结构。推断过程是由一些已知属性值推断未知变量的概率分布。

下面在大鸭岛实验情景下应用贝叶斯信任网络实现传感部件的故障检测。环境监测中有 5 个属性：温度(T)、相对湿度(H)、气压(P)、光照强度(L)、节点电压(V)。它们的关系如图 5-42 所示，气压和相对湿度受温度影响，而电压影响了所有其他属性。

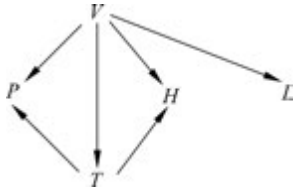


图 5-42 属性间的条件依赖条件

一个节点所有属性的联合概率分布为 $P(V,T,H,P,L)=P(V)P(L|V)P(T|V)P(H \mid V,T)P(P \mid V,T)$。设所有属性被划分成两个不重叠的子区间 r_1 和 r_2，通过大量数据训练得到以下概率。

$P(S)$：传感器读数落入某个区间的概率。

$P(N|S)$：给定当前读数落入某区间的概率的情况下，邻居读数落入某区间的概率。在大鸭岛实验情景下通过训练所得概率如表 5-10 和表 5-11 所示。

表 5-10 温度、电压的概率分布

S	$P(S_T)$	$P(S_V)$
r_1	0.9	0.6
r_2	0.1	0.4

表 5-11 气压、相对湿度的概率分布

S_T	S_V	$P(S_P)$		$P(S_H)$	
		r_1	r_2	r_1	r_2
r_1	r_1	0.5	0.5	0.6	0.4
r_1	r_2	0.9	0.1	0.3	0.7
r_2	r_1	0.2	0.8	0.2	0.8
r_2	r_2	0.7	0.3	0.6	0.4

可以根据这些条件，推测未知的情况。例如，已知节点的相对湿度感应值 $H \in r_2$，电压 $V \in r_1$，由此来推测是否 $T \in r_1$，如表 5-12 所示。

表 5-12 计算推理

T	$P(H|V,T)\ P(T)\ P(V)$	$P(T|V=r_1,H=r_2)$
r_1	$0.4\times0.9\times0.6=0.216$	0.4
r_2	$0.2\times0.9\times0.6=0.108$	0.8

2. 节点故障检测

在传感部件的故障检测中，假设节点发生故障时仍能与外界正常通信。实际部署中，由于电池耗尽或通信故障导致节点不能与网络正常连通，这类故障通常需要其他节点来检测。根据检测过程是否集中进行，节点故障检测可分为集中式和分布式两种。

1) 集中式故障检测

集中式故障检测通过在 Sink 节点放置检测程序，实时监测网络状态。Sink 节点需要

收集的内容如表 5-13 所示。

表 5-13 Sink 节点需要收集的信息

名　　称	描　　述
邻居列表	由邻居 ID 组成的一个列表
链路质量	用 0(100%丢失)至 100(100%传输)间的一个数表示
字节数	节点传输和收到的字节数
下一跳(路由表)	路由的下一跳节点
路径丢失(路由表)	从节点到 Sink 节点的链路质量的一种衡量

网络初始化完成后,Sink 节点保存节点的路由表、邻居列表、链路质量等参数值。在网络运行过程中,Sink 节点可以向其他节点发送收集信息的指令,然后其他节点向 Sink 节点上报消息;或者由节点周期性地上报信息。这样,Sink 节点能够利用这些信息判断出发生了什么事件。事件列表及判断依据如表 5-14 所示。

表 5-14 事件列表及判断依据

事件名	描　　述	用来识别事件的信息
节点丢失	节点没有出现在任何节点的邻居列表中	所有邻居表
孤立节点	节点没有任何邻居	此节点的邻居表
路由改变	比较当前路由表与上次路由表的变化	此节点的路由表信息
邻居表改变	比较当前邻居表与上次邻居表的变化	此节点的邻居表
链路质量改变	此节点与邻居的链路质量低于统计定义的门槛值。把当前的和以前的链路质量写入日志	此节点的邻居表

集中式检测要求 Sink 节点的处理能力较高。如果网络规模很大,这些信息的传播也会消耗大量的网络资源。

2) 分布式故障检测

分布式故障检测不是由 Sink 节点统一检测,而是由每个节点分别自行检测。隐藏终端(Hidden Terminals)、拥塞、链路不对称是几种常见的节点通信故障。

节点发现数据率下降后,询问路由表中的子节点是否也有同样的现象,如果答案是肯定的,那么继续询问下去;当遇到否定的回答时,这个父节点就触发诊断程序,把诊断到的原因及可能的措施发往基站以写入日志。

图 5-43 给出了一种分布式故障诊断算法的基本步骤。整个算法分为多个阶段,每个阶段检测一种故障。每个阶段由两部分组成,一部分判断某一故障是否发生;另一部分是处理故障的措施。由于不同故障可能导致相同的现象,所以检测算法各阶段的顺序要根据实际情况安排。如图 5-43 所示,隐藏终端和网络拥塞都会引起网络层传输队列缓冲区溢出和无线信道的高冲突(High Contention of the Wireless Channel)。假如检测算法先判断是网络拥塞,那么就可能得出错误的结论。而利用 MAC 层提供的信息先判断是否是隐藏终端,如果不是则再判断是否为网络拥塞,这样就可以避免误判。

5.5.4 故障修复

为了提高容错能力,可以在无线传感器网络部署之初放置一些冗余节点。当有节点失效时,冗余的节点移动到指定位置,从而弥补失效节点所造成的连接割断或覆盖漏洞。

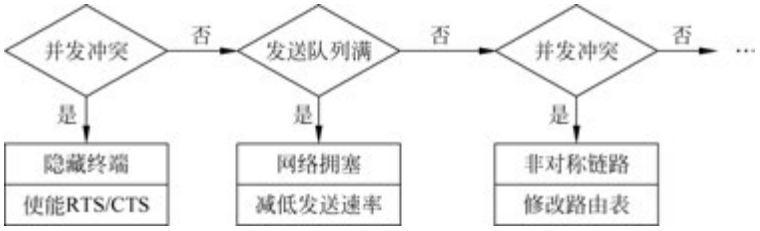


图 5-43 分布式故障诊断算法

1. 基于连接的修复

1) 部署 k 连通网络

无线传感器网络中有些节点一旦出现故障,网络就会被断开。如图 5-44 所示,如果发生故障,网络就被划分成 3 个独立部分。为了使无线传感器网络能够容忍节点发生故障,一种方法是在任意两个节点之间部署多条路径。

一种建立容错拓扑的方法是构造 k 连通网络。k 连通网络是指网络中任意两点之间都至少有 k 条不相交的路径,k 连通网络中任意 $k-1$ 个节点发生故障时网络仍然保持连通。图 5-45 所示为三连通网络,它能容忍任意两个节点发生故障而保持网络的连通性。

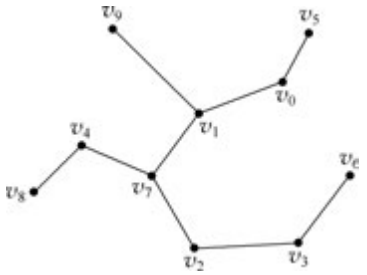


图 5-44 节点失效会导致网络断开

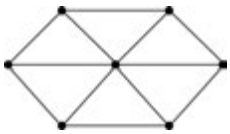


图 5-45 三连通网络

在完全图中找最小代价的 k 连通子图是很难的,目前有很多近似算法。其中 $FGSS_k$ 和 $FLSS_k$ 分别是在网络中维持 k 连通的全局和局部近似算法。

$FGSS_k$(Fault-tolerant Global Spanning Subgraph)是一个集中式贪婪算法,它使得网络中的最大能耗最小化。$FGSS_k$ 算法把网络模型化为一个图,节点是图中的点,存在直接通信连接的节点间作一条边,两个节点间的距离作为该边的权重。$FGSS_k$ 算法分为 3 步,首先按权重对所有边排序;然后按边权重从大到小依次考虑边是否应该加入生成子图,判断准则是该边的两个端点还没有 k 条路径连通时,就把这条边加入生成子图;最后判断是否所有节点都达到 k 连通,如果不是则重复第二步,否则算法结束。

$FLSS_k$(Fault-tolerant Local Spanning Subgraph)是一个分布式算法,它通过局部信息调整发射功率和维护邻居集,由信息收集、拓扑构造、决定传输能量 3 个阶段组成。在信息收集阶段,每个节点以最大的能量广播自己的 ID 和位置信息,同时收集周围节点发来的信息,形成自己的邻居集;在拓扑构造阶段,使用 $FGSS_k$ 算法,每个节点生成一个子图,在这个子图中它与其他邻居都是 k 连通的;在决定传输能量阶段,通过去掉单向边或加强为双向边改进生成的连通图。

2) 非 k 连通网络

网络维持 k 连通需要消耗大量资源,所以很难在大规模的网络中达到 k 连通。那么在

非 k 连通网络中,需要有特定的方法处理节点故障。如图 5-46 所示,一个节点或一片节点发生故障时,基站将不能收到它们的消息。

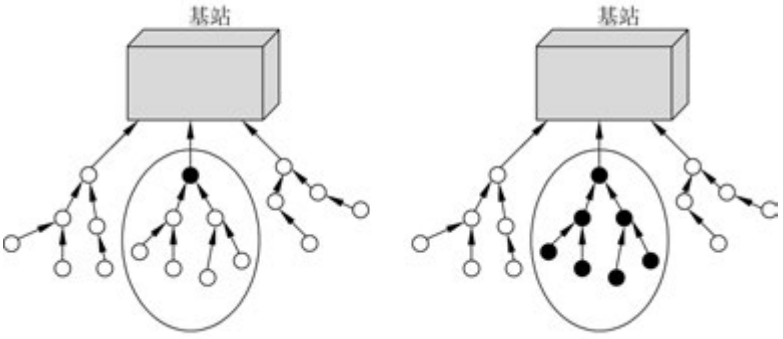


图 5-46 基站收不到某些节点的消息

基站查询到从某个节点开始通信中断了,可以把它们的下一跳重新定向到能与基站保持通信的最近节点。图 5-47 给出了两种路由选择的新方案。

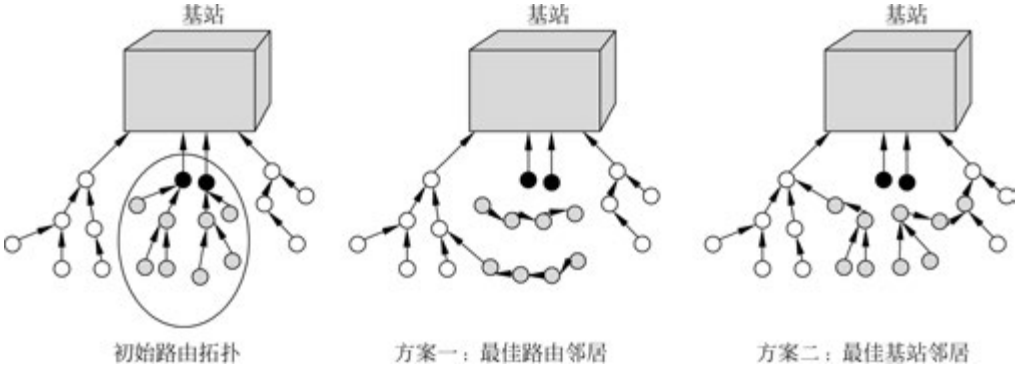


图 5-47 重新路由

为了延长整个网络的生命期以及降低对网络带宽的要求,无线传感器网络通常不需要所有节点都保持活动状态。这样可以用冗余节点或睡眠节点弥补故障节点所造成的网络连接断开。容错节点是指一种可以替换失效活动节点的睡眠节点或冗余节点,表 5-15 给出了活动节点选择容错节点的方法。

表 5-15 基于连接的容错节点选择(处理节点 S_k)

节点 S_k 的状态	动 作
活动(Active)	S_k 对每对没有互连的邻居指定一个容错节点(FT Node),这个容错节点与这对邻居都有连接
睡眠(Sleep)	(1) 如果 S_k 的两个没有互连邻居节点中,两个都是容错节点,则 S_k 为容错节点 (2) 如果 S_k 的两个没有互连邻居节点中,一个是容错节点,另一个是睡眠节点,则 S_k 为容错节点 (3) 如果 S_k 的邻居节点中没有容错节点,则 S_k 为容错节点

活动节点失效会造成某些邻居的连接断开,所以需要为它们指定容错节点,即活动节点

失效时，它的邻居可以通过指定的容错节点来通信。如图 5-48 所示，S_k 指定 S_3 为 S_1、S_2 的容错节点。当 S_k 失效时，S_1 和 S_2 可以通过 S_3 保持连通。

2. 基于覆盖的修复

无线传感器网络监测环境时，节点失效会造成某些区域不被覆盖，这时需要采取措施弥补造成的覆盖空洞。节点的覆盖区域定义为整个感知区域去掉与其他节点重叠的感知区域，可以在网络中部署一部分可以移动的节点，当其他节点失效时，这些冗余的节点就移动到某个区域以弥补失效节点对网络造成的影响。如图 5-49 所示，节点 A 的感知区域为粗线围成的区域，它失效后，它的覆盖区域需要其他节点来弥补。表 5-16 给出了覆盖区域和移动区域的定义。

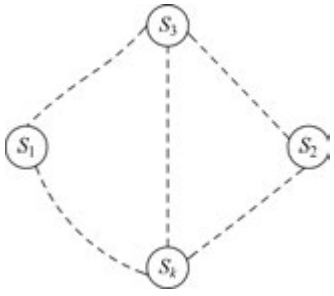


图 5-48 容错节点

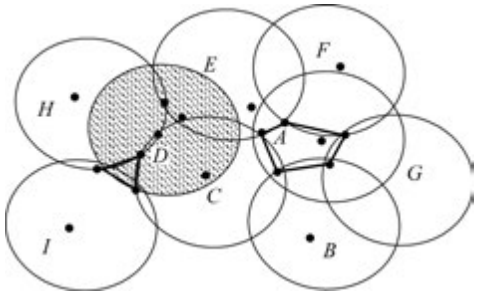


图 5-49 节点 A、节点 D 的感知区域

表 5-16 覆盖区域和移动区域的定义

名 称	定 义	计 算 方 法
覆盖区域	节点单独覆盖的区域	整个感知区域去掉与其他节点重叠的感知区域
移动区域	有效节点移动到该区域即可重新覆盖漏掉的区域	以遗漏区域的边界为圆心，节点感知半径为半径作圆所形成的区域

假设网络中的节点具有移动能力，将覆盖修复过程分为 4 个阶段。

(1) 初始化阶段。节点计算自己的覆盖区域，以及每个覆盖区域对应的移动区域。

(2) 恐慌请求阶段。垂死节点广播求助消息。

(3) 恐慌回应阶段。垂死节点的邻居收到求助消息后，计算如果自己移动到垂死节点的移动区域，是否会影响到自身的覆盖区域，如果不影响则给求助节点返回消息。

(4) 决策阶段。垂死节点根据收到的回应信息，决定让哪个节点移动。

在节点地理位置未知时，提供了失效节点寻找冗余节点的算法。冗余节点通过移动弥补垂死节点以保证覆盖和连通。由于地理位置未知，主要考虑用试探的方法探测到移动的方向。

5.6 QoS 保证

5.6.1 QoS 概述

1. 服务质量定义

随着网络多媒体技术的飞速发展，Internet 上的多媒体应用层出不穷，如 IP 电话、视频

会议、视频点播(Video on Demand,VoD)、远程教育等多媒体实时业务、电子商务等。Internet 已逐步从单一的数据传输网演化为数据、语音、图像等多媒体信息的综合传输网。这些应用对网络中的带宽、延时以及分组丢失率等传输质量参数提出了不同要求。显然,尽力传输服务已无法满足这些要求,需要 Internet 提供相应的机制满足应用对服务质量(QoS)的要求,确保数据传输的适当服务级别。表 5-17 列出了目前 Internet 上的一些主要应用的业务特征及其 QoS 需求。

表 5-17 Internet 业务特征及 QoS 需求

应 用	业 务 特 征	QoS 需求
电子邮件/文件传输、远程终端	数据量小、批文件的传输	允许延时;带宽需求低;尽力而为传输
HTML 网页浏览	一系列小的、突发的文件传输	允许适当的延时;带宽需求变化;尽力而为传输
客户端/服务器电子商务	许多小的双向传输	对延时、丢失率敏感;带宽需求适当;必须可靠传输
基于 IP 的语音实时音频	连续或变化地传输	对延时、抖动非常敏感;带宽需求低;需要可预计的延时和丢包率
流媒体	变化的位速	对延时、抖动非常敏感;带宽需求高;需要可预计的延时和丢包率

目前网络界针对如何定义网络 QoS 并没有一个统一的标准。QoS 论坛将 QoS 定义为网络元素(包括应用、主机或路由器等网络设备)对网络数据的传输承诺的服务保证级别。RFC2386 则将 QoS 看作网络在从源节点到目的节点传输分组流时需要满足的一系列服务要求。同样,在网络分层模型中,不同的层对 QoS 也有不同的解释。为了更好地理解 QoS 的含义,从应用层和网络层的角度对 QoS 定义进行分析。在应用层,QoS 通常是指用户或应用所获取具体业务的服务质量。而在网络层,QoS 则定义为对网络提供给应用及用户的服务质量的量度,网络提供特定 QoS 的能力依赖于网络自身及其采用的网络协议的特性。图 5-50 给出了一个简单的 QoS 模型。在这个模型中,应用/用户并不关心网络如何利用自身资源提供 QoS 支持,他们只关心直接影响应用质量的网络服务。而网络的目标则是最大限度地利用网络资源提供 QoS 支持。为了实现这个目标,网络需分析应用 QoS 需求,并通过调度自身资源满足提供 QoS 支持。

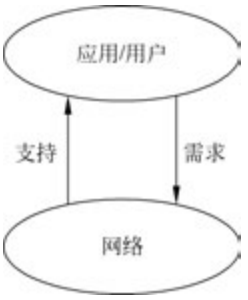


图 5-50 简单的 QoS 模型

在网络 QoS 研究中,人们比较关注的服务质量标准主要包括可用性、吞吐量、延时、延时变化和丢包率等几个参数。

可用性指综合考虑网络设备的可靠性与网络生存性等网络失效因素,当用户需要时网络即能开始工作的时间百分比。

吞吐量又称为带宽,是在一定时间段内对网络流量的量度。一般而言,吞吐量越大越好。

延时是指一项服务从网络入口到出口的平均经过时间。许多实时应用,如语音和视频

等服务对延时的要求很高。产生延时的因素很多，包括分组延时、排队延时、交换延时和传播延时等。

延时变化是指同一业务流中不同分组所呈现的延时不同。高频率的延时变化称作抖动，而低频率的延时变化称作漂移。抖动主要是由于业务流中相继分组的排队等候时间不同引起的，是对服务质量影响最大的一个问题。

丢包率是指网络在传输过程中数据包丢失的比例。造成数据包丢失的主要原因包括网络链路质量较差、网络发生拥塞等。

针对不同应用对网络 QoS 的需求，研究人员主要通过在不同网络协议层使用各种不同的算法与机制最大限度地利用网络资源提供端到端的 QoS 支持。

2. 服务质量支持机制

当前，Internet 如何提供 QoS 支持(即 IP QoS)已成为业界关注的焦点，研究人员开展了大量的工作。但是，在网络 QoS 支持的研究中仍然存在着许多挑战和困难。首先，网络提供 QoS 支持的最大挑战是网络拥塞。在网络发生拥塞时，由于数据包在每跳的队列缓冲区中都将等待更长的时间，所以会导致网络延迟增大。当队列缓冲区发生溢出时，新到数据包将被丢弃，导致网络丢包率上升，同时网络的吞吐量也将下降。其次，为提高网络数据包传输的可靠性，通常采用多路径路由。这将带来新的问题，当两个数据包从源端发往目的端，没有任何保障时，它们会采用相同的路径到达目的端。当其中某个数据包由于所选择的路由路径较长或所经过的节点发生拥塞时，就无法保证两个数据包到达目的端的时间，这样会带来不可接受的延时与抖动。

为解决上述问题，更好地满足应用的需求，研究人员提出了许多 QoS 支持机制。下面只简单介绍其中 3 种最主要的体系结构。

1) Int-serv 集成业务

Int-serv 主要引入了一个重要的网络控制协议——资源预留协议(Resource Reservation Protocol，RSVP)。RSVP 的引入使得 IP 网络为应用提供所要求的端到端的 QoS 保证成为可能。Int-serv 尽管提供 QoS 保证，但其扩展性差。因为其工作方式是基于每个流的，这就需要保存大量的与分组队列数成正比的状态信息。此外，RSVP 的有效实施必须依赖于分组所经过路径上的每个路由器。在骨干网中，业务流的数目可能很大，因此要求路由器的转发速率很高，这使得 Int-serv 难以在骨干网中得到实施。

2) Diff-serv 区分业务

因特网工程任务组(Internet Engineering Task Force，IETF)在 RFC22475 中提出 Diff-serv 体系结构，旨在定义一种能实施 QoS 且更易扩展的方式，以解决 Int-serv 扩展性差的问题。Diff-serv 简化了信令，对业务流的分类粒度更粗。Diff-serv 通过汇聚(Aggregate)和 PHB (Per Hop Behavior)的方式提供 QoS。汇聚是指路由器把 QoS 需求相近的业务流看成一个大类，以减少调度算法所处理的队列数。PHB 是指逐跳的转发方式，每个 PHB 对应一种转发方式或 QoS 要求。由于 Diff-serv 采用对数据流分类聚集后提供差别服务的方法实现对数据流的可预测性传输，所以对 QoS 的支持粒度取决于传输服务的分级层次，各网络节点中存储的状态信息数量仅正比于服务级别的数量，而不是数据流的数量，由此 Diff-serv 获

得了良好的扩展性。

3) 多协议标签交换

多协议标签交换(Multi-Protocol Label Switching,MPLS)将灵活的3层IP选路和高速的两层交换技术完美地结合起来,从而弥补了传统IP网络的许多缺陷。它引入了"显式路由"机制,对QoS提供了更可靠的保证。MPLS在路由寻址方面同传统路由器有明显的不同。MPLS支持特殊路由,到达同一目的地的数据包可沿不同路径进行转发。MPLS网络主要由标签边缘路由器(Label Edge Router,LER)和标签交换路由器(Label Switching Router,LSR)组成。相比于集成业务与区分服务,MPLS是目前最全面的服务质量保证体系。

以上3种体系结构仅仅提供了一种在区域网络内实施QoS支持的框架结构,而具体的一些策略和相应的实现机制则由不同的厂商来决定。目前有关IP QoS的4种实现机制大致可归纳为队列管理机制、队列调度机制、基于约束的路由(Constraint Based Routing,CBR)和流量工程。

5.6.2 QoS研究

当无线自组织网络出现时,人们也很自然地有了在自组织网络上传输不同类型业务的需求,并且希望自组织网络能像固定的有线网络一样为不同业务的服务质量提供保障。然而,与固定的有线网络不同,在无线自组织网络中,无线链路的带宽相对较低,移动节点的内存、能源等资源都相对受限。因此,传统有线网络中的QoS支持机制无法直接应用于无线自组织网络。如何合理、有效地利用无线网络的资源,以获取更好的数据传输性能,进而为多媒体业务的服务质量提供保障,是无线自组织网络QoS研究中需要解决的首要问题。

无线传感器网络是一种新兴的无线自组织网络,由大量无线传感器节点通过自组织的方式组成网络。节点的传感器负责采集监测数据,无线射频收发装置负责节点之间以及节点与基站之间的数据交互。无线传感器网络的工作方式以及承载的业务类型都与传统网络以及无线通信网络有着很大的区别。

(1) 无线传感器网络中,节点资源非常有限,其中尤以节点能量为最大限制。节点通常是电池供电的,并且工作在不易更换电池的环境中。因此,计算复杂度以及开销过高的算法都不适用于无线传感器网络。

(2) 在大多数无线传感器网络应用中,网络中占多数的节点需要将采集的传感数据发送到基站节点,而基站节点仅发送少量的查询控制数据回传感器节点。因此,在设计无线传感器网络算法协议时需要考虑网络通信负载的不均衡性。

(3) 无线传感器网络中通常有大量冗余节点,虽然这些冗余节点能够提高传输数据的可靠性,但也导致了网络额外的能量消耗。如何在保证数据可靠性的同时减少数据冗余,是无线传感器网络有待解决的一个重要问题。

(4) 无线传感器网络是一个异构网络,网络节点包含各种不同类型的传感器,如光、温度、湿度、压力、加速度以及图像传感器等。这将导致传感器采集传感数据的频率以及传输数据所需带宽等方面都有很大的不同。

上述无线传感器网络自身的特点,使得现有QoS支持机制都无法适用于无线传感器网

络。实际上，自无线传感器网络出现以来，人们一直在对如何定义无线传感器网络的服务、如何衡量无线传感器网络的服务质量以及无线传感器网络究竟是否需要服务质量支持而争论不休。目前，研究人员普遍达成共识：无线传感器网络需要 QoS 支持，但是不同的应用对 QoS 有不同的理解和需求，无法对无线传感器网络的 QoS 形成统一的定义。

Holger Karl 在 *Protocols and Architecture for Wireless Sensor Network* 一书中将可靠性视作最重要的质量因素。他认为在无线传感器网络中，可靠性包含检测可靠性(Detection Reliability)以及可靠数据传输(Reliable Data Transport)两方面。检测可靠性是指无线传感器网络能否准确检测到监测区域发生的事件，即网络中所有活动节点传感范围能否覆盖整个监测区域。这类问题通常归结为覆盖问题。可靠数据传输是指如何保证传感器节点监测到的数据可靠地传输到 Sink 节点。Dazhi Chen 对当前无线传感器网络的 QoS 支持研究进行了总结和归纳。他将当前无线传感器网络中的 QoS 研究总结为 3 类。①传统端到端 QoS 支持研究：针对实时性无线传感器网络应用，提供延迟服务保证；②可靠性保证：保证数据包传输的可靠性；③应用相关 QoS：包括传感覆盖和如何控制网络活动节点数量等问题。Mohamed Younis 则针对无线传感器网络中的 QoS 路由协议以及提供 QoS 支持的 MAC 层协议分别进行了介绍。无线传感器网络 QoS 需求参数及定义如表 5-18 所示。

表 5-18 无线传感器网络 QoS 需求参数及定义

网络体系结构	QoS 需求参数	QoS 参数定义
应用层 (Application Layer)	网络生存寿命	从网络撒布工作到其不能满足用户需求的时间间隔
	网络响应时间	向网络发出查询命令到其获得响应的延时
	数据时效性 (Data Freshness)	传感器节点发现数据到相应传感数据传输到基站节点的延时
	检测概率	现实环境中的事件被网络发现并报告给用户的概率
	数据可信度 (Data Fidelity)	用户从网络中所获取的数据的可信度
	数据精细度 (Data Resolution)	用户从网络中所获取的数据的时空精细粒度
数据存放与发现 (Data Placement and Discovery)	丢失率	由于数据存放和发现所导致的信息丢失百分比
	出错率	由于数据存放和发现所导致的信息出错百分比
	通信代价	向中间存储节点或基站节点存放和提取信息所耗费的能量
	发现延迟 (Discovery Latency)	向中间存储节点或基站节点发送用户查询请求并成功提取信息的延迟
	存放延迟 (Placement Latency)	将数据传输到中间存储节点或基站节点的时间延迟
网内数据处理 (In-network Data Processing)	计算开销	为生成一个样本所进行的网内数据处理的能量消耗
	数据抽象 (Data Abstraction)	为描述现实世界现象所需处理的数据量
	数据完整性	所处理数据中包含事件发现信息的比例
	数据精度	所处理传感数据的精度
	处理延迟	预处理数据延迟

续表

网络体系结构	QoS需求参数	QoS参数定义
传输层 (Transport Layer)	N-to-End Reliability	从所有传感数据源成功接收的单一数据包个数占实际传输数据包个数的比例
	N-to-End Bandwidth	单位时间从所有传感数据源成功接收单一数据包数
	N-to-End Latency	从所有传感数据源传输至目的地的单一数据包中最短延时
	N-to-End Cost	从所有发送源获取单一数据包所需的传输次数
网络层 (Network Layer)	路径延迟	网络所有源到目的路径的平均跳段数
	能量效率	沿路由路径传输一个数据包所消耗的能量
	路由维护开销	维护网络中所有源到目的路径的能量消耗率
	拥塞概率	路径流量负载超过路径所有链路的瓶颈容量的概率
	路由鲁棒性	一对节点之间的最少路径数
连通保持层 (Connectivity Maintenance Layer)	网络半径	网络中两个节点之间的最大传输延迟
	网络容量	网络中可以并发传输的最大数据包个数
	平均路径代价	所有源到目的节点对中,传输一个数据包的平均能量消耗
	连通保持代价	保持一个网络拓扑连通的能量消耗率
	连通鲁棒性	保持一个网络连通所能容许的失效节点个数
MAC层	通信距离	一跳数据所能够传输的最大距离
	吞吐量	单位时间MAC层能够成功传输的数据帧最大个数
	能量效率	一跳范围内成功传输一帧数据所耗费的能量
	传输可靠性	成功传输帧的比例
覆盖保持层 (Coverage Maintenance Layer)	观测精度	监测到事件发生的传感器节点而产生数据的准确度
	采样量	监测到事件发生的传感器节点单位时间所产生的数据量
	覆盖鲁棒性	覆盖质量,监测事件发生的传感器节点个数
	覆盖保持代价	保持所需网络覆盖的能量消耗率
定位与时钟服务层	定位与时钟精确度	网络中传感器节点定位/时钟信息的精确度
	能量消耗	提供定位与时钟服务的能量消耗比率
物理层	无线单元功能 (Wireless Unit Capabilities)	信道速度、编码、射频功耗
	传感单元功能 (Sensing Unit Capabilities)	采样精度与频率、测量精确度、传感距离与传感功耗
	处理单元功能 (Processor Capabilities)	处理速度、计算功耗、定位功能与时钟同步功能

Jun Lu认为前面提到的无线传感器网络QoS支持研究都只针对网络协议中的某个层或某个特定的应用,没有从整体上考虑网络的各部分对QoS的影响。他认为无线传感器网络QoS研究应该首先从整体上明确无线传感器网络QoS的需求,然后分析这些QoS需求与网络各功能组件之间的关系。他详细地定义并分析了网络的QoS需求及各需求之间的

关系，并给出了一个无线传感器网络 QoS 分析模型。该模型是目前无线传感器网络 QoS 研究中最为全面和详细的概括和分析，涵盖了目前无线传感器网络中的各协议层及基本服务。利用该框架，应用设计者可以非常清晰地确定当前应用的 QoS 需求，而系统工程师在设计网络时则能够从全局协调受当前 QoS 需求影响的各功能模块之间的关系。

通过上述工作的分析，发现在目前大多数无线传感器网络应用中，人们关注较多的主要有两个问题：①如何保证网络能够及时可靠地发现所实施应用中相关事件的发生？②如何保证采集的传感数据在网络中传输时满足应用需求？这两个问题可以归结为感知服务质量（传感覆盖）和网络传输服务质量。

5.7 安全技术

5.7.1 安全攻击

WSN 易受各种攻击，根据 WSN 的安全要求，对 WSN 的攻击归类如下。

（1）对秘密和认证的攻击。标准加密技术能够保护通信信道的秘密和认证，使其免受外部攻击，如偷听、分组重放攻击、分组篡改、分组哄骗。

（2）对网络有效性的攻击。对网络有效性的攻击常常称为拒绝服务（Denial of Service，DoS）攻击，可以针对传感器网络任意协议层进行 DoS 攻击。

（3）对服务完整性的秘密攻击。在秘密攻击中，攻击者的目的是使传感器网络接收虚假数据，如攻击者威胁一个传感器节点的安全，并通过这个节点向网络注入虚假数据。

在这些攻击中，使传感器网络继续发挥其预定作用是必要的。DoS 攻击通常就是攻击者针对网络进行的破坏、扰乱、毁灭。DoS 攻击可以是削弱或消除网络执行其预定功能的能力的任何事件。由于能够针对传感器网络任意协议层进行 DoS 攻击，所以层次化体系结构使得 WSN 在面对 DoS 攻击时很脆弱。下面按照 WSN 协议层次结构分析 WSN 的安全攻击。

1. 物理层攻击

物理层负责频率选择、载波频率生成、信号检测、调制/解调、数据加密/解密。传感器网络是 Ad-Hoc 大规模网络，主要采用无线通信，无线传输介质是开放式媒介，因此在 WSN 中有可能存在人为干扰。对于布置在敌方环境或不安全环境中的 WSN 节点，攻击者很容易进行物理访问。

1）人为干扰

对无线通信的一种众所周知的攻击就是采用干扰台干扰网络节点的工作频率。一个干扰源只要功率足够大，能够破坏整个 WSN；如果功率比较小，只能破坏网络中一个较小区域。即使采用功率较低的干扰源，假如干扰源随机分布在网络中，那么攻击者仍然有可能破坏整个网络。攻击者使用 k 个随机分布的干扰节点就能够破坏整个网络，使 k 个节点处于服务之外，k 比 N 小得多。对于单个频率的网络，这种攻击既简单又有效。

抗人为干扰的典型技术就是采用各种扩频通信技术（如跳频、码扩）。跳频扩频（Frequency Hopping Spread Spectrum，FHSS）就是发送信号时使用发射机和接收机均知道的伪随机序列在许多频率之间迅速切换载波频率。攻击者若不能跟踪频率选择序列，则

不能及时干扰给定时刻的工作频率。但是,由于工作频率范围是有限的,所以攻击者可以干扰工作频带的很大一部分甚至整个工作频带。

码扩是用来对抗人为干扰的另一种技术,通常用于移动网络中。码扩设计复杂性较高,能量需求也较高,从而限制了其在 WSN 中的应用。一般地,为了维护低成本和低功耗要求,传感器装置采用单频率工作,因此极易受人工干扰攻击。

假如攻击者持久性采用干扰台干扰整个网络,那么就会得到有效而完整的 DoS 效果。因此,传感器节点应该采用对抗人工干扰的策略,如切换到较低占空因数,尽量节省能量。

节点周期性苏醒,检查人工干扰是否已经结束。传感器节点通过节省能量可能能够承受得住攻击者的人工干扰,此后攻击者必须以更高的成本进行人工干扰。

假如人工干扰是断断续续的干扰,那么传感器节点可以采用高功率给中心节点发送几条高优先级的消息,将人工干扰报告给中心节点。各传感器节点应该相互协作,共同努力将这些消息交付给中心节点。传感器节点也可以不定期地缓存高优先级消息,等待在人工干扰间隙将其中继给其他传感器节点。

对于大规模 WSN,攻击者要成功干扰整个网络比较困难;假如进行干扰的只是被攻击者攻克的原网络节点,那么要成功干扰整个网络就更加困难了。

2) 物理篡改

攻击者也可以从物理上篡改、询问和危害 WSN 节点,这些是针对 WSN 不断恶化的安全威胁。实际上,实施对分布在数千米范围内的几百个传感器节点的访问控制是极困难的,甚至是不可能的。WSN 不仅要承受武力破坏,而且还要承受较复杂的分析攻击。攻击者可以毁坏 WSN 节点,使其丧失正常工作能力;可以替换 WSN 节点中的关键组件(如传感器硬件、计算硬件,甚至软件),将 WSN 节点变成失密节点,从而对其实现掌控;也可以提取 WSN 节点中的敏感组件(如加密密钥),以便能够自由访问高层通信。节点被毁、节点故障静默这两种情形可能无法区分。

物理篡改的一种对抗措施是篡改验证节点的物理层分组。这种对抗措施的成功依赖于:①WSN 设计者在设计 WSN 时就精确、完整地考虑可能存在的物理安全威胁;②可用于设计、结构、测试的有效资源;③攻击者的智慧高低和果断程度。但是,这种对抗措施通常假定在 WSN 中,由于额外的成本开销,传感器节点是不能篡改验证的。这就意味着安全机制必须考虑传感器节点被危害的情形。

2. 链路层攻击

MAC 层为相邻节点到相邻节点的通信提供信道仲裁,基于载波侦听的协作性 MAC 协议特别易受 DoS 攻击。

1) 碰撞

攻击者只需要发送一字节就可能产生碰撞,从而损坏整个分组。分组中的数据部分发生变化,则在接收方不能通过校验和检验。ACK 控制消息被损坏会引起有些 MAC 协议退避时间呈指数增长。除了旁听信道发送之外,攻击者需要的能量极少。

采用差错纠错机制能够容忍消息在任意协议层次上遇到不同程度的损伤,差错纠错编码本身存在额外的处理开销和通信开销。对于一个给定的差错纠错编码,恶意节点仍然能够使其损坏的分组多于网络能够纠正的分组,但是开销较高。

网络可以采用碰撞检测技术识别恶意碰撞,恶意碰撞会产生一种链路层人为干扰,但是

迄今为止还没有彻底有效的防护措施和技术。正当发送仍然需要节点之间的相互协作,以期避免互相损坏对方发送的分组。一个被攻击者彻底颠覆的节点能够故意、反复拒绝信道访问,而其能耗比全时段人工干扰低得多。

2) 能量消耗

链路层可能采用反复重传技术。即使被一个异常延迟的碰撞(如在本帧即将结束时引起的碰撞)所触发,也可能会进行重传。这种主动 DoS 攻击会耗尽附近节点的电池储能,危害网络的可用性(即使攻击者不再进行攻击)。随机退避只能降低无意碰撞概率,却不能防止这种攻击。

时分复接给每个节点分配一个发送时隙,不需要为发送每个帧而进行信道访问仲裁。这种方法能够解决退避算法中的不确定性延迟问题,但是仍然易受碰撞攻击。

可以利用大多数 MAC 协议的交互式特性进行询问攻击。例如,基于 IEEE 802.11 的 MAC 协议采用 RTS/CTS/DTA/ACK 交互方式预留信道访问和发送数据,因此节点可以反复利用 RTS 请求信道访问,得到目标相邻节点的 CTS 响应,持续发送最终耗尽发送节点和目标相邻节点的能量资源。

一种解决方法是限制 MAC 准入控制速率,网络对过多信道访问请求不予理睬,不进行能耗甚高的无线发送。这种限制策略不会使准入速率下降到网络所能支持的最大数据速率以下(但是会发生这种情况)。防止电池能量消耗攻击的一个策略是限制无关紧要却是 MAC 协议所需要的响应。为了提高总体效率,设计人员常常在系统中实现这种能力,但是处理攻击的软件代码需要额外逻辑。

3) 不公平性

不公平性是一种较弱形式的 DoS 攻击。断断续续地运用碰撞攻击和电池能量消耗攻击,或者滥用协作性 MAC 层优先权机制会引起不公平性。这种安全威胁尽管不能完全阻止合法的信道访问,但是会降低服务质量,如导致实时 MAC 协议的用户发生时间错位。

一种对付不公平性攻击的方法是采用短帧结构,因此每个节点占用信道的时间较短。但是,假如网络经常发送长消息,那么这种方法导致成帧开销上升。在竞争信道访问时,攻击者采取欺骗手段很容易突破这种防护措施——攻击者迅速作出响应,而其他节点则随机延迟其响应。

3. 网络层(路由)攻击

由于 WSN 常常依靠电池供电,而电池能量非常有限,所以许多传感器网络路由协议设计得很简单,节省能量,使节点寿命、网络寿命达到最大,因此有时易受攻击。各种 WSN 网络层攻击的主要差异表现在是试图直接操作用户数据的攻击还是试图影响低层路由拓扑的攻击。针对 WSN 进行的网络层攻击分成以下几类:对路由信息的哄骗、篡改、重放,选择性转发,污水池攻击,女巫攻击,蠕虫攻击,Hello 泛洪攻击,确认哄骗。

1) 对路由信息的哄骗、篡改、重放

针对路由协议最直接的攻击就是以节点之间交换的路由信息为目标进行攻击。攻击者通过对路由信息的哄骗、篡改、重放,能够创建路由闭环、吸引或抵制网络流量、延长或缩短源路由、产生虚假错误消息、分割网络、增大端到端延时等。

2) 选择性转发

多跳网络常常假定参与节点安全、正确地转发收到的消息。在选择性转发攻击中,攻击

者可能拒绝转发某些消息,简单地将这些消息丢掉,确保这些消息不会进一步传播。当恶意节点的表现类似黑洞、拒绝转发通过其传递的每个分组时,就是这种简单形式的选择性转发攻击。攻击者采用这种形式攻击存在风险:由于接收不到攻击者节点发送的消息,所以相邻节点将会认为攻击者节点已经失效,因而决定寻找另一条路由。另一种表现形式稍有不同的选择性转发攻击是攻击者选择性地转发分组,其兴趣在于抑制或篡改若干精选节点产生的分组,但是仍然可靠转发其余流量分组,从而降低了其攻击行为被怀疑的可能性。

当攻击者直接处于数据流传输路由上时,选择性转发攻击通常是非常有效的。攻击者旁听通过相邻节点的数据流量,因此通过人为干扰或碰撞其感兴趣的每个转发分组就能够模仿选择性转发。这种攻击机制需要高超技巧,因此很难施行。例如,如果网络中每个相邻节点对使用唯一一个密钥初始化跳频通信或扩频通信,那么攻击者要施行这种攻击极其困难。因此,攻击者很可能沿着抗攻击能力最弱的路径,并且尽量包含自身的数据流实际传输路径进行选择性转发攻击。

3) 污水池攻击

在污水池攻击中,攻击者的目的是引诱来自某个特定区域的附近所有流量通过一个失密节点,从而产生一个比喻性的"污水池",中心位置就是攻击者。由于分组传输路径上的节点及其附近的节点有很多机会篡改应用数据,所以污水池攻击能够同时伴随许多其他攻击,如选择性转发攻击。

污水池攻击的工作原理是使失密节点对路由算法和周围节点看上去很有吸引力。例如,攻击者可以哄骗或重放到达中心节点的极高质量路由广播消息。有些路由协议可能会采用端到端应答(包含可靠性、延时信息)真正验证路由的质量。此时,微型计算机类攻击者采用大功率发射机直接对中心节点发送(发射功率足够高,单跳可达)或采用蠕虫攻击,就能够提供到达中心节点的真正高质量路由。由于存在通过失密节点的真正或虚假高质量路由,所以攻击者的每个相邻节点很可能将传递给中心节点的分组转发给攻击者,并且又将这种高质量路由信息传播给自己的相邻节点,攻击者由此有效创建一个巨大的影响球吸引传递给中心节点的所有数据流(来自离失密节点数个转发跳的节点)。

进行污水池攻击的一个动机是进行选择性转发攻击,攻击者通过确保特定目标区域的所有数据流传递通过失密节点,就能够选择性抑制或篡改来自该区域任意节点的分组。

传感器网络特别易受污水池攻击的原因在于其特殊的通信模式。因为所有分组的最终目的节点只有一个中心节点(在只有一个中心节点的 WSN 中),所以失密节点只需要提供单跳可达中心节点的高质量路由就有可能影响大量传感器节点。

4) 女巫攻击

女巫攻击是指一个恶意装置非法占用多个网络身份。将一个恶意装置的额外身份称为女巫节点。女巫攻击会大幅度降低路由协议、拓扑维护中的容错功效。认为使用不相交节点的各条路由实际上包含冒充多个身份的那个攻击者节点。

一个女巫节点可以采取以下方法获取身份。一种方法是伪造一个新的身份。在有些情况下,攻击者可以简单、任意地产生新的女巫身份。例如,假如使用一个 32 位整数表示每个节点的身份,那么攻击者可以给每个女巫节点分配一个 32 位随机整数。另外一种获取身份的方法是窃取某个合法节点的身份。给定一个合法节点身份识别机制,那么攻击者可能无法伪造新的身份。此时攻击者需要将其他合法节点的身份分配给女巫节点。假如攻击者摧

毁了假扮节点或使假扮节点临时失效，那么可能无法察觉这种身份窃取行为。

女巫节点直接与合法节点通信，当一个合法节点向一个女巫节点发送一条消息时，其中一个恶意装置在无线信道上侦听此消息。女巫节点发送的消息实际上是其中一个恶意节点发送的。假如合法节点不能与女巫节点直接通信，那么其中一个或多个恶意装置声明能够到达女巫节点。女巫节点发送的消息通过其中一个恶意节点传递，后者假装将消息传递给女巫节点。

女巫攻击对地理路由协议威胁极大。位置意识路由为了高效地利用地理路由传递分组，一般要求节点与其相邻节点交换位置坐标信息，攻击者运用女巫攻击就能够“立即出现在多个地点”。

5）蠕虫攻击

一条蠕虫就是一条连接两个网络子区域的低延时链路，攻击者在这条链路上中继网络消息。蠕虫可以由单个节点创建，即该节点位于两个相邻或不相邻节点之间，转发其间的消息；也可以由一对节点创建，即这两个节点分别位于两个不同的网络子区域，并且相互进行通信。

在蠕虫攻击中，攻击者接收到某个网络子区域的消息，然后沿着低延时链路（蠕虫）将这些消息重放到网络其他区域中。特别是在同一个通信节点对之间，通过蠕虫发送的分组传输延时小于采用正常多跳路由的分组传输延时。最简单的蠕虫攻击就是一个节点位于另外两个节点之间，转发这两个节点之间的消息。但是，蠕虫攻击通常涉及两个相距较远的恶意节点，这两个恶意节点共同有意低估相互之间的距离，沿着只有攻击者才能够使用的带外信道中继分组。

假如攻击者离中心节点较近，那么攻击者通过精心设计和布置的蠕虫就有可能彻底破坏路由。攻击者可能使离中心节点数个转发跳的节点相信通过蠕虫只距离一跳或两跳。这就能够产生污水池——处在蠕虫另一边的攻击者能够提供到达中心节点的虚假高质量路由，如果备用路由没有竞争力，那么附近区域中的所有流量有可能通过蠕虫传递，当蠕虫的端点离中心节点相对较远时就很可能总是如此。

较一般的情况是，蠕虫可以充分利用路由竞争条件。当一个节点根据其接收的第一条消息而忽略随后消息采用某种操作时通常就会出现路由竞争条件。在这种情况下，如果攻击者能够使节点在多跳路由正常到达时间前接收某种路由信息，那么攻击者就能够影响最后得到的拓扑。蠕虫正是这样实现的，即使路由信息被加密和需要认证，蠕虫也仍然有效。蠕虫通过中继两个相距甚远的节点之间的分组使这两个节点相信是相邻节点。

蠕虫攻击很可能与选择性转发或偷听一起使用。当蠕虫攻击与女巫攻击一起使用时，可能很难检测到蠕虫攻击。

6）Hello 泛洪攻击

Hello 泛洪攻击就是攻击者利用 WSN 路由协议中使用的 Hello 消息进行的攻击。很多 WSN 路由协议要求节点广播 Hello 消息，以向其相邻节点声明自己的存在和广播自己的一些信息（如身份、地理位置）。接收到 Hello 消息的节点则可假定自己处在该 Hello 消息发送节点的覆盖范围内。这个假设条件有可能是虚假的，如微型计算机类的攻击者采用足够大的发射功率广播路由或其他信息，就能够使网络中每个节点相信攻击者就是其相邻节点。

攻击者给每个网络节点广播到达中心节点的质量极高的路由,这样就可能使大量节点使用这条路由,但是离攻击者甚远的所有节点发送的分组就会被湮没,从而导致网络处于混乱状态。节点认识到到达攻击者的这条链路是虚假链路后几乎没有什么可选择的处理办法,其所有相邻节点都可能将分组转发给攻击者。那些依靠相邻节点间位置信息交换维护网络拓扑或进行流量控制的协议也易受 Hello 泛洪攻击。

攻击者进行 Hello 泛洪攻击时不必建立合法分组流。攻击者只需采用足够大的发射功率重复广播开销分组,使每个网络节点能够接收到这个广播,也可以认为 Hello 泛洪是单方广播蠕虫。

"泛洪"经常用来表示一条消息在多跳拓扑上迅速传播给每个网络节点。但是 Hello 泛洪攻击采用单跳广播将一条消息发送给大量接收节点,所以两者之间是有差别的。

7) 确认哄骗

有些 WSN 路由协议依靠间接或者直接的链路层应答,由于 WSN 传输媒介的固有广播特性,所以攻击者可以旁听传递给相邻节点的分组,并对其做出链路层哄骗应答。应答哄骗的目的包括使发送节点相信一条质量差的链路是一条质量高的链路、一个失效节点或者被毁节点是一个活动节点。例如,路由协议可以运用链路可靠性选择传输路径的下一个转发跳。在应答哄骗攻击中,攻击者故意强迫使用一条质量差的链路或者一条失效的链路。因为沿着质量差的或失效的链路传递的分组将会丢失,所以攻击者运用应答哄骗能够有效地进行选择性转发攻击,鼓励目标节点在质量差的或失效的链路上发送分组。

4. 传输层攻击

传输层负责管理端到端连接。传输层提供的连接管理服务可以是简单的区域到区域的不可靠任意组播传输,也可以是复杂、高开销的可靠按序多目标字节流。WSN 一般采用简单协议,使应答和重传的通信开销最低。WSN 传输层可能存在两种攻击:泛洪和去同步。

1) 泛洪

要求在连接端点维护状态的传输协议易受泛洪攻击,泛洪攻击会引起传感器节点存储容量被耗尽的问题。攻击者不断反复提出新的连接请求,直到每个连接所需的资源被耗尽或达到连接最大限制条件为止。此后,合法节点的连接请求被忽略。假如攻击者没有无穷资源,那么这是不可能的——攻击者建立新连接的速度快到足以在服务节点上产生资源饥饿问题。

2) 去同步

去同步是指打断一个既存的连接。例如,攻击者反复向一个端主机发送哄骗消息,使这个主机申请重传丢失分组。假如时间同步正确,那么攻击者可以削弱端主机数据交换能力,甚至阻止端主机交换数据,从而导致端主机浪费能量,试图从实际上并不存在的错误中恢复过来。一种对抗措施是要求认证端主机之间通信的所有分组。假定认证方法本身是安全的,那么攻击者就不能给端主机发送哄骗消息。

5.7.2 安全协议

SPINS 安全协议框架采用安全网络加密协议(Secure Network Encryption Protocol, SNEP)提供数据机密性、数据认证、数据完整性、数据新鲜度,采用 μTESLA 协议提供广播认证。

1. SNEP

SNEP 中，通信双方共用两个计数器，分别代表两种数据传输方向。每发送一组数据后，通信双方各自增加计数器。通过共享计数器状态，SNEP 通信开销较低。基于计数器交换协议，通信双方可以进行计数器同步。

SNEP 使用消息认证码提供认证和数据完整性服务，MAC 认证可以保证点到点认证和数据完整性。采用加密认证方式，可以加快接收者认证数据包的速度。接收者在收到数据包后可以马上对加密文件进行认证，发现问题直接丢弃，无须对数据包进行解密。另外，逐跳认证方式只能选择加密认证方式，因为中间节点没有端到端的通信密钥，不能对加密的数据包进行解密。

SNEP 支持数据的新鲜性认证，通过在消息中嵌入计数器值，可以实现提供“弱数据新鲜性”。在 MAC 计算中加入一个随机数就可实现“弱数据新鲜性”保证。“弱数据新鲜性”是指消息中只提供消息块的顺序，不携带延时信息。

2. μTESLA 协议

μTESLA 是为低功耗设备传感器节点专门打造的实现广播认证的微型化 TESLA (Timed Efficient Stream Loss-tolerant Authentication)协议版本。在基站和节点松散同步的假设情况下，基于对称密钥体制，μTESLA 协议通过延迟公开广播认证密钥模拟对称认证。μTESLA 协议实现了基站广播认证数据过程和节点广播认证运算过程。

在基站广播认证数据过程中，基站用一个密钥计算消息认证码。基于松散时间同步，节点知道同步误差上界，因而了解密钥公开时槽，从而知晓特定消息的认证密钥是否已经被公开。如果该密钥未公开，节点可以确信在传输过程中消息不会被篡改。节点缓存消息直至基站广播公开相应密钥。如果节点收到正确密钥，就用该密钥认证缓存中的消息；如果密钥不正确或消息晚于密钥到达，该消息可能被篡改，将会被丢弃。

μTESLA 协议中，基站的消息认证码密钥来自一个单向散列密钥链，单向散列函数 F 公开。首先基站随机选择密钥 K_n 作为密钥链中第一个密钥，重复运用函数 F 产生其他密钥，即

$$K_i = F(K_{i+1}), \quad 0 \leqslant i \leqslant n-1$$

密钥链中每个密钥都关联一个时槽，基站可以根据消息发送消息的相应时槽选择密钥计算消息认证码，如图 5-51 所示。

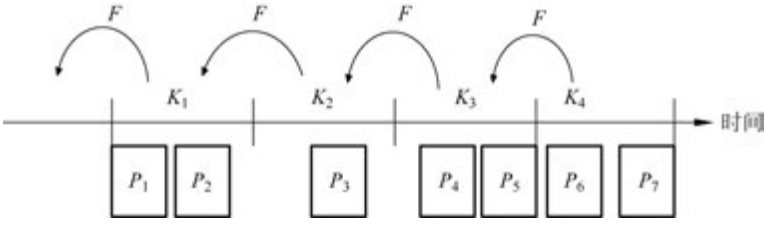


图 5-51 μTESLA 单向密钥链

假设接收者已知 K_0，并与发送方实现松散时间同步，密钥延迟两个时槽后公开。数据包 P_1 和 P_2 在时槽 1 使用密钥 K_1 计算各自的消息认证码。同理，P_3 用 K_3 计算消息认证码。接收者在两个时槽后才能进行验证。假设 P_4、P_5、P_6 丢失，这使接收者不能验证接收到的 P_1 和 P_2。如果接收者收到 P_7，此时接收者可验证 P_3，同时恢复 K_1，如果满足

$K_0=F[F(K_2)]$,则 $K_1=F(K_2)$,这样接收者就可以验证 P_1 和 P_2。

5.7.3 安全管理

安全管理包含安全体系建立(即安全引导)和安全体系变更(即安全维护)两部分。安全体系建立表示一个传感器网络从一堆分立的节点(或者说一个完全裸露的网络),通过一些共有的知识和协议过程,逐渐形成一个具有坚实安全外壳保护的网络。安全体系变更主要是指在实际运行中,原始的安全平衡因为内部或外部的因素被打破,传感器网络识别并去除这些异构的恶意节点,重新恢复安全防护的过程。这种平衡的破坏可能由敌方在某一个范围内进行拥塞攻击形成路由空洞造成。也可能由敌方俘获合法的无线传感器节点造成。还有一种变更的情况是增加新的节点到现有网络中以延续网络生命期的网络变更。

SPINS 安全协议框架对安全管理没有过多的描述,只是假定节点之间以及节点和基站之间的各种安全密钥已经存在。在基本安全外壳已经具备的情况下,如何完成机密性、认证、完整性、新鲜性等安全通信机制,对于传感器网络来说是不够的。试想一个由上万节点组成的传感器网络,随机部署在一个未知的区域内,没有节点知道自己周围的节点会是谁。在这种情况下,要想预先为整个网络设置好所有可能的安全密钥是非常困难的,除非对环境因素和部署过程进行严格控制。

安全管理最核心的问题就是安全密钥的建立过程(Bootstrap)。传统解决密钥协商过程的主要方法有信任服务器分配模型(Center of Authentication,CA)、自增强模型和密钥预分布模型。信任服务器分配模型使用专门的服务器完成节点之间的密钥协商过程,如 Kerberos 协议;自增强模型需要非对称密码学的支持,而非对称密码学的很多算法,如 Diffie-Hellman(DH)密钥协商算法,都无法在计算能力非常有限的传感器网络上实现;密钥预分布模型在系统布置之前完成了大部分的安全基础的建立,对系统运行后的协商工作只需要很简单的协议过程,所以特别适合传感器网络安全引导。对随机密钥模型的各算法,从下面几方面进行评价和比较。

(1) 计算复杂度评价。

(2) 引导过程的安全度评价。

(3) 安全引导成功概率。

(4) 节点被俘后,网络的恢复力评价。

(5) 节点被复制后,或者不合法节点插入现有网络,网络对异构节点的抵抗力。

(6) 支持的网络规模评价。

在介绍安全引导模型之前,首先引入一个概念——安全连通性。安全连通性是相对于通信连通性提出的。通信连通性主要是指在无线通信环境下,各节点与网络之间的数据互通性。安全连通性主要是指网络建立在安全通道上的连通性。在通信连通性的基础上,节点之间进行安全初始化的建立,或者说各节点根据预共享知识建立安全通道。如果建立的安全通道能够把所有节点连接成一个网络,则认为该网络是安全连通的。图 5-52 描述了网络连通和安全连通的关系。

图 5-52 中所有节点是通信连通的,但不全是安全连通的,因为节点 4 以及节点对 9 和 13 无法与它们周围通信的节点建立安全通道。有的安全引导模型从设计之初就同时保证网络的通信连通性和安全连通性,如预共享密钥模型;另外一些安全引导模型则不能同时

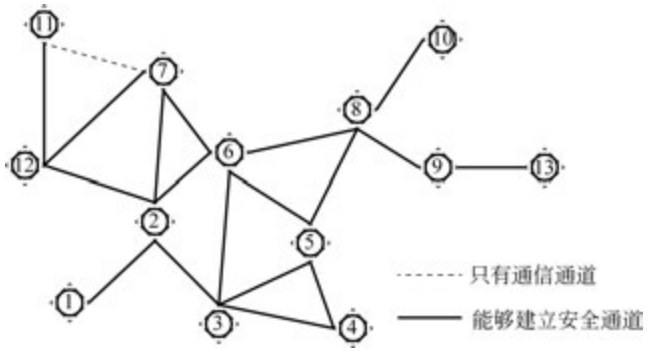


图 5-52 安全连通和网络连通的关系

保证通信连通性和安全连通性。有一点可以确定,安全连通的网络一定是通信连通的,反过来不一定成立。

1. 预共享密钥模型

预共享密钥是最简单的一种密钥建立过程,SPINS 使用的就是这种建立过程。预共享密钥有以下几种主要模式。

(1) 每对节点之间都共享一个主密钥,以保证每个节点之间的通信都可以直接使用这个预共享密钥衍生出来的密钥进行加密。该模式要求每个节点都存放与其他所有节点的共享密钥。这种模式的优点包括:不依赖于基站,计算复杂度低,引导成功率为 100%;任何两个节点之间的密钥是独享的,其他节点不知道,所以一个节点被俘,不会泄露非直接建立的任何安全通道。但这种模式的缺点也很多:扩展性不好,无法加入新的节点,除非重建网络对复制节点没有任何防御力;网络的免疫力很低,一旦有节点被俘,敌人将很容易通过该节点获得与所有节点之间的秘密并通过这些秘密攻破整个网络;支持的网络规模小,假设节点之间使用 64 位主共享密钥(8 字节),那么 1000 个节点规模的网络就需要每个节点有 8 字节的主密钥存储空间,如果考虑各种衍生密钥的存储,整个用于密钥存储的空间就是一个非常庞大的数字(一个合理的网络规模在几十个到上百个节点)。

(2) 每个普通节点与基站之间共享一对主密钥,参考 SPINS 协议框架描述。这样,每个节点需要存储的密钥空间将非常小,计算和存储的压力全部集中在基站上。该模式的优点包括:计算复杂度低,对普通节点资源和计算能力要求不高;引导成功率高,只要节点都能够连接到基站就能够进行安全通信;支持的网络规模取决于基站的能力,可以支持上千个节点;对于异构节点基站可以进行识别,并及时将其排除在网络之外。这种模式的缺点包括:过分依赖基站,如果有节点被俘,会暴露与基站的共享密钥,若基站被俘则整个网络被攻破,所以要求基站被布置在物理安全的位置;整个网络的通信或多或少都要通过基站,基站可能成为通信瓶颈;如果基站可以动态更新,网络能够扩展新的节点,否则将无法扩展。这种模式对于收集型的网络比较有效,因为所有节点都与基站(Sink 节点)直接联系;而对于协同型的网络,如目标跟踪型的应用网络,效率会比较低。在协同型网络的应用中,数据要安全地在各节点之间通信,一种方法是通过基站,但会造成数据拥塞;另一种方法是通过基站建立点到点的安全通道。在通信对象变化不大的情况下,建立点到点安全通道的方式还能够正常运行;如果通信对象频繁切换,安全通道的建立过程也会严重影响网络运行效率。最后一个问题就是在多跳网络环境下,这种协议对于 DoS 攻击没有任何防御能

力。在节点与基站之间通信的过程中,中间转发节点没有办法对信息包进行任何的认证判断,只能透明转发。恶意节点可以利用这一点伪造各种错误数据包发送给基站,因为中间节点透明传输,数据包只能在到达基站以后才能够被识别出来。基站会因此而不能提供正常的服务,这是相当危险的。

预共享密钥引导模型虽然有很多不尽如人意的地方,但因其实现简单,所以在一些网络规模不大的应用中可以得到有效实施。

2. 随机密钥预分配模型

1) 基本随机密钥预分配方案

随机密钥预分配方案是由 Eschenauer 和 Gligor 最早提出的,它的主要思想是根据经典的随机图理论,控制节点间共享密钥的概率,并在节点被捕获之后撤销节点的密钥链,以及更新节点间的共享密钥。随机密钥预分配方案的具体实施过程如下。

(1) 密钥预分配阶段。部署前,部署服务器首先生成一个密钥总数为 S 的大密钥池及密钥标识,每个节点从密钥池里随机选取 K 个不同密钥以及密钥对应的标识存入节点的存储器内,K 个密钥称为节点的密钥链,K 的选择应保证每两个节点间至少拥有一个共享密钥的概率大于预先设定的概率。

(2) 共享密钥发现阶段。随机部署后,每个节点都广播自己密钥链中所有密钥标识,周围的邻居节点收到信息后查看自己的密钥链,如有相同密钥标识,则存在共享密钥,就随机选取其中的一个作为双方的对密钥(Pairwise Key)。否则,进入密钥路径建立阶段。

(3) 密钥路径建立阶段。当节点与邻居节点没有共享密钥时,节点通过与其他存在共享密钥的邻居节点经过若干跳后建立双方的一条密钥路径。

(4) 当检测到一个节点被捕获时,为了保证网络中其他未被捕获节点之间的通信安全,必须删除被捕获节点密钥链中的密钥。因此,控制节点广播被捕获节点密钥链中的所有密钥标识,其他节点收到信息后删除自己密钥链中含有相同密钥标识的对应密钥,与删除密钥相关的密钥连接将会消失,因此受影响的节点需要重新进入共享密钥发现阶段或密钥路径建立阶段。

随机密钥预分配方案存在着一个概率问题,即有可能存在一些节点与周围邻居节点没有共享密钥,也没有密钥路径,所以不能保证网络的密钥连接性。影响网络的密钥连接性的因素有网络的部署密度、目标区域的状况、密钥池的大小 S 以及节点密钥链的大小 k。k/S 越大,邻居节点之间存在共享密钥的概率就越大。但 k/S 太大会导致网络安全变得脆弱,因为 k 太大会占用节点的太多资源,S 太小会容易让攻击者通过捕获少量节点就获得大部分密钥池中密钥,进而危及网络安全。

根据经典的随机图理论,节点的度 d 与网络节点总数 n 存在以下关系。

$$d=\frac{n-1}{n}[\ln(n)-\ln(-\ln p_c)] \tag{5-41}$$

其中,p_c 为全网连通概率。对于一个给定密度的无线传感器网络,假设 n' 为节点通信半径内邻居节点个数的期望值,则相邻两个节点共享一个密钥的概率 p' 为

$$p'=d/(n'-1) \tag{5-42}$$

该方案的优点:①节点仅存储少量密钥就可以使网络获得较高的安全连通概率,计算复杂度低;②密钥预分配时不需要节点的任何先验信息,如节点的位置信息和连通关系等;

③密钥管理具有良好的分布特性。

2）q-Composite 随机密钥预分配方案

Chan 等在 Eschenauer 和 Gligor 方案的基础上，提出了 q-Composite 随机密钥预分配方案，该方案要求相邻节点间至少有 q 个共享密钥，通过提高 q 值提高网络的抗毁性。q-Composite 随机密钥预分配方案的具体实施过程如下。

（1）密钥预分配阶段。部署服务器首先生成一个密钥总数为 S 的大密钥池及密钥标识，每个节点从密钥池里随机选取 k 个不同密钥以及密钥对应的标识存入节点的存储器内，k 个密钥称为节点的密钥链，k 的选择应保证每两个节点间至少拥有 q 个共享密钥的概率大于预先设定的概率。

（2）共享密钥发现阶段。与随机密钥预分配方案类似，节点广播自己密钥链中的密钥标识，找出位于自己通信半径内与自己有共享密钥的节点。

（3）共享密钥发现完成后。每个节点确定与自己邻居节点有 t 个共享密钥，$t>q$，则可以使用单向散列函数建立通信密钥 $K=\mathrm{hash}(k_1||k_2||\cdots||k_t)$。

该方案的网络连通性概率也是基于概率论和随机图理论计算的。

$$p(i)=\frac{\binom{S}{i}\binom{S-i}{2(k-i)}\binom{2(k-i)}{k-i}}{\binom{S}{k}^2} \tag{5-43}$$

其中，$p(i)$为从 S 个密钥中抽取 k 个预存储给节点时，两个邻居节点有 i 个公共密钥的概率。根据全概率公式，任意两个相邻节点能够直接建立共享密钥的概率为

$$p=1-(p(0)+p(1)+\cdots+p(q-1)) \tag{5-44}$$

网络中所有节点都从同一个密钥池中抽取密钥，所以未被捕获的节点间可能使用被捕获节点泄露的密钥通信，这对网络安全构成重大威胁。使用量化指标“x 个节点被捕获时，一对未被捕获的节点间共享密钥泄露的概率”评估方案的抗攻击能力，该值也等价于“x 个节点被捕获时，剩余网络不安全部分的比例”。在 Eschenauer 和 Gligor 的随机密钥预分配方案中，每个节点携带任意一个预分发密钥的概率为 k/S，x 个节点被捕获时，任意一对未被捕获的节点间共享密钥泄露的概率为

$$P_{\text{compromised}}=1-(1-k/S)^x \tag{5-45}$$

在该方案中，抗攻击能力计算与 Eschenauer 和 Gligor 提出的方案类似，但不同在于要考虑 $k-q+1$ 种可能性。由全概率公式可知，x 个节点被捕获时，任意一对未被捕获的节点间共享密钥泄露的概率为

$$P_{\text{compromised}}=\sum_{i=q}^{k}[1-(1-k/S)^x]^i\frac{p(i)}{p} \tag{5-46}$$

该方案的优点是相较随机密钥预分配方案，网络抗毁性比较好，少量节点被捕获不会影响网络中其他节点间的通信。缺点是要想网络中相邻节点间至少有 k 个共享密钥的概率达到预先设定的概率，就必须缩小整个密钥池，增大节点间共享密钥的重叠度，从而限制了网络的可扩展性，攻击者捕获少量节点就能获得密钥池中大部分密钥。

3. 基于分簇式的密钥管理

1）低能耗密钥管理方案

低能耗密钥管理方案是由 Jolly 等提出的，它假定基站有入侵检测机制，可以检测出恶意节点，并能触发删除节点的操作。但是，对传感器节点不作任何信任的假设，簇头之间可

以通过广播或单播与节点进行通信。该方案具体实施步骤如下。

(1) 预分配阶段。每个节点预先存储两个密钥和两个ID标识符,其中一个密钥是与某个簇头共享的,另一个密钥是与基站共享的,两个ID标识符分别表示该簇头和节点自身的ID。由于节点是不可信任的,且节点的存储空间有限,所以在节点存储少量密钥不仅可以节省节点的存储空间,也能提高网络安全性。另外,所有簇头共享一个密钥用于簇头间的广播通信,且每个簇头还分配一个与基站共享的密钥和随机选择$|S|/|G|$个传感器节点的密钥,其中$|S|$表示传感器节点的个数,$|G|$表示簇头的个数,而基站则要存储所有密钥。

(2) 初始化阶段。

① 首先,节点广播自己的信息,格式如下。

$$S_i \rightarrow *:\mathrm{ID}_{G_j} \| \mathrm{ID}_{S_i} \| K_{S_i,G_j}(\mathrm{nonce} \| \mathrm{sdata})$$

② 簇头接收到信息后,找出与自己没有节点密钥的所有节点的ID标识符,并在簇头间广播,格式如下。

$$G_i \rightarrow G:\mathrm{ID}_{G_j} \| K_G(\mathrm{nonce} \| \{\mathrm{ID}\})$$

③ 每个簇头接收到信息后,在自己存储的密钥中查找与信息中节点标识符对应的密钥,然后将密钥信息发送回源簇头,格式如下。

$$G_j \leftarrow G_k:K_{G_j,G_k}(\mathrm{nonce} \| (\mathrm{ID}_{S_i},K_{S_i,G_k}))$$

④ 最后,簇头发送信息给节点,指定节点的归属,格式如下。

$$S_i \leftarrow G_j:\mathrm{ID}_{G_j} \| K_{S_i,G_k}(\mathrm{nonce} \| \mathrm{ID}_{G_j} \| \mathrm{msg})$$

(3) 加入新节点阶段。

加入新节点时,基站首先随机选择一个簇头,将新节点的密钥发送到该簇头,格式如下。

$$C \rightarrow G_h:K_{C,G_h}(\mathrm{nonce} \| \mathrm{ID}_{S_j} \| K_{S_i,G_h})$$

然后通过初始化阶段,新节点就可以加入网络。

该方案的优点是节点只要预存两个密钥和两个ID标识符,对节点的存储空间要求不高,且计算复杂度低,网络的抗毁能力强。缺点是网络的扩展性差,通信依赖于簇头,如果多个相邻簇头被捕获,则整个网络就会瘫痪。而且,当簇头被捕获时,该方案重新指定一个新的簇头代替旧的簇头,然后把该簇内的所有节点都分配给新簇头。但是,这在实际应用中是不可行的,因为不能保证新簇头正好部署在旧簇头的位置上,所以也就不能保证新簇头能包含所有旧簇内的节点。

2) LEAP密钥管理方案

2003年Zhu等提出的LEAP(Localized Encryption and Authentication Protocol)是一个既能支持网内处理(In-Network Processing),又具有较好抗捕获性的密钥管理协议,这种协议支持4类密钥的生成和管理,提供了较好的低能耗的密钥建立和更新方案,同时还提供了基于单向密钥链的网内节点认证方案,并在不丢失网内处理功能和被动参与(Passive Participation)的情况下支持源认证操作。

Zhu等认为应该在网络节点中设立多种密钥以适应不同的需要,因此在LEAP中建立了4种类型的密钥:个体密钥(Personal Key)、对密钥(Pairwise Key)、簇密钥(Cluster Key)、组密钥(Group Key)。每种密钥都有不同的作用,个体密钥、对密钥、簇密钥、组密钥建立过程的具体步骤如下。

(1) 个体密钥。个体密钥为节点与基站所共享的密钥,由节点在部署前通过预分配的主

密钥 K^{m} 和伪随机函数 f 来生成，用于节点向基站发送秘密信息，节点 u 的个体密钥如下。

$$K_u^{m}=f_{K^{m}}(u) \tag{5-47}$$

(2) 对密钥。对密钥是相邻节点间单独共享的密钥，用于节点间单独交换秘密信息，是通过交换其标识符及使用预分配的主密钥和单向散列函数计算得到的，具体产生步骤如下。

① 密钥预分配。管理节点产生一个初始化密钥，每个节点预存，并按式(5-48)计算出节点自身的主密钥。

$$K_u=f_{K_I}(u) \tag{5-48}$$

② 邻居发现。部署后，节点广播自己的标识符 ID，邻居节点接收到信息后回复源节点，格式如下。

$$u \rightarrow *:u$$
$$v \rightarrow u:v,\mathrm{MAC}(K_v,u \mid v)$$

③ 对密钥建立。节点收到邻居节点的回复后就可以计算对密钥：

$$K_{uv}=\begin{cases} f_{K_v}(u), & u<v \\ f_{K_v}(v), & u \geqslant v \end{cases} \tag{5-49}$$

④ 撤销密钥。对密钥建立周期过后，每个节点都撤销密钥。

(3) 簇密钥。簇密钥为同一簇内相邻节点所共享，由簇头产生一个随机密钥作为簇密钥，然后使用与邻居节点的对密钥逐一地把簇密钥加密后发送给邻居节点，邻居节点把簇密钥解密后保存下来。

(4) 组密钥。组密钥为基站与所有节点共享的通信密钥，基站首先把组密钥使用与其子节点共享的簇密钥加密后广播给子节点，子节点获取最新的组密钥后，与其下一级子节点共享的簇密钥加密组密钥后广播给其子节点。以此类推，直到所有节点都获取最新的组密钥为止。

LEAP 方案的优点是任何节点的受损都不会影响其他节点的安全。缺点是节点部署后，在一个特定的时间内必须保留全网通用的主密钥。主密钥一旦被暴露，整个网络的安全都受到威胁。此外，在对密钥生成阶段，因为只有单向认证，因此还存在 Hello 攻击，即当攻击者 f 假冒除 v 外的网络中任何节点向节点 v 广播协商请求时，按照协议节点 v 将生成对所有节点的对密钥。

4. 基于本地协作的组密钥分发方案

基于本地协作的组密钥分发方案(Group Key Distribution via Local Collaboration)的基本思想是：网络生存时间被划分为许多时间间隔，称为会话(Sessions)，每次会话阶段由基站发起组密钥更新；组密钥更新时，基站向全组进行广播，合法节点可以通过预置的密钥信息和广播消息包获得一个私有密钥信息，节点通过和一定数目的邻居节点进行协作利用私有密钥信息计算获得新的组密钥。一个会话阶段的组密钥更新过程如下。

1) 初始化

基站随机选择一个度数为 $2t$ 的隐藏多项式 $h(x)=a_0+a_1x+\cdots+a_{2t}x^{2t}$ 和一个度数为 t 的加密多项式 $l(x)$，并为每个节点 i 预置密钥信息 $h(i)$ 和 $l(i)$。

2) 广播组密钥信息

集合 $R=\{r_i\}$，$|R|=w\leqslant t$ 代表基站知道的被捕获节点的个数。基站向外广播的消息

包 $B=\{R\}\cup\{w(x)=g(x)f(x)+h(x)\}$。其中,$f(x)$为度数为 t 的私有密钥多项式,$g(x)=(x-r_1)(x-r_2)\cdots(x-r_w)$ 为剔除多项式。

3) 获得私有密钥

当合法节点 i 收到广播信息包 B 后,将其节点 ID 代入广播多项式 $w(x)$,能够计算出其私有密钥 $f(x)=[w(i)-h(i)]/g(i)$。相反,任意被捕获节点 j 均不能获得私有密钥,因为 $g(x)=0$,将 ID 代入 $w(x)$后只能得到其本身存储的密钥信息 $h(x)$。

4) 本地协作

为了获得新的组密钥,节点需要同至少 t 个邻居节点进行协作才能获得新的组密钥。节点向其邻居节点广播私有密钥请求,邻居节点收到请求信息后,如果信任该节点,则将其加密后的私有密钥 $s(i)=f(i)+l(i)$发送给请求节点。

5) 生成组密钥

当节点获得至少 t 个节点的私有密钥后,加上其自身存储的私有密钥,节点可获得 $t+1$ 个加密后的私有密钥。节点利用这 $t+1$ 个信息通过拉格朗日插值,可获得一个组密钥多项式 $s(x)=f(x)+l(x)$。从而,节点可计算出新的组密钥 $K=s(0)$。

该方案使得只有组中的合法节点才能获得私有密钥 $f(i)$,以及只有被一定数目的邻居节点信任的节点才能够通过本地协作的方式获得新的组密钥。网络有多个会话阶段,节点需要存储所有会话阶段的 $h_j(i)$和一个固定的 $l_j(i)$,$h_j(i)$和 $l_j(i)$代表第 j 次会话用到的密钥信息。

该方案的优点是:实现比较简单,只需在节点部署之前给每个传感器节点预置所有会话阶段的密钥信息即可进行组密钥更新;安全性较好,能很好地抵制部分节点被捕获时对其他节点安全通信造成的影响;支持网络的动态变化,加入节点只需预置目前传感器网络所处阶段的组密钥及之后阶段的密钥信息即可参与网络协同操作。缺点是:存在孤立节点以及计算开销较大的问题;存储开销较大,因为每个节点需要存储所有阶段的密钥信息,而网络的会话次数一般很大;通信开销较大,组密钥更新通过广播方式,同时每个节点需要和一定数目的邻居节点进行本地协作。

习题 5

1. 典型的时钟由一个稳定的石英振荡器和_____组成,这个计数器随着每次石英晶体的振荡递减。

2. 同步消息的延时包含 4 部分:_____、_____、传播延时、接收延时。

3. 在无线传感器网络中,需要定位的节点称为未知节点(Unknown Node),即不知道自身位置的节点,在一些资料中也称为_____;而已知位置并协助未知节点定位的节点称为_____,在一些资料中也称为参考节点、信标节点。

4. 定位计算的基本方法包括_____、_____、极大似然估计法、最小最大法。

5. 距离无关的定位算法有_____、凸规划定位算法等。

6. 典型室内定位系统有_____、Active Office 等。

7. 根据处理融合信息方法的不同,数据融合系统可分为_____、_____和_____3 种。

8. 根据融合处理的数据种类，数据融合系统可分为_____、_____和时空融合。

9. 检测和决策理论是通过把被测对象的测量值与被选假设进行比较，以确定哪个假设能最佳地描述观测值，代表方法为_____、_____、马尔可夫随机域理论等。

10. 无线传感器网内数据融合主要有两个融合层次，即_____和应用级。

11. 数据包级的融合操作有两种方法：_____和无损的。

12. 数据融合算法中应用较多且比较典型的融合方法包括_____、_____、贝叶斯估计、D-S 证据推理、模糊逻辑、统计决策理论、产生式规则、聚类分析法、神经网络等。

13. 目前人们采用的节能策略主要有_____和数据融合等，它们应用在计算单元和通信单元的各环节。

14. 容错领域有几个基本概念：_____、_____、差错(Error)。

15. 无线传感器网络容错设计需要考虑三方面：_____、故障检测与诊断、修复机制。

16. 故障检测分为_____和节点检测。

17. 在网络 QoS 研究中，人们比较关注的服务质量标准主要包括可用性、吞吐量、_____、_____和丢包率等几个参数。

18. 传感器网络实现时间同步的作用是什么？

19. 传感器网络常见的时间同步机制有哪些？

20. 简述 TPSN 时间同步协议的设计过程。

21. 传感器网络定位问题的含义是什么？

22. 如何对传感器网络的定位方法进行分类？

23. 简述以下概念术语的含义：锚点、测距、连接度、到达时间差、接收信号强度指示、视线关系。

24. 如何定义传感器网络的平均定位误差？

25. 如何评价一种传感器网络定位系统的性能？

26. RSSI 测距的原理是什么？

27. 简述 ToA 测距的原理。

28. 举例说明 TDoA 的测距过程。

29. 举例说明 AoA 测角的过程。

30. 试描述传感器网络多边定位法的原理。

31. 简述最小最大法定位的原理。

32. 简述质心定位算法的原理及其特点。

33. 举例说明 DV-Hop 算法的定位实现过程。

34. 什么是数据融合技术？它在传感器网络中的主要作用是什么？

35. 简述数据融合技术的不同分类方法及其类型。

36. 什么是数据融合的综合平均法？

37. 常见的数据融合方法有哪些？

38. 无线通信的能量消耗与距离的关系是什么？它反映出传感器网络数据传输的什么特点？

39. 简述节能策略休眠机制的实现思想。

40. 简述传感器网络节点各单元能量消耗的特点。
41. 动态电源管理的工作原理是什么?
42. 传感器网络的安全性需求包括哪些内容?
43. 什么是传感器网络的信息安全?
44. 简述在传感器网络中实施蠕虫攻击的原理过程。
45. SPINS 安全协议框架能提供哪些功能?
46. 如何选择传感器网络安全协议的加密算法?
47. 如何设计安全协议的随机数发生器?