



视频

网络层是位于 MAC 层之上与应用层交互的一个协议层。网络层的主要功能是设备的发现和配置、网络的建立与维护、路由的选择以及广播通信,并具有自我组网与自我修复功能。路由是指通过网络把信息从源(source)传递到目的地的行为,是网络层的基本问题。路由技术由两项最基本的活动组成,即决定最优路径和传输数据包。在各种路由协议的设计中,数据包的传输路径选择比较复杂,路径选择优化是路由设计的主要任务,而数据交换相对比较简单。按照不同依据,路由算法可以分为很多种,例如静态与动态、平面与分层、单路与多路等。本章重点讨论平面路由协议和好的路由协议。

## 5.1 路由协议概述



视频

### 5.1.1 网络层的服务实体

网络层位于 MAC 层和应用层之间,为了与应用层交互,网络层逻辑上包含两个服务实体:数据服务实体(NLDE)和管理服务实体(NLME)。NLDE-SAP 是网络层提供给应用层的数据服务接口,用于将应用层提供的数据打包成应用层协议数据单元,并将其传输给相应节点的网络层;或者将接收到的应用层协议数据单元进行解包,并将解包后得到的数据传送给本节点的应用层,也就是说,NLDE-SAP 实现两个应用层之间的数据传输。NLDE-SAP 的主要任务如下:

- (1) 发起一个网络并且分配网络地址(网络协调器)。
- (2) 向网络中添加设备或者从网络中移除设备。
- (3) 将消息路由到目的节点。
- (4) 对发送的数据进行加密。
- (5) 在网状网络中执行路由寻址并存储路由表。

网络层帧即网络协议数据单元(NPDU),由两个部分组成: NWK 头和 NWK 有效载荷。NWK 头部分包含帧控制、地址和序号信息; NWK 有效载荷部分包含的信息因帧类型的不同而不同,长度可变。NWK 帧结构见表 5-1-1。

表 5-1-1 NWK 帧结构

2B	2B	2B	1B	1B	可变长度
帧控制域	目的地址	源地址	半径域	序号	帧有效载荷
	路由域				
NWK 头					NWK 有效载荷

网络层定义了两种类型的设备：全功能设备(Full Function Device, FFD)和简化功能设备(Reduced Function Device, RFD)。FFD作为网络的协调器,支持各种拓扑结构的网络的建立,也可以有效地降低成本和功耗。

网络层支持的网络拓扑结构有3种:星状(star)结构、树状(tree)结构和网状(mesh)结构,如图5-1-1所示。

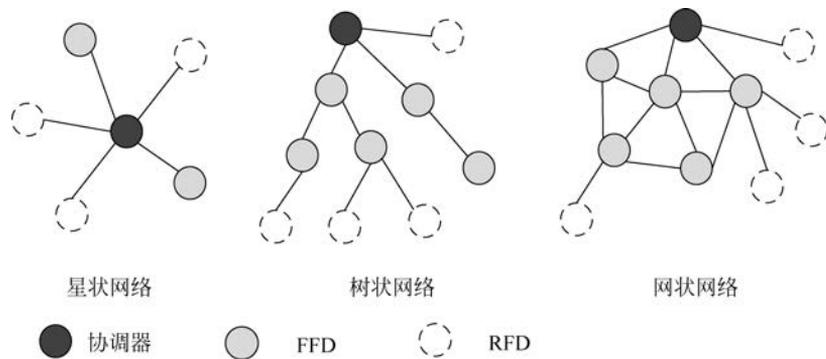


图 5-1-1 网络拓扑结构

(1) 星状网络为主从结构,由单个网络协调器和多个终端设备组成,网络的协调者必须是 FFD,由它负责管理和维护网络。

(2) 树形网络可以看成是扩展的单个星状网或者相当于互联的多个星状网络。

(3) 网状网络中的每一个 FFD 还可以作为路由器,根据网络路由协议来优化最短和最可靠的路径。

### 5.1.2 路由协议的基本问题

路由协议主要负责路由选择和数据转发。路由选择是指寻找一条符合一定条件的路径作为从源节点到目的节点的传输路径,数据转发是指将数据分组沿着选择的传输路径进行转发。

在对无线传感器网络节点进行数据交换的路径选择中,每个节点维护一个路由表,路由算法初始化并维护包含路径信息的路由表。路由算法根据目的地址和下一跳地址等诸多信息填充路由表。节点在通信过程中通过交换路由信息各自维护自身的路由表,路由更新信息通常包含全部或者部分路由表信息,通过分析来自其他节点的路由更新信息来进行优化分析,选择最优路径。每个节点可以建立一个网络拓扑图,节点之间还发送链接状态广播信息来通知其他节点数据源的链路状态,根据链接信息来构建完整可靠的网络拓扑图,使节点路由算法能够选择最优路径。

无线传感器网络中的数据交换相对比较简单,而且各种不同的路由算法具有相同的数据交换算法。一般而言,当源节点向目的节点发送数据,并通过某些方法获得中继节点的地址以后,源节点向中继节点发送数据包,该数据包中的 IP 地址指向目的节点。中继节点查看了数据包的目的节点地址后,确定是否知道如何转发该包;如果不能知道,则丢弃该包;如果知道,则按照路由表信息进行转发,把目的物理地址变为下一条的物理地址并向之发送。下一条可能是最终的目的节点,也有可能是下一级中继节点,当数据包在网络中流动



视频

时,其物理地址在改变,但是其 IP 地址始终不变。

网络数据传输离不开路由协议,但是,传统无线自组织网络的路由协议却不能适用于无线传感器网络。

(1) 无线传感器网络是以数据为中心进行路由的,不同于无线自组织网络的点对点通信模式。

(2) 无线传感器网络随应用需求而变化,所以无线传感器网络路由协议是基于特定应用进行设计的,很难设计通用性强的路由协议。

(3) 无线传感器网络邻居节点间采集的数据具有相似性,存在冗余信息,需经数据融合(Data Fusion)处理后再进行路由。

(4) 传统网络(包括有线和无线)的每一个节点具有唯一的标识号(ID)。而无线传感器网络是基于属性进行寻址的(Attribute-Based Addressing),不需给每一个节点分配唯一的地址。

(5) 由于无线传感器网络节点能量有限,所以路由设计一般将能效高放在首位,将服务质量(QoS)放在第三位考虑。

因此,设计符合无线传感器网络自身特点的路由协议时必须考虑以下问题。

(1) 低功耗: 传感器节点一般采用电池供电,通常部署在恶劣环境下,很难更换电池。

(2) 可靠性: 传感器节点之间通过无线信道传输数据容易出错,必须提供可靠的差错控制和校正机制来确保数据在交付时正确无误。

(3) 自组织性: 传感器节点在部署后自动组织成网络,当发生故障的节点需要退出或补充新的节点时,网络也能适应拓扑结构变化正常运行。

(4) 信道利用率: 无线传感器网络带宽资源有限,应该有效地利用带宽,提高信道利用率。

(5) 容错性: 传感器节点由于无人操作和部署环境的恶劣容易发生故障,所以无线传感器网络应该具有一定的容错能力,允许节点进行自我测试和自我修复。

(6) 安全性: 无线的传输环境很容易被恶意攻击,无人看管的节点很容易被窃取数据,所以无线传感器网络应该提供有效的安全机制确保信息安全。

(7) QoS 保障: 在无线传感器网络中,针对不同的应用需求往往要提供不同的服务质量。



视频

### 5.1.3 路由的过程

无线传感器网络的路由过程主要分为以下 4 个步骤:

(1) 某一个设备发出路由请求命令帧,启动路由发现过程。

(2) 对应的接收设备收到该命令后,回复应答命令帧。

(3) 对潜在的各条路径开销(跳转次数、延迟时间)进行评估比较。

(4) 将评估确定之后的最佳路由记录添加到此路径上各个设备的路由表中。

网络中的各个节点都会保持一个路由表,该表由目的(节点)和下一跳地址所组成。对于某一个节点来说,当它收到一个数据分组时,该节点将检查该分组的地址并将此地址与路由表中的目的地址相匹配,找出下一跳地址,并将此分组转发给对应的节点。路由器之间会相互通信通过交换路由信息维护其路由表,路由更新信息通常包含全部或部分路由表,通过分析其他路由器的更新信息,此路由器可以建立网络拓扑图。

### 1. 初始化路由查找

无论网络层接收到的帧是来自应用层,还是来自 MAC 子层,若帧的目的地址不等于当前设备地址或广播地址,则启动路由查找过程节点将发布路径请求命令帧,而每个发布路径请求命令帧的设备都保留有一个计数器,该计数器用于产生路径请求标识符(ID)。当建立了一个新的路径请求命令帧时,路径请求计数器被装载。其值存储在路径查找表的路径请求标识符域。同时装载的还有一个路径请求计时器。该计时器限定了查找路径可用的时间,当计时器终止时,设备会删除路径查找表中相关的记录。

### 2. 建立虚拟路径

在接收到路径请求命令帧后,设备首先要判定自己还有没有路由容量。接着,它将对接收到的帧进行检查,看其是否沿着有效的路径传输。如果帧由设备的子节点发送后,系统判定路径有效,那么设备会进一步判定是否设备本身或其他设备的子节点就是路径请求命令帧的目的地,如果是,则设备会对路径查找表进行搜索,查找与路径请求标识符(ID)及源地址一致的记录项;若记录项不存在,路径查找表会为之创建一个新的记录项;否则,设备将比较路径请求命令帧中的路径损耗值与查找表中的向前损耗域值,选择更合适的路径。

如果设备不是路径请求命令帧的目的地,那么它将搜索路径查找表,查找与路径请求 ID 及源地址域一致的记录。当路径请求计时器终止时,设备将删除查找表中的路径请求记录项,若路由表中相关记录项的状态仍为查找进行中,则该项将被删除。如果查找表中原来就含有与目的地址相对应的记录,则该记录中的前向损耗值会与路径请求命令帧中的路径损耗值进行比较。若路径损耗大于前向损耗,则路径请求命令会被抛弃;否则,记录中的前向损耗及上一级发送地址将被请求命令帧中的新路径损耗及前一级设备地址所更新代替。注意,此时的新路径损耗应为路径请求命令帧中的路径损耗与前一级链接损耗的总和。

### 3. 确认路径

接收到路径应答帧后,设备将首先检查是否自身就是路径应答帧的目的地。如果是,那么设备将搜索它的路径查找表,查找与应答帧的路径请求 ID 相对应的记录;接着搜索它的路由表,查找与应答帧的回应地址相一致的记录。若路由表与路径查找表中都存在与应答相关的记录,且路由表中相关记录的状态域为查找进行中,则该状态将被重置为激活,且路由表中相关记录的下一跳地址即为传输路径应答帧的前一级设备。路径查找表中相关记录的后向损耗值即为路径应答帧中的路径损耗。

若接收到应答帧的设备并不是应答帧的目的地,那么设备将搜索路径查找表,查找与应答帧的源设备地址及路径请求 ID 相一致的记录项。若查找失败,则路径应答帧会被抛弃,否则应答帧中的路径损耗值将与查找表中相关记录的后向损耗值进行比较。当后向损耗小于路径损耗值时,路径应答帧被抛弃;当后向损耗大于路径损耗时,设备将接着在路由表中查找与应答帧中的回应地址相对应的记录项。若路由查找表中存在相关项,而路由表中不存在,则错误产生,应答帧被抛弃;否则,路由表中相关项的下一跳地址会被传输应答帧的前一级设备地址所更新,而路径查找表中相关项的后向损耗域也会被应答帧中的路径损耗所更新,对相关记录更新完毕后,设备会继续传输路径应答帧。

设备可通过搜索路径查找表中相关项的前一级发送地址域,来获得通往应答目的地的下一跳地址。同时,可用下一跳地址来计算链接损耗,该损耗加入应答帧的路径损耗域并更新该域,应答帧可通过数据请求原语发送到下一地址,原语中的目的地址参数为路由查找表

中的下一跳地址。



视频

### 5.1.4 传感器网络路由的评价标准

对于给定的传感器路由协议,需要从多个方面来评价它的优劣,既要考虑应用对路由提供的服务质量的要求,又要考虑传感器网络、节点特性带来的要求。下面列举一些常用的评价标准。

(1) 能耗: 传感器节点能量受限,路由协议应当以能量高效的方式实现数据传输。能量高效包括两个方面的要求: 首先,路由协议本身消耗的能量不能过大; 另外,路由需要从整个网络的角度考虑,均衡网络的能量消耗。

(2) 鲁棒性: 路由机制针对网络拓扑结构变化需要具备一定的容错能力。在实际的部署环境中,由于能量耗尽或环境因素,传感器节点可能提前死亡; 由于节点加入、移动或链路变化,导致链路稳定性差、时有时无。设计路由协议时,需要考虑这些因素造成的网络拓扑动态变化,提供稳定的路由服务。

(3) 快速收敛性: 由于传感器网络的拓扑结构动态变化,传感器节点能量和通信带宽等资源有限,要求路由机制要能够快速收敛,以适应网络拓扑的动态变化,减少通信开销,提高消息传输的效率。

(4) 服务质量要求: 传感器网络路由面向应用设计,因此要满足应用本身的服务质量要求,如端到端延迟、吞吐率、可靠性等方面; 网络状态动态变化,因此要求路由协议能够在适应这种变化的同时,保障服务质量。

(5) 可扩展性: 传感器网络路由应当能适应网络规模、密度的变化。在不同的应用中,监测区域范围和节点部署密度不同,要求路由机制具有可扩展性,能够适应网络结构的变化。为进一步减少通信开销,路由协议还可以支持网内处理来减少传输数据量。



视频

### 5.1.5 路由协议分类

由于无线传感器网络路由协议是面向应用的,因此对于不同的网络应用环境,研究者提出了大量的路由协议。下面对路由协议进行分类整理:

(1) 根据拓扑结构,路由协议可分为平面路由协议和分层路由协议。平面路由协议一般节点对等、功能相同、结构简单、维护容易,但是它仅适合规模小的网络,不能对网络资源进行优化管理,而分层路由协议中节点功能不同,各司其职,网络的扩展性好,适合较大规模的网络。

(2) 根据路径的多少,路由协议可分为单路径路由协议和多路径路由协议。单路径路由协议是将数据沿一条路径传递,数据通道少、消耗低,容易造成丢包且错误率高。多路径路由协议是将单个数据分成若干组,沿多条路径进行传递,即便有一条路径报废,数据也会经由其他路径传递,可靠性较好,但重复率高、能量消耗大,适合对传输可靠性要求较高且初始能量高的应用场合。

(3) 根据路由模式,路由协议可分为时钟驱动型、事件驱动型和查询驱动型。时钟驱动型是传感器节点周期性地、主动地把采集到的数据信息报告给汇聚节点,如环境监测类的无线传感器网络。事件驱动型是传感器节点感应到数据后进行判断,若超过事先设定的阈值,则认为触发了某种事件,需要立即传送数据给汇聚节点,如用于预警的无线传感器网络。在查询驱动型路由协议中,仅当传感器节点收到用户感兴趣的查询时,传感器节点才向汇聚节

点发送数据。

(4) 根据目的节点的个数,路由协议可分为单播路由协议和多播路由协议。单播路由协议只有一个发送方和一个目的节点。多播路由有多个目的节点,节点采集到的数据信息并行地以多播树方式进行传播,在树的分叉处复制和转发数据包。多播路由协议数据包发送次数变少,网络带宽的使用效率提高。

(5) 根据是否进行数据融合,路由协议可分为融合路由协议和非融合路由协议。如果在数据传输过程中,根据预先制定的规则对多个数据包的相关信息合并和压缩,就属于数据融合路由协议,这类协议降低了数据冗余度,减少了网络通信量,节省了能量消耗,但是相应地增加了传输的时延。非融合路由协议在传递过程中,不做任何处理,传递量大,消耗能量大,甚至引起“拥堵”。

## 5.2 平面路由协议

### 5.2.1 Flooding 协议和 Gossiping 协议

Flooding(泛洪)协议是一种传统的网络路由协议,是网络中各节点不需要掌握网络拓扑结构和计算的路由算法。节点接收感应消息后,以广播的形式向所有邻居节点转发消息,直到数据包到达目的节点或预先设定的生命期限变为零为止。泛洪路由协议实现起来简单、鲁棒性高,而且时延短、路径容错能力高,可以作为衡量标准去评价其他路由算法,但是该协议很容易出现消息“内爆”、盲目使用资源和消息重叠的情况,消息传输量大,加之能量浪费严重,这在资源受限的无线传感器网络是非常不利的。

如图 5-2-1 所示,节点 A 向节点 B 传送消息,当节点 A 广播后,节点 A 和 B 中间的每一个节点都会收到该信息。对于规模较大的网络,其中需要传递信息的节点数量会很多,因此最终的结果会导致网络中的每个节点都参与到每次的数据传输并接收到每次传输的数据,这种可能性将使得网络中的无效数据传输急剧增加,出现信息爆炸现象,消耗本来就紧张的能量、存储空间等资源。

在泛洪法中,信息重叠的情况也会大量出现。如图 5-2-2 所示,节点 A 和节点 B 收集的信息有相同重叠的部分 Z,当它们把信息传送给节点 C 时,节点 C 会接收到信息 Z 的两个副本。如果源节点远离目的节点时,这种现象尤其严重,而且收到的副本数量可能会远大于 2。

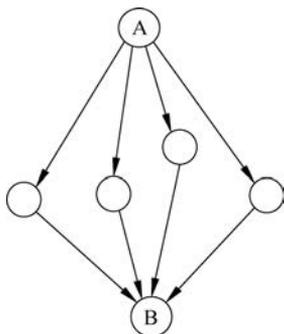


图 5-2-1 Flooding 协议的信息爆炸问题

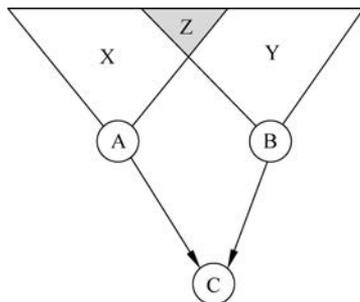


图 5-2-2 Flooding 协议的信息重叠问题



视频

Gossiping(闲聊)协议是对泛洪路由协议的改进,节点在收到感应数据后不是采用广播形式而是随机选择一个节点进行转发,这样就避免了消息的内爆,但是随机选取节点会造成路径质量的良莠不齐,增加了数据传输时延,并且无法解决资源盲目利用和消息重叠的问题。另外,由于采用随机选择接收节点的方式,使得数据传输不可能按照最短路径进行,甚至会出现南辕北辙的现象,所以数据传输平均时延拉长,传输速度变慢,无谓的资源消耗依然很多。



视频

## 5.2.2 SPIN 路由协议

SPIN(Sensor Protocol for Information via Negotiation,信息协商传感器协议)是第一个以数据为中心的自适应路由协议,针对 Flooding 协议中的“内爆”和“重叠”问题,它通过协商机制来解决。传感器节点监控各自能量的变化,若能量处于低水平状态,则必须中断操作转而充当路由器的角色,所以在一定程度上避免了资源的盲目使用。但在传输新数据的过程中,没有考虑到邻居节点由于自身能量的限制,只直接向邻居节点广播 ADV 数据包,不转发任何新数据。如果新数据无法传输,就会出现“数据盲点”,影响整个网络数据包信息的收集。

SPIN 路由协议包含 ADV 广告消息、REQ 请求消息和 DATA 数据消息 3 种消息类型。其路由过程可分为 3 个阶段,如图 5-2-3 所示。

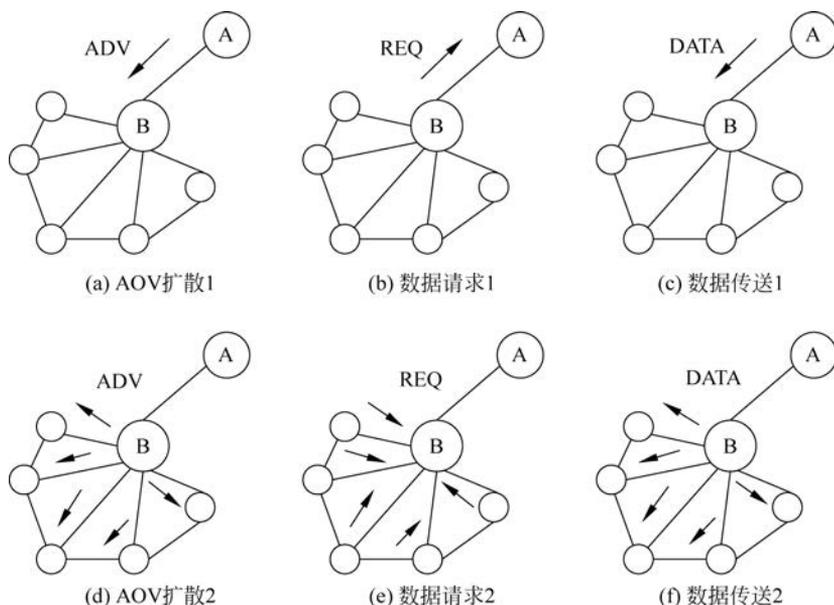


图 5-2-3 SPIN 路由的过程

### 1. 广告扩散

节点采集到数据后向邻居节点发送 ADV 消息,如图 5-2-3(a)所示,其中 ADV 消息包含数据的描述信息,描述信息通常比原始信息要短得多,因此广告信息消耗的能量较小,这样的设计符合能量受限的传感器网络。

### 2. 请求

邻居节点接收到 ADV 消息后,如果对信息感兴趣且尚未收到过 ADV 消息中的描述信

息所对应的数据,则给发送 ADV 消息的节点发送数据请求消息 REQ,如图 5-2-3(b)所示;如果已经收到或不感兴趣,则丢弃 ADV 消息,不做处理。

### 3. 传输

当节点收到邻居节点返回的数据请求消息 REQ 后,将数据封装到 DATA 消息中发送给该邻居节点,如图 5-2-3(c)所示。

上述 3 个步骤不断执行,将数据扩散到网络中那些希望得到数据的节点,如图 5-2-3(d)~(f)所示。SPIN 最初的版本称为 SPIN-PP(Point-to-Point),协议假设节点之间采用单播通信,且既不考虑节点的剩余能量,也不考虑信道丢包。这样的设定不适合传感器网络,其原因包括:首先,因为无线通信具有广播特性,节点一旦发送数据,在一定范围内的所有节点都能够接收到;其次,无线通信相比有线通信,丢包率要高得多,消息丢失可能造成 SPIN-PP 协议无法正常运行;最后,传感器节点能量受限,节点需要根据自身剩余能量决定是否参与数据转发。

## 5.2.3 DD 路由协议

DD(Directed Diffusion,定向扩展)路由协议多用于查询的扩散路由协议,与其他路由协议相比,其最大特点就是引入了梯度的理念,表明网络节点在该方向的深入搜索,来获得匹配数据的概率。它以数据为中心,生成的数据常用一组属性值来为其命名。兴趣扩散、初始梯度场建立和数据传输组成了 DD 路由协议的 3 个阶段,如图 5-2-4 所示。



视频

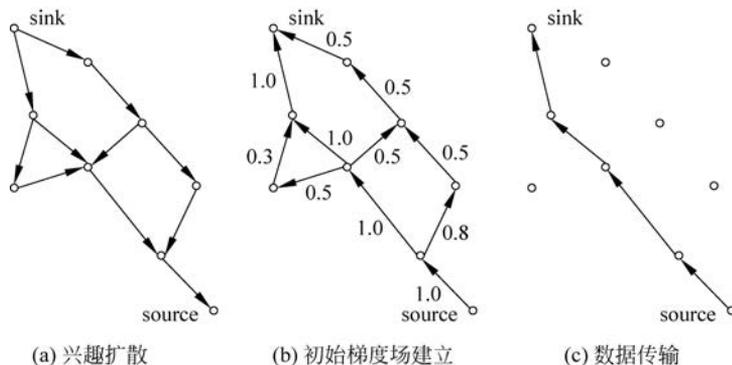


图 5-2-4 定向扩散路由机制图示

### 1. 兴趣扩散阶段

汇聚节点下达查询命令多采用泛洪方式,传感器节点在接收到查询命令后对查询消息进行缓存并执行局部数据的融合。

### 2. 初始梯度场建立

随着兴趣查询消息遍布全网,梯度场就在传感器节点和汇聚节点间建立起来,于是多条通往汇聚节点的路径也相应形成。

### 3. 数据传输阶段

通过加强机制发送路径加强消息给最新发来数据的邻居节点,并且给这条加强信息赋予一个值,最终梯度值最高的路径为数据传输的最佳路径。

DD 路由协议多采用多路径,鲁棒性好;节点只需与邻居节点进行数据通信,从而避免

保存全网的信息；节点不需要维护网络的拓扑结构，数据的发送是基于需求的，这样就节省了部分能量。DD 路由协议的不足是建立梯度时花销大，多 sink 的网络一般不建议使用。



视频

### 5.2.4 Rumor 路由协议

在有些传感器网络应用中，节点产生的数据量较少，如果采用定向扩散路由，需要经过查询消息的洪泛传播和路径增强机制，才能确定一条优化的数据传输路径，这样路由的开销就过大了，Rumor(谣传)路由协议正是为解决此问题而设计的。该协议借鉴了欧氏平面图上任意两条曲线交叉概率很大的思想。当节点监测到事件后将其保存，并创建称为 Agent (代理消息) 的生命周期较长的包括事件和源节点信息的数据包，将其按一条或多条随机路径在网络中转发。收到 Agent 的节点根据事件和源节点信息建立反向路径，并将 Agent 再次随机发送到邻居节点，并可在再次发送前在 Agent 中增加其已知的事件信息。汇聚节点的查询请求也沿着一条随机路径转发，当两条路径交叉时则路由建立；如不交叉，汇聚节点可 flooding 查询请求。在多汇聚节点、查询请求数目很大、网络事件很少的情况下，Rumor 协议较为有效。但如果事件非常多，维护事件表和收发 Agent 带来的开销会很大。

Rumor 协议被认为是 SPIN 路由协议与 DD 定向扩散路由协议的折中，并加入了 Gossiping 的随机转发机制。该协议引入了 Agent 代理消息的概念，使用了单播随机转发的方式。Rumor 协议的原理如图 5-2-5 所示，灰色区域表示发生事件的区域、圆点表示传感器节点，黑色圆点表示代理消息经过的传感器节点，灰色节点表示查询消息经过的传感器节点，连接灰色节点和部分黑色节点的路径表示事件发生区域到汇聚节点的数据传输路径。

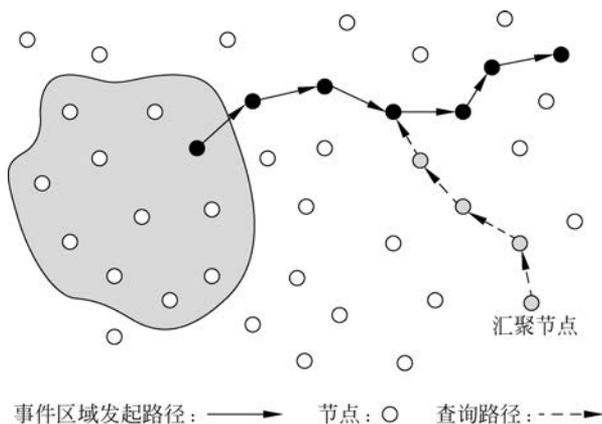


图 5-2-5 Rumor 路由原理图



视频

## 5.3 分层路由协议

分层拓扑，又称为层次型拓扑，分层拓扑把网络中的节点分为两类：骨干节点和普通节点。一般来说，普通节点把数据发送给骨干节点，骨干节点负责协调其区域内的普通节点的通信和进行数据融合等工作。骨干节点的能量消耗相对较大，因而需要经常更换骨干节点。

分层拓扑结构具有很多优点。例如，由簇头节点担负数据融合的任务，减少了数据通信量；有利于分布式算法的应用，适合大规模部署的网络；由于大部分节点在相当长的时间

内关闭了通信模块,所以显著地延长了整个网络的生存时间等。

采用分层路由的出发点是节省能量,在分层路由中网络划分成多个簇,每个簇包含多个簇成员和一个簇头,簇头节点管理协调簇内成员之间的通信。成员节点和簇头可以看作两个不同层次,簇头节点还可以进一步划分成多个更高层次的簇,这样网络就形成了多层结构。分层路由的过程实际上就是分簇的过程,网络完成分簇后,每个节点都只需要将数据传输到簇头节点,由簇头节点存储、融合后再发往更高层次的簇头节点,直到汇聚节点为止。因此,分层路由的关键在于如何分簇和簇内节点间的协同操作。

### 5.3.1 LEACH 路由协议

LEACH(Low Energy Adaptive Clustering Hierarchy,低功耗自适应分簇层次型)路由协议是一种经典的基于簇的自适应分簇拓扑协议,这是第一个提出数据融合的分层协议。为平衡网络各个节点的能耗,簇头是周期性按轮随机选举的。LEACH 协议定义了“轮”的概念,每轮循环分为簇的建立阶段和稳定的数据通信阶段。在簇的建立阶段,邻居节点动态地形成簇,随机产生簇头;在数据通信阶段,簇内节点把数据发送给簇头,簇头进行数据融合并把结果发送给汇聚节点。LEACH 协议示意图如图 5-3-1 所示。由于簇头需要完成数据融合、与汇聚节点通信等工作,所以能量消耗大。LEACH 算法能够保证各节点等概率地担任簇头,使得网络中的节点相对均衡地消耗能量。簇头选举算法如下:

(1) 每个节点产生一个  $0 \sim 1$  的随机数,如果这个数小于阈值  $T(n)$ ,则该节点称为簇头。

$$T(n) = \begin{cases} \frac{p}{1 - p(r \bmod (1/p))}, & n \in G \\ 0, & \text{其他} \end{cases} \quad (5-3-1)$$

其中, $p$  为网络中簇头数与总节点数的百分比, $r$  为当前的选举轮数, $r \bmod (1/p)$  表示这一轮循环中当选过簇头的节点个数, $G$  是最近  $1/p$  轮未当选过簇头的节点集合。

(2) 选定簇头后,通过广播告知整个网络。网络中的其他节点根据接收信息的信号强度决定从属的簇,并通知相应的簇头节点,完成簇的建立。最后簇头节点采用 TDMA 方式为簇中每个节点分配向其传送数据的时间片。

(3) 稳定阶段,传感器节点将采集的数据传送到簇头,簇头对数据进行融合后再传送到基站。稳定阶段持续一段时间后,网络重新进入簇的建立阶段,进行下一轮的簇头选举。

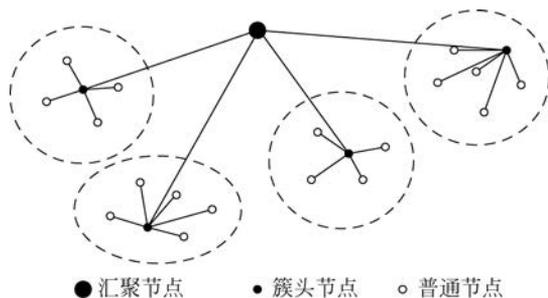


图 5-3-1 LEACH 路由协议



视频

在 LEACH 算法中,节点等概率地承担簇头角色,较好地体现了负载均衡思想,减小了能耗,提高了网络的生存时间。但是由于簇头位置具有较强的随机性,簇头分布不均匀,致使骨干网的形成无法得以保障,不适合大范围的应用;簇头同时承担数据融合、数据发送的“双重”任务,因此能量消耗很快。频繁的簇头选举引发的通信增加了能量消耗。

簇头选出后,就要向全网广播当选成功的消息,其他节点根据接收到信号的强度来选择它要加入哪个簇并递交入簇申请,信号强度越强表明离簇头越近。当完成簇成型后,簇头根据簇成员的数量多少,需要发送给本簇内的所有成员一份 TDMA 时间调度表。簇成员在数据采集时就根据事先设置的 TDMA 时间表进行操作、采集信息,并上传给簇头。簇头将接收到数据进行数据融合后直接传向汇聚节点。在数据采集达到规定时间或次数后,网络开始新一轮的工作周期,簇头依然根据上述步骤进行再一次的选举。

该协议实现起来简单,由于利用了数据融合技术,在一定程度上减少了通信流量,节省了能量;随机选举簇头,平均分担路由任务量,减少了能耗,延长了系统的寿命。同时,LEACH 路由协议也存在不可忽视的缺点,如由于簇头选举是随机地依据本地信息自行来决定,避免不了出现位置随机、分布不均的情况;每轮簇头的数量和不同簇中节点数量不同,导致网络整体负载的不均衡;多次分簇带来了额外开销以及覆盖问题;簇头选取时没有考虑节点的剩余能量,有可能导致剩余能量很少的节点随机当选成为簇头。如果汇聚节点位置与目标区域有较大的距离,且功率足够大,则通过单跳通信传送数据会造成大量的能量消耗,所以单跳通信模式下的 LEACH 协议比较适合小规模网络。另外,该协议在单位时间内一般发送数量基本固定的数据,不适合突发性的通信场合。



视频

### 5.3.2 HEED 路由协议

针对 LEACH 算法中节点规模小,簇头选举没考虑节点的地理位置等不完善的地方,在 LEACH 算法的基础上,有学者提出了 LEACH 的改进算法 HEED(Hybrid Energy-Efficient Distributed clustering,混合能量高效分布式分簇)路由协议,有效解决了 LEACH 算法中簇头可能分布不均匀的问题。以簇内平均可达能量作为衡量簇内通信成本的标准,节点用不同的初始概率发送竞争消息,节点的初始化概率  $CH_{prob}$  根据式(5-3-2)确定:

$$CH_{prob} = \max(C_{prob} + E_{resident}/E_{max}, P_{min}) \quad (5-3-2)$$

式中, $C_{prob}$  和  $P_{min}$  是整个网络统一的参量,它们影响到算法的收敛速度。簇头竞选成功后,其他节点根据在竞争阶段收集到的信息选择加入哪个簇。HEED 算法在簇头选择标准以及簇头竞争机制上与 LEACH 算法不同,成簇的速度有一定的改进,特别是考虑到成簇后簇内的通信开销,把节点剩余能量作为一个参量引入算法中,使得选择的簇头更适合担当数据转发的任务,形成的网络拓扑更趋合理,全网的能量消耗更均匀。

HEED 综合考虑了生存时间、可扩展性和负载均衡,对节点分布和能量也没有特殊要求。显然 HEED 执行并不依赖于同步,但是不同步却会严重影响分簇的质量。



视频

### 5.3.3 TEEN 路由协议

TEEN(Threshold sensitive Energy Efficient sensor Network protocol,能量有效的阈值敏感)路由协议是一个事件驱动的响应型聚类路由协议。根据簇头与汇聚节点间距离的远近来搭建一个层次结构。TEEN 中有两个重要参数:硬阈值和软阈值。

硬阈值设置一个检测值,只有传送的数据值大于硬阈值时,节点才允许向汇聚节点上传数据。而软阈值设置一个检测值的变化量值,规定只有当传送数据的改变量大于设定的软阈值时,才同意再次向汇聚节点上传数据,这两个阈值决定了节点何时能够发送数据。

TEEN 路由协议的工作原理如下:当首次发送的数据值大于硬阈值时,下一级节点向上一级节点报告,并将数值保存起来;此后当发送数据值大于硬阈值且变化量大于软阈值时,低一级节点才会再次向上一级节点报告数据。TEEN 路由协议如图 5-3-2 所示。

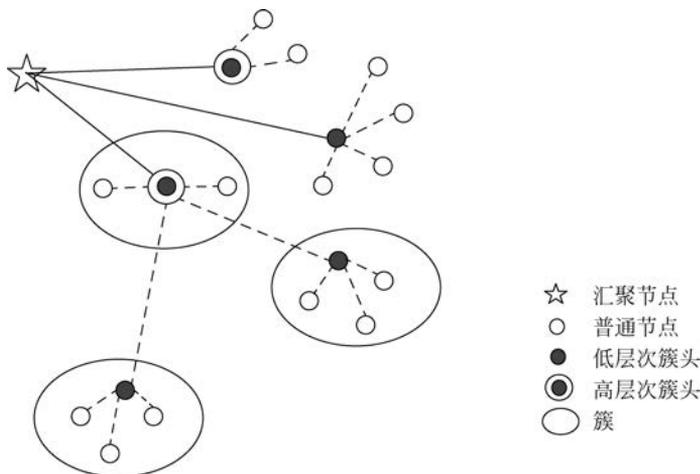


图 5-3-2 TEEN 路由协议图

TEEN 路由协议的优点是软、硬阈值的存在,具有了过滤功能,精简数据传输量,数据传输量比主动网络少,节省了大量的能量,适合于响应型应用。分层的簇头结构无须所有节点具有大功率通信能力,更适合无线传感器网络的特点。

TEEN 路由协议的缺点是多层次簇的构建非常复杂。如果某个节点的检测数据达不到硬阈值,那么用户将无法获知这个感应数据,也无法判断这个节点是否失效,因此这个方法在周期性采样的网络中要谨慎使用,如果每个节点都需要较高的通信功率与汇聚节点通信,则仅适合小规模的系统。

### 5.3.4 TTDD 路由协议

TTDD(Two-Tier Data Dissemination, 双层数据分发)路由协议是一种主要针对网络中存在的多 sink 和 sink 移动问题的,基于网格的分层路由协议。

TTDD 协议包括 3 个阶段:构建网格阶段、发送查询数据阶段和传输数据阶段。其中构建网格阶段是 TTDD 协议的第一步,也是最关键、最核心的一步。协议中的节点都清楚自身所处的位置,当有事件发生信号时,就近选择一个节点作为源节点,源节点将自身所处位置作为网格的一个交叉点,基于此点,先计算出相邻交叉点的位置,利用贪婪算法计算出距离该位置最近的节点,最近节点就成为新交叉点,以此铺展开构建成为一个网格。

事件信息和源节点信息被保存在网格的各个交叉点。数据查询时,汇聚节点在所达范围内,依次找出最近的交叉点,经由交叉点传播数据直至源节点,源节点收到查询命令后,将数据沿最短路径传向汇聚节点。有时,在等待数据回传时,汇聚节点可以采用代理机制保持



视频

移动,以保证数据可靠地进行传输。

TTDD 路由协议采用的是单路径,可以延长网络的生命周期;采用代理机制很好地解决了汇聚节点的移动性问题。但是在 TTDD 路由协议中,节点必须知道自身位置的所在,要求节点密度比较高,计算与维护网络的开销成本较大,网格构建、查询请求和数据传递过程都会造成传输的延迟,所以这种协议在目标高速移动和高实时性需求的场合应慎用。



视频

### 5.3.5 PEGASIS 路由协议

PEGASIS(Power Efficient Gathering in Sensor Information System,传感器信息系统中的节能收集)路由协议假设:每个节点都知道其余节点的位置信息;每个节点都能和汇聚节点直接通信。在这些假设前提下,PEGASIS 将网络中所有节点组合成一条链,链中只有一个节点充当簇头节点;节点沿着链将数据发送给靠近簇头节点的邻居节点,邻居节点将接收到的数据和自己的数据融合后,再将数据沿着链发送给它的邻居节点。簇头节点接收到数据后将数据发送给汇聚节点,示意图如图 5-3-3 所示。

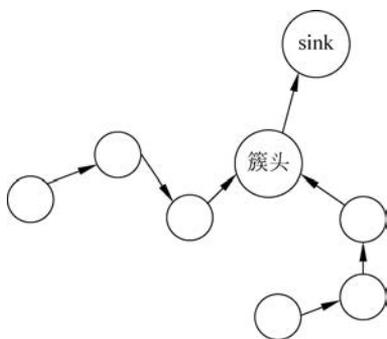


图 5-3-3 PEGASIS 路由协议

#### 1. 链的构造

链的构造可以由节点自己完成,也可以由汇聚节点来完成,然后通过广播告知所有节点在链的构造过程中,节点选择距离自己最近的节点作为数据传输的下一跳。为了确保距离汇聚节点较远的节点有较近的邻居节点作为下一跳,链的构造从距离汇聚节点最远的节点开始。因为在链的构造过程中已经在链中的节点不能再次被访问,链中邻居节点间的距离可能会越来越大。如图 5-3-4 所示,节点 0 距离 sink 最远,节点 0 选择距离自己最近的节点 3 连接,节点 3 在剩余的节点中选择

和距离较近的节点 1 连接,最后节点 1 和节点 2 连接,这样就组成了一条 0-3-1-2 的链。当链中有节点能量耗尽的时候整条链再重构一次。每条链中只有一个簇头节点,且簇头节点周期更换。在第  $i$  个周期中,PEGASIS 使用链条上第  $i \bmod N$  个节点作为簇头节点,其中  $N$  为节点的总数。

#### 2. 数据传输

PEGASIS 使用令牌消息(token)来控制数据传输,令牌消息一般很小,带来的开销也较小。如图 5-3-5 所示,节点  $c_2$  为本轮的簇头节点,节点  $c_2$  沿着链先给节点  $c_0$  发送一个令牌消息。节点  $c_0$  接收到令牌消息后将自己的数据发送给  $c_1$ , $c_1$  先将  $c_0$  的数据和自己的数据融合,再将融合后的数据发送给节点  $c_2$ ;  $c_2$  接收到  $c_1$  发送的数据后,再沿着链的另一个方向给节点  $c_4$ ,发送令牌消息,节点  $c_4$ 、 $c_3$  的数据采用和前面相同的方式发送给节点  $c_2$ ,最后节点  $c_2$  将  $c_1$ 、 $c_3$  发送过来的数据和自己的数据融合,并将结果发送给汇聚节点。有些节点可能和自己邻居节点的距离比较大,例如图 5-3-4 中的节点 1 和节点 2,如果让这些节点作簇头节点,那么在数据传输阶段的开销会比较大。为了防止这些节点因为能量耗尽而过早失效,可以设置一个阈值,当节点和邻居节点的距离大于此阈值时,该节点就不能成为簇头节点。

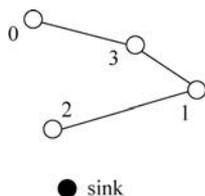


图 5-3-4 基于贪婪算法的链构造

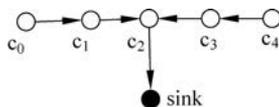


图 5-3-5 令牌传递过程

PEGASIS 和 LEACH 相比能够节省很多能量。首先,PEGASIS 更换簇头后不会像 LEACH 一样,要广播告知网络中的其余节点;其次,在数据传输阶段,PEGASIS 中节点将数据发送给邻居节点,LEACH 中节点将数据发送给簇头节点,而一般说来前者的传输距离比后者的传输距离要小;接着,PEGASIS 每轮中只有一个簇头节点,LEACH 中有多个簇头节点,而簇头节点和汇聚节点之间的通信开销很大;最后,PEGASIS 中簇头节点只需接收两个节点的数据,而 LEACH 中簇头节点要接收很多节点的数据。

PEGASIS 虽然能够有效地延长网络的寿命,但是 PEGASIS 也有一些缺点:首先,簇头节点必须要等到链两边的数据到达后才能将数据发送给汇聚节点,因此 PEGASIS 的延迟会比较大;其次,数据在沿途过程中会经过多次融合,这将导致汇聚节点接收到的数据可能不精确。

## 5.4 其他路由协议

### 5.4.1 地理位置路由

在传感器网络的一些应用中,采集到的数据需要与地理位置联系起来置。地理位置信息路由假设所有节点都知道自己的地理位置信息。以及目的节点或目的区域的地理位置,利用这些地理位置信息作为路由选择的依据,节点按照一定策略转发数据到目的节点。在地理位置信息路由中,节点的物理位置已经隐含了向哪个邻居节点转发数据分组的信息,所以它只需用很小的路由表(甚至不需要路由表),这可以大大简化路由协议。地理位置信息路由的关键在于如何根据节点的位置信息选择下一跳邻居节点,以及在出现路由空洞时如何绕路。

#### 5.4.1.1 GAF 路由协议

GAF(Geographical Adaptive Fidelity,地域自适应保真)路由协议是基于节点地理位置的分簇协议。该协议首先把部署区域划分成若干虚拟单元格,将节点按照地理位置划入相应的单元格,然后在每个单元格中定期地选举一个簇头节点。在 GAF 算法中,每个节点可以处于 3 种不同状态:休眠、发现和活动状态。状态间的转换过程如图 5-4-1 所示。

在初始状态下,所有节点处于发现状态。此时节点交换 Discovery 消息来获得同一虚拟单元格中其他节点的信息。

当节点进入发现状态时,为其设置一个定时长度为  $T_d$  的定时器 D,一旦 D 定时时间达到  $T_d$ ,节点广播 Discovery 消息,同时转换到活动状态。如果在计时器超时之前节点收到其他节点成为簇头的声明,则取消计时器,进入休眠状态。



视频



视频

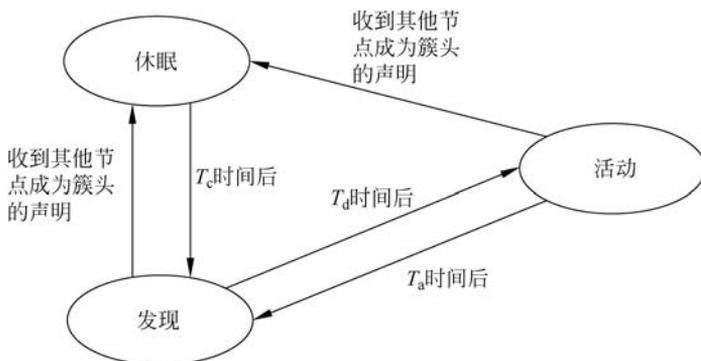


图 5-4-1 GAF 算法节点状态转换图

当节点进入活动状态时,为其设置一个定时长度为  $T_a$  的定时器 A,表示节点处于活动状态的时间。一旦 A 定时时间达到  $T_a$ ,节点转换到发现状态。在节点处于活动状态期间,以时间间隔  $T_d$  重复广播 Discovery 消息,以便压制其他处于发现状态的节点进入活动状态。

当节点转入休眠状态时,就关闭收发信机。处于休眠状态的节点在休眠一段时间  $T_c$  之后被唤醒,同时重新转入发现状态。

GAF 算法基于平面模型,以节点间的距离来度量是否能够通信,而在实际应用中距离邻近的节点可能因为各种因素不能直接通信。此外,该算法也没有考虑节点能耗均衡的问题。

#### 5.4.1.2 GEAR 协议

GEAR(Geographical and Energy-Aware Routing,地理和能源敏感路由)协议是基于位置的一种路由协议,根据事件区域的地理位置信息和节点的剩余能量,建立汇聚节点到事件区域的优化路径,避免了洪泛传播方式,从而减少了路由建立的开销,使能量得以有效利用。

GEAR 路由假设已知事件区域的位置信息;每个节点知道自己的位置信息和剩余能量信息,并通过一个简单的 HELLO 消息交换机制获取所有邻居节点的位置信息和剩余能量信息;节点间的无线链路是对称的。GEAR 路由中查询消息传播包括两个阶段:

- (1) 汇聚节点发出查询命令,并根据事件区域的地理位置和剩余能量,将查询命令传送到区域内距汇聚节点最近的节点;
- (2) 获得查询命令的节点将查询命令广播到区域内的其他所有节点。来自查询区域内节点的监测数据沿查询消息的反向路径传送到汇聚节点。

#### 5.4.1.3 GPSR 协议

GPSR(Greedy Perimeter Stateless Routing,贪婪周边无状态路由)协议是一个典型的基于位置的路由协议。使用 GPSR 协议,网络节点都知道自身地理位置并被统一编址,各节点利用贪婪算法尽量沿直线转发数据,产生或收到数据的节点向以欧氏距离计算最靠近目的节点的邻居节点转发数据,但由于数据会到达没有比该节点更接近目的节点的区域(称为空洞),导致数据无法传输,当出现这种情况时,空洞周围的节点能够探测到,并利用右手法则沿空洞周围传输来解决此问题。该协议避免了在节点中建立、维护、存储路由表,只依赖直接邻居节点进行路由选择,几乎是一个无状态的协议;且使用接近于最短欧氏距离的



视频



视频

路由,数据传输时延小;并能保证只要网络连通性不被破坏,一定能够发现可达路由。但缺点是当网络中汇聚节点和源节点分别集中在两个区域时,由于通信量不平衡易导致部分节点失效,从而破坏网络连通性;需要 GPS 或其他定位方法协助计算节点位置信息。

## 5.4.2 QoS 路由协议

QoS(Quality of Service)路由是指除了需要减少能耗以外,还要满足某些其他服务质量需求的路由。常用的服务质量指标包括延迟、吞吐率和可靠性等。值得注意的是,QoS 路由并非仅仅要优化网络的某项性能,而是要保证网络提供的路由服务必须满足一个给定的服务质量界限。例如,在有一定实时性要求的系统中,端到端传输延迟不得高于给定阈值,因此建立的路由转发路径必须满足这个条件。由于每个节点只有局部信息,为了达到服务质量要求,每个节点在选择下一跳节点时都只能估计整条路径可能达到的服务质量。QoS 路由的关键在于如何将全局服务质量转换为局部路由指标,进而分布式实现路由转发过程。

### 5.4.2.1 SPEED 路由协议

SPEED(速率)是一种为了拥塞控制和软实时提供保证利用地理位置信息的 QoS 路由,主要针对传输速率有一定要求的应用场景。SPEED 不是仅仅根据地理位置选择下一跳节点,而是还加入了传输速率。由于传输速率本身受到负载的影响,即负载越大转发延迟越高、传输速率越低,所以 SPEED 通过选择传输速率较高的下一跳,可以间接起到均衡网络负载的作用。SPEED 是一个实时路由协议,在一定程度上实现了端到端的传输速率保证、网络拥塞控制以及负载平衡机制。为了实现上述目标,SPEED 协议首先交换节点的传输延迟,以得到网络负载情况;然后节点利用局部地理信息和传输速率信息做出路由决定,同时通过邻居反馈机制保证网络传输速率在一个全局定义的传输速率阈值之上。节点还通过反向压力路由变更机制避开延迟太大的链路和路由空洞。但是,SPEED 并未考虑网络中有多种不同传输速率要求的应用并存的情况。

SPEED 协议主要由以下 4 部分组成:

- (1) 延迟估计机制,用来得到网络的负载情况,判断网络是否发生拥塞;
- (2) SNGF(Stateless Non-deterministic Geographic Forwarding)算法,用来选择满足传输速率要求的下一跳节点;
- (3) 邻居反馈环策略(Neighborhood Feedback Loop,NFL),在 SNGF 路由算法中找不到满足要求的下一跳节点时采取的补偿机制;
- (4) 反向压力路由变更机制,用来避免拥塞和路由空洞。

SPEED 协议中各部分之间的关系如图 5-4-2 所示。

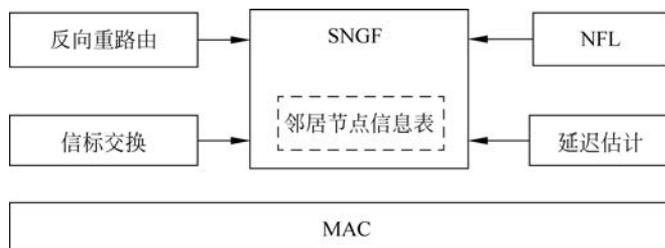


图 5-4-2 SPEED 协议框架



视频



视频



视频

### 5.4.2.2 SAR 协议

SAR(Sequential Assignment Routing,有序分配路由)协议是第一个在无线传感器网络中保证 QoS 的主动路由协议。汇聚节点的所有一跳邻居节点都以自己为根创建生成树,在创建生成树的过程中考虑节点的时延、丢包率等 QoS 参数以及最大数据传输能力,各个节点从而反向建立了到汇聚节点的具有不同 QoS 参数的多条路径,节点发送数据时选择一条或多条路径进行传输。该协议能够提供 QoS 保证,但缺点是节点中的大量冗余路由信息耗费了存储资源,且路由信息维护、节点 QoS 参数与能耗信息的更新均需要较大的开销。



视频

### 5.4.3 多径路由协议

源节点与 sink 之间存在多条可用传输路径,数据可以沿着其中一条路径传输,也可以同时使用多条路径传输以提高数据传输的鲁棒性。多径路由通常分为两类,即相交多径和不相交多径。相交多径路由中一个中间节点可能处在多条路径上;不相交多径路由中除源节点与汇聚节点外,其他任意中间节点只能位于一条路径之上。多径路由的构造可以看作单条路径路由在数量上的扩展,节点可能需要同时维护多条路径。多径路由的关键在于如何以较低的开销构造和维护多条路径。



视频

#### 5.4.3.1 MMSPEED 路由协议

MMSPEED(Multi-path MultiSPEED,多路径多速率)路由协议是在 SPEED 基础上提出的一种同时考虑传输延迟和丢包率的 QoS 路由协议。与 SPEED 相比,MMSPEED 做了多方面的改进:

(1) SPEED 仅能支持一种速率要求,而 MMSPEED 可以同时支持不同应用的多种速率要求;

(2) SPEED 不考虑端到端传输的可靠性,而 MMSPEED 在速率的基础上,还支持不同的端到端可靠性要求;

(3) SPEED 没有考虑估计误差对端到端延迟的影响,MMSPEED 在每一跳节点上都需要进行速率补偿和可靠性补偿,不断纠正由于估计误差造成的影响。

具体来说,协议分为两个基本组件:时延控制和可靠性控制。时延控制即满足某种特定应用对端到端传输延迟的需求,这种控制是通过路径选择完成的。在源节点和目标节点之间存在多条可选路径,不同路径的端到端传输延迟不同,对于那些延迟要求高的应用,使用高速路径转发,而延迟要求低的应用使用低速路径转发。可靠性控制是指满足端到端传输可靠性,是通过多径转发实现的。对于可靠性要求高的应用,使用多条路径同时转发,这样即使其中部分路径失败,目标节点仍然能够接收到数据。对于可靠性要求低的应用,可以只使用单条路径转发。



视频

#### 5.4.3.2 MP-EA 路由协议

多路径路由还可以用于均衡节点的能量消耗。传统网络的路由机制往往选择源节点到目标节点的跳数最小的路径传输数据,但在传感器网络中,如果频繁使用同一条路径传输数据,就会造成该路径上的节点因能量消耗过快而过早失效,从而使整个网络分割成互不相连的孤立部分,减少了网络的生存期。为此,Rahul C. Shah 等人提出了一种 MP-EA 路由协议(Multi-path Routing Protocol with Energy-Aware,能量敏感多路径路由协议)。该机制在源节点和目标节点之间建立多条路径,根据路径上节点的通信能量消耗以及剩余能量情

况,给每条路径赋予一定的选择概率,数据传输均衡消耗整个网络的能量,延长网络的生存期。

Shah 等人提出的能量多路径路由协议的目的主要在于改善定向扩散协议的耗能情况,采用地理位置和数据类型(即节点类型)标识节点。Shah 等人认为该协议是按需路由协议,但其含义更多的是查询驱动的,将其与定向扩散协议都列为主动路由协议。汇聚节点( $\text{Cost}(\text{sink})=0$ )利用受控的 Flooding 方式发起建立路由请求,产生或转发路由请求节点  $N_i$  的所有邻居节点  $N_j$  测量与  $N_i$  的通信开销以及  $N_i$  的剩余能量:  $\text{Metric}(N_j, N_i)$ 。  $N_j$  根据式(5-4-1)计算代价,  $N_j$  节点选择  $C$  较小的一些邻居节点反向构造路由表  $\text{FT}_j$ , 邻居节点  $N_i$  被赋予由式(5-4-4)计算的路由概率,此后  $N_j$  节点由式(5-4-5)计算自身代价  $\text{Cost}(N_j)$ , 然后,  $N_j$  转发包含自身代价信息的请求。在通信阶段,节点  $N_j$  根据选择一条路径进行数据发送,与定向扩散协议相比,该协议虽然存在多条路径,但只选用一条,能够有效节约能源 40% 以上;随机选择路由的方式平衡了通信量。其缺点是:汇聚节点需要周期性以 Flooding 方式维护路由信息;需要进行节点间收发开销和剩余能量测量;根据概率随机选择一条路径导致其可靠性不如定向扩散协议。

能量多路径路由协议包括路径建立、数据传输和路由维护 3 个过程。路径建立过程是该协议的重点内容。每个节点需要知道到达目标节点的所有可选下一跳节点,并计算选择每个下一跳节点传输数据的概率。选择概率是根据节点到目标节点的通信代价来计算的,在下面的描述中用  $\text{Cost}(N_i)$  表示节点  $i$  到目标节点的通信代价。因为每个节点到达目的节点的路径很多,所以这个代价值是各个路径的加权平均值。

能量多路径路由的主要流程描述如下。

### 1. 发起路径建立

目的节点广播路径建立消息,启动路径建立过程,广播消息中包含一个代价域,表示发出该消息的节点到目的节点路径上的能量信息,设初始值为零。

### 2. 判断是否转发路径建立消息

当某一个节点接收到邻居节点发送的路径建立消息时,与发送该消息的节点进行比较,只有在自己距离源节点更近,并且距离目的节点更远的情况下,才转发该路径建立消息,否则丢弃该消息。

### 3. 计算能量代价

如果节点决定转发路径建立消息,需要计算新的代价值来替代原来的代价值。当路径建立消息从  $N_i$  发送到  $N_j$  时,该路径的通信代价值为  $N_i$  的代价值加上两个节点间的通信能量消耗,即有

$$C_{N_j, N_i} = \text{Cost}(N_i) + \text{Metric}(N_j, N_i) \quad (5-4-1)$$

其中,  $C_{N_j, N_i}$  表示节点  $N_j$  发送数据经由节点  $N_i$  路径到达目的节点的代价值,  $\text{Metric}(N_j, N_i)$  表示节点  $N_j$  到节点  $N_i$  的通信能量消耗,计算公式如式(5-4-2)所示

$$\text{Metric}(N_j, N_i) = e_{ij}^\alpha R_i^\beta \quad (5-4-2)$$

其中,  $e_{ij}^\alpha$  表示节点  $N_j$  到节点  $N_i$  直接通信的能量消耗,  $R_i^\beta$  表示节点  $N_i$  的剩余能量,  $\alpha, \beta$  是与网络有关的常量。

### 4. 节点加入路径条件

代价太大的路径对网络生存时间没有益处,因此并非每个路径都是可用的,节点需要丢

弃代价太大的路径。 $N_j$  将  $N_i$  加入本地路由表  $FT_j$  中的条件是

$$FT_j = \{i | C_{N_j, N_i} \leq \alpha (\min_{k \in FT_j} (C_{N_j, N_k}))\} \quad (5-4-3)$$

其中,  $\alpha$  为大于 1 的系统参数。

### 5. 节点选择概率计算

为了均衡网络中节点的能量消耗, 节点选择概率需与能量消耗成反比,  $N_j$  使用式(5-4-4)来计算选择  $N_i$  的概率。

$$P_{N_j, N_i} = \frac{1/C_{N_j, N_i}}{\sum_{k \in FT_j} 1/C_{N_j, N_k}} \quad (5-4-4)$$

### 6. 代价平均值计算

节点根据路由表中的能量代价和下一跳节点选择概率计算本身到目的节点的代价  $Cost(N_j)$ , 定义为经由路由表节点到目的节点代价的平均值:

$$Cost(N_j) = \sum_{k \in FT_j} P_{N_j, N_k} \cdot C_{N_j, N_k} \quad (5-4-5)$$

其中,  $N_j$  将用  $Cost(N_j)$  代替消息中原有的代价值, 然后向邻居节点广播该路由建立消息。

数据传播时, 节点对于每一个接收的数据分组, 都利用概率选择某一个下一跳节点, 并转发该分组。网络中路由的维护则是通过周期性地从目的节点到源节点实施扩散查询来保证所有路由处于有效状态的。从该协议原理可以看出, 由于选择节点的概率和剩余能量成反比, 因此能量多路径路由协议可以在各节点之间均衡通信能耗, 从而保证整个网络中各节点的能量消耗平稳而均衡地进行, 最大限度地延长网络的生存期。

## 习题 5

1. 简述无线传感器网络路由协议的考虑因素。
2. 简述无线自组织网络和无线传感器网络的路由协议的区别。
3. 简述无线传感器网络的路由过程的步骤。
4. 简述无线传感器网络路由协议分类方法和内容。
5. 简述定向扩散路由机制的基本过程。
6. 简述 LEACH 算法的实现思想。