

长链型数据收集传感器网络的负载平衡研究*

于永华, 李丽芬, 朱永利

(华北电力大学 计算机科学与技术学院, 河北 保定 071003)

摘要: 为解决传感器网络的漏斗效应, 依据减少跳步数原则, 提出了一种新的拓扑控制算法, 算法对网络中的局部区域突发的大量事件包进行数据分流, 实现了网络的区域负载平衡, 很好地解决了网络的漏斗效应。实验结果证明了该方法的有效性。

关键词: 数据收集传感器网络; 服务质量; 漏斗效应; 负载平衡

中图分类号: TP393

文献标志码: A

文章编号: 1001-3695(2010)07-2724-03

doi: 10.3969/j.issn.1001-3695.2010.07.091

Research on load balancing of long-chain structure data collection sensor network

YU Yong-hua, LI Li-fen, ZHU Yong-li

(School of Computer Science & Technology, North China Electric Power University, Baoding Hebei 071003, China)

Abstract: With the funnel effect of the sensor network in mind, and based on the principle of reducing the number of hop-step, this paper designed a new topology control algorithm. The algorithm could shunt data streaming in the local area of network, which was caused in unexpected events way, then achieved load balancing of the network. It can be well solved the problem of the funnel effect of sensor network. Experimental results show that the method is effective.

Key words: data collection sensor network; quality of service(QoS); funnel effect; load balancing

无线传感器网络(WSN)因其广泛的应用前景而备受人们关注,传感器网络可被部署在各种非布线区、电源供给困难的区域和人员不易到达的区域。实时监测是传感器网络的一项重要应用,传感器节点需要对监测区域的被监测对象进行实时感知,当传感器节点有特殊事件包产生,网络能够及时将采集的事件包以多跳的方式发送到基站。监测中心也可通过查询方式查询监测区域某节点的监测对象信息,继而上传。

面向输电线检测^[1]、地下矿井监测以及公路交通监测等应用的无线传感器网络的拓扑结构一般为长链状,这种传感器网络多具有集中式数据接收、多跳传输、多对一流量模式等特征,通常被称做数据收集传感器网络^[2]。

漏斗效应是影响长链型数据收集传感器网络服务质量提升的一个重要问题。网络中越靠近基站,数据量越大,传感器节点承担的转发任务也就越重,基站邻近的几跳节点成为制约网络传输容量的瓶颈,即漏斗效应^[3],网络传输过程中也必然会出现拥塞和热点问题。研究者针对该问题已提出了一些解决方案,如汇聚节点动态功耗管理^[4]、负载平衡^[5]、数据融合^[6]等,但这些方法都只能在一定程度上缓解漏斗效应,不能彻底改善网络的漏斗效应。

1 网络模型

传感器网络可看做一个图 $G = (V, E, \text{BASE})$ 。其中:BASE 表示基站; V 为传感器节点集合; 如果节点 u 与节点 v 能够直接通信,则在集合 E 中包含一条边 (u, v) 。为实现本文的拓扑控制算法,每个节点需要建立四个列表:邻居节点列表 Nlist、

父节点列表 Plist、子节点列表 Clist 和兄弟节点列表 Blist。

本文的研究基于数据收集传感器网络,这种网络被广泛应用于各种监测领域。网络中节点可以定时地对特定被监测对象进行感知、采集数据,也可对被监测区域的特殊事件进行响应。针对传感器网络在输电线路监测方面的应用,本文中的节点特性作如下假设:

a) 基站配置多天线支持多频率接收,基站能量不受限但通信范围不能覆盖全网。

b) 传感器网络由分布于平面上的大量传感器节点组成,每个节点都有惟一标志。

c) 传感器节点电池能量较充足,最大发射功率下通信距离可达 5~6 个单位跳距(某一固定长度)。

d) 所有传感器节点一经部署之后不再移动,节点具有较低的失效率。

本文拓扑控制算法的思想是,将长链状数据收集传感器网络进行层次划分,将网络中的节点按固定地理位置间隔平均抽取来组成一个数据传输层次,整个传感器网络将被组织成多个相对独立的数据传输层,每层采用不同的频段传输数据。这样网络的每个层次平均覆盖整个监测区域,起到对网络中的局部区域节点上的大量事件包进行层次分流的作用,由拓扑控制算法得到的网络模型可通过图 1 加以理解。网络层次划分完成后可在其上实现各种路由协议。

2 节点间干扰分析

传感器节点能量相对充足,网络传输数据时可采用最大发射功率、最大单跳传输距离进行数据传输,从而降低数据传输

收稿日期: 2009-12-02; 修回日期: 2010-01-02 基金项目: 国家自然科学基金资助项目(60974125)

作者简介: 于永华(1984-),男,内蒙古赤峰人,硕士研究生,主要研究方向为无线传感器网络(hbdlxxy@yahoo.cn);李丽芬(1970-),女,河北保定人,讲师,博士,主要研究方向为计算机网络;朱永利(1964-),男,教授,博导,主要研究方向为无线传感器网络在电力系统中的应用。

时延。但节点通信覆盖范围较大也会带来一些负面影响,最主要的是会增加节点之间的通信干扰,干扰不但增加信道竞争,造成严重的包碰撞、包丢失,而且会造成大量能量资源浪费。

在长链状数据收集传感器网络中,由图 2(a)可知,假设节点通信覆盖范围为 3 个单位跳距,在网络未分层的情况下,一个节点通信范围内的节点数目约为 20 个,通信过程中这 20 个节点共同竞争无线信道,且数据包传输所经历跳距为 1~3 个单位跳距不等。

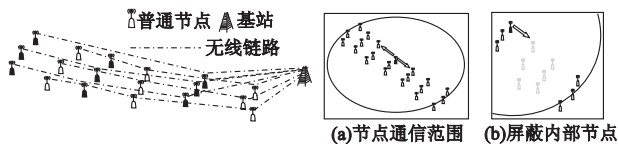


图1 网络层次划分示意图

图2 节点通信信道竞争

通过网络层次划分将节点覆盖范围区域内部的一些节点划归到另外的工作在其他频段的层次中,只与内边缘节点通信,竞争信道节点数量将减少为图 2(b)中的 8 个,数据包传输跳距均为 3 个跳距。

在分布式协调 DCF 访问控制方式下,MAC 层采用虚拟载波侦听来确定无线信道状态,根据 CSMA/CA 协议,当一个节点要传输一个分组时,它需要通过一个帧间间隔 DIFS 来确定信道是否空闲。CSMA/CA 协议工作模式如图 3 所示。

20 个节点竞争信道的情况下,将 20 个节点对网络造成的信道竞争压力划归到一个节点覆盖范围内,则每个节点为转发一个分组所用的平均信道侦听时间为 $(1 + 2 + 3 + \dots + 20) / (2 \times 20) \approx 10$ (DIFS),减少到 8 个节点时的平均信道侦听时间为 $(1 + 2 + 3 + \dots + 8) / (2 \times 8) \approx 4$ (DIFS)。所以分层后每个节点传输一个分组的平均通信竞争引发的推迟发送时间变为未分层时的 2/5,大大缩短了数据的推迟发送时间,同时数据传输跳步数由 1 个单位跳距增加到 3 个单位跳距,数据传输总跳数的减少也会在一定程度上降低网络传输时延。所以说拓扑控制分层在实现负载均衡的同时,获得了网络服务质量的提升。

3 网络的负载均衡

一般情况下,用于监测的无线传感器网络覆盖范围较广,被监测区域内的环境因素差别很大,如局部区域出现大雾、雷电天气,传感器节点对该环境变化引发的事件进行数据采集,这时可能会造成该区域内传感器节点采集的数据包急剧增加,网络中传输的数据主要以突发事件数据包为主。上层网络对传感器网络信息查询也多是基于对某一节点信息的查询,所以网络数据源具有集中的特性,对这种网络负载的平衡,借助文献[7]构建负载均衡网络的方法是无法解决的。

本文的拓扑控制算法将传感器网络中的节点按固定地理位置间隔平均抽取来组成一个数据传输层次,整个传感器网络将被组织成多个相对独立的数据传输层。

如图 4 所示,这意味着对于一个事件多发区域,区域内数据源节点将被平均分配在每个传输层。实质是数据已经在数据源节点所属网络层次处被划归到不同层次,从而达到层次分流数据的目的。

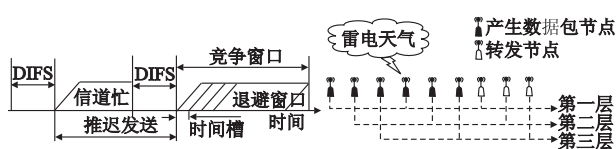


图3 CSMA/CA协议工作模式

图4 数据层次分流

网络进行分层后,整个网络的数据平均分配在网络的每个传输层,此时网络瓶颈仍然存在,每个传输层邻近基站的几跳节点会形成一个瓶颈子网,网络的瓶颈由单瓶颈转换为多瓶颈,其对网络传输效果的影响是相同的。但网络分层后,每个子网邻近基站的最后几跳节点上的数据包数量基本相同,起到了负载均衡的作用,解决了局部负载过重的问题。本文方法对于文献[7]中的每个节点产生等量数据包的数据收集方式同样适用,具有更广的应用范围。

4 算法描述

拓扑控制算法步骤:a)对网络进行初始化,控制节点发射功率使其传输范围为 1 个单位跳距,得到各普通节点到基站的跳步数;b)利用上一步得到的跳步数,基站向网络中发送组网消息,选择某一合适的正整数 N_L 作为模值,其中模值小于或等于天线数量。对节点的跳步数进行作模取余,取相同余数的节点组成同一层次,不同天线对应不同余数层。

首先在节点被部署后,每个节点都设置自己的跳数为无穷,节点发射功率控制在 1 个单位跳距,网络以概率 1 连接。然后,每个节点广播自己的消息,这样,一个节点就可以知道其所有邻居节点的存在,并将这些信息存储到 Nlist 列表之中。接着,设置基站跳数值为零,随后让基站向网络中广播一个 hello 消息,消息中包含基站的跳数值和模值。当一个节点 u 从节点 v 接收到一个 hello 消息后,从中得到跳数值 N_v ,然后利用自己的跳数值 N_u 进行如下比较并执行相应操作:

```
void HandleHelloMsg( NETMsg msg)
{
    int Nv = msg. Nv;
    NL = msg. NL;
    if (Nu < Nv - 1)
    {
        Nu = Nv + 1;
        broadcast hello message;
        promote power of the local node;
        clean Nlist;
    }
    return;
}
(NL 为全局变量, msg. NL 消息携带的模值)
```

经过一定时延后,每个节点再次发送广播通知其邻居节点报告其存在,目的是获取节点通信覆盖范围扩大后节点的新 Nlist 列表,随后将基站天线从 0 到 N_L 依次编号,向网络中发送 deal 消息进行组网,利用第一步得到的跳步数和节点获取的模值对节点的跳步数进行作模取余,取相同余数的节点组成同一层次。路由表建立过程如下,当一个节点 u 从节点 v 接收到一个 deal 消息后,从中得到跳数值 N_v ,然后利用自己的跳数值 N_u 进行如下比较并执行相应操作:

```
void HandleDealMsg( NETMsg msg)
{
    int Nv = msg. Nv;
    if (Nu % NL = Nv % NL)
    {
        add msg. ID into Nlist;
        if (Nu / NL > Nv / NL)
            add msg. ID into Plist;
        else
            if (Nu / NL < Nv / NL)
                add msg. ID into Clist;
        else
            add msg. ID into Blist;
    }
    broadcast deal message;
    return;
}
```

每层子网的不同通信频段可利用基站的多天线和作模取得的余数值加以实现。基站一点的瓶颈问题通过采用支持 MIMO^[8] 的多天线技术可有效解决,文中不再赘述。下面将针对邻近基站的几跳节点的瓶颈问题,通过实验验证本文拓扑控制算法的区域负载平衡特性。

5 仿真实验及结果分析

本实验算法采用 VC++ 开发,实验所使用的传感器网络包含 100 个节点,随机分布在面积为 200 × 500 m² 的带状平面上。所有节点的初始通信距离为 50 m,扩展后通信距离可达 150 m。节点的初始能量为 50 J,节点用于感知、收集一个数据包消耗的能量分别为 2 × 10⁻⁵ J 和 2 × 10⁻⁵ J,一个单位跳距下传输一个数据包消耗能量 1 × 10⁻⁴ J,通信距离扩展后,传送一个数据包消耗的能量可根据关系 $P_r = \beta R^\alpha$ 确定,实验中取模值为 2。图 5 分别给出了利用算法得到的二层次网络的节点距离基站的跳步数、兄弟节点关系和最终实现拓扑控制算法的网络链路关系。图 5 中较大黑色点代表基站的天线参与数量,带有标号的小点代表普通传感器节点,最终网络节点间的链路条数远多于图 5(c) 中所示。本处为将结果更明确地加以演示,将节点间的通信距离控制在 100 ~ 120 m。

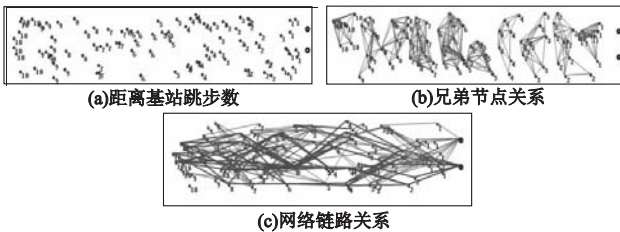


图 5 多层次传感器网络

至此根据拓扑控制算法构建了多层次网络,为了说明网络性能,本文将在层次网络上运行简单的基于表的路由协议,如根据传感器节点的父节点列表 Plist,遍历表方式选择路径转发数据。通过与采用仅满足覆盖的拓扑控制的无层次网络进行对比,来验证依据本文算法构建的层次网络的负载平衡效果。

随机产生五个传感器网络,事件包产生区域设定为横向 $X \in [100, 200]$,纵向 $Y \in [0, 100]$,区域内每个节点一个环境变化周期产生 20 个数据包。图 6(a) 给出了这五个传感器网络在采用仅满足覆盖的拓扑控制和采用本文拓扑控制算法两种情况下,同一路由协议所对应的负载最重节点的数据包总数关系。图 6(b) 给出了功率调整时 β 不变、 $\alpha = 2$ 这一条件下,网络划分层次多少对网络寿命的影响,全网每个传感器节点采用每间隔 10 s 产生一个数据包,图中 N_L 代表网络划分的层次数。

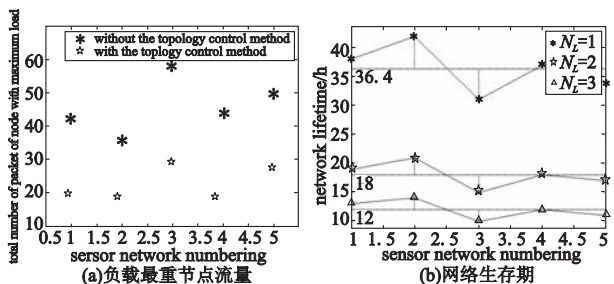


图 6 多层次网络性能

在图 6(a) 中,两种网络的负载最重节点的数据包数量浮动较大,原因在于网络中节点的部署是随机的,不同传感器网

络事件包产生区域内的节点数量均不相同,同时网络靠近基站的关键节点数量也存在差异,致使网络负载最重节点数据包流量有所差异,但可以看出,两种网络的负载最重节点的数据包流量存在近似 2 倍的关系。这就说明本文的拓扑控制算法将区域事件数据包进行了层次间分流,从而起到了负载平衡的作用。因此可断定随着网络层次数的增加,负载最重节点的拥塞度将变为一个单位跳距情况下的 $1/N_L$ 倍,与网络模型分析得出的结论相同。

由图 6(b) 可以看出,网络层次划分得越多,网络生存期越短。根据模型分析可知,能量的消耗主要体现在数据包的发射功率上,表明网络中数据包的单跳传输距离增加,这将会降低非端到端传输时延,提升网络的服务质量。实验中数据收集采用周期性上传方式,上传频繁高,能量消耗较快。对于查询频率较低或主要用来收集网络突发事件数据包 of 的传感器网络而言,网络生存期会延长很多。

综合考虑上述结果,可以得出结论:本文拓扑控制算法能够很好解决局部负载过重问题,但网络服务质量的提升是以电池能量的损耗为代价。针对传感器网络的特殊应用,需要综合考虑传感器电池容量、网络数据量和信道扰动等多方面因素,根据网络漏斗效应的严重程度来决定 N_L 的取值,使网络最终满足用户的应用需要。

6 结束语

本文提供的拓扑控制算法主要是为应对用于输电线路监测、油气管道监测等长链型数据收集传感器网络漏斗效应问题而提出的,算法起到了很好的局部区域大量突发事件数据包分流的作用。根据算法生成层次网络后,可在其上运行负载平衡路由协议,对网络资源进行两级优化,更好地平衡网络负载,从而有效地解决传感器网络的漏斗效应问题,起到缓解拥塞、提升网络服务质量的作用。

参考文献:

- [1] 张帆,熊兰,刘希钰.基于加速度传感器的输电线舞动监测系统[J].电测与仪表,2009,46(1):30-33.
- [2] SUN Li-min, LI Jian-zhong, CHEN Yu, et al. Wireless sensor networks[M]. Beijing: Tsinghua University Press, 2005.
- [3] EE C T, BAJCSY R. Congestion control and fairness for many-to-one routing in sensor networks[C]//Proc of the 2nd ACM Conference on Embedded Networked Sensor Systems. New York: ACM, 2004:148-161.
- [4] 王毅,张德运,陈海荣.无线传感器网络汇聚节点动态功耗管理方法[J].西安交通大学学报,2008,42(2):138-142.
- [5] CAI Wen-yu, JIN Xin-yu, ZHANG Yu, et al. A load-balanced minimum energy routing algorithm for wireless Ad hoc sensor networks[J]. Journal of Zhejiang University Science A, 2006, 7(4): 502-506.
- [6] PETROVIC D, SHAH R C, RAMCHANDRAN K, et al. Routing with aggregation and compression for wireless sensor networks[C]//Proc of the 1st IEEE Int'l Workshop on Sensor Network Protocols and Applications. Piscataway: IEEE Press, 2003: 156-162.
- [7] 张重庆,李明禄,伍民友.数据收集传感器网络的负载平衡网络构建方法[J].软件学报,2007,18(5):1110-1121.
- [8] 徐侃如,刘威,杨宗凯,等.无线传感器网络中一种基于分簇的能量高效虚拟 MIMO 传输策略[J].小型微型计算机系统,2008,29(2):193-198.