

面向低功耗无线传感器网络的动态重传算法

吴伟男 刘建明

(桂林电子科技大学电子工程与自动化学院 广西 桂林 541004)

摘 要 由于信道通信质量具有时变性,为了保障数据传送具备高可靠性,需要在低功耗传输网络中加入能量利用率更高的重传机制。低功耗无线传感器网络对数据实时性的要求不高,但整体能量有限,因此重传的时机及重传的有效性显得额外重要。在静态发送队列的基础上,提出一种可靠且稳定的动态重传算法。随机出现丢包的节点利用自身产生的随机数作为序列,将其添加到重传队列中。该算法利用随机生成发送时机的方式避免随机出现的通信不佳的情况。实践表明,动态重传算法在降低能耗的同时,能够提高数据传送的成功率。

关键词 无线传感器网络,动态重传,低功耗

中图分类号 TP393.06 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.06.016

Dynamic Retransmission Algorithm in Low-power Wireless Sensor Networks

WU Wei-nan LIU Jian-ming

(School of Electronic Engineering & Automation, Guilin University of Electronic & Technology, Guilin, Guangxi 541004, China)

Abstract The quality of channel communication is time-varying. In order to improve the reliability of data transmission, the retransmission mechanism with higher energy utilization should be introduced into low-power wireless sensor network. In low-power wireless sensor networks, the real-time data requirements are not high, but the overall energy is limited. Therefore, the timing and the validity of retransmission are particular important. On the basis of static queue transmission, this paper proposed a reliable and stable dynamic retransmission algorithm. The nodes which fail to send create a random number as a sequence added to the retransmission queue. The algorithm use the random transmission timing to retreat the harsh communication. The experimental results show that the dynamic retransmission algorithm improves the success rate of data transmission while reducing the energy consumption.

Keywords WSN, Dynamic retransmission, Low-power

1 引言

在信息采集的应用环境中,通过部署大量的传感器节点组成庞大的无线传感器网络系统,其中节点的位置一般比较分散且不易甚至不可回收,因此节点使用寿命的长短直接影响着整个传感网络的完整性^[1]。在实际应用中,在保证通信可靠的前提下,尽可能地降低节点的能耗显得尤为重要^[2-3]。在无线传感器网络的应用中,由于大量的无线传感节点均使用广播的方式进行通信,共享无线信道的节点间将产生竞争和通信干扰,从而使节点在数据传送的过程中出现不同程度的丢包。此外,通过实际检测发现无线信道具有很大的时变性与不稳定性,通信质量在不同时段内有较大的差异。因此,有效的数据重传机制才能保障数据传送的可靠性^[4]。在一些特定的应用中,往往不允许无线收发器更换信道,在此情况下,数据传输之间的相互规避及重传处理显得更为重要。

在传统的端到端的重传方案中,若节点接收不到基站发送的 ACK 信号,则立即进行重传处理,直到节点接收到源自

基站的 ACK 信号或等待接收数据超时后结束传送^[5]。此方法虽然在一定程度上提高了数据传输的可靠性,但并没有充分考虑到重传的有效性及重传时机的把握。在实际检测中,当节点因为信道通信质量不佳而导致数据发送失败时,链路通常需要持续一段时间才能恢复正常的通信。这段时间是随机的或未知的,因此在没有充分考虑发送时机的情况下依次进行多次发送,将提高能量的消耗量。低功耗无线传感器节点的能量有限,过多的重传开销会大大降低网络的耐用性。针对以上情况,在实时性要求不高,但对低功耗和无线网络耐用性要求较高的应用中,将静态发送队列^[6]与动态重传队列相结合作为传输机制,组成具有一定抗干扰能力的改进型发送机制。该机制能有效规避多节点通信竞争引起的丢包,同时兼顾较低的能耗开销,选择正确的发送时机能有效地完成数据重传。

2 信道通信质量时变分析

2.1 实验环境搭建

由于信号强弱分布不均,实验中使用了多跳网络拓扑。

到稿日期:2017-04-18 返修日期:2017-08-05 本文受国家自然科学基金(61262074),桂林电子科技大学研究生创新项目(YJCS201542),桂林电子科技大学计算机软件创新团队项目(桂电科 2012(04)号)资助。

吴伟男(1991—),男,硕士生,主要研究方向为无线传感器网络,E-mail:1501755307@qq.com;刘建明(1975—),男,教授,博士生导师,主要研究方向为网络性能分析及 QoS 机制、无线传感器网络,E-mail:jmliu@guet.edu.cn(通信作者)。

在高层公寓中分散部署 100 个无线传感器节点和数个路由节点。基站放置于距离该公寓 300 m 以外的另一座高层中,用于接收和处理数据。部署节点前,在相应的位置测试信号的强弱分布情况。实验中,基站向对应的终端节点发送心跳包,并开始对该次通信进行计时,最终系统能记录该次通信所需的时长。

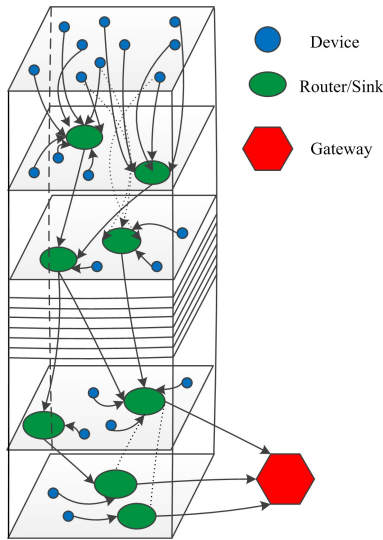


图 1 无线传感器节点在公寓楼中的硬件拓扑

Fig.1 Hardware topology of wireless sensor network in apartment building

2.2 结果分析

实验中,以 15 min 为一个周期,遍历 100 个节点进行通信时间测试,采集过程不间断进行。随机抽取 3 天的数据,在各时段内对通信质量进行实验分析。由于终端节点根据实际情况布设于公寓中,每个节点周围的环境均有可能发生不可预测的改变,如人员的暂留与走动、货物的堆砌、空中传播时风力的变化、雨水天气等外部的因素都会对该次通信造成一定的影响。

图 2 给出了某节点在某时段内单独进行 20 次通信及其双向通信的时长记录。

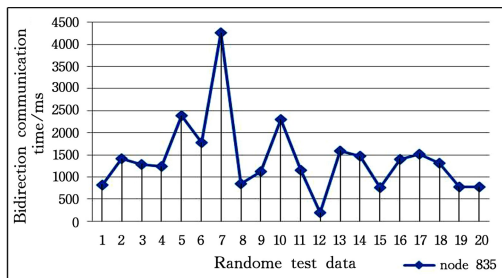


图 2 某节点随机进行 20 次通信的通信时间

Fig.2 Communication time for a node to communicate randomly 20 times

由图 2 可知,其双向通信时间具有一定的时变性,通信时长范围是 0~4300 ms,但总体集中于 1000~1500 ms。在所有的统计数据中发现,100 个节点均出现过通信时间为 0,即通信失败的情况。因此,如何在通信质量不佳的情况下,使数据尽可能有效地进行重传性是研究的重点。

3 静态发送队列重传问题

在低功耗多节点单信道通信的应用中,若实时性要求不高,节点在网络层可以使用静态发送队列进行数据发送。在传感网络通信中把时间分割成互不重叠的时间块,再将每个时间块分配给对应的用户,然后依据时隙来区分不同地址的用户信号,从而完成多址连接^[7]。在使用该方法的大部分应用中发现,为了节约电能,低功耗节点的大部分活动周期处于深度休眠中,不接收外部的唤醒信号。图 3 给出了终端节点与网关的通信机制。

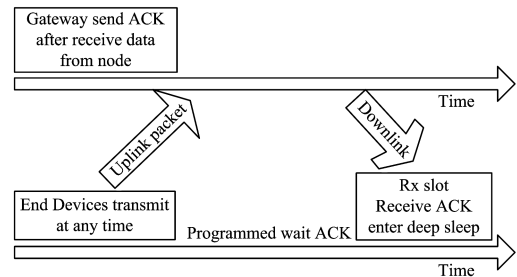


图 3 终端节点与网关的通信机制

Fig.3 Communication mechanism of terminal nodes and gateway

实验表明,网络中采用静态发送队列进行发送,可以保证大量的集群节点成功完成信息发送,同时也能在一定程度上确保数据发送的可靠性与高效性。但是,考虑到信道通信质量具有时变性,在数据发送过程中无可避免地会发生数据丢失的情况。若把立即重传加到节点各自发送的时隙中,则会出现以下问题:1)在时隙分配中,每个节点的分配时隙都将扩大,从而使整体发送周期变长。为了预留重传的时间,整个静态发送队列的发送周期都会相应增加。而发生重传的节点可能只占大量节点中的一小部分,在该种机制下,每个节点的时隙都要预留重传时间,如此一来,为了解决数个节点而去延长发送周期,将使网络不具有高效性。2)在首次发送失败后立即进行重传,这种重传可能是无效的。根据实验测试,一旦该节点与基站通信失败而链路可能需要持续一段时间才能恢复正常通信。结合实际分析,节点的通信可能会受到外界的影响,如果该环境没被恢复,仍有可能导致通信失败。因此,在低功耗终端节点能量有限的情况下,需减少无效的操作,以降低能耗开销。

4 动态重传分配策略

本节给出一种灵活且高效的动态重传策略,其基本思想是:基于静态发送队列,对数据发送失败的终端节点进行后续重传。然而,终端节点间是相互独立通信的,各终端节点间并不能得知哪些终端节点与自身一样发送数据失败而进入重传队列中。在单信道通信中,应尽可能避免通信冲撞或竞争信道失败引起的数据发送失败。使用静态发送队列的方式进行数据发送能在一定程度上避免这种冲突。现在面临的问题是如何将静态发送队列中发送失败的节点从整个网络中分离出来,对其重新进行整理与排序后,使其进行重传发送。

首先,终端节点主要有深度休眠、数据发送、等待接收 3 种状态,而深度休眠占节点生命周期的绝大部分时间,严格来

说,终端节点只与基站进行通信。网络中节点间均使用了时分多址的通信方式,在一个节点活动中,网络中其他节点均处于深度休眠状态,期间不能与其他终端节点进行数据交互,所以终端节点间是相互独立工作的。因此,终端节点在静态发送队列中发送,仅仅利用预先分配的时隙与基站进行通信,无法得知它前后的节点有无数据发送失败。终端节点在加入网络时,都必将经过一次与基站的时间同步,获得一个基站的当前时间。因此,所有的终端节点都是以基站时间为参考时间,结合自身ID号与发送间隔等信息,可以得出属于该节点的发送时隙,而在其他时段内节点均处于休眠模式。当休眠至属于该节点的发送时段时,节点自动唤醒并进行数据发送。一旦数据发送失败,节点自行进入重传机制,根据算法重新计算下一次唤醒前进入休眠的时长,当到达一定的时间后,再次自行唤醒,然后进行数据重传。我们此过程中加入了动态调整时机的重传机制,既降低了功耗,又保证了数据发送的可靠性,防止了数据丢失。

如图4所示,条纹部分是全网静态发送队列的发送区域,重传不允许发生在条纹区域,以防止打乱有序的发送队列。条纹区域是由各终端节点的时隙堆叠而成的发送区域。白色区域共分为5个小块,分别为5个重传区域。当某个节点发送数据失败,即没有收到基站回复的ACK信号时,进入重传机制,计算出下一次休眠的时长,等待下次唤醒后进行数据发送。

$$B_1 = R \text{ Mod } NS \quad (1)$$

$$ST = B + (Nm - Nn) * S_1 + B_1 * S_2 \quad (2)$$

式(1)中, R 为单片机生成的16位随机数, NS 为网络中的最大节点容量, B_1 是容量数值范围内的一个随机值。式(2)中, Nm 为网络中最大节点的ID号; Nn 是当前节点的ID号; B_1 为预先设置的缓冲时间; S_1 为静态发送队列发送时,相邻节点间的时间间隔; S_2 为重传区域内相邻节点间的时间间隔; ST 为该节点即将进入休眠的时长。

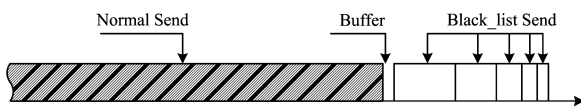


图4 正常发送与重传发送时机示意图

Fig. 4 Normal send and retransmission send timing

首次数据传送失败后,利用式(1)、式(2)计算下次休眠的时长。式中,通过网络中最大的节点ID及自身产生的随机数,绕过静态发送队列区域,随机数可以将该节点添加到动态发送队列中。各节点间是否发送失败都是未知的,但可以通过计算整体静态发送队列所需时长,来绕过静态发送队列发送区域;随后产生的随机数负责把发送失败的节点添加到动态重传队列中。以图5为例,在100个网络节点的实验中,一旦某节点发送失败(如第一段弧线箭头所示),它能获得一个1~99之间的随机数作为加入到重传队列的参数。由于随机数的产生具有随机性,样本容量较大时,多个节点产生相同随机数的概率也将大大降低,在数据传输失败也是随机发生的情况下,利用这种策略能较好地规避冲撞,并将多个发送失败的节点加入到重传队列中。

第一个重传区域进行重传,若重传失败,则执行式(3)、式(4)计算下一次休眠的时间。图5中第二段弧线为避开第一个重传区域的路线,方法同样是计算出重传区域的长度以绕过该区域。在再次主动唤醒后,进入下一个重传队列,方可进行重传。从图中亦可知道,重传区域不断缩小,因为随着重传次数的增多,数据发送失败的节点数逐渐减少,加入到下一轮的重传队列的节点也会相应减少,为了加快重传的速率,重传区域不断减小。最后,达到一定重传次数后,若依旧存在不能上传的数据,则放弃该轮数据,等待下一周期的发送队列。式(5)、式(6)设定最大重传数为 K ,计算下次进入休眠的时长 ST 。而重传最大次数 K 则根据实际情况进行调整。

$$B_2 = (R \text{ Mod } NS) / 2 \quad (3)$$

$$ST = B + (NS - B_1) * S_2 + B_2 * S_2 \quad (4)$$

$$B_K = (R \text{ Mod } NS) / 2^{K-1} \quad (5)$$

$$ST = B + (NS / 2^{K-1} - B_{K-1}) * S_2 + B_K * S_2 \quad (6)$$

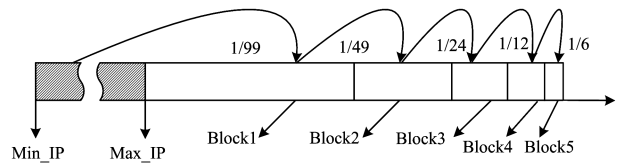


图5 示例网络中某终端节点的重传时机分配图

Fig. 5 Retransmission time distribution diagram of node in sample network

如图6所示,若某节点在紧随静态发送队列后的重传区域进行发送时依旧不能发送成功,则在下一周期静态发送到来前,选择时间跨度更大的两次随机重传。这种重传分配策略适用于某种干扰持续较长的一段时间后链路才恢复正常的情况。因此,该策略可防止该周期的数据丢失。

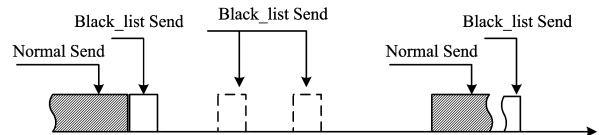


图6 两个静态发送队列在发送周期的重传发送的时段分配

Fig. 6 Time distribution of retransmission during transmission cycle of two static sending queue

5 重传区域长度分配策略

重传区域主要分布在静态发送队列之后。重传区域会随着网络节点容量的增大而增大^[8],而重传区域越长,该网络的容量越小。因此重传区域的长度与网络负载能力是相互制衡的。一方面,重传是受环境因素的影响,不同的布设环境下通信质量的优劣不一,网络中需要重传的终端节点也会有所不同。因此,需要设计一种具备统计传输成功率且自适应调整重传区域长度的重传机制。若要进行重传区域调整,则需依托于基站对网络中节点发送成功率的统计。基站通过统计一次上传成功率、一次重传成功率、二次重传成功率等信息,对当前该网络通信质量进行评估,符合要求后,在回复各终端节点的ACK信号中加入重配置信息,让终端节点在下一周期的数据发送使用新的重传机制。调整重传区域的方略有以下两种。

1)调整重传的最大次数。首次发送失败的节点,可能通过一两次的重传就传输成功。而初始值设定最大重传次数为 K ,当信道通信质量较好时,预留靠后的几个重传区域并没有作用,重传区域 $(K-3) \sim K$ 中可能完全没有节点进入。因此,可对这些区域进行削减,从而减少重传区域的长度。

2)调整重传区域队列中相邻节点间的发送间隔。在调取随机数获得重传时隙的策略中,随着重传节点数量的减少,若样本容量保持不变,则抓取相邻数值的概率会大幅下降。根据插入到重传发送队列中的随机数,多个重传的终端节点在重传队列中会相隔数个单位时间。由于终端节点在队列中的位置是随机产生的,因此在重传过程中终端节点间相隔未知个单位时间。在重传终端节点不多的情况下,重传的终端节点间相隔大于或等于 1 个单位时间,因此可通过压缩单位时间来缩短重传区域的长度。

图 7、图 8 中每个块状为一个单位时间。由于队列中发送节点的数量不多,压缩单位时间后,两节点间也会由于相隔数个单位时间而使重传的终端节点间依旧相隔较长的时间。

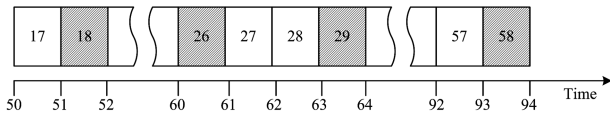


图 7 重传区域队列中待发送的 4 个排序节点

Fig. 7 4 sorted nodes to be sent in retransmission region queue

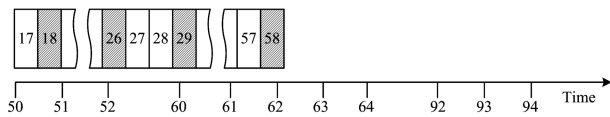


图 8 压缩后重传区域队列中待发送的 4 个排序节点

Fig. 8 4 sorted nodes in to be sent retransmission region queue after compression

对比两种方法可知,第二种方法能够更加科学、可靠地完成重传任务。在重传的过程中,重传大部分出现在前三个重传区域,若直接减少重传次数,则存在节点的随机数相同而使节点加入队列失败。若此时重传结束,则竞争失败的节点将无法重传数据。采用第二种方法,网络中重传的节点突发增多,当出现大量节点在遍历前三个重传区域后,依旧加入队列失败。由于重传区域的增多,节点加入队列的概率将增加,因此在上一轮竞争失败的节点,在下轮中可以尝试重新加入队列。

6 性能测试与分析

在相同的网络拓扑和相同的网络布设环境中,分别测试 3 种不同重传机制的发送成功率。每种发送机制均在实际布设环境中运行 5 天。实验中,通过修改数据发送的时段,故意在网络中引入干扰源,例如引入楼道中人流不同造成的环境差异、楼道中随机物体的堆砌等外界干扰。因此,在此实验中测试的成功率为多种干扰条件下多次接收成功率的平均值。

由图 9 可知,在没有加入重传机制的发送系统中,如图中浅灰色柱状图所示,其发送成功率为 $85\% \sim 90\%$ 。在该机制下,端到端的发送成功率完全以当前链路通信质量为依托,没有对外界干扰作出任何响应。这是最节约能耗的工作机制,

但数据的发送可靠性最低。相比之下,在加入立即重传机制的系统中,其发送成功率有所提升,如图中白色柱状图所示,成功率为 $92\% \sim 96\%$ 。该模式采用了限定最大次数的立即重传机制,通过多次发送,有效地提高了数据发送的成功率,其对环境干扰有一定的抵抗性,但能耗开销有所增加。但由于重传发送的时机过于集中,一旦链路在该时段内无法通信,则不能规避该时段的通信不佳并且不能将数据发送至基站端。加入动态重传机制的系统有较高且稳定的发送成功率,如图中深灰色柱状图所示,其成功率为 $99\% \sim 100\%$ 。该重传机制在重传时机上优于立即重传,可靠性较高。

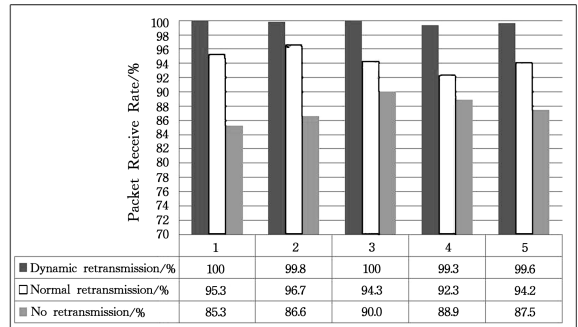


图 9 3 种发送机制的发送成功率柱状图

Fig. 9 Send success rate histogram of three kinds of sending mechanism

在能耗方面,由于实际运行中动态重传等于或低于立即重传机制的发送次数,因此能耗会有所降低。两种机制都设置了相同的最大重传次数,区别则在于发送时机不同。对比两种重传时机的选择,立即重传在一小段等待接收基站 ACK 信号超时后,重传数据,若通信依旧失败,则循环重传 K 次,直到某次成功后退出重传机制。因此,可能在通信环境没有变好的情况下做了多次相同重复的重传,而数据依旧没有成功到达基站,从而增大了能量开销。而对于动态重传,在一次重传通信失败后,其离散性地选择下一次重传时机,在下次重传产生之前,因为相隔时间较长,信道通信质量改变的概率大大增加,所以提高了每次重传的成功率,同时也降低了重传所带来的额外开销。

图 10 给出了节点部署于相同的位置,在随机加入扰动后,3 种不同重传机制在 10 天内的电能消耗总量。

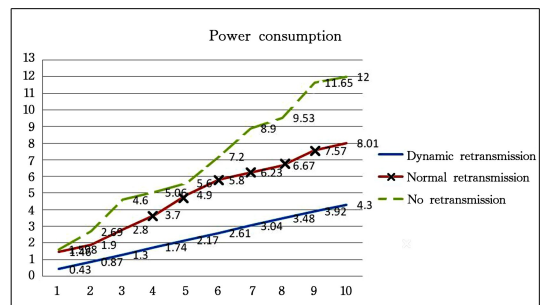


图 10 引入大量干扰源后 3 种不同重传机制的电能消耗统计
Fig. 10 Electricity consumption statistics of three different retransmission mechanisms after introducing a large number of interference sources