

文章编号:1007-6735(2013)03-0215-06

无线传感器网络自适应拥塞控制的路由算法分析

李路伟, 杨洪勇

(鲁东大学 信息与电气工程学院, 烟台 264025)

摘要: 无线传感器网络中节点的覆盖范围有限,因而采用多跳路由传输方式.无线自组网中的多跳路由是由普通节点协作完成的,选择不同的转发节点,会对网络的信息传输产生不同的影响.对不同路由(洪泛路由、最短路径等)算法下的网络自适应拥塞控制进行了分析,研究了不同路由算法下的网络性能和拥塞控制效果.根据节点跳数与缓存占用的关系,提出一种基于节点跳数和缓存占用的性能函数的改进最短路径算法,算法选取使性能函数值最小的节点作为转发节点.最后,通过实验比较了最短路径算法与改进路由算法的网络性能,发现改进路由算法相比最短路径算法,具有较好的网络性能和服务质量.

关键词: 无线传感器网络; 自适应; 拥塞控制; 路由算法
中图分类号: TP 3 **文献标志码:** A

Analysis on Routing Algorithms of Adaptive Congestion Control of Wireless Sensor Networks

LI Lu-wei, YANG Hong-yong

(College of Information and Electrical Engineering, Ludong University, Yantai 264025, China)

Abstract: Since the coverage of wireless sensor network nodes is limited, the transmission of multi-hop routing is adopted. The multi-hop routing of wireless ad hoc networks is realized by common nodes cooperation and selecting different forwarding nodes will have a different effect upon the information transmission. By analyzing the network adaptive congestion control with different routing algorithms such as the flooding routing and the shortest path algorithm, the performances of network and congestion control with different kinds of routing algorithms were studied. According to the relationship between hop count and cache occupied, a kind of improved shortest path algorithm based on the performance function of node hop count and cache occupied was proposed, where the node with minimum value of function was selected as the forwarding node. Experiment examples were used to compare the network performances of the shortest path algorithm and the modified routing algorithm. It is shown that the modified one has better network performance and service quality than the shortest path algorithm.

Key words: *wireless sensor networks; adaptability; congestion control; routing algorithms*

收稿日期: 2012-10-21
基金项目: 国家自然科学基金资助项目(61273152); 山东自然科学基金资助项目(ZR2011M017)
第一作者: 李路伟(1988-), 男, 硕士研究生. 研究方向: 网络通信技术. E-mail: luweili1988@126.com
通讯作者: 杨洪勇(1967-), 男, 教授. 研究方向: 网络拥塞控制、复杂网络、多智能体协调控制等. E-mail: hyyang@yeah.net

与物理世界紧密结合的无线传感器网络具有局部区域大规模部署、节点资源有限、无线带宽小、拓扑结构动态变化等特点^[1].此外,还具有多对一的通信方式、无线链路的相互干扰、网络的动态变化和资源受限等特性,使得无线传感器网络极易产生拥塞,严重影响网络的服务质量和生存周期,因此拥塞控制成为保障无线传感器网络服务质量的关键技术之一^[2-5].

网络路由是指设备从一个接口上收到数据包,并根据数据包的目的地址进行定向并转发到另一个接口的过程.在无线传感器网络的应用中,每一个节点都是一个终端,其路由问题就是产生数据的节点如何通过其它节点与 sink 节点建立起链接的问题^[6-9].

目前,无线传感器网络的应用环境千差万别,使得没有一个路由机制适合所有的应用,因此无线传感器网络的路由协议具有很强的应用相关性.另外,WSNs(wireless sensor networks)节点随机分布,是以数据为中心的网络,所以传统的无线路由协议在此并不适用.进而,根据不同应用对 WSNs 敏感性要求的不同,可将无线传感器网络路由协议分为以下5种类型:数据查询路由、能量最优路由、位置信息路由、可靠路由和可编程路由^[10].总的来说,目前WSNs路由技术的研究还仅仅处于起步阶段,虽然发展迅速、成果显著,但距离很好地支持更高要求的无线传感器网络应用,还有很大的差距^[11-13].

本文将路由技术与拥塞控制技术二者结合起来考虑,比较了洪泛路由、最短路径算法下自适应拥塞控制的网络性能.针对最短路径算法存在的不足之处,提出一种基于跳数和缓存占用的改进路由算法,并通过实验比较和分析了最短路径算法与改进路由算法在自适应拥塞控制下的网络性能.

1 无线传感器网络的自适应拥塞控制

在研究中,假设在一定区域内所有节点随机自组形成网状结构模型.此时,一个节点可以与多个节点进行通信.实际应用中,通常采用播撒一定数量节点到某一区域内的做法来实现对某一区域特定事件的观测,这个时候的网络结构趋向为图1所示的网状拓扑结构.

本文将以网状拓扑结构为模型,进行不同路由算法下基于自适应拥塞控制的研究.假设拥塞调节策略为带上、下跳节点协同进行拥塞调节的机制,该

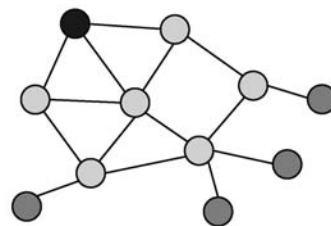


图1 网状拓扑结构图

Fig.1 Mesh topology diagram

拥塞控制算法包含以下几部分:

a. 节点发送分组之前,判断是否有增加发送速率的请求.

令 $request$ 表示节点增加发送速率的请求.当 $request = 1$ 时,有增加发送速率的请求;否则,请求无效,按原速率进行发送.

Begin

if ($0 \leq breg < b2$)

$request = 0;$

$packet_send(ratio_present);$

end

if ($b2 \leq breg$)

$request = 1;$

$request_send(request);$

end

End

其中, $breg$ 表示节点缓存占用, $b2$ 是算法中对节点缓存空间进行划分的标记值之一,此外还有 $b1$ 和 $b3$. $b1, b2, b3$ 依次增大,主要用来对缓存空间进行划分,便于节点根据缓存占用处于不同的划分区间对传输速率、操作请求等进行自适应控制. $request_send()$ 表示请求发送操作, $packet_send()$ 为分组发送操作, $ratio_present$ 表示节点的当前分组发送速率.

b. 接收节点对增加发送速率的请求作出反馈.

假设允许节点增加发送速率的最大值为 $ratio_initial$, $gain_range$ 表示允许增加发送速率的幅度.

Begin

if ($request = 1$)

if ($0 \leq breg < b1$)

$gain_range = ratio_initial * \beta1/node_max;$

end

if ($b1 \leq breg < b2$)

$gain_range = ratio_initial * (\beta1 -$

```

         $\beta)/node\_max;$ 
    end
    if (  $b2 \leq breg$  )
         $gain\_range = 0;$ 
    end
    request_feedback( $gain\_range$ );
end
End

```

其中, β_1 是在研究中预设的两个拥塞阈值之一; 另一个阈值为 β_2 , 两者依次增大, 主要用来控制缓存占用的拥塞程度; β 表示节点的拥塞程度. 首先, 判断是否有增加发送速率的请求. 若 $request = 1$, 有请求, 此时, 若接收节点缓存占用的值 $breg \in [0, b1)$, 则此时发送节点发送速率增加的幅度为一个平均值, 即 $ratio_initial * \beta_1 / node_max$; 若接收节点缓存占用的值 $breg \in [b1, b2)$, 则将拥塞阈值 β_1 与节点当前拥塞程度 β 间的差值在多个发送节点间进行平均分配, 并乘上发送速率增加的最大值. $node_max$ 表示可以发送分组到该节点的所有邻居节点数目. 若接收节点缓存占用的值 $breg \in [b2, \infty)$, 则拒绝节点请求, 不允许增加. request_feedback() 表示请求反馈函数.

c. 节点收到分组后, 根据缓存占用 $breg$ 计算拥塞程度值 β . β 计算算法^[14]为

```

Begin
    if (  $0 \leq breg < b1$  )
         $\beta = 0;$ 
    else if (  $b1 \leq breg < b2$  )
         $\beta = (breg - b1) * \beta_1 / (b2 - b1);$ 
    else if (  $b2 \leq breg < b3$  )
         $\beta = (breg - b2) * (\beta_2 - \beta_1) / (b3 - b2) + \beta_1;$ 
    else if (  $b3 \leq breg$  )
         $\beta = \beta_2;$ 
    else
        return;
    end
End

```

其中, $breg$ 表示节点缓存占用, $bi (i = 1, 2, 3)$, β_1 与 β_2 如前所述.

d. 根据节点拥塞程度 β , 执行速率调节算法, 计算下一分组接收的最大速率. 分组速率调节算法为

```

Begin

```

```

    if (  $0 \leq breg < b1$  )
         $ratio = ratio\_initial;$ 
    end
    if (  $b1 \leq breg < b2$  )
         $ratio = ratio\_initial * (1 - \beta);$ 
    end
    if (  $b2 \leq breg < b3$  )
         $ratio = ((\beta_2 - \beta) / (\beta_2 - \beta_1)) * ratio\_initial * (1 - \beta);$ 
        packet_lose( $a$ );
    end
    if (  $b3 \leq breg$  )
         $ratio = 0;$ 
        packet_lose( $b$ );
    end
End

```

其中, 初始分组发送速率为 $ratio_initial$. packet_lose(x) 表示分组丢弃操作, $x \in (0, 1)$, 表示节点将以 $x * ratio_initial$ 的速度丢弃分组. 参数 a, b 是预设的两个分组丢失比例, 取值较大时, 节点丢弃分组的速率也会很大, 有助于迅速减小缓存占用. 下一分组发送时, 只准许分组发送速率小于或等于 $ratio$.

2 3 种路由算法下的自适应拥塞控制性能分析

对 3 种不同的路由算法: Flooding 算法、Gossiping 算法、最短路径算法 (Shortest) 进行分析, 比较基于网络自适应拥塞控制下 3 种路由算法的性能.

2.1 3 种路由算法简介

Flooding 是一种较传统的网络通信路由协议. Flooding 算法对网络的拓扑结构和相关的路由计算并没有要求, 所以其路由算法下的节点采用广播方式来传递分组. 为克服 Flooding 算法的固有缺陷, Hedetniemi 等^[15] 提出闲聊式策略 Gossiping 算法, 使用随机性原则, 即节点发送数据时不再采用广播形式, 而是随机选取一个相邻节点转发它接收到的数据副本. 最短路径算法可以使每个节点在分组发送之前, 选择最小跳数的节点作为转发节点, 继而完成分组的发送和接收工作.

由于无线传感器网络多跳的数据传输方式, 其拓扑结构更像一个分布式动态系统. 本文基于有向图的基本理论对无线传感器网络的拓扑结构进行建模.

2.2 3种路由算法下的自适应拥塞控制性能分析

通过比较自适应拥塞调节机制下3种路由算法的网络性能,分析产生性能差异的原因,如图2所示.

假设100个节点随机均匀地播撒在 $100\text{ m} \times 100\text{ m}$ 的方形区域内,sink节点位于该方形区域的右边界中点上.每次实验随机选取20,40,60或80个节点作为源节点进行分组的发送.节点缓存队列划分: $b_1 =$

200, $b_2 = 400$, $b_3 = 500$;节点缓存最大容量: $cache = 600$;拥塞阈值: $\beta_1 = 0.4$, $\beta_2 = 0.6$;分组丢弃比例: $a = 0.1$, $b = 0.3$;节点初始分组发送速率: $ratio_initial = 10$;源节点数目: $source\ node_num = 20/40/60/80$;所有节点数目: $sum_node_num = 100$;sink数目: $sink_num = 1$;网络发送 packets 任务: $400/node$;分布区域面积: $100\text{ m} \times 100\text{ m}$.

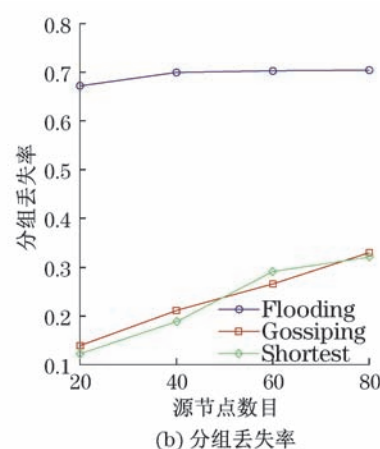
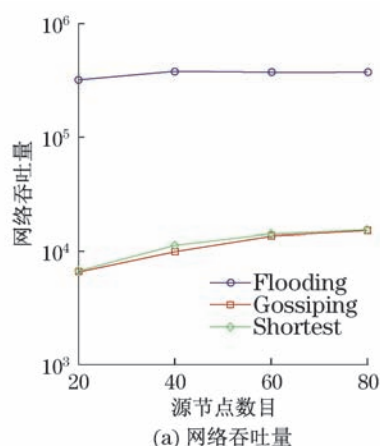


图2 3种路由算法的网络性能对比图

Fig.2 Network performance comparison between the three routing algorithms

图2(a),(b)分别表示3种路由算法下,随机选取源节点数目为20,40,60,80时,网络的吞吐量和分组丢失率对比图.由图2(a)可知,最短路径算法的网络吞吐量要高于Gossiping算法而低于Flooding算法.由于Flooding算法下,节点在进行分组传输时,除对其发送分组的节点外,节点会向其所有邻居节点发送相同数目的分组,该做法会使网络中分组数量成倍增加,网络吞吐量大幅增长.而Gossiping算法使用随机性原则,放弃广播形式,随机选取一个邻居节点转发分组,相比Flooding算法,网络吞吐量下降较大.最短路径算法介于两者之间,选取到sink节点的最短路径节点进行分组传输,相比Gossiping算法,网络吞吐量有所增加.

由图2(b)可见,Flooding算法下的分组丢失率一直很高.而最短路径算法在分组丢失率上要明显好于Flooding算法,且与Gossiping算法接近.由于“消息”内爆,Flooding算法会产生大量分组丢失.与之相比,Gossiping策略解决了“消息”内爆问题,降低了分组丢失率.最后,最短路径算法具有实现简单、路径最短、时延最短等优点,能够将分组快速传递到sink节点,但也存在某一路径节点传输压力过大,造成分组大量丢失的问题.因而在保证分组快速传输的同时,解决某一路径负载过重问题,有利于改

善最短路径算法的网络性能.

3 改进最短路径算法下的拥塞控制

通过研究发现,最短路径算法在节点选择时,可能会选择缓存空间剩余较小的节点作为其接下来的分组转发节点,而这无疑会增加节点的拥塞处理压力,造成分组的丢失.针对最短路径算法的上述不足之处,提出一种基于节点跳数和缓存占用的改进最短路径算法(Modified Shortest).

基于有向图的基本理论对无线传感器网络的拓扑结构进行建模.假设 n 阶加权有向图 $G(V, E, K)$,其中,节点集 $V = \{v_i\}$, $i \in L = \{1, 2, \dots, n\}$,边集 $E = \{e_{ij} = (i, j)\}$,加权邻接矩阵 $K = (k_{ij}) \in \mathbb{R}^{n \times n}$ ($k_{ij} \geq 0$, $i, j \in L$ 且 $i \neq j$).

3.1 改进最短路径算法

基于节点跳数和缓存占用的路由算法,通过将跳数与下一跳节点缓存占用两者综合考虑,实现适当降低传输速率的同时,缓解某一路径传输压力,降低分组丢失率的目的.

改进最短路径算法以缓存占用和节点跳数的函数 $weight(pocketnum, hopcount)$ 为依据,动态确定路由路径,主要包括以下几个部分:

a. 缓存占用与节点跳数的性能函数

如分组发送节点为 S , 其邻居节点分别为 A, B, C , 则它们的缓存占用与节点跳数分别为 $packetnum_A, packetnum_B, packetnum_C$ 和 $hopcount_A, hopcount_B, hopcount_C$. 节点在完成分组发送或接收后, 计算其 $weight$ ($packetnum, hopcount$) 值, 权值计算公式为

$$weight(packetnum, hopcount) = \alpha * hopcount / max_hopcount + (1 - \alpha) * packetnum / cache \quad (1)$$

其中, $max_hopcount$ 表示节点到 sink 的最大跳数. α 表示控制函数值 $weight$ 对节点跳数和缓存占用的侧重程度, α 愈大, 对跳数越为侧重, 相应传输速度也大; 反之, 则小. $cache$ 表示节点的缓存大小. 显然, 节点权值不再是一个预先设计好的定值而是一个随网络状况不断变化的自适应值.

b. 请求 $weight$ 值

节点发送分组前, 请求各邻居节点的 $weight$ 值, 邻居节点反馈 $weight$ 值请求

Begin

send($weight_request$); % $weight$ 值查询请求操作

send($weight$); % 节点反馈查询请求操作

End

其中, $weight_request$ 表示对邻居节点 $weight$ 值的查询请求. 节点反馈查询请求, 将自己的 $weight$ 值发送给请求节点.

c. 确定转发节点

节点 S 在收到 3 个邻居节点的 $weight$ 值 $weight_A, weight_B, weight_C$ 后, 执行算法为

Begin

$mini_weight = weight_A$;

if ($mini_weight \leq weight_B$)

if ($mini_weight \leq weight_C$)

choose(A);

else

choose(C);

else

if ($weight_B \leq weight_C$)

choose(B);

else

choose(C);

end

End

其中, $mini_weight$ 表示最小 $weight$ 值, 节点 S 根据比较, 选定 $weight$ 值最小的节点, 并向其发送分组. $weight$ 值愈小, 说明其距离 sink 节点愈近或分组占用缓存越少. 节点选定其下一跳节点后, 完成分组的发送, 相应的下一跳节点完成分组的接收. choose() 为节点选取操作.

假设 $\alpha = 0.8, cache = 20$, 各节点缓存占用 $packetnum = 10$ 时, 根据前述算法, 可以得到类似图 3 所示的节点 $weight$ 值.

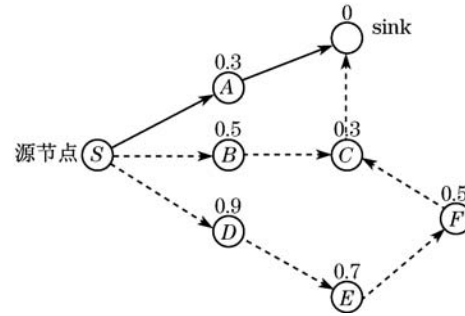


图3 改进最短路径算法下分组传输

Fig.3 Packet transmission of improved shortest path algorithm

由图 3 可以看到, 从源节点到汇聚节点 sink, 需要经过多个中间节点, 每个节点都有一个 $weight$ 值, 该值由上述算法计算得到. 分组发送时, 选择 $weight$ 值最小的节点作为其下一跳节点, 即值为 0.3 的节点. 对于 sink, 假定其缓存无限, 故其缓存大小可以始终设为 0, 所以其权值也为 0.

由求最短路径的 Dijkstra 算法可以得知, 分组总是可以在一次传递过程中, 找到一条到达 sink 节点的最短路径, 而这条路径则是改进后的最短路径.

3.2 改进最短路径算法下自适应拥塞控制性能分析

实验中, 通过随机选取一定数量的源节点进行分组的发送, 比较基于节点跳数和缓存占用算法与最短路径算法的网络吞吐量和分组丢弃率.

假设 100 个节点随机均匀地播撒在 $100\text{ m} \times 100\text{ m}$ 的方形区域内, sink 节点位于该方形区域的右边界中点上. 实验设置与上一实验相同. 此外, α 的取值为 0.8.

图 4(见下页)给出了不同路由算法下, 随机选取源节点数目为 20, 40, 60, 80 时, 网络采用自适应拥塞控制机制所得到的网络性能比较图.

由图 4(a) 可见, 随着源节点数目的增多, 两种算法的网络吞吐量都不断增加, 但超过一定的节点数量后, 改进算法的网络吞吐量要高于最短路径算

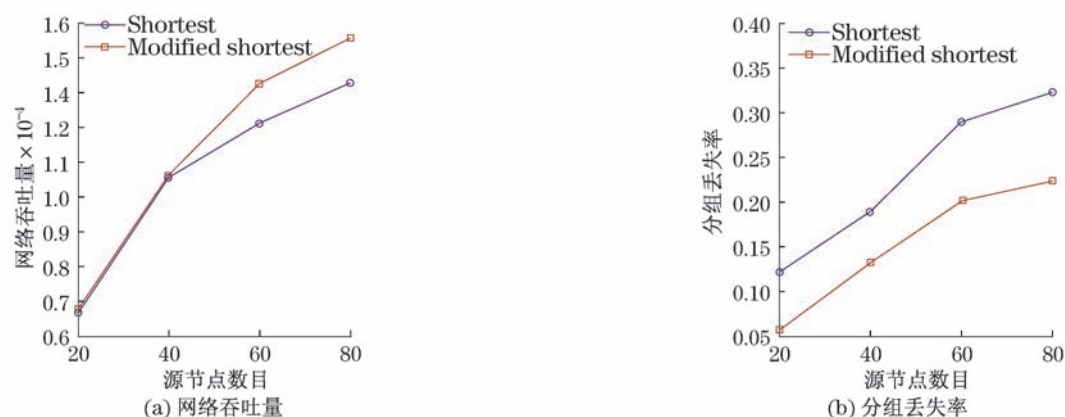


图4 两种路由算法下的网络性能对比图

Fig.4 Network performance comparison of the two routing algorithms

法.由图4(b)可以看出,随着源节点数目的增多,两者的分组丢失率都在不断上升,但很显然,改进最短路径算法的分组丢失率要低于最短路径算法.其中,“Shortest”表示最短路径算法,“Modified shortest”表示改进最短路径算法.由图4不难看出,选取相同数目源节点的情况下,改进最短路径算法的网络性能相比最短路径算法要好一些.改进最短路径算法,在下一跳节点的选择上,侧重于选择那些缓存空间剩余较多的节点,以避免过多分组的丢失,由此,在网络性能上较最短路径算法要好.

4 总结

首先分析了Flooding、Gossiping、最短路径等路由算法在自适应拥塞控制机制下的网络性能.然后,由于最短路径算法可能导致某一路径节点负载过重,综合考虑节点跳数与缓存占用,提出一种改进最短路径算法,并以图论思想予以证明和讨论.最后,通过实验比较了最短路径算法与改进最短路径算法的网络性能,发现基于自适应拥塞控制机制下的改进最短路径算法相比最短路径算法,具有较好的网络性能.

参考文献:

[1] 李凌,周兴社,李士宁,等.基于无线传感器网络的拥塞控制算法的研究与比较[J].计算机应用研究,2006,23(3):11-13.

[2] 孙利民,李波,周新运.无线传感器网络的拥塞控制技术[J].计算机研究与发展,2008,45(1):63-72.

[3] 崔莉,鞠海玲,苗勇,等.无线传感器网络研究进展[J].计算机研究与发展,2005,42(1):163-174.

[4] Wang C, Li B, Sohraby K, et al. Upstream congestion control in wireless sensor networks through cross-layer optimization [J]. IEEE Journal on Selected Areas in Communication, 2007, 25(4): 786-795.

[5] 文浩,林闯,任丰原,等.无线传感器网络的QoS体系结构[J].计算机学报,2009,32(3):432-440.

[6] Cheng Z, Heinzelman W B. Searching strategy for multi-target discovery in wireless networks [C] // Proceedings of the 4th Workshop on Applications and Services in Wireless Networks, ASWN '04. Boston, 2004.

[7] Rezaiifar R, Makowski A M. From optimal search theory to sequential paging in cellular networks [J]. IEEE Journal on Selected Areas in Communication, 1977, 15(7): 1253-1264.

[8] 于海斌,曾鹏,梁犇.智能无线传感器网络系统[M].北京:科学出版社,2006.

[9] 胡仕强.无线传感器网络的路由协议分析研究[J].机械与电子,2010,7(1):26-28.

[10] 廖鹰,王育勤,陈楚湘.无线传感器网络路由技术研究[J].计算机工程与设计,2008,29(9):2156-2159.

[11] 孙姬,陈霞,谈振辉.无线传感器网络路由技术浅析[J].传感器世界,2005(11):30-34.

[12] 郭春学,肖丽.基于蚁群算法的低能耗LEACH协议分析[J].上海理工大学学报,2010,32(1):99-102.

[13] 于林峰,肖丽.一种基于WSN的协议改进算法分析[J].上海理工大学学报,2009,31(4):406-408.

[14] 李路伟,杨洪勇.基于RED的无线传感器网络的拥塞控制[J].计算机仿真,2012,29(3):13-17.

[15] Hedetniemi S M, Hedetniemi S T, Liestman A L. A survey of gossiping and broadcasting in communication networks [J]. Networks, 1988, 18(4): 319-349.

(编辑: 丁红艺)