

# 一种自适应的无线传感器网络拥塞缓解机制

罗娟, 唐文胜, 王威

(湖南师范大学计算机教学部, 长沙 410081)

**摘要:** 提出一种自适应的拥塞缓解机制。当无线传感器网络发生拥塞时, 尝试在拥塞节点附近建立新路径, 路径建立失败时引入公平汇聚机制, 按汇聚数据流权重大小比例限制上游节点速率以减少源节点数据发送率。仿真实验表明, 该机制减少了能耗, 降低了丢包率, 使重要的数据能实时、公平地传送到目的节点。

**关键词:** 传感器网络; 拥塞缓解; 公平汇聚

## Adaptive Scheme for Alleviate Congestion in Wireless Sensor Networks

LUO Juan, TANG Wen-sheng, WANG Wei

(Department of Computer Teaching, Hunan Normal University, Changsha 410081)

**【Abstract】** This paper proposes an adaptive scheme to alleviate congestion in sensor networks. It attempts to alleviate the congestion by creating more routing paths. When creating new path is failed, the paper introduces aggregate fairness model to reduce forwarding rate in proportion to upstream link's aggregate flow weight and to alleviate the congestion. Simulations demonstrate that the scheme can improve energy efficiencies, reduce the number of dropped packets and ensure that some important data packets can reach sink node timely and fairly.

**【Key words】** sensor networks; alleviate congestion; aggregate fairness

无线传感器网络(WSN)具有大规模密集部署、节点资源受限、无线带宽小、拓扑结构动态变化等特点。由于感知节点的数据发送距离有限、感知节点距离Sink节点较远, 因此无线传感器网络是基于多跳(multi-hop)的数据传输方式和多对一的通信模式。而被感知事件(如地震或火灾)产生的突发数据流, 可能导致网络拥塞<sup>[1-2]</sup>, 尤其是靠近基站(base station)附近<sup>[2]</sup>。拥塞节点缓冲区丢包导致了全局信道质量的下降和丢包率的上升, 增大了延迟, 降低了能源的利用率。感知事件产生的突发数据流通常包含了重要的信息, 这些数据包的丢失可能产生严重的后果。同样, 如果在拥塞后只是简单地限制源速率将会使重要的信息不能实时地到达sink节点, WSN的实时监测功能受到限制。所以, 发生拥塞后在缓解拥塞的过程中如何保证数据包能够公平、可靠、实时地到达sink节点是WSN中亟待解决的热点问题之一。

### 1 拥塞缓解机制现状及分析

ESRT<sup>[1]</sup>采用基于节点的本地缓冲监测的拥塞检测机制, 根据网络的当前状态调节节点速率, 该算法不适用于发生短暂拥塞、大规模的网络。文献[2]提出了能源有效的拥塞控制模式CODA(Congestion Detection and Avoidance), 该模式由3种机制组成: 基于接收端的拥塞检测, 开环Hop-by-Hop的后压(Backpressure)机制和闭环多源调节。文献[3]中提出的融合策略研究了3种拥塞控制技术: Hop-by-Hop的流控制, 源速率限制模式和有优先级的MAC层协议。它们实现于传统协议栈的不同层次, 三者的有机结合明显地改善了网络的有效性、公平性和信道丢失率。但是, 除了文献[3]中源速率限制模式考虑到了公平性, 其他的几种策略都没有考虑公平性。文献[4]提出的公平汇聚的局部算法根据拥塞节点的局部信息计算各源节点的数据量权重, 并按权重比例减少各源节点

的数据发送率。该办法使信道分配达到了一定的公平性。但文献[5]认为很多拥塞是由突发的感知信息的数据流引起, 是短暂的, 简单地限制源速率是不合适的。在文献[5]中提出了在发生拥塞时增加资源供应, 形成一条或多条额外的路径参与传送数据直到拥塞解除。该方法在一定程度上有效地利用了整个网络的资源, 减少了丢包率。但当找不到符合条件的节点建立新路径的时候, 不但没有缓解拥塞, 而且增加了能耗。同时在建立新路径以后, 也可能形成新的拥塞节点。因此, 通过一种合适的拥塞缓解机制来解除拥塞问题是当前无线传感器网络急需解决的热点问题之一。

### 2 自适应的拥塞缓解机制

本文提出了一种自适应的拥塞缓解机制。当某个节点发生拥塞以后, 首先由拥塞节点搜索邻居节点表, 从第1个邻居节点开始比较拥塞度和剩余能量(比较拥塞度是为了避免建立新路径以后形成新的拥塞, 比较剩余能量大小是为了避免建立新路径以后, 节点能量不足而使路径失效)。如果当前搜索节点满足条件即拥塞度 $C$ 小于设定的阈值, 剩余能量 $E$ 大于阈值时, 可以作为新路径的一跳。然后, 从该节点开始按照同样的方法继续寻找下一跳节点, 直到建立起新路径。此方法的缺点是当找不到满足条件的节点建立新路径时, 拥塞无法解除, 同时还增加了能耗。本文引入了公平汇聚算法<sup>[4]</sup>, 当新路径建立失败以后, 根据数据量的比例减少上游节

**基金项目:** 湖南省自然科学基金资助项目(07JJ3129); 湖南省教育厅基金资助项目(06C521); 湖南省青年骨干教师培养基金资助项目

**作者简介:** 罗娟(1983-), 女, 硕士研究生, 主研方向: 无线传感器网络; 唐文胜(通信作者), 副教授、在职博士研究生; 王威, 硕士研究生

**收稿日期:** 2007-03-30 **E-mail:** tangws@hunnu.edu.cn

点的数据发送率以缓解拥塞。

### 2.1 建立新路径

每一个节点都有邻居节点的信息表。如表 1 是一个邻居节点信息样表，包括唯一的标识号 ID、节点的拥塞度、剩余能量和与不同数据流的距离。当一条初始路径建立以后，这些节点将自己的信息作为数据包的报头周期性地发送给邻居节点。邻居节点收到信息以后更新该表，以维护其实时性。新路径建立过程如图 1 所示。

表 1 邻居节点信息样表

邻居节点	拥塞度	剩余能量
5	0.8	0.6
9	0.4	0.2
10	0.5	0.8
6	0.6	0.7

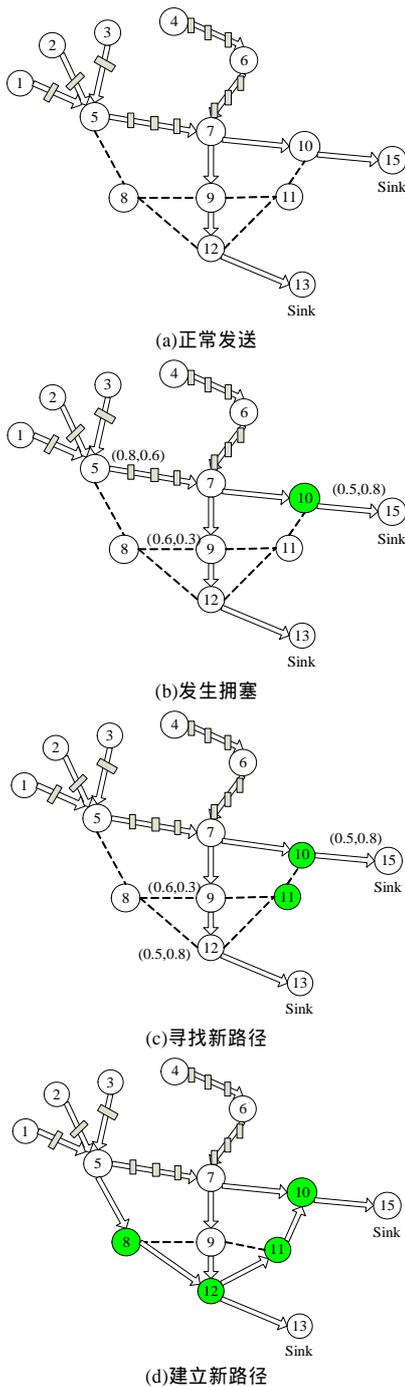


图 1 新路径建立过程

在图 1(a)中，节点 1~节点 4 均为源节点。其中节点 1~

节点 3 分别发送了一个数据包，但节点 4 发送了 3 个数据包，2 个数据流都经过节点 7。当节点 7 发生拥塞，在其邻居表中找到一个满足条件的候选节点。此处设定满足条件的节点拥塞度的阈值为 0.7，剩余能量阈值为 0.6。查找邻居节点信息表，节点 5 的拥塞度为  $0.8 > 0.5$ ，故先排除。节点 9 的拥塞度  $0.6 < 0.7$  满足条件，但剩余能量  $0.3 < 0.6$ ，不能作为新路径中的节点。节点 10 满足条件，可以作为新路径的发起者如图 1(b)。然后，节点 10 按照同样的方法找到节点 11，节点 11 可以作为新路径的下一跳节点。同理，找到节点 12 和节点 8。至此，新路径建立成功。

新路径建立以后，节点 5 将部分信息发送给新路径中节点 8。路径的发起者(节点 10)继续接收信息，查找信息表。如果发现初始路径上游节点的拥塞已经缓解，则向新路径的上游节点发出反馈信息取消该路径。

如果邻居节点信息表中找不到满足条件的节点，新路径建立不成功，不但拥塞无法缓解，还增加了能耗。所以，检测到找不到满足条件的节点时，停止寻找新路径，自适应拥塞缓解机制对源速率进行限制。

### 2.2 源速率限制

当检测到找不到满足条件的节点时，本文引入了文献[4]提出的公平汇聚算法的思想按比例减少上游节点的数据发送率，在每个节点发送的数据包权重相同的情况下，保证每个节点都有同等的到达 sink 节点的机会。如图 1(a)中，节点 1~节点 4 均为源节点。其中，节点 1~节点 3 发送了一个数据包，节点 4 发送了 3 个数据包。拥塞在节点 7 发生后，要减少源节点的发送率以缓解拥塞。只有分配 75%的信道分配给节点 5，25%分配给节点 6，才能够最好地保证每个节点都能将数据公平的传到 sink 节点。但如何实现是本文工作的难点，引入的公平汇聚思想，按汇聚数据流权重的比例分配信道能够较好的解决这个问题。描述如下：

同一个节点在同一时刻可能有不同的数据流经过，某个链路  $i, j$  在同一时刻也可能有多个数据流经过。令  $r_{i,j}$  表示所有数据流经过链路  $i, j$  的速率之和； $r_i$  表示所有数据流经过  $i$  点的速率之和； $d(s)$  表示某个源节点产生数据流发送速率。

$$r_{i,j} = \sum_{s \in N} r_{i,j}(s) \quad (1)$$

$$r_i = \sum_{s \in N} r_i(s) \quad (2)$$

$w(i)$  表示节点  $i$  产生数据流权重。令  $F_i$  为经过节点  $i$  所有数据流的汇聚数据流权重之和，如式(3)：

$$F_i = \sum_{s \in N} f_i(s) = \sum_{s \in N} \frac{r_i(s) \times w(s)}{d(s)} = \sum_{k \in U_i} F_{k,i}(s) + w(i) \quad (3)$$

其中， $d(s)$  表示数据在节点转发过程中没有任何延迟和丢包时的传播速率。式(3)表示经过节点  $i$  所有数据流的汇聚数据流权重等于  $i$  点上游节点数据流权重之和加上  $i$  本身产生数据的权重。

令  $U_i$  表示节点  $i$  的上游节点， $D_i$  表示节点  $i$  的下游节点。 $k \in U_i$ ， $j \in D_i$ ， $F_{k,i}$ ， $F_{i,j}$  分别表示  $i$  点的上游和下游节点汇聚数据流权重。

$$F_{k,i} = \sum_{s \in N} f_{k,i}(s) = \sum_{s \in N, s \neq i} f_{k,i}(s) = \sum_{s \in N, s \neq i} \frac{r_{k,i}(s) \times w(s)}{d(s)} \quad (4)$$

$$F_{i,j} = \sum_{s \in N} f_{i,j}(s) = \sum_{s \in N} \frac{r_{i,j}(s) \times w(s)}{d(s)} = F_i \frac{r_{i,j}}{r_i} \quad (5)$$

先令  $F_i$ ， $F_{k,i}$ ， $F_{i,j}$  初值都为 0，利用式(1)计算出  $F_i$ ，然

后通过式(5)计算出  $F_{i,j}$  的值,  $i$  将  $F_{i,j}$  发送给节点  $j$ , 以备下一轮计算使用。同样,  $i$  可以从接收到来自上游节点  $k$  的  $F_{k,i}$ 。经过几次重复计算以后,  $F_i$  和  $F_{k,i}$  达到一个稳定值。一旦在节点  $i$  发生拥塞。按式(6)给链路  $k,i$  分配信道。

$$l_{k,i} = \frac{F_{k,i}}{F_i} r_i \quad (6)$$

如果  $i$  节点本身就是源节点, 则按式(7)给本地分配信道。

$$l_i = \frac{w(i)}{F_i} r_i \quad (7)$$

对源速率的限制可以通过令牌桶算法实现。

如果建立新路径不成功, 即 CreatNewPath!=1

CurrentTokenNum=Min(DateTime.Now-t)\*l<sub>k,i</sub>, MaxBucket Size);

1// Tokennum 为当前令牌数, MaxBucketSize 为令牌桶最大容量

2 T = DateTime.Now;

3 if (CurrentTokenNum >= CurrentPackets) // Packets 为

//当前数据包数目

4 Send(CurrentPackets)

5 Else

6 WaitFor(CurrentPackets - CurrentTokenNum)/l<sub>k,i</sub>

7 Jump 5; //WaitFor() 等待到某一条件满足 Jump 跳转函数

如何在邻居节点信息表中, 选取满足条件的节点作为新

路径的一跳:

1 Do While (I <=N) //N 为邻居节点信息表信息条数

2 If (C<sub>i</sub> < C && E<sub>i</sub> > E) //C为拥塞度的阈值, E为剩余能//

量阈值

3 { If (Sleeping(i)=1)

4 Wakeup(i); //如果节点 i 在休眠状态则唤醒它

5 AddtoPath(i); //将节点 i 作为新路径的一跳

6 }

7 Else

8 {

9 i++;

10 }

### 3 仿真实验及分析

本文将自适应机制在 glomosim 中进行了模拟测试。将 500 个传感器节点任意洒在  $1000 \times 1000$  的区域。节点的发送范围是 100, 每个节点的数据发送率为 512 Kb/s。任意选取其中的 100 个节点作为源节点。数据包的初始产生率为每个单位时间 25 个包, 每个数据包长度为 30 B。节点的缓存空间为 30 个数据包。实验结果如图 2~图 5 所示。在图 2 和图 3 中, 对无拥塞控制、建立新路径<sup>[5]</sup>和本文提出的自适应机制进行了对比。可以看出建立新路径有效地缓解了拥塞, 减少了丢包数目, 但因为建立新路径可能失败而无法缓解拥塞, 丢包数目仍然比较多。而采用本文提出的新的拥塞缓解机制以后, 丢包的数目大大降低。图 3 比较了 3 种不同情况下的网络能量消耗情况。建立新路径减少了冲突和拥塞带来的能量浪费, 有效的提高了能源的利用率, 节约了能量。本文提出的自适应拥塞缓解机制在建立新路径不成功时降低源速率的发送, 减少了不断搜索新路径的能量浪费, 故在能耗方面更显优势。图 4 和图 5 表明采用自适应机制源节点发送的数据包权重不一样时, 数据的发送率也不一样。图 4 中 id 号为 1~50 的节点数据的权重为 3, 这些节点发送数据包的平均发送率为 23%, id 号为 50~100 的节点数据的权重为 1, 数据包地平均发送率为 6%。图 5 中 id 号为 1~30 的权重为 3, 31~65 权重为 1, 66~100 权重为 2, 数据包的平均发送率分别

为 24%、15%、6%。因为在拥塞缓解过程中, 对源速率的限制考虑了权重。权重越大即数据越重要, 数据的发送率越高。有效的保证了重要数据能可靠的传到 sink 节点。

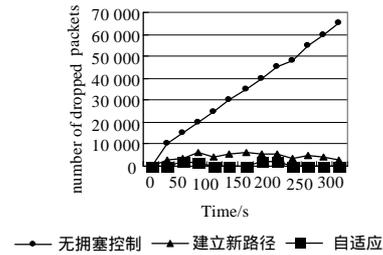


图 2 丢包数的比较

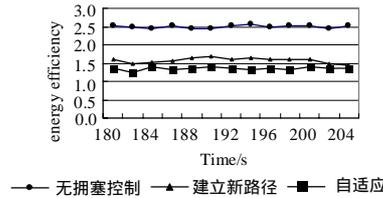


图 3 能耗的比较

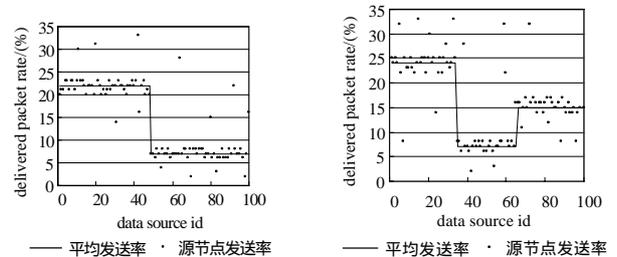


图 4 不同节点发保率比较(1)

图 5 不同节点发保率比较(2)

### 4 结束语

本文在综合分析已有的拥塞缓解机制和建立新路径<sup>[5]</sup>的基础上, 提出了一种自适应的拥塞缓解机制。该机制弥补了建立新路径失败和建立以后形成新的拥塞点等一些缺陷。实验结果表明, 该机制有效地提高了能量利用率, 降低了丢包率, 使重要数据能实时地传到 sink 节点。但是, 在能耗方面无明显优势, 如何进一步减低能耗, 提高能量的利用率是今后的工作重点之一。

### 参考文献

- [1] Sankarasubramaniam Y, Akan O, Akyildiz I. ESRT: Event-to-sink Reliable Transport in Wireless sensor Networks[C]//Proceedings of the ACM MobiHoc Conference. New York, USA: ACM Publish, 2003.
- [2] Wan C Y, Eisenman S B, Campbell A T. CODA: Congestion Detection and Voidance in Sensor Networks[C]//Proc. of SenSys'03. Los Angeles, California, USA: [s. n.], 2003-11.
- [3] Hull B, Jamieson K, Balakrishnan H. Mitigating Congestion in Wireless Sensor Networks[C]//Proc. of ACM SenSys'04. Baltimore, MD: [s. n.], 2004-11.
- [4] Chen Shigang, Zhang Zhan. Localized Algorithm for Aggregate Fairness in Wireless Sensor Networks[C]//Proceedings of the ACM MobiHoc Conference. New York, USA: ACM Publish, 2006-09.
- [5] Kang Jaewon. Adaptive Resource Control Scheme to Alleviate Congestion in Sensor Networks[C]//Proc. of the 5th Workshop on Broadband Advanced Sensor Networks. Boston, MA: [s. n.], 2004.