

# $\epsilon$ -近似和加权公平性保证的无线传感器网络 拥塞控制算法

李国华 李建中 高 宏

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

**摘 要** 针对现有的无线传感器网络拥塞控制算法中很少考虑数据压缩和加权公平性的问题,提出了一种  $\epsilon$  近似和加权公平性保证的拥塞控制算法. 首先,我们对传感器节点产生的时间序列采用贪心分段常值  $\epsilon$  近似(Greedy Piecewise Constant  $\epsilon$ -Approximation, GPCA)的策略,极大地减少了数据通信量. 其次,我们给出了具有加权公平性保证的拥塞控制算法(Weighted Fairness Guaranteed Congestion Control Algorithm, WFCC),并首次给出了公平性度量的下界  $1-(10c/9)^2$ ,其中  $c$  为常数且  $0 < c < 0.2$ . 实验结果表明,GPCA 具有很好的压缩性能,WFCC 达到了较高的吞吐量和加权公平性(95%以上).

**关键词** 无线传感器网络;拥塞控制; $\epsilon$ -近似;加权公平性;拥塞度量

中图法分类号 TP393 DOI号: 10.3724/SP.J.1016.2011.02197

## $\epsilon$ -Approximation and Weighted Fairness Guaranteed Congestion Control Algorithm for Wireless Sensor Networks

LI Guo-Hua LI Jian-Zhong GAO Hong

(Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

**Abstract** In a wireless sensor network, congestion problem not only causes packet loss, but also leads to delay increase and energy consumption. Most of the existing congestion control algorithms for wireless sensor networks seldom concern the combined problem of data compression and weighted fairness. In order to address this problem, a greedy piecewise constant  $\epsilon$ -approximation algorithm, GPCA, and a novel decentralized congestion control algorithm WFCC are proposed in this paper. GPCA approximates a subsequence using a constant and guarantees that the error between the real sequence and the approximation sequence is less than or equal to  $\epsilon$ . In addition, we prove the optimality of constant  $\epsilon$ -approximation in theory. WFCC not only mitigates congestion, but also guarantees the weighted fairness among all sensor nodes. Importantly, we give a lower bound of weighted fairness metric  $1-(10c/9)^2$  where  $c$  is a constant and  $0 < c < 0.2$ . We evaluate GPCA using Intel Lab Data, and the results show that GPCA has great compression performance. In addition, we evaluate the algorithm WFCC in a 22-node wireless sensor network by TOSSIM. The results of simulation demonstrate that WFCC achieves high throughput and weighted fairness (above 95% on average).

**Keywords** wireless sensor networks; congestion control;  $\epsilon$ -approximation; weighted fairness; congestion metric

收稿日期:2011-08-29;最终修改稿收到日期:2011-09-15. 本课题得到国家自然科学基金重点项目(61033015,60933001)、国家自然科学基金基金(60831160525,61100030)、中国博士后科学基金(20110491060)和中央高校基本科研业务费专项资金(HIT.NSRIF.201179)资助.  
李国华,男,1984年生,博士研究生,主要研究方向为无线传感器网络和物联网. E-mail: lghhit1@126.com. 李建中,男,1950年生,教授,博士生导师,主要研究领域为物联网、无线传感器网络、数据库和海量数据处理. E-mail: lijzh@hit.edu.cn. 高宏,女,1966年生,博士,教授,博士生导师,主要研究领域为无线传感器网络、物联网、海量数据管理和数据挖掘.

## 1 引 言

无线通信技术、微电子技术及嵌入式计算技术的快速发展使得无线传感器网络被广泛应用于环境监测、健康监护、智能家居及战场监控等领域. 无线传感器网络由许多分布于特定区域且具有一定计算、存储和通信能力的传感器节点组成. 每个节点只能与附近几跳的节点通信, 数据通过多跳转发才能到达 sink 节点. 当传感器节点接收数据包的速率大于其转发或调度速率时, 需对过剩的数据包进行缓存; 当缓存的数据包队列满时, 过剩的数据包就会被丢弃, 即引发无线传感器网络中的拥塞问题. 拥塞不仅增加包的延迟, 降低网络吞吐量且浪费能量. 因此如何有效地减缓拥塞是无线传感器网络技术研究的一个重要方面.

目前, 几乎所有的拥塞控制算法都没有考虑对数据进行预处理后再进行传输的问题. 在无线传感器网络应用中, 连续稠密采样能极大地提高对现实世界的感知程度, 然而如果不加处理将采集的原始数据存储再转发将带来很大的存储和通信开销. 为了从源头上减少数据的传输量, 我们对传感器节点产生的时间序列采取贪心分段常值  $\epsilon$  近似 (Greedy Piecewise Constant  $\epsilon$ -Approximation, GPCA) 的策略, 使得一个连续的子序列仅需要一个常值来近似且保证其与真实值的误差不大于  $\epsilon$ . 这样既没有严重损害数据的保真度, 还减少了数据传输的通信开销, 从而延长了网络的生命周期. 节点在得到时间序列的常值近似后, 将相应的常值及近似的起止时间存入缓冲队列, 等待调度传输.

衡量拥塞控制算法有效性主要有两个指标: 一是服务质量, 包括吞吐量、时延; 二是加权公平性<sup>[1-2]</sup>. 目前的大多数拥塞控制算法主要考虑了服务质量, 而很少考虑如何保证数据传输的加权公平性问题, 即如何保证 sink 节点接收到不同传感器节点产生的数据包数目与传感器节点的重要性成正比. 保证加权公平性可以使 sink 节点从重要的数据源获得更多的数据包. 然而, 现有的拥塞控制算法均没有全面地从服务质量和加权公平性这两个方面进行考虑. 文献[3-4]主要考虑了如何提高数据传输的效率, 即服务质量, 而未涉及加权公平性. 文献[5-6]不仅考虑了服务质量, 还考虑了公平性, 但只是简单的公平性, 即保证 sink 节点在某段时间内接收每个节点的数据包一样多. 文献[1-2]考虑了服务质量和加权公平性, 但只是启发式方法, 没有具体的加权公平

性度量和理论结果保证. 针对上述不足, 我们提出了一个新的拥塞控制算法, 不仅考虑了数据传输的服务质量, 还首次给出了一个加权公平性度量, 并在理论上给出了其的一个下界.

由于不同的节点可能装有不同的传感器, 所产生的数据的重要性不一样. 故一个好的拥塞控制算法应该使 sink 从重要的传感器节点接收较多的数据包. 我们给每个节点  $i$  赋予一个权值  $w_i$ ,  $w_i$  越大表示节点  $i$  产生的数据越重要. 我们采用下面的公式<sup>[7]</sup>作为公平性度量.

$$f_t = \frac{\left(\sum_{i=1}^N P_{i,t}/w_i\right)^2}{N \sum_{i=1}^N (P_{i,t}/w_i)^2},$$

这里,  $P_{i,t}$  表示 sink 节点在时间段  $[0, t]$  接收来自节点  $i$  的数据包数量,  $N$  表示节点总数. 当  $P_{i,t}/w_i = P_{j,t}/w_j$  ( $j \in [1, N], j \neq i$ ) 成立时,  $f_t = 1$ , 此时达到了最大的公平性. 当  $P_{i,t}/w_i$  为正数而  $P_{j,t}/w_j = 0$  ( $j \in [1, N], j \neq i$ ) 时,  $f_t = 1/N$ , 此时达到了最小的公平性. 我们在理论上严格证明了  $f_t \geq 1 - (10c/9)^2$ , 其中  $c$  为常数且  $0 < c < 0.2$ .

综上所述, 目前的拥塞控制算法各有其针对性和局限性, 均是对原始数据进行操作, 且没有给出加权公平性一个量化表达式. 基于此, 本文联合考虑了数据压缩和拥塞控制技术, 提出了一个  $\epsilon$ -近似和加权公平性保证的拥塞控制算法, 其由两个子算法组成: on-line 的贪心分段常值  $\epsilon$  近似算法 (GPCA)<sup>[8]</sup> 和加权公平性保证的拥塞控制算法 (Weighted Fairness Guaranteed Congestion Control Algorithm, WFCC). GPCA 分布式地运行在每个非 sink 节点, 对节点产生的时间序列进行贪心分段近似, 使得每一个连续子序列仅需要一个常值来描述, 这样不仅减少了存储开销, 更重要的是减少了通信开销. 我们证明了在常值  $\epsilon$  近似算法中, GPCA 是最优的. WFCC 以进入节点的平均数据包速率除以流出节点的平均数据包速率作为拥塞度量, 对进入节点的数据包速率进行调节, 使得单位时间收到孩子节点传输的数据包的数量与其权值大小成正比. 我们随后分析说明通过这种局部公平调速, 可以达到全局的公平性, 并给出了理论证明. 综上, 本文的主要贡献总结如下:

(1) 本文设计了一个贪心分段常值  $\epsilon$  近似算法, 并证明其在所有满足  $\epsilon$  近似的分段常值近似中是最优的.

(2) 本文首次给出了无线传感器网络中加权公平性度量的定义.

(3) 本文提出了一个分布式的加权公平性保证的拥塞控制算法, 并首次给出了加权公平性的一个下界  $1 - (10c/9)^2$ , 其中  $c$  为常数且  $0 < c < 0.2$ , 且给出了其严格的理论推导.

(4) 本文进行了广泛的实验来评估算法性能, 实验结果表明本文提出的拥塞控制算法不仅达到了较高的吞吐量, 且其加权公平性平均达到了 95%, 验证了我们的理论分析.

本文第 2 节讨论相关工作; 第 3 节介绍算法 GPCA; 第 4 节给出算法 WFCC; 第 5 节对 WFCC 的加权公平性进行分析; 第 6 节是实验和分析; 第 7 节是本文结论部分.

## 2 相关工作

$\epsilon$  时间序列近似<sup>[8-10]</sup> 是时间已久一种有损数据压缩方法, 虽然丢失了一些数据精度, 但仍可以满足查询用户的需求且减少数据的存储、计算和通信开销. 文献[8]考虑在数据流模型中, 给出了两个空间有效的且有质量保证的  $L_\infty$  柱状近似算法. 文献[9]考虑如何动态地融合多个传感器节点的数据, 进行数据压缩以减少通信开销. 文献[10]给出了一个能以任意精度近似原始时间序列的算法 EAQ, 其能支持多种不同的查询. 文献[8]虽给出了贪心分段常值  $\epsilon$  近似, 但并没有给出其方法最优性的严格的数学证明, 我们在本文给出了其最优性的严格证明. 文献[9-10]的方法适用于对数据延迟不敏感的传感器网络应用. 而对数据延迟敏感的应用一般使用 on-line 近似时间序列方法, 因为它可以极大地减少数据的延迟. 本文的  $\epsilon$  近似时间序列方法为 on-line 近似时间序列方法.

早期研究拥塞问题的主要文献有 CODA<sup>[8]</sup>、Fusion<sup>[9]</sup>、CCF<sup>[11]</sup>、IFRC<sup>[6]</sup> 和文献[12]. 文献[8]给出了最早的被动拥塞控制机制 CODA, 它通过检测传感器节点缓冲区队列的长度或信道利用率来侦测网络中是否发生拥塞. 如果发生了拥塞, 则通过局部地降低上一跳节点的发送速率来缓解局部拥塞, 同时采用闭环多源调节机制来缓解大范围拥塞和长期拥塞. 文献[9]给出的 Fusion 拥塞控制机制综合利用 3 种技术来进行拥塞控制: 一是在拥塞发生时逐跳地向源节点发反馈信息, 以减少数据包的丢失和避免能量浪费; 二是限制源节点的发包速率; 三是用带优先权控制的 MAC 协议给缓冲区占有率高的节点以更高的通信信道访问权. Fusion 机制能有效地缓解拥塞现象, 但没有给出保证事件公平性的措施.

CCF<sup>[11]</sup> 和 IFRC<sup>[12]</sup> 能达到非常好的简单的公平性, 但是未考虑加权公平性. 文献[12]提出了一种拥塞避免的方法, 即发送者只有在接收者通知其缓冲区未满足的条件下才发送数据包. 文中还考虑了可能由于隐终端问题而引起的拥塞, 并给出了一种通过广播  $1/k$  剩余缓冲区的方法来避免由于隐终端问题引起的拥塞. 尽管该方法能有效地避免拥塞, 但由于每个包都得携带关于剩余缓冲区大小的信息, 所以通信消耗的能量开销较大. 最近的关于拥塞问题的文献主要有 PCCP<sup>[1]</sup>、RCRT<sup>[13]</sup> 和 RCCAP<sup>[14]</sup>. 文献[1]以数据包的平均服务时间与数据包的平均到达时间的比值作为拥塞度量, 每个节点每接收到一定数目包, 就通过广播将自己的拥塞度告知给子节点, 子节点根据父亲节点的拥塞度量值相应地调节自己的发送速率和采样速率. 文献[13]以 RTT (Round-Trip Time) 的整数倍作为拥塞度量, 端对端地通过对节点的传输速率进行 AIMD (Additive Increase/Multiplicative Decrease) 的方法来缓解拥塞. 文献[14]通过单神经元和 PID 控制结合的方法减缓拥塞.

在上述拥塞控制的相关研究中, 均没有考虑数据压缩或近似技术, 且或者没有考虑公平性, 或者只考虑简单的公平性和加权公平性, 没有理论上的分析和证明. 基于此, 本文联合考虑数据压缩技术和拥塞控制技术, 提出一种保证  $\epsilon$  近似和加权公平性的拥塞控制算法.  $\epsilon$  近似减少了数据的存储和通信开销, 加权公平性的保证提高了传感器网络的负载平衡能力, 从而延长网络的生命周期; 且我们证明了 GPCA 的最优性, 并首次给出了拥塞控制算法中加权公平性度量的下界,  $1 - (10c/9)^2$ , 其中  $c$  为常数且  $0 < c < 0.2$ .

## 3 拥塞控制算法概述

在无线传感器网络中, 节点通信所消耗的能量远远大于计算所消耗的能量<sup>[1]</sup>, 所以减少数据的通信量可以极大地减少能量的消耗, 而且还可以减缓拥塞. 基于此, 我们提出了一个对数据先进行压缩然后再传输的拥塞控制算法. 其基本思想是: (1) 先对每个节点的原始采样数据进行压缩, 使得多个连续的采样数据可被一个常值  $c$  近似且保证近似值与原始采样值之间的误差小于  $\epsilon$ , 然后将三元组  $\langle c, [t_a, t_b] \rangle$  压入队列, 等待调度传输. (2) 由于有一些传感器产生的数据比较重要, 因此需要更高的数据传输速率(带宽). 我们首先量化了每种传感器节点产生的数

据的重要性,赋予每种传感器一个权值,然后在进行数据的调度传输时,不仅保证重要的数据源获得的带宽比较高,且尽量使得每个数据源获得的带宽与其权值成正比.

据我们所知,本文是第一篇不仅保证了对原始数据的  $\epsilon$  近似,且首次给出了加权公平性的一个下界的论文.

下面分两个小节对算法的基本思想进行详细的说明.其中第 4 节的贪心分段常值  $\epsilon$  近似算法 GPCA 对应基本思想中的(1);第 5 节的加权公平性保证的拥塞控制算法 WFCC 对应基本思想中的(2).

## 4 贪心分段常值 $\epsilon$ 近似算法

### 4.1 预备知识

假设  $S^{(n)} = \langle x[t_1], x[t_2], \dots, x[t_n] \rangle$  为一个传感器节点产生的时间序列,其中  $t_h (1 \leq h \leq N)$  为节点采样的时刻; $x[t_h] (1 \leq h \leq N)$  为节点采样得到的属性值.  $S[t_a : t_b] = \langle x[t_a], x[t_{a+1}], \dots, x[t_b] \rangle$  表示  $S^{(n)}$  的一个连续子序列.我们使用贪心分段常值近似的方法近似  $S^{(n)}$ ,使得每一段的子序列仅需要一个常值来描述.这样我们对每一个子序列只需记录常值及子序列的开始时间和结束时间.从而大大减少了记录整个时间序列的开销.

假设最后将产生  $H$  个常值  $(c_1, c_2, \dots, c_H)$  近似序列  $S^{(n)}$ ,其中  $c_1$  近似子序列  $\langle x[t_1], x[t_2], \dots, x[t_{j_1}] \rangle$ ,  $c_h (2 \leq h \leq H)$  近似子序列  $\langle x[t_{j_{h-1}+1}], x[t_{j_{h-1}+2}], \dots, x[t_{j_h}] \rangle$ ,其中  $j_H = n$ .假设  $\hat{S}^{(n)} = \langle \hat{x}[t_1], \hat{x}[t_2], \dots, \hat{x}[t_n] \rangle$  为  $(c_1, c_2, \dots, c_H)$  近似  $S^{(n)}$  得到的时间序列,我们用下面的  $E_\infty$  范数来量化两个序列的误差.其表达式为

$$E_\infty(S^{(n)}, \hat{S}^{(n)}) = \max_{1 \leq p \leq n} |x[t_p] - \hat{x}[t_p]|.$$

我们的优化问题是:对于给定的时间序列  $S^{(n)} = \langle x[t_1], x[t_2], \dots, x[t_n] \rangle$ ,误差界  $\epsilon$  和误差函数  $E_\infty(\cdot)$ ,找到最小长度的常值序列  $(c_1, c_2, \dots, c_H)$ ,使得  $E_\infty(S^{(n)}, \hat{S}^{(n)}) \leq \epsilon$ .

### 4.2 贪心分段常值 $\epsilon$ 近似算法

贪心分段常值  $\epsilon$  近似算法(GPCA)的主要思想描述如下.对于子序列  $S[t_a : t_b]$  中的最大值和最小值我们分别用  $\max_{a \leq p \leq b} x[t_p]$  和  $\min_{a \leq q \leq b} x[t_q]$  表示,当  $\max_{a \leq p \leq b} x[t_p] - \min_{a \leq q \leq b} x[t_q] \leq 2\epsilon$  时,我们用  $c = (\max_{a \leq p \leq b} x[t_p] + \min_{a \leq q \leq b} x[t_q]) / 2$  来近似子序列  $S[t_a : t_b]$ .此时,易得对  $\forall a \leq m \leq b$ ,  $|c - x[t_m]| \leq \max\{|c - \min_{a \leq q \leq b} x[t_q]|,$

$$\begin{aligned} & |c - \max_{a \leq p \leq b} x[t_p]| \} \\ & = (\max_{a \leq p \leq b} x[t_p] - \min_{a \leq q \leq b} x[t_q]) / 2 \\ & \leq \epsilon. \end{aligned}$$

当新来数据点  $(t_{b+1}, x[t_{b+1}])$  时,

(1) 若  $\max_{a \leq p \leq b+1} x[t_p] - \min_{a \leq q \leq b+1} x[t_q] \leq 2\epsilon$  时,我们用

$$c_0 = (\max_{a \leq p \leq b+1} x[t_p] + \min_{a \leq q \leq b+1} x[t_q]) / 2$$

来近似子序列  $S[t_a : t_{b+1}]$ .

(2) 否则,若  $\max_{a \leq p \leq b+1} x[t_p] - \min_{a \leq q \leq b+1} x[t_q] > 2\epsilon$ ,则子序列  $S[t_a : t_{b+1}]$  不能被一个常数  $c$  近似且保证误差不大于  $\epsilon$ . 因为  $|\max_{a \leq p \leq b} x[t_p] - c| + |c - \min_{a \leq q \leq b+1} x[t_q]| > 2\epsilon$ , 所以  $|\max_{a \leq p \leq b} x[t_p] - c|$  与  $|c - \min_{a \leq q \leq b+1} x[t_q]|$  两者必有一个的值大于  $\epsilon$ . 故我们用

$$c_1 = (\max_{a \leq p \leq b} x[t_p] + \min_{a \leq q \leq b} x[t_q]) / 2$$

近似子序列  $S[t_a : t_b]$ ,且在时间  $t_{b+1}$  开始一个新的子序列的近似.

为了证明贪心常值  $\epsilon$  近似的最优性,我们先证明下面的引理.

**引理 1.** 给定任意时间序列  $S[t_a : t_b] = \langle x[t_a], x[t_{a+1}], \dots, x[t_b] \rangle$ ,误差界  $\epsilon$  和误差函数  $E_\infty(\cdot)$ .

(1) 若  $S[t_a : t_b]$  可用一个常数近似且保证误差小于等于  $\epsilon$ . 则对  $\forall a_1, b_1, a \leq a_1 < b_1 \leq b$ ,  $S[t_{a_1} : t_{b_1}]$  可用常数近似且保证误差小于等于  $\epsilon$ .

(2) 若  $S[t_a : t_b]$  不能被一个常数近似且保证误差小于等于  $\epsilon$ . 则对  $\forall a_2, b_2, a_2 \leq a < b \leq b_2$ ,  $S[t_{a_2} : t_{b_2}]$  也不能被一个常数近似且保证误差小于等于  $\epsilon$ .

证明.

(1) 设  $\max_{ab}$  和  $\min_{ab}$  分别为  $S[t_a : t_b]$  的最大值和最小值,  $\max_{a_1 b_1}$  和  $\min_{a_1 b_1}$  分别为  $S[t_{a_1} : t_{b_1}]$  的最大值和最小值. 则有

$$\min_{ab} \leq \min_{a_1 b_1} \leq \max_{a_1 b_1} \leq \max_{ab},$$

从而得到

$$\max_{a_1 b_1} - \min_{a_1 b_1} \leq \max_{ab} - \min_{ab} \leq 2\epsilon.$$

由算法描述可知,我们可用

$$c = (\max_{a_1 \leq p \leq b_1} x[t_p] + \min_{a_1 \leq q \leq b_1} x[t_q]) / 2$$

来近似子序列  $S[t_{a_1} : t_{b_1}]$  保证其误差小于等于  $\epsilon$ .

(2) 假设  $S[t_{a_2} : t_{b_2}]$  能被一个常数近似且保证误差小于等于  $\epsilon$ , 因为  $a_2 \leq a < b \leq b_2$ , 则由(1)可知,  $S[t_a : t_b]$  也能被一个常数近似且保证误差小于等于  $\epsilon$ . 得出矛盾. 故  $S[t_{a_2} : t_{b_2}]$  也不能被一个常数近似且保证误差小于等于  $\epsilon$ . 证毕.

**定理 1**(最优性证明). 给定时间序列  $S^{(n)} = \langle x[t_1], x[t_2], \dots, x[t_n] \rangle$ , 误差界  $\epsilon$  和误差函数

$E_\infty(\cdot)$ . 在所有满足  $E_\infty(S^{(n)}, \hat{S}^{(n)}) \leq \epsilon$  的分段常值  $\epsilon$  近似中, 贪心分段常值近似算法产生最小长度的常值序列  $(c_1, c_2, \dots, c_H)$ .

证明. 反证法. 假设有一个更小长度的常值序列  $(d_1, d_2, \dots, d_{H'})$  ( $H' < H$ ) 近似  $S^{(n)}$  且满足  $E_\infty(S^{(n)}, \hat{S}'^{(n)}) \leq \epsilon$ , 其中,  $\hat{S}'^{(n)}$  为  $(d_1, d_2, \dots, d_{H'})$  近似  $S^{(n)}$  得到的时间序列. 假设  $c_1$  近似子序列  $\langle x[t_1], x[t_2], \dots, x[t_{j_1}] \rangle$ ,  $c_h$  ( $2 \leq h \leq H$ ) 近似子序列  $\langle x[t_{j_{h-1}+1}], x[t_{j_{h-2}+1}], \dots, x[t_{j_h}] \rangle$ , 其中  $j_h = n$ .  $d_1$  近似子序列  $\langle x[t_1], x[t_2], \dots, x[t_{l_1}] \rangle$ .  $d_{h'}$  ( $2 \leq h' \leq H'$ ) 近似子序列  $\langle x[t_{l_{h'-1}+1}], x[t_{l_{h'-1}+2}], \dots, x[t_{l_{h'}}] \rangle$ ,  $l_{H'} = n$ .

下面我们用数学归纳法证明对  $\forall 1 \leq m \leq H' - 1$ ,  $l_{H'-m} > j_{H-(m+1)}$  成立.

(1) 当  $m = 1$  时,  $l_{H'-1} > j_{H-2}$  成立. 否则, 如果  $l_{H'-1} \leq j_{H-2}$ , 则子序列  $S[t_{j_{H-2}}: t_{j_H}] \subseteq S[t_{l_{H'-1}}: t_{l_{H'}}]$ , 故由引理 1 可知子序列  $S[t_{j_{H-2}}: t_{j_H}]$  可用  $d_{H'}$  近似且误差不大于  $\epsilon$ . 但是由算法可知  $S[t_{j_{H-2}}: t_{j_{H-1}+1}]$  ( $\subseteq S[t_{j_{H-2}}: t_{j_H}]$ ) 不能被一个常数近似且保证误差小于等于  $\epsilon$ . 故由引理 1 可知  $S[t_{j_{H-2}}: t_{j_H}]$  也不能被一个常数近似保证误差小于等于  $\epsilon$ . 得出矛盾, 故  $m = 1$  时,  $l_{H'-1} > j_{H-2}$  成立.

(2) 假设  $m = k < H'$  时,  $l_{H'-k} > j_{H-(k+1)}$  成立. 则  $l_{H'-(k+1)} > j_{H-(k+2)}$  成立. 否则, 若  $l_{H'-(k+1)} \leq j_{H-(k+2)}$  成立, 则由归纳假设知  $l_{H'-(k+1)} \leq j_{H-(k+2)} < j_{H-(k+1)} < l_{H'-k}$ , 从而  $S[t_{j_{H-(k+2)}}: t_{j_{H-(k+1)}+1}] \subseteq S[t_{l_{H'-(k+1)}}: t_{l_{H'-k}}]$ , 由引理 1 可知,  $S[t_{j_{H-(k+2)}}: t_{j_{H-(k+1)}+1}]$  可用  $d_{H'-k}$  近似且误差小于等于  $\epsilon$ . 但是由算法可知  $S[t_{j_{H-(k+2)}}: t_{j_{H-(k+1)}+1}]$  不能被一个常数近似且保证误差小于等于  $\epsilon$ . 得出矛盾, 故  $l_{H'-(k+1)} > j_{H-(k+2)}$  成立.

由(1)、(2)可知对  $\forall 1 \leq m \leq H' - 1$ ,  $l_{H'-m} > j_{H-(m+1)}$  成立. 特殊地, 当  $m = H' - 1$ ,  $l_1 > j_{H-H'}$  成立, 从而有  $S[t_1: t_{j_{H-H'+1}}] \subseteq S[t_1: t_{l_1}]$ , 由引理 1 可知,  $S[t_1: t_{j_{H-H'+1}}]$  可被  $d_1$  近似且误差小于等于  $\epsilon$ . 而  $H - H' \geq 1$ , 从而有  $S[t_1: t_{j_1+1}] \subseteq S[t_1: t_{j_{H-H'+1}}]$ , 故由引理 1 可知  $S[t_1: t_{j_1+1}]$  可被  $d_1$  近似且误差小于等于  $\epsilon$ . 但是由算法可知  $S[t_1: t_{j_1+1}]$  不能被一个常数近似且保证误差小于等于  $\epsilon$ . 从而得出矛盾. 故不存在长度小于  $H$  的常值序列近似  $S^{(n)}$  且满足  $E_\infty(S^{(n)}, \hat{S}^{(n)}) \leq \epsilon$ . 证毕.

### 4.3 时间和空间复杂性分析

由于 GPCA 是一个 on-line 算法, 每当一个新的采样到达, 只需要  $O(1)$  的空间和  $O(1)$  的计算时

间, 因为新采样值只需与之前未结束的子序列的最大值与最小值比较大小. 故 GPCA 的时间复杂度为  $O(n)$ , 空间复杂度为  $O(1)$ , 其中  $n$  为时间序列  $S^{(n)}$  的长度.

## 5 加权公平性保证的拥塞控制算法 WFCC

### 5.1 节点模型和术语

假设传感器节点连续地采集数据, 中间节点不仅采集数据还转发来自其他节点的数据. 每个节点在其应用层维护一个最大长度为  $L$  的循环队列作为缓存自身采样得到的和来自子孙节点的数据包的缓冲区, 每个数据包的 payload 域由 GPCA 算法得到的  $\langle t_s, t_e, c \rangle$  三元组组成, 其中  $c$  为  $[t_s, t_e]$  时间段内子序列的近似值; 每个节点有一个唯一的编号. 并且, 由于传感器节点所监测区域的重要性不同以及可能不同的节点所装传感器不一样, 每个节点被分配给一个反映其重要性的正权值<sup>[1-2]</sup>. 传感器节点间的数据传输通过单路径的树状拓扑结构  $\mathcal{T}$  来实现.

在路由结构  $\mathcal{T}$  中, 如图 1 所示, 根节点为 sink 节点, 其编号为 0; 编号为  $i$  的节点关联一个二元组  $\langle \omega_i, s\omega_i \rangle$ , 其中,  $\omega_i$  是节点  $i$  的权值,  $s\omega_i$  是  $\mathcal{T}$  的以节点  $i$  为根的子树  $\mathcal{T}_i$  中所有节点的权值和, 即  $s\omega_i = \sum_{j \in \mathcal{T}_i} \omega_j$ .  $s\omega_i$  的计算在路由构造过程中完成. WFCC 算法将利用二元组  $\langle \omega_i, s\omega_i \rangle$  来分配流入节点的总速率以满足拥塞控制过程中的加权公平性.

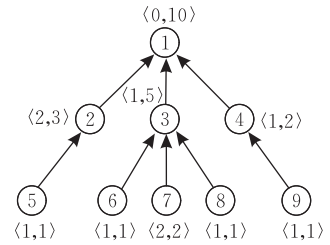


图 1 路由拓扑示意图

我们用  $O(i)$  表示节点  $i$  的子孙节点总数, 将直接传输数据包给节点  $i$  的节点  $j$  称为节点  $i$  的孩子节点, 相应地称节点  $i$  为节点  $j$  的父亲节点.  $r_{\text{sam}}^{(i)}$  和  $r_{\text{send}}^{(i)}$  分别表示节点  $i$  的采样速率和发送速率.  $r_{\text{in}}^{(i)}$  表示进入节点的总速率, 且有  $r_{\text{in}}^{(i)} = r_{\text{sam}}^{(i)} + \sum_{j \in \text{child}(i)} r_{\text{send}}^{(j)}$ , 其中  $\text{child}(i)$  为节点  $i$  的孩子节点编号集合.

### 5.2 拥塞度量的计算

我们使用文献[8]中的拥塞度量来测量每个节点的局部拥塞情况, 与文献[8]中不同的是, 我们将

流入的数据包速率公平地分配给孩子节点和自身节点分别作为其在下一个时间段的发送速率和采样速率来实现加权公平性. 最后, 我们从理论上证明了通过这种局部的公平速率分配策略可以达到很好的全局公平性, 并给出了一个下界. 下面我们先给出几个定义, 然后给出拥塞度量的表达式.

**定义 1**(数据包的服务时间). 数据包的服务时间  $t_s$  是指从数据包到达 MAC 层的时间到数据包的最后一个比特成功发送的时间间隔.

**定义 2**(数据包的间隔到达时间). 数据包的间隔到达时间  $t_a$  是指两个连续到达节点网络层的数据包之间的间隔时间.

单个的  $t_s$  和  $t_a$  只反映短时间内的包的服务时间和间隔到达时间, 为了更长时间更准确反映包的平均服务时间和平均到达时间, 每发送一个数据包, 我们使用指数加权移动平均方法更新节点  $i$  的数据包平均服务时间  $\bar{t}_s^{(i)}$ , 其具体计算表达式如下:

$$\bar{t}_s^{(i)}(h+1) = (1-q_s) \cdot \bar{t}_s^{(i)}(h) + q_s \cdot t_{\text{new}} \quad (1)$$

其中,  $0 < q_s < 1$  (实验中  $q_s = 0.1$ ) 是一个加权常数,  $t_{\text{new}}$  是第  $h+1$  个数据包的服务时间. 类似地, 每收到一个数据包, 我们使用指数移动加权平均方法更新数据包平均到达时间  $\bar{t}_a^{(i)}$ , 其具体计算表达式如下:

$$\bar{t}_a^{(i)}(m+1) = (1-q_a) \cdot \bar{t}_a^{(i)}(m) + q_a \cdot t'_{\text{new}} \quad (2)$$

其中,  $0 < q_a < 1$  (实验中  $q_a = 0.1$ ),  $t'_{\text{new}}$  是第  $m$  个与第  $m+1$  个数据包到达的间隔时间.

**定义 3**(拥塞度量). 节点的拥塞度量值是指数据包的平均服务时间与数据包的平均到达时间之间的比值. 其具体计算式如下:

$$d(i) = \bar{t}_s^{(i)}(h+1) / \bar{t}_a^{(i)}(m+1) \quad (3)$$

从其表达式易见, 当数据包的平均到达时间小于平均服务时间时,  $d(i) > 1$ , 即流入的速率大于流出的速率, 说明节点拥塞程度较严重. 反之,  $d(i) < 1$ , 即流入的速率小于流出的速率, 说明节点拥塞程度较轻.  $d(i)$  能根据网络的变化情况自适应地进行调节, 为实现拥塞控制提供了动态指示信息. 下面, 我们描述如何利用拥塞度量来进行速率的公平分配.

### 5.3 速率的公平分配

WFCC 算法是一个分布式算法, 在每一个节点上周期地运行. 时间轴被划分成长度为  $T$  的周期序列. 第  $k$  个周期是指  $(k-1)T$  到  $kT$  这个时间段.  $r_{\text{sam}}^{(i)}(k)$  表示节点  $i$  在第  $k$  个周期的采样速率,  $r_{\text{send}}^{(i)}(k)$  表示节点  $i$  在第  $k$  个周期的发送速率.  $r_{\text{in}}^{(i)}(k)$  表示在第  $k$  个周期流入节点的数据包总速率. 其值等于在第  $k$  个周期自身采样速率和孩子节点的发送速率之和, 即

$$r_{\text{in}}^{(i)}(k) = r_{\text{sam}}^{(i)}(k) + \sum_{j \in \text{child}(i)} r_{\text{send}}^{(j)}(k).$$

下面我们分 sink 节点和非 sink 节点描述 WFCC 算法在一个周期内的运行情况. 首先我们对算法在非 sink 节点上的运行过程进行描述.

(1) 非 sink 节点  $i$  的初始采样速率和发送速率分别设置为  $r_{\text{sam}}^{(i)}(k) = r_0 \cdot \omega_i$  和  $r_{\text{send}}^{(i)}(k) = r_0 \cdot s\omega_i$ , 其中  $r_0$  为初始设定的基准速率. 下面我们以一个周期为例, 来说明节点的行为. 我们将节点分成两类来描述: 一类是有孩子节点; 另一类是没有孩子节点. 先讨论 WFCC 算法在第 1 类节点 (有孩子节点) 上运行时的行为. ① 在  $kT$  时刻, 即第  $k+1$  个周期的开始时间, 节点  $i$  比较拥塞度  $d(i)$  与 1 的大小, 若  $d(i) > 1$ , 则更新

$$r_{\text{in}}^{(i)}(k+1) = r_0 \cdot s\omega_i \cdot \text{Factor}_1 \quad (4)$$

其中,  $0.9 \leq \text{Factor}_1 < 1$  (实验中  $\text{Factor}_1 = 0.9$ ) 为减速因子. 若  $d(i) \leq 1$ , 则更新

$$r_{\text{in}}^{(i)}(k+1) = r_0 \cdot s\omega_i \cdot \text{Factor}_2 \quad (5)$$

其中,  $1 < \text{Factor}_2 \leq 1.1$  (实验中  $\text{Factor}_2 = 1.1$ ) 为增速因子. 得到  $r_{\text{in}}^{(i)}(k+1)$  的值后, 我们将其公平地分配给节点  $i$  及其孩子节点分别作为其在下一个周期的采样速率和发送速率. 具体计算式如下:

$$r_{\text{sam}}^{(i)}(k+1) = r_{\text{in}}^{(i)}(k+1) \cdot \omega_i / s\omega_i \quad (6)$$

$$r_{\text{send}}^{(j)}(k+1) = r_{\text{in}}^{(i)}(k+1) \cdot s\omega_j / s\omega_i \quad (7)$$

其中节点  $j$  为节点  $i$  的孩子节点且  $s\omega_i = \omega_i + \sum_{j \in \text{child}(i)} s\omega_j$ .  $r_{\text{sam}}^{(i)}(k+1)$  的计算在本地即可完成, 而  $r_{\text{send}}^{(j)}(k+1)$  的计算实现是通过将二元组  $\langle s\omega_i, 1000 / r_{\text{in}}^{(i)}(k+1) \rangle$  捎带在数据包的头部, 节点  $i$  在广播这个数据包的同时, 其孩子节点  $j$  收到这个包后, 就可以计算  $r_{\text{send}}^{(j)}(k+1)$ . 这里我们没有传  $r_{\text{in}}^{(i)}(k+1)$  而是传  $1000 / r_{\text{in}}^{(i)}(k+1)$ , 是因为在 TinyOS 操作系统平台中, 包只存整数形数据. 这样对节点  $i$  来说, 流入的数据包速率被公平地分配给自身和其孩子节点分别作为其在下一个周期的采样速率和发送速率. ② 若节点  $i$  属于第 2 类节点, 即没有孩子节点的节点. 只需计算  $r_{\text{sam}}^{(i)}(k+1) = r_{\text{send}}^{(i)}(k+1)$ . 其中  $r_{\text{send}}^{(i)}(k+1)$  的计算如①所述是在收到其父亲节点的广播包后进行的.

(2) sink 节点 0 采用 AIMD (Additive Increase/Multi- plicative Decrease) 的方法来调节流入的数据包总速率. 其具体调节过程如下: 在  $kT$  时刻, 比较  $r_{\text{in}}^{(0)}(k)$  与  $\sum_{1 \leq j \leq N} \omega_j \cdot r_0$  大小, 若  $r_{\text{in}}^{(0)}(k) < \sum_{1 \leq j \leq N} \omega_j \cdot r_0$ , 为提高链路的利用率, 对  $r_{\text{in}}^{(0)}(k+1)$  进行加性增调节, 其计算式如下:

$$r_{\text{in}}^{(0)}(k+1) = r_{\text{in}}^{(0)}(k) + \text{AddFactor},$$

其中,  $AddFactor > 0$  (实验中视网络大小而定, 一般在  $[2r_0, 5r_0]$ ) 为加性增因子. 若  $r_{in}^{(0)}(k) \geq \sum_{1 \leq j \leq N} \omega_j \cdot r_0$ , 为减缓拥塞, 对  $r_{in}^{(0)}(k+1)$  进行乘性减调节, 其计算式如下:

$$r_{in}^{(0)}(k+1) = r_{in}^{(0)}(k) \cdot MulFactor,$$

其中,  $0 < MulFactor < 1$  (实验中一般在  $[0.9, 0.98]$  中取值) 为乘性减因子. 得到  $r_{in}^{(0)}(k+1)$  的值后, 我们将其公平地分配给节点 0 的孩子节点分别作为其在下一个周期的发送速率. 具体计算式如下:

$$r_{send}^{(j)}(k+1) = r_{in}^{(0)}(k+1) \cdot s\omega_j / \sum_{j \in child(0)} s\omega_j,$$

其中,  $\sum_{j \in child(0)} s\omega_j = \sum_{1 \leq j \leq N} \omega_j$ , 节点  $j$  为节点 0 的孩子节点. 其具体实现同(1)一样.

下面我们给出 WFCC 算法在第  $k+1$  个周期内的运行伪代码.

### 算法 1. 加权公平性保证的拥塞控制算法.

输入:  $\omega_i, s\omega_i (1 \leq i \leq N), r_{in}^{(0)}(k), d(i) (NumOfChild(i) \neq 0), AddFactor, MulFactor, Factor_1, Factor_2$

输出:  $r_{sam}^{(i)}(k+1)$  和  $r_{send}^{(j)}(k+1) (j \in child(i))$

/\*  $N$  表示节点总数,  $NumOfChild(i)$  表示节点  $i$  的孩子节点总数 \*/

1. if ( $NodeID \neq 0$ )
2. if ( $NumOfChild(i) \neq 0$ )
3. if ( $d(i) > 1$ )
4.  $r_{in}^{(i)}(k+1) \leftarrow r_0 \cdot s\omega_i \cdot Factor_1$ ;
5. else
6.  $r_{in}^{(i)}(k+1) \leftarrow r_0 \cdot s\omega_i \cdot Factor_2$ ;
7. end;
8.  $r_{sam}^{(i)}(k+1) \leftarrow r_{in}^{(i)}(k+1) \cdot \omega_i / s\omega_i$ ;
9. for  $j \in child(i)$
10.  $r_{send}^{(j)}(k+1) \leftarrow r_{in}^{(i)}(k+1) \cdot s\omega_j / s\omega_i$ ;
11. end;
12. else
13.  $r_{sam}^{(j)}(k+1) \leftarrow r_{send}^{(j)}(k+1)$ ;
14. end;
15. else
16. if ( $r_{in}^{(0)}(k) < \sum_{1 \leq j \leq N} \omega_j \cdot r_0$ )
17.  $r_{in}^{(0)}(k+1) \leftarrow r_{in}^{(0)}(k) + AddFactor$ ;
18. else
19.  $r_{in}^{(0)}(k+1) \leftarrow r_{in}^{(0)}(k) \cdot MulFactor$ ;
20. end;
21. for  $j \in child(0)$
22.  $r_{send}^{(j)}(k+1) \leftarrow r_{in}^{(0)}(k+1) \cdot s\omega_j / \sum_{j \in child(0)} s\omega_j$ ;
23. end;
24. Output  $r_{sam}^{(i)}(k+1)$  and  $r_{send}^{(j)}(k+1) (j \in child(i))$ .

在 WFCC 算法中, 每一个周期内流入节点  $i$  的数据包速率被完全公平地分配给自身及其孩子节点  $j$  分别作为其在本周期的采样速率和发送速率, 且有  $r_{sam}^{(i)}(k+1) / \omega_i = r_{in}^{(i)}(k+1) / s\omega_i = r_{send}^{(j)}(k+1) / \omega_j$ . 这

样带宽按节点  $i$  的权值及以其孩子节点  $j$  为根的子树的权值比例公平地分配给它们. 这种局部公平的速率分配方案间隔地保证了全局的加权公平性. 我们将在之后的加权公平性分析中严格证明. 下面我们先分析算法的复杂性, 之后再行加权公平性分析.

## 5.4 WFCC 算法复杂性分析

我们分析 WFCC 算法在一个周期内的计算、存储和通信复杂性. 从上面的算法描述可知, 在一个周期内, WFCC 算法在每个节点只需执行  $\alpha T / r_0$  次的四则运算操作, 其中  $\alpha \leq 8 = 6 + 2$ , 6 是指更新一次  $\bar{t}_s^{(i)}(h+1)$  和  $\bar{t}_a^{(i)}(m+1)$  所需的四则运算次数, 2 是指计算  $t_{new}$  和  $t'_{new}$  所各需的 2 次减法操作;  $T / r_0$  一般小于 1000. 在无线传感器网络中, 传感器节点用无线方式传播 1bit 到 100m 远所消耗的能量可供执行 3000 条左右加法指令<sup>[15]</sup>. 故相比于缓冲区溢出引起重传数据包的浪费的能量, 运行 WFCC 算法的计算开销是值得的. 除了正常的数据存储外, 额外的所需的变量个数不超过 15 个, 其存储开销小于 60 字节. 对一般的传感器节点 (如 Telosb 节点的扩展内存有 1 兆字节), 是可以承受的. 我们重点关注的是算法的通信复杂性, 为降低通信代价, 我们采用隐式的拥塞通知方法: 即将拥塞信息 ( $s\omega_i, 1000 / r_{in}^{(i)}(k+1)$ ) 捎带在数据包的头部, 利用无线信道的广播特性, 孩子节点也能接收到这个数据包, 之后就可以对包进行解析和计算. 隐式的拥塞通知方法并没有影响正常的数据传输, 只需要将必要的拥塞信息捎带在数据包的头部即可. 故 WFCC 算法所需的额外通信较低.

## 6 加权公平性分析

### 6.1 加权公平性度量

如前所述, 由于传感器节点所监测区域的重要性不同以及可能不同的节点所装传感器不一样, 每个节点被分配给一个反映其重要性的正权值. 我们以  $\omega_i$  表示节点  $i$  的正权值. 直观地, 如果  $\omega_i > \omega_j$ , 我们希望单位时间内 sink 节点接收来自节点  $i$  的包的数目  $P_i$  大于来自节点  $j$  的包的数目  $P_j$ , 最理想的情况 (也是最公平的情况) 是  $\omega_i / \omega_j = P_i / P_j$ . 为了量化这种加权公平性, 我们使用和文献[5]一样的公平性度量, 其表达式如下:

$$f(y_1, y_2, \dots, y_N) = \frac{\left(\sum_{p=1}^N y_p\right)^2}{N \sum_{p=1}^N y_p^2}, y_p \geq 0 \text{ 且 } \sum_{p=1}^N y_p > 0 \quad (8)$$

其中,  $N$  表示传感器网络中节点总数. 这里的  $y_p$  不是文献[5]中的数据包速率, 而是  $TP_p$  与  $\omega_p$  的比值. 这里  $TP_p$  表示在前  $M$  个周期 sink 节点接收来自节点  $p$  的数据包总数. 当  $TP_p/\omega_p = TP_j/\omega_j$  ( $1 \leq j \leq N, j \neq p$ ) 成立时, 式(8)的值等于最大值 1, 此时达到了最好的加权公平性. 当  $TP_p/\omega_p > 0$ , 而  $TP_j/\omega_j = 0$  ( $1 \leq j \leq N, j \neq p$ ) 时, 式(8)的值等于最小值  $1/N$ . 此时的加权公平性最差.

基于式(8), 我们定义在前  $M$  个周期内的加权公平性度量如下:

$$f_M = f\left(\sum_{k=1}^M \frac{r_{\text{sam}}^{(1)}(k) \cdot T}{\omega_1}, \sum_{k=1}^M \frac{r_{\text{sam}}^{(2)}(k) \cdot T}{\omega_2}, \dots, \sum_{k=1}^M \frac{r_{\text{sam}}^{(N)}(k) \cdot T}{\omega_N}\right) \\ = \frac{\left(\sum_{p=1}^N \sum_{k=1}^M \frac{r_{\text{sam}}^{(p)}(k) \cdot T}{\omega_p}\right)^2}{N \sum_{p=1}^N \left(\sum_{k=1}^M \frac{r_{\text{sam}}^{(p)}(k) \cdot T}{\omega_p}\right)^2} \quad (9)$$

## 6.2 加权公平性分析

下面我们先给出几个重要的引理及其证明, 然后基于这些引理, 证明  $f_M \geq 1 - (10c/9)^2$ .

**引理 2.** 节点  $i$  和  $j$  为路由树中的任意 2 个节点, 则有

$$\frac{r_{\text{sam}}^{(i)}(k)}{\omega_i} - \frac{r_{\text{sam}}^{(j)}(k)}{\omega_j} \leq cr_0, \quad 1 \leq i, j \leq N,$$

$$\Delta f = f(y_1, y_2, \dots, y_{p-1}, y_p + \Delta y_p, y_{p+1}, \dots, y_N) - f(y_1, y_2, \dots, y_{p-1}, y_p, y_{p+1}, \dots, y_N) \\ = \frac{\left[(y_p + \Delta y_p) + \sum_{j \neq p} y_j\right]^2}{N \left[(y_p + \Delta y_p)^2 + \left(\sum_{j \neq p} y_j\right)^2\right]} - \frac{\left(y_p + \sum_{j \neq p} y_j\right)^2}{N \left[y_p^2 + \left(\sum_{j \neq p} y_j\right)^2\right]} = \frac{\left[(y_p + \Delta y_p) + b_1\right]^2}{N \left[(y_p + \Delta y_p)^2 + b_2\right]} - \frac{(y_p + b_1)^2}{N (y_p^2 + b_2)} \\ = \frac{\left[(y_p + b_1) + \Delta y_p\right]^2 (y_p^2 + b_2) - (y_p + b_1)^2 [(y_p + \Delta y_p)^2 + b_2]}{N (y_p^2 + b_2) [(y_p + \Delta y_p)^2 + b_2]} \\ = \frac{\left[(\Delta y_p)^2 + 2\Delta y_p (y_p + b_1)\right] (y_p^2 + b_2) - (y_p + b_1)^2 [(\Delta y_p)^2 + 2y_p \Delta y_p]}{N (y_p^2 + b_2) [(y_p + \Delta y_p)^2 + b_2]} \\ = \frac{(b_2 - b_1^2 - 2b_1 y_p) (\Delta y_p)^2 + 2(b_1 b_2 + b_2 y_p - b_1^2 y_p - b_1 y_p^2) \Delta y_p}{N (y_p^2 + b_2) [(y_p + \Delta y_p)^2 + b_2]} \\ = \frac{(b_2 - b_1^2 - 2b_1 y_p) (\Delta y_p)^2 - 2b_1 (y_p + b_1) \left(y_p - \frac{b_2}{b_1}\right) \Delta y_p}{N (y_p^2 + b_2) [(y_p + \Delta y_p)^2 + b_2]} \quad (10)$$

由于  $b_2 - b_1^2 - 2b_1 y_p = (b_2 + y_p^2) - (b_1^2 + y_p^2 + 2b_1 y_p) = \sum_{j=1}^N y_j^2 - \left(\sum_{j=1}^N y_j\right)^2 \leq 0$ , 且由式(8)可知  $y_p + b_1 > 0$ , 又  $b_1 > 0$  和  $N(y_p^2 + b_2) [(y_p + \Delta y_p)^2 + b_2] > 0$ . 故  $\Delta f < 0$  当且仅当下面 3 个条件之一成立.

- (1)  $\Delta y_p > 0$  且  $y_p > y_0$ .
- (2)  $\Delta y_p < 0$  且  $0 \leq y_p < y_0$ .
- (3)  $y_p = y_0$  且  $\sum_{j=1}^N y_j^2 < \left(\sum_{j=1}^N y_j\right)^2$ .

其中,  $y_0 = b_2/b_1$ .

证毕.

其中,  $c = |Factor_1 - Factor_2|$ .

证明. 由式(4)、(5)和式(6)可知

$$r_{\text{sam}}^{(i)}(k) = r_{\text{in}}^{(i)}(k) \cdot \frac{\omega_i}{s\omega_i} = r_0 \cdot s\omega_i \cdot Factor_m \cdot \frac{\omega_i}{s\omega_i}, \\ = r_0 \cdot \omega_i \cdot Factor_m,$$

其中,  $m \in \{1, 2\}$ . 则由上式易知

$$\frac{r_{\text{sam}}^{(i)}(k)}{\omega_i} - \frac{r_{\text{sam}}^{(j)}(k)}{\omega_j} \leq |Factor_1 - Factor_2| \leq cr_0, \\ 1 \leq i, j \leq N$$

其中,  $c = |Factor_1 - Factor_2|$ .

证毕.

**引理 3.** 关于式(8), 如果只改变  $y_p$  为  $y_p + \Delta y_p$ , 而不改变  $y_j$  ( $j \neq p, 1 \leq j \leq N$ ), 则

$$f(y_1, y_2, \dots, y_{p-1}, y_p + \Delta y_p, y_{p+1}, \dots, y_N) < \\ f(y_1, y_2, \dots, y_{p-1}, y_p, y_{p+1}, \dots, y_N)$$

成立当且仅当下面 3 个条件之一成立.

- (1)  $\Delta y_p > 0$  且  $y_p > y_0$ .
- (2)  $\Delta y_p < 0$  且  $0 \leq y_p < y_0$ .
- (3)  $y_p = y_0$  且  $\sum_{j=1}^N y_j^2 < \left(\sum_{j=1}^N y_j\right)^2$ .

其中,  $y_0 = b_2/b_1$ ,  $b_2 = \sum_{j \neq p} y_j^2$ ,  $b_1 = \sum_{j \neq p} y_j$ .

证明. 由式(8)可知

**推论 1.** 若  $0 < y_{\min} \leq y_p \leq y_{\max}$  ( $1 \leq p \leq N$ ), 则式(8)中的  $f$  取最小值  $f_{\min}$  时,  $y_p = y_{\min}$  或  $y_p = y_{\max}$  两者必成立一个.

证明. 因为  $b_2 = \sum_{j \neq p} y_j^2$ ,  $b_1 = \sum_{j \neq p} y_j$ ,  $y_p + b_1 > 0$ . 若  $b_1 = 0$ . 则  $y_j = 0$  ( $j \neq p, 1 \leq j \leq N$ ). 此时  $y_p > 0$  且  $y_p = y_{\min} = y_{\max}$ , 命题成立. 若  $b_1 \neq 0$ , 由表达式易知  $y_{\min} b_1 \leq b_2 \leq y_{\max} b_1$ , 即  $y_{\min} \leq y_0 \leq y_{\max}$ . 下面分两种情况分析: (1) 若  $y_{\min} < y_0 < y_{\max}$ , 则由引理 3 可知, 对  $\forall y_0 < y_p < y_{\max}$ , 我们可以减少  $f$  的值通过增加  $y_p$

直到  $y_p = y_{\max}$ ; 对  $\forall y_0 < y_p < y_{\max}$ , 我们可以减少  $f$  的值通过减少  $y_p$  直到  $y_p = y_{\min}$ ; 对  $\forall y_p = y_0$ , 我们可以减少  $f$  的值通过增加或减少  $y_p$  直到  $y_p = y_{\min}$  或  $y_p = y_{\max}$ . (2) 若  $y_0 = y_{\min}$  或  $y_0 = y_{\max}$ , 则  $y_1 = y_2 = \dots = y_N = y_{\min} = y_{\max}$ . 如若不然, 当  $y_0 = y_{\min}$  成立时, 如果  $\exists p_0 \in [1, N]$ , 使得  $y_{p_0} > y_{\min}$ , 则有  $y_{p_0}^2 > y_{\min} y_{p_0}$ . 又  $y_p^2 \geq y_{\min} y_p$  ( $p \in [1, N]$  且  $p \neq j, p_0$ ), 可得  $\sum_{p \neq j} y_p^2 > (\sum_{p \neq j} y_p) y_{\min}$ , 即  $y_0 > y_{\min}$ , 与  $y_0 = y_{\min}$  矛盾, 故  $y_1 = y_2 = \dots = y_N = y_{\min} = y_{\max}$ . 同理可知, 当  $y_0 = y_{\max}$  成立时,  $y_1 = y_2 = \dots = y_N = y_{\min} = y_{\max}$ . 证毕.

基于推论 1, 我们来分析式(8)中的  $f$  的下界, 得到如下的引理.

**引理 4.** 如果  $0 < y_{\min} \leq y_p \leq y_{\max}$  ( $1 \leq p \leq N$ ), 则式(8)中的  $f$  有下界  $4K/(1+K)^2$ , 即  $f \geq 4K/(1+K)^2$ . 其中  $K = y_{\max}/y_{\min}$ .

证明. 由推论 1 可知, 当  $f$  取最小值  $f_{\min}$  时, 对  $\forall p, 1 \leq p \leq N$  或者  $y_p = y_{\min}$  或者  $y_p = y_{\max}$ . 假设  $f = f_{\min}$  时,  $y_{\min}$  和  $y_{\max}$  的个数分别为  $m_0$  和  $N - m_0$ . 为方便分析, 我们令  $m_0 = \lambda N$  ( $0 < \lambda < 1$ ),  $K = y_{\max}/y_{\min}$ . 则

$$f_{\min} = \frac{[m_0 y_{\min} + (N - m_0) y_{\max}]^2}{N[m_0 y_{\min}^2 + (N - m_0) y_{\max}^2]} = \frac{[\lambda + (1 - \lambda)K]^2}{\lambda + (1 - \lambda)K^2}$$

$$= \frac{(1 - K)^2 \left(\lambda + \frac{K}{1 - K}\right)^2}{(1 - K^2) \left(\lambda + \frac{K^2}{1 - K^2}\right)} = \frac{(1 - K) \left(\lambda + \frac{K}{1 - K}\right)^2}{(1 + K) \left(\lambda - \frac{K^2}{K^2 - 1}\right)},$$

令  $t = \lambda - \frac{K^2}{K^2 - 1}$ , 则  $t \in \left(\frac{-K^2}{K^2 - 1}, \frac{-1}{K^2 - 1}\right)$ , 从而有

$$f_{\min} = \frac{(1 - K) \left(\lambda + \frac{K}{1 - K}\right)^2}{(1 + K) \left(\lambda - \frac{K^2}{K^2 - 1}\right)}$$

$$= \frac{1 - K}{1 + K} \cdot \frac{\left(t + \frac{K^2}{K^2 - 1} + \frac{K}{1 - K}\right)}{t}$$

$$= -\frac{K - 1}{K + 1} \cdot \frac{\left(t - \frac{K}{K^2 - 1}\right)^2}{t}$$

$$= \frac{K - 1}{K + 1} \cdot \left[-t + \frac{\left(\frac{K}{K^2 - 1}\right)^2}{-t} + \frac{2K}{K^2 - 1}\right]$$

$$\geq \frac{K - 1}{K + 1} \cdot \frac{4K}{K^2 - 1} = \frac{4K}{(K + 1)^2},$$

其中, 等号成立条件是  $t = t_{\min} = -\frac{K}{K^2 - 1} \in \left(\frac{-K^2}{K^2 - 1}, \frac{-1}{K^2 - 1}\right)$ , 从而  $\lambda_{\min} = t_{\min} + \frac{K^2}{K^2 - 1} = \frac{K}{K + 1}$

时,  $f$  取最小值  $\frac{4K}{(K + 1)^2}$ .

证毕.

基于上面的引理和推论, 我们得到下面的定理.

**定理 2.** 如果我们以式(9)中  $f_M$  作为网络在前  $M$  个周期的加权公平性度量, 则

$$f_M \geq 1 - (10c/9)^2,$$

其中,  $c = |Factor_1 - Factor_2| \leq 0.2$ .  $Factor_1$  和  $Factor_2$  分别为式(4)和(5)中的减速因子和增速因子.

证明. 令  $S_p(M) = \sum_{k=1}^M \frac{r_{\text{sam}}^{(p)}(k)}{\omega_p}$  ( $1 \leq p \leq N$ ), 则

由式(9)可知  $f_M$  可重写为如下表达式

$$f_M = \frac{\left[\sum_{p=1}^N S_p(M)\right]^2}{N \sum_{p=1}^N [S_p(M)]^2}.$$

由引理 2 可得

$$|S_p(M) - S_N(M)| \leq \sum_{k=1}^M \left| \frac{r_{\text{sam}}^{(p)}(k)}{\omega_p} - \frac{r_{\text{sam}}^{(N)}(k)}{\omega_N} \right| \leq cMr_0,$$

令  $S_{\min}(M) = S_N(M) - cMr_0$ ,  $S_{\max}(M) = S_N(M) + cMr_0$ , 则对  $\forall p, 1 \leq p \leq N$ , 有

$$S_{\min}(M) \leq S_p(M) \leq S_{\max}(M),$$

又  $S_N(M) = \sum_{k=1}^M \frac{r_{\text{sam}}^{(N)}(k)}{\omega_N} \geq Factor_1 \cdot M \cdot r_0 \geq 0.9M \cdot r_0$ , 故  $S_{\min}(M) \geq 0.7Mr_0$ , 故对  $\forall p, 1 \leq p \leq N$ , 有

$$0 < S_{\min}(M) \leq S_p(M) \leq S_{\max}(M).$$

令  $K_M = \frac{S_{\max}(M)}{S_{\min}(M)} = \frac{S_N(M) + cMr_0}{S_N(M) - cMr_0} > 1$ , 则由引

理 4 可得

$$f_M \geq \frac{4K_M}{(K_M + 1)^2} = \frac{4}{K_M + \frac{1}{K_M} + 2},$$

因为  $K_M = \frac{S_{\max}(M)}{S_{\min}(M)} = \frac{S_N(M) + cMr_0}{S_N(M) - cMr_0} > 1$ , 故可得

$$f_M \geq \frac{4}{K_M + \frac{1}{K_M} + 2}$$

$$= \frac{4}{\frac{S_N(M) + cMr_0}{S_N(M) - cMr_0} + \frac{S_N(M) - cMr_0}{S_N(M) + cMr_0} + 2}$$

$$= 1 - \left[\frac{cMr_0}{S_N(M)}\right]^2.$$

又  $S_N(M) \geq 0.9M \cdot r_0$ , 故  $f_M \geq 1 - (10c/9)^2$ .

证毕.

**注.** 由于在证明下界时, 对于  $|S_p(M) - S_N(M)| \leq$

$\sum_{k=1}^M \left| \frac{r_{\text{sam}}^{(p)}(k)}{\omega_p} - \frac{r_{\text{sam}}^{(N)}(k)}{\omega_N} \right| \leq cMr_0$ , 我们取的是其上界, 而实际运行中  $S_p(M) \approx S_N(M)$ , 故实际运行时  $f_M$

的值非常接近 1. 其具体接近程度在接下来的实验中可以看到.

## 7 实验和分析

下面我们通过实验评价算法 GPCA 和算法 WFCC 的性能. 其中, 我们使用 Intel Lab Data<sup>①</sup> 来检验算法 GPCA 的压缩效率; 在 TOSSIM 模拟平台下, 通过与目前最好的基于速率的拥塞控制算法 PCCP<sup>[1]</sup> 进行对比实验来检验算法 WFCC 的性能.

### 7.1 算法 GPCA 性能评价

为了衡量算法 GPCA 的压缩性能, 我们给出两个性能指标, 其定义如下.

**压缩率.** 其值等于  $n/H$ , 其中  $n$  为原始时间序列  $S^{(n)}$  的长度,  $H$  为近似  $S^{(n)}$  的常值序列的长度.

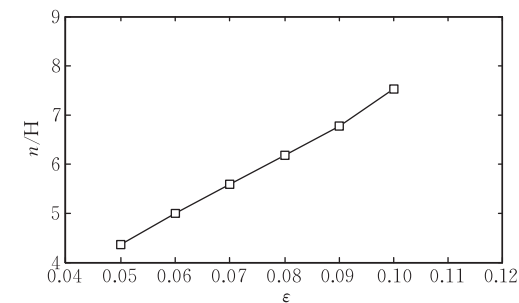
**平均绝对值误差.**  $SumOfAbsoError^{(n)}/n$ , 其中  $SumOfAbsoError^{(n)}$  为原始数据与近似数据的绝对值之差的累积和,  $n$  为原始序列长度.

下面, 分两组实验来检验算法 GPCA 的性能. 第 1 组: 在  $\epsilon$  不固定情况下, 检验算法 GPCA 在单源数据上的运行性能; 第 2 组: 在  $\epsilon$  固定情况下, 检验算法 GPCA 在单源数据上的运行性能.

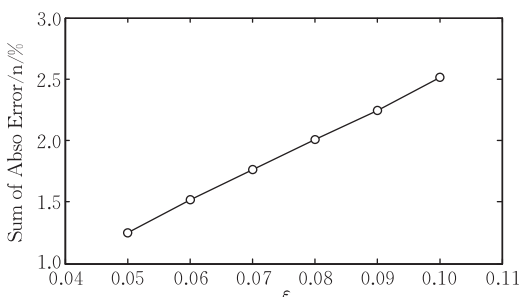
图 2 为算法 GPCA 在节点 1 上运行时其压缩比和平均绝对值误差随误差界  $\epsilon$  变化的情况. 这里我们使用的数据为 Intel Lab Data 中节点 1 采集的前 10000 个温度数据. 图 2(a) 为误差界  $\epsilon$  从 0.05 递

增到 0.1 时, 压缩比的变化情况. 易见, 随着误差界  $\epsilon$  的放宽 (即  $\epsilon$  增大), 压缩比逐渐增大. 当  $\epsilon=0.1$  时, 其压缩比等于 7.5301, 大大减少了数据的存储和通信开销. 图 2(b) 为误差界  $\epsilon$  从 0.05 递增至 0.1 时, 平均绝对值误差的变化情况. 当误差界  $\epsilon$  从 0.05 递增至 0.10 时, 平均绝对值误差从 0.01253 递增至 0.02515. 由此可见平均绝对值误差与最大误差之比接近于 1/4.

图 3 为在  $\epsilon=0.05$  的条件下, 算法 GPCA 在多个节点上的温度数据上运行时的性能. 这里我们使用的数据为 Intel Lab Data 中节点 1~10 (节点 5 除外, 因为其温度传感器故障) 各自采集的前 10000 个数据. 图 3(a) 为在  $\epsilon=0.05$  的条件下, 算法 GPCA 在各个节点上运行所达到的压缩比. 其中 GPCA 在节点 7 上压缩比最大, 为 7.7101; 在节点 1 上的压缩比最小, 为 4.3535. 由此可见节点 7 所监测区域的温度变化比节点 1 所监测区域缓慢. 图 3(b) 为在  $\epsilon=0.05$  的条件下, 算法 GPCA 在各个节点上运行产生的平均绝对值误差. 节点 2 的平均绝对值误差最小, 为 0.01221; 节点 4 的平均绝对值误差最大, 为 0.01349. 由图易见, 所有节点的平均绝对值误差基本维持在  $\epsilon/4$ , 即 0.0125 左右变化.

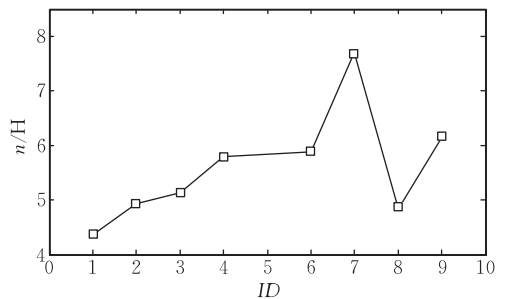


(a) 压缩比

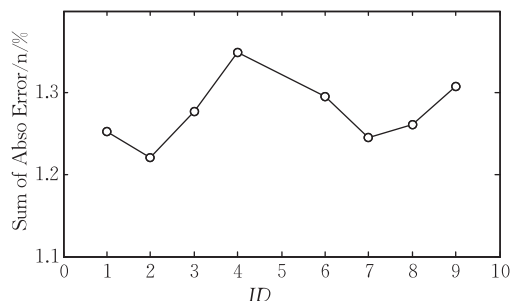


(b) 平均绝对值误差

图 2 单个节点,  $\epsilon$  变化时 GPCA 性能.



(a) 压缩比



(b) 平均绝对值误差

图 3 多个节点,  $\epsilon$  固定时 GPCA 性能

由图 2 和图 3 可知, 算法不仅达到了很好的压缩效率, 且其平均绝对值误差远低于最大值误差  $\epsilon$ .

① <http://db.csail.mit.edu/labdata/labdata.html>

## 7.2 算法 WFCC 评价

我们用 TOSSIM 来进行模拟实验来检验算法 WFCC 性能. 其中, TOSSIM (TinyOS simulator) 是 TinyOS (UC Berkeley 开发的开放源代码操作系统, 专为嵌入式无线传感网络设计) 自带的一个仿真工具, 可以支持大规模的网络仿真. 由于 TOSSIM 运行和传感器硬件相同的代码, 所以仿真编译器能直接从 TinyOS 应用的组件表编译仿真程序.

### 7.2.1 WFCC 和 PCCP 的对比实验

为了研究算法 WFCC 的性能, 我们和目前最好的基于速率控制的拥塞控制协议 PCCP<sup>[1]</sup> 进行基于 TOSSIM 平台的模拟对比实验. 下面我们简要介绍 PCCP 的核心思想. PCCP 的拥塞度量和本文一样, 为  $d(i) = \bar{t}_s^{(i)}(h+1) / \bar{t}_a^{(i)}(m+1)$ , 但其  $\bar{t}_a^{(i)}(m+1)$  是每收到  $N_p (= 50, \text{实验中})$  个数据包进行一次更新,  $\bar{t}_s^{(i)}(h+1)$  是每发送完一个数据包更新一次. 节点  $i$  触发拥塞控制的条件是: (1) 收到数据包数目为  $O(i) \cdot N_p$ , 其中  $O(i)$  为节点子孙节点总数; (2) 收到父亲节点发来的包含拥塞信息的数据包. 在条件 (1) 下, 节点更新拥塞度量为  $d_{\text{new}}(i)$  并把更新前的拥塞度量保存为  $d_{\text{old}}(i)$ . 若  $d_{\text{new}}(i) \leq d_{\text{old}}(i)$ , 则节点  $i$  通过广播包含拥塞信息的数据包的方法将其孩子节点  $j$  的发送速率更新为  $r_{\text{send}}^{(j)} = r_{\text{send}}^{(j)} / d_{\text{new}}(i)$ , 采样速率更新为  $r_{\text{sam}}^{(j)} = r_{\text{send}}^{(j)} \cdot \omega_j / s\omega_j$ ; 若  $d_{\text{new}}(i) > d_{\text{old}}(i)$ , 则其孩子节点发送速率限制为  $r_{\text{send}}^{(j)} = \text{base\_rate} \cdot s\omega_j / \omega_j$ . 在条件 (2) 下, 节点收到父亲的广播包, 进行条件 (1) 中的相应发送速率和采样速率的更新.

在下面的实验中, 我们采用的路由拓扑如图 4 所示, 其中  $\omega_3 = \omega_5 = \omega_{18} = 2, \omega_i = 1 (1 \leq i \leq 21, i \neq 3, 5, 18)$ .

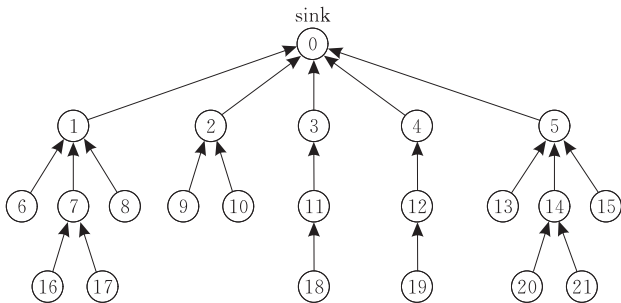


图 4 传感器网络(22 个节点)的路由拓扑

我们进行 4 组比较实验, 实验参数如表 1 和表 2 所示, 其中表 1 中 4 个参数两算法都有, 表 2 中 4 个参数中  $N_p$  为算法 PCCP 中参数,  $T, Factor_1$  和  $Factor_2$  为算法 WFCC 中参数.

表 1 对比实验参数设置(1)

算法/参数	AddFactor	MulFactor	$\omega_s$	$\omega_a$
WFCC ( $r_0 = 0.25 \text{ pkts/s}$ )	1.5	0.97	0.1	0.1
PCCP ( $r_0 = 0.25 \text{ pkts/s}$ )	1.5	0.97	0.1	0.1
WFCC ( $r_0 = 0.30 \text{ pkts/s}$ )	1.8	0.97	0.1	0.1
PCCP ( $r_0 = 0.30 \text{ pkts/s}$ )	1.8	0.97	0.1	0.1
WFCC ( $r_0 = 0.35 \text{ pkts/s}$ )	2.0	0.96	0.1	0.1
PCCP ( $r_0 = 0.35 \text{ pkts/s}$ )	2.0	0.96	0.1	0.1
WFCC ( $r_0 = 0.40 \text{ pkts/s}$ )	2.3	0.96	0.1	0.1
PCCP ( $r_0 = 0.45 \text{ pkts/s}$ )	2.3	0.96	0.1	0.1

表 2 对比实验参数设置(2)

算法/参数	$T$	$N_p$	$Factor_1$	$Factor_2$
WFCC ( $r_0 = 0.25 \text{ pkts/s}$ )	200	/	0.9	1.1
PCCP ( $r_0 = 0.25 \text{ pkts/s}$ )	/	50	/	/
WFCC ( $r_0 = 0.30 \text{ pkts/s}$ )	200	/	0.9	1.1
PCCP ( $r_0 = 0.30 \text{ pkts/s}$ )	/	50	/	/
WFCC ( $r_0 = 0.35 \text{ pkts/s}$ )	200	/	0.9	1.1
PCCP ( $r_0 = 0.35 \text{ pkts/s}$ )	/	50	/	/
WFCC ( $r_0 = 0.40 \text{ pkts/s}$ )	200	/	0.9	1.1
PCCP ( $r_0 = 0.45 \text{ pkts/s}$ )	/	50	/	/

实验中我们主要考察 2 个性能指标: sink 节点的吞吐量; 加权公平性. 在后面的对比实验中 will 比较这 2 个性能指标.

sink 节点的吞吐量是指 sink 节点单位时间内接收来自所有节点的数据包总数.

加权公平性用式(9)中的  $f_M$  来表示.  $f_M = f_{\text{max}} = 1$  时, 加权公平性最好;  $f_M = f_{\text{min}} = 1/N$  时, 加权公平性最差.

图 5 给出了在初始基准速率  $r_0 = 0.25 (\text{pkts/s})$  条件下, 本文算法 WFCC 与算法 PCCP 的吞吐量和加权公平性随实验时间变化的对比情况. 其中节点  $i$  的初始采样速率和发送速率分别为  $r_0 \cdot \omega_i = 0.25\omega_i (\text{pkts/s})$  和  $r_0 \cdot s\omega_i = 0.25s\omega_i (\text{pkts/s})$ . WFCC 和 PCCP 均有的参数加性增因子 AddFactor 和乘性减因子 MulFactor 分别设为 1.5 (pkts/s) 和 0.97. 其余参数设置如表 1 和表 2 所示. 数据为每隔 50 s 统计得到. 从图 5(a) 中可以看到: (1) 随着时间的增加, WFCC 的吞吐量基本维持在 4.35 到 4.5 (pkts/s), 变化不是很大, 体现了 WFCC 在吞吐量上的保障; (2) 在与 PCCP 的吞吐量的对比中, 可以看到随着时间的增加, PCCP 吞吐量逐渐在下降. 这是由于 PCCP 中子节点得知父节点拥塞度增大时, 进行了限速措施, 从而子节点流出速率减少, 逐渐拥塞, 进一步影响其子节点的采样速率和发送速率, 故如此逐层限速, 最后使得整个网络吞吐量逐渐下降. 从图 5(b) 中可以看到: (1) WFCC 的加权公平性基本维持在 0.96 以上, 与我们的理论分析相符; (2) 而 PCCP 的加权公平性由第 2 个 50 s 时的 0.966 逐渐下降至第 10 个 50 s 时的 0.836. 这是由

于 WFCC 在计算得到流入总速率后严格按权值大小比例将此速率分配给自身和孩子节点;而 PCCP 没有保障流入速率的公平分配.

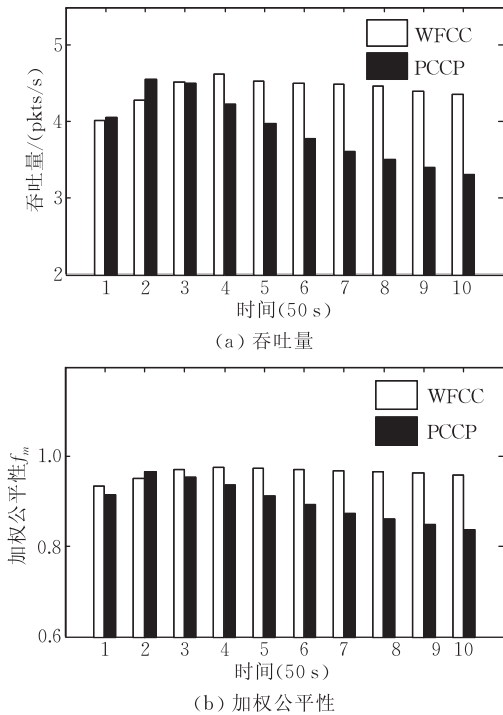


图 5 初始条件  $r_{\text{init\_sam}}^{(i)} = 0.25\omega_i, r_{\text{init\_send}}^{(i)} = 0.25s\omega_i$

图 6~图 8 分别给出了在初始基准速率  $r_0 = 0.30(\text{pkts/s})$ 、 $0.35(\text{pkts/s})$  和  $0.40(\text{pkts/s})$  条件下,本文算法 WFCC 与算法 PCCP 的吞吐量和加权公平性随实验时间变化的对比情况.其中对比实验参数设置如表 1 和表 2 所示,实验时间均为 500 s.从 3 个图的图(a)可以看到 WFCC 的吞吐量变化比较慢且平稳,且高于 PCCP 的吞吐量;而 PCCP 的吞吐量变化则比较快,随着时间增加逐渐下降,这是由于 PCCP 的限速所致.从 3 个图的图(b)可以看到 WFCC 的加权公平性基本维持在 0.95 以上,验证了我们的理论分析;而 PCCP 的加权公平性仅在前 100 s 内维持在 0.9 以上,之后随时间增加逐渐减少,最低时其值小于 0.8.

### 7.2.2 WFCC 的参数变化对性能的影响

为了进一步检验算法 WFCC 的性能,我们研究了两个重要参数周期  $T$  和初始基准速率  $r_0$  对算法 WFCC 性能的影响.具体方法是:在其中一个参数值固定另一个参数值变化的条件下,研究 WFCC 吞吐量和加权公平性的变化情况.

图 9 给出了在周期  $T=200\text{ s}$ , WFCC 的吞吐量和加权公平性随初始基准速率  $r_0$  变化的情况.其中实验时间为 500 s.从图 9(a)可以看出吞吐量随着初始基准速率增加逐渐增大,且吞吐量与初始速率比

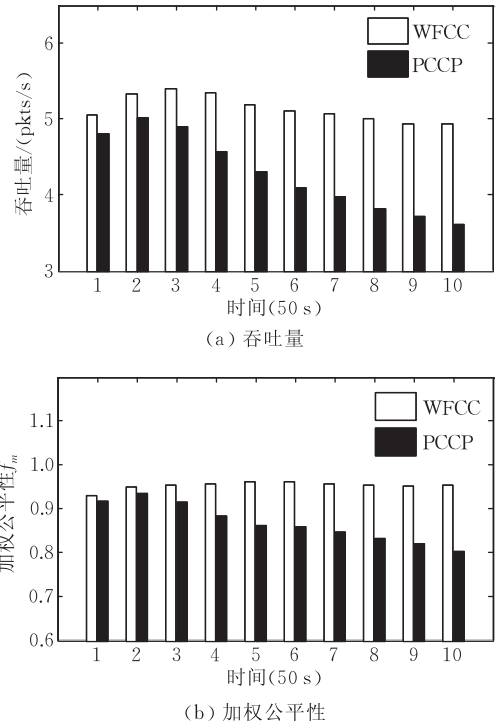


图 6 初始条件  $r_{\text{init\_sam}}^{(i)} = 0.30\omega_i, r_{\text{init\_send}}^{(i)} = 0.30s\omega_i$

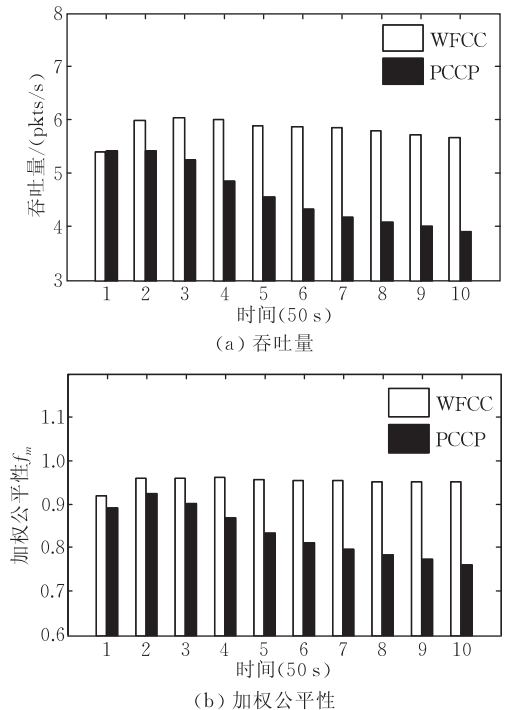
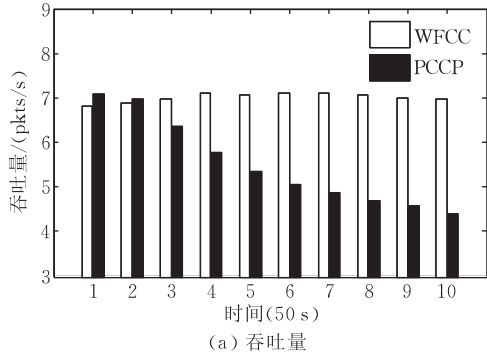


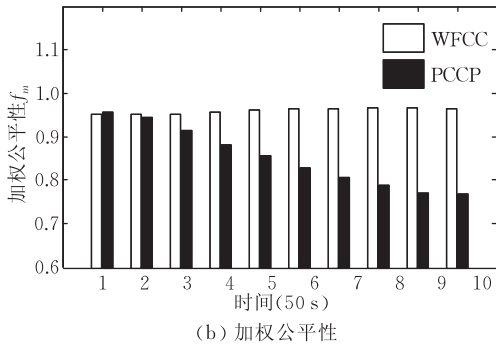
图 7 初始条件  $r_{\text{init\_sam}}^{(i)} = 0.35\omega_i, r_{\text{init\_send}}^{(i)} = 0.35s\omega_i$

例接近正比关系;从图 9(b)可以看出加权公平性基本维持在 0.95 以上,符合我们的理论分析.因此在周期  $T$  固定,初始基准速率  $r_0$  变化条件下,WFCC 的各方面性能表现良好.

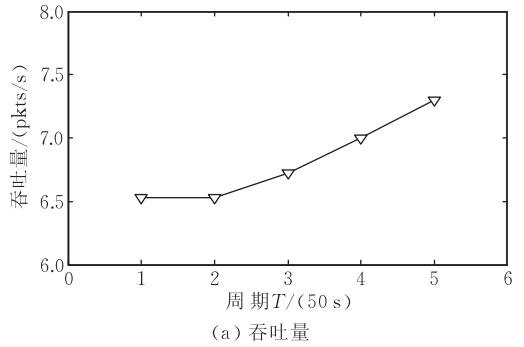
图 10 给出了在初始基准速率  $r_0 = 0.4\text{ pkts/s}$ , WFCC 的吞吐量和加权公平性随周期  $T$  变化的情



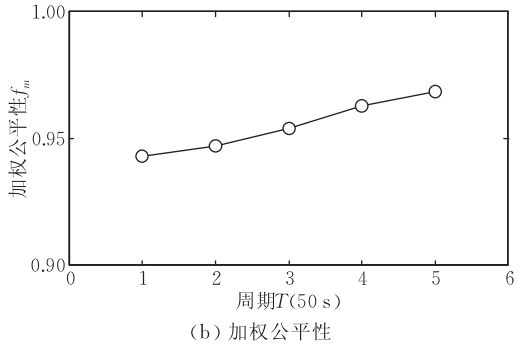
(a) 吞吐量



(b) 加权公平性

图 8 初始条件  $r_{init\_sam}^{(i)} = 0.40\tau\omega_i$ ,  $r_{init\_send}^{(i)} = 0.40s\tau\omega_i$ 

(a) 吞吐量



(b) 加权公平性

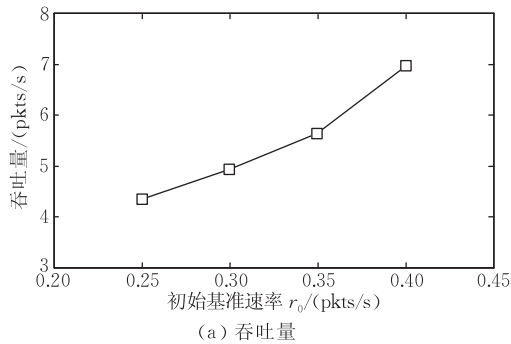
图 10  $r_0 = 0.4$  (pkt/s), 周期  $T$  变化

## 8 结 论

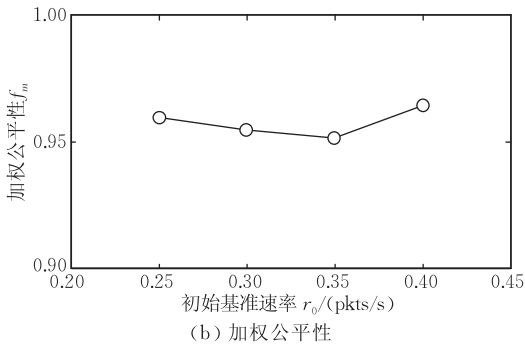
本文针对无线传感器网络的拥塞问题提出了一个  $\epsilon$  近似和加权公平性保证的拥塞控制算法, 其由两个子算法组成: 算法 GPCA 和算法 WFCC. GPCA 采取贪心分段常值  $\epsilon$  近似的策略, 使得一个连续子序列仅需要一个常值来近似且保证其与真实值的误差不大于  $\epsilon$ , 这样不仅减少传感器网络的数据传输量, 而且保证数据的精确度. 最后, 我们在理论上证明了这种常值近似的最优性. WFCC 以进入节点的平均数据包速率除以流出节点的平均数据包速率作为拥塞度量, 将流入的数据包速率公平地分配给孩子节点和自身节点分别作为其在下一个周期的发送速率和采样速率来实现加权公平性. 最后, 我们从理论上证明了通过这种局部的公平速率分配策略可以达到很好的全局公平性且得到了一个理论上的下界. 实验结果表明, GPCA 具有良好的压缩性能; WFCC 不仅达到了较高的吞吐量, 且其加权公平性基本维持在 0.95 以上, 验证了我们的理论分析.

## 参 考 文 献

- [1] Wang C, Li B, Sohrawy K, Daneshmand M, Hu Y. Upstream congestion control in wireless sensor networks through cross-layer optimization. IEEE Journal on Selected Areas in Communications, 2007, 25(4): 786-795



(a) 吞吐量



(b) 加权公平性

图 9  $T=200$  s, 初始基准速率  $r_0$  变化

况. 其中实验时间为 500 s. 从图 10(a)可以看出吞吐量随周期增大逐渐增加, 其原因在于短周期的频繁调速降低了部分节点的采样速率和发送速率; 从图 10(b)可以看出加权公平性随周期增加逐渐增大, 且维持在 0.95 左右. 因此在进行参数设置时, 周期  $T$  不宜太小且不宜太大, 一般设置在 200 s 比较合适.

- [2] Chen S, Zhang Z. Localized algorithm for aggregate fairness in wireless sensor networks//Proceedings of the 12th Annual International Conference on Mobile Computing and Networking. Los Angeles, USA, 2006; 274-285
- [3] Wan S, Eisenman S B, Campbell A T. CODA: Congestion detection and avoidance in sensor networks//Proceedings of the ACM Conference on Embedded Networked Sensor Systems. Los Angeles, USA, 2003; 266-279
- [4] Sankarasubramaniam Y, Akan O, Akyildiz I. ESRT: Event-to-sink reliable transport in wireless sensor networks//Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing. Annapolis, USA, 2003; 177-188
- [5] Hull B, Jamieson K, Balakrishnan H. Mitigating congestion in wireless sensor networks//Proceedings of the ACM Conference on Embedded Networked Sensor Systems. Baltimore, USA, 2004; 134-147
- [6] Rangwala S, Gummadi R, Govindan R. Interference-aware fair rate control in wireless sensor networks//Proceedings of the ACM Conference on Special Interest Group on Data Communication. Pisa, Italy, 2006; 63-74
- [7] Jain R K. The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation, and modeling. Hoboken, USA, Wiley, 1991
- [8] Buragohain C, Shrivastava N, Suri S. Space efficient streaming algorithms for the maximum error histogram//Proceedings of the International Conference on Data Engineering, Istanbul, Turkey, 2007; 1026-1035
- [9] Gandhi S, Nath S, Suri S, Liu J. GAMPS: Compressing multi sensor data by grouping and amplitude scaling//Proceedings of the ACM SIGMOD International Conference on Management of Data. Providence, USA, 2009; 771-784
- [10] Li Y, Li J, Gao H. Enabling  $\epsilon$ -approximate querying in sensor networks//Proceedings of the International Conference on Very Large Data Bases. Lyon, France, 2009; 169-180
- [11] Ee C-T, Bajcsy R. Congestion control and fairness for many-to-one routing in sensor networks//Proceedings of the ACM Conference on Embedded Networked Sensor Systems. Baltimore, USA, 2004; 148-161
- [12] Chen S, Yang N. Congestion avoidance based on lightweight buffer management in sensor networks. IEEE Transactions on Parallel and Distributed Systems, 2006, 17(9): 934-946
- [13] Paek J, Govindan R. RCRT: Rate-controlled reliable transport for wireless sensor networks//Proceedings of the ACM Conference on Embedded Networked Sensor Systems. Sydney, Australia, 2007; 305-319
- [14] Li G, Li J. Global weighted fairness guaranteed congestion avoidance protocol for wireless sensor networks//Proceedings of the IEEE International Conference on Computer Communications. Shanghai, China, 2011; 601-605
- [15] Raghunathan V, Schurgers C, Park S, Srivastava M-B. Energy-aware wireless microsensor networks. IEEE Signal Processing Magazine, 2002, 19(2): 40-50



**LI Guo-Hua**, born in 1984, Ph. D. candidate. His research interests include wireless sensor networks, cyber-physical systems.

**LI Jian-Zhong**, born in 1950, professor, Ph. D. supervisor. His research interests include wireless sensor networks, cyber-physical systems, database, massive data processing etc.

**GAO Hong**, born in 1966, Ph. D., professor, Ph. D. supervisor. Her research interests include wireless sensor networks, cyber-physical systems, massive data management and data mining.

## Background

This paper focuses on research for congestion control problem in wireless sensor networks. Congestion control is an important problem in wireless sensor networks. Because congestion causes information loss, throughput reduction, and energy waste. Therefore congestion in WSNs needs to be controlled in order to prolong system lifetime. Most of the existing congestion control algorithms for wireless sensor networks seldom concern the combined problem of data compression and weighted fairness. In order to address this problem, a greedy piecewise constant  $\epsilon$ -approximation algorithm GPCA and a novel decentralized congestion control algorithm WFCC are proposed in this paper. GPCA approximates a subsequence using a constant and guarantees that the error between the real sequence and the approximation sequence is less than or equal to  $\epsilon$ . In addition, we prove the optimality of constant  $\epsilon$ -approximation in theory. WFCC not only mitigates congestion, but also guarantees the weighted fairness among all sensor nodes. Importantly, we give a lower bound of weighted fairness metric  $1 - (10c/9)^2$  where  $c$  is a constant

and  $0 < c < 0.2$ . We evaluate GPCA using Intel Lab Data, and the results show that GPCA has great compression performance. In addition, we evaluate the algorithm WFCC in a 22-node wireless sensor network by TOSSIM. The results of simulation demonstrate that WFCC achieves high throughput and weighted fairness (above 95% on average).

This work is supported in part by the Key Program of the National Natural Science Foundation of China under Grant Nos. 61033015, 60933001; the National Natural Science Foundation of China under Grant Nos. 60831160525, 61100030; the China Postdoctoral Science Foundation under Grant No. 20110491060 and the Fundamental Research Funds for the Central Universities under Grant No. HIT.NSRIF.201179. Our group has been working on the research of wireless sensor networks for many years. Many good papers have been published in worldwide conferences and transactions. such as SIGMOD, VLDB, ICDE, KDD, INFOCOM, TPDS, IPL, TKDE, VLDB Journal et al..