

## QoS Mechanism of WSN Based on Rate Adjustment and DiffServ \*

ZHU Chang, JIN Xin-yu \*, ZHANG Yu, TANG Jun

(Dept. of Information Science and Electronic Eng, Zhejiang Univ, Hangzhou 310027, China)

**Abstract :** The modeling and analysis of QoS of a multi-hop WSN consisting of clusters arranged linearly are performed by employing the queueing theory. Formula of non-end-to-end QoS parameters called CDL (Collective Data Loss) and CDD (Collective Data Delay) are given. Furthermore, a rate adjustment algorithm by reducing the data generating rate at the clusters from the sink is proposed to ensure the QoS. In order to adopt "DiffServ", a preemptive priority scheduling algorithm called PPS is proposed so that different services have different QoS. The result of the simulation proves that the rate adjustment algorithm has improved the QoS so much that the CDL decreased by 23.0% ~ 100% and the maximum reduction of CDD is up to 89.6%, meanwhile the PPS scheduling algorithm has realized the "differentiated services" QoS mechanism better than SATS algorithm in ensuring the important data's CDD and the low network jitter.

**Key words :** WSN; QoS; queueing system; rate adjustment; diffServ

**EEACC :** 6150P; 7230

## 基于速率调整和区分服务的 WSN QoS 机制 \*

朱 畅, 金心宇 \*, 张 昱, 唐 军

(浙江大学信息与电子工程系, 杭州 310027)

**摘 要 :** 基于排队论对多跳线性拓扑结构的无线传感器网络 QoS 进行了建模和分析, 提出了非端到端的集体数据丢包率 CDL 和时延 CDD 的 QoS 指标的具体公式。进而提出了从 Sink 节点开始逐级减少的速率调整算法来保证网络 QoS。又引入“区分服务”到本文模型, 提出了基于抢先优先制的 PPS 调度算法以实现不同的业务有不同的 QoS 参数。仿真表明, 速率调整算法能明显提高网络的 QoS 性能, CDL 下降达 23.0% ~ 100%, CDD 最多减小 89.6%, 而 PPS 算法在重要数据的时延 CDD 和网络抖动等 QoS 参数上明显优于 SATS 算法, 从而更好地实现了“区分服务”的 QoS 机制。

**关键词 :** 无线传感器网络; 服务质量; 排队系统; 速率调整; 区分服务

**中图分类号 :** TP393

**文献标识码 :** A

**文章编号 :** 1004-1699(2009)05-0694-06

目前, 传感器网络的研究大部分集中在能量有效的机制上<sup>[1]</sup>。然而, 随着研究和应用的深入, 尤其是需要传播视频、语音等对 QoS 参数敏感的实时业务的无线多媒体传感器网络 (WMSN) 的出现<sup>[2]</sup>, 使传感器网络 QoS 机制的研究成为热点。但是无线传感器网络在网络自身和应用需求上有明显特点, 比如无线传感器网络所保障的 QoS 是非端到端的 (non-end-to-end) 概念<sup>[5]</sup>, 网络所要保证的 QoS 不是汇聚节点 Sink 与单个传感器节点的端到端之间,

传统的端到端的 QoS 衡量指标便失去了意义。

当前, 速率控制在传统 IP 网络中得到了较大的规模的应用, 比如 TCP 中使用了慢启动、加性增、乘性减等速率控制思想实现了拥塞控制<sup>[3]</sup>, 但是有关速率控制在 WSN 领域的应用研究工作还很少。文献[4]提出了一种通过控制速率实现到 Sink 节点的可靠传输机制 (ESRT)。

此外, 现有无线传感器网络研究大都假定基于单一的“尽力而为” (best effort) 数据传输服务模型。

基金项目: 浙江省科技计划项目资助 (2005CS31001)

收稿日期: 2008-12-31 修改日期: 2009-01-31

该模型下,各类业务平等竞争网络资源,网络拥塞时,业务被不加区分地丢弃,这样不利于服务质量要求高的业务传输。因此,提供不同业务服务质量的“区分服务”QoS 机制正显示出广阔应用前景,受到越来越多的关注<sup>[5-6]</sup>。文献[7]将“区分服务”思想引入到 WSN 中,基于带宽划分针对两种 QoS 需求不同的数据包采用了不同的调度策略。文献[8]提出了一种服务感知业务调度算法 SATS,并通过仿真分析表明 SATS 算法比文献[7]的带宽划分调度算法整体上提供了更为有效的 QoS 保证<sup>[8]</sup>。

总之,传感器网络 QoS 研究面临许多新的挑战,国内外对该领域的研究才刚刚起步,有价值的成果相对较少<sup>[9]</sup>。本文针对传感器网络特点,给出了非端到端 QoS 指标的具体公式,并基于速率控制和区分服务提出了速率调整算法和调度算法实现保障 WSN QoS 的机制。

### 1 系统模型

本文研究的传感器网络模型<sup>[10]</sup>如图 1 所示。该模型由  $M$  个簇线性排列组成,簇  $i$  有  $n_i$  个传感器节点和一个簇头,簇头负责管理该簇里  $n_i$  个传感器节点和转发前一个簇  $i + 1$  过来的数据流到下一个簇  $i - 1$ ,最终由簇 1 转发到到汇聚节点 Sink。Sink 节点到簇 1 的距离为  $d_1$ ,簇  $i - 1$  到簇  $i$  的距离为  $d_i$ 。该模型代表了簇和路由已经形成的 WSN 模型,具有普遍意义<sup>[10]</sup>。

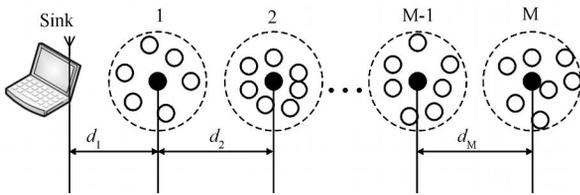


图 1 基于分簇的线性传感器网络模型

每个簇的簇头受限于其有限的处理能力和内存等,当簇头中等待处理的数据包超过其容量后,继续到达的数据包会被丢弃,因此可以看成是一个有限容量损失制的排队系统。排队规则遵守 FCFS(先到先服务)原则,具有一个服务台。簇  $i$  的簇头记为簇头  $i$ ,簇头  $i$  的服务时间服从强度为  $\mu_i$  负指数分布,簇  $i$  的容量记为  $K_i$ 。簇  $i$  中每个传感器节点的数据产生过程可以认为是一个泊松过程并且相互独立<sup>[11]</sup>,设簇  $i$  中每个传感器节点有相同的数据产生强度,记为  $g_i$ (并且假定该值可以通过簇头发送命令给传感器节点而调整),由泊松过程的可加性可以得到所有这些传感器节点产生的数据到达簇头的过程也是一个泊松过程,强度记为  $\lambda_i$ 。因此每个簇头可

以建模为一个  $M/M/1/K_i$  的排队系统,数据到达速率为  $\lambda_i$ ,服务速率为  $\mu_i$ 。而整个线性网络可以看成是一个串联开排队网络系统。由 Jackson 定理<sup>[12]</sup>可以得到:

$$\lambda_i = \begin{cases} n_M g_M & i = M \\ n_i g_i + \lambda_{i+1} [1 - p_{i+1}(K_{i+1})] & \\ n_i g_i + \sum_{j=i+1}^M n_j g_j [1 - p_k(K_k)] & \\ & 1 \leq i \leq M - 1 \end{cases} \quad (1)$$

对于簇头  $i$ ,稳态时系统队长等于  $j$  的概率记为  $p_i(j)$  如公式(2)所示<sup>[13]</sup>,当  $j = K_i$  系统容量满,该簇头阻塞。

$$p_i(j) = \frac{(1 - \rho_i)^j}{1 - \rho_i^{K_i+1}}, \text{ 其中 } \rho_i = \frac{\lambda_i}{\mu_i}, 0 \leq j \leq K_i \quad (2)$$

### 2 速率调整算法

与传统 IP 网络端到端的 QoS 度量指标不同,无线传感器网络采用的是非端到端的 QoS 指标,非端到端的概念所针对的是无线传感器网络面向以数据为中心(data-centric)的应用,应用所决定的任务交给一组传感器节点来执行,即网络层面的 QoS 问题映射到节点层面来解决;节点层面的 QoS 所要保证的是 sink 节点与执行任务的多个传感器节点之间的性能指标是否满足 QoS 的要求,而不是针对具体的某个传感器节点。非端到端 QoS 指标定义如下:集体性丢包率 CDL (collective data loss):在信息传递过程中与监测事件相关的数据包丢失的数量,而不是单个节点的丢包率。集体性时延 CDD (collective data delay):监测到事件发生后传感器节点产生的第一个数据包与最后一个到达 Sink 的数据包之间的时间差<sup>[5]</sup>。

根据上述定义和本文的模型可以定义出反映传感器网络丢包率和延迟的非端到端的 QoS 具体指标。到达 Sink 节点的流量是用户接收到的真正有效的流量,因此可以定义数据的集体丢包率公式  $CDL = (\text{总的产生流量} - \text{到达 Sink 节点的流量}) / \text{总的产生流量} = 1 - \text{到达 Sink 节点的流量} / \text{总的产生流量}$ ,可得

$$CDL = 1 - \lambda_1 [1 - p_1(K_1)] / \sum_{i=1}^M n_i g_i \quad (3)$$

其中  $p_1(K_1)$  可由公式(1)、式(2)得到。簇  $i$  的延迟  $Delay_i = \text{排队延迟} + \text{传输延迟} + \text{处理延迟} + \text{传播延迟}$ ,对一个多跳的传感器网络来说处理延迟和传播延迟相对于排队和传输延迟非常小,可忽略不计。而排队和传输延迟之和就是分组在  $M/M/1/K_i$  的驻留时间,因此

$$\text{Delay}_i = \frac{1}{(\mu_i - \lambda_i) [1 - p_i(K_i)]} - \frac{(K_i + 1) \lambda_i^{K_i+1}}{\mu_i (1 - \lambda_i^{K_i+1}) [1 - p_i(K_i)]} \quad (4)$$

定义集体数据延迟公式

$$\text{CDD} = \sum_{i=1}^M \text{Delay}_i \quad (5)$$

这些指标反应了网络的整体状况,而非单个节点状况,具有非端到端的特点。随着每个传感器节点数据产生率  $g$  的增加,由公式(3)分析得 CDL 减少,由公式(5)分析得 CDD 增大, QoS 质量下降,因此可以采取某种机制(比如簇头给传感器节点发送相应命令)减少数据产生率,从而提高 CDL 和 CDD。由于越靠近 Sink 节点的簇头必须承担自身簇流量以及它之后的所有簇流量的转发工作,工作量较大,数据阻塞率就较大,因此降低靠近 Sink 节点的簇的数据产生率对系统 QoS 参数的提高效果更明显(数学分析也是如此)。减少到一定程度如果还达不到 QoS 要求,则减少它之后的簇的流量的产生率,逐级往后。算法如下:

用户设定好期望的丢包率  $\text{CDL}_r$ ; 每个簇设定好一个最小数据产生率  $g_{\min}$  (简便起见,设每个簇最小数据产生率都相同);

簇  $i$  每次调整的数据产生率调整量为  $i$

SUB 命令为要求减少数据产生率命令

Sink 节点上运行如下算法:

While (TRUE)

{ 计算出当前的 CDL;

  If (CDL >  $\text{CDL}_r$ )

    给簇 1 发送 SUB 命令;

}

簇  $i$  上运行如下算法:

While (TRUE)

{ 接收到上一簇 ( $i-1$ ) 发送的命令 Command

  If (Command == SUB)

    If ( $g_i > g_{\min}$ )

      减少簇内传感器节点的数据产生率;

      使之  $g_i = g_i - i$ ;

    Else

      发送 SUB 命令给下一个簇 ( $i+1$ );

}

其中计算 CDL 会需要较大的计算资源和能耗,但是这一步放在汇聚节点上进行,汇聚节点的能量和资源一般不受限。而在资源和能量受限的各簇头节点上运行的算法十分简单,开销小。因此本文提出的算法适合分簇的传感器网络的结构特点。

### 3 区分服务的 QoS 机制

传感器网络的不同业务有不同的 QoS 需求,比

如文献[5]根据传输模型的不同分为 3 种业务类型——基于事件驱动的应用、基于查询驱动的应用和持续不断传输的应用,并且一般来讲,事件驱动的应用不容许有时延,查询驱动的应用可以设定一定时延,持续不断传输应用则容许一定时延,对时延的 QoS 需求依次降低。

SA TS 算法采用非抢先优先制的调度思想,而抢先优先制的调度思想在实时操作系统等领域已经常采用<sup>[14]</sup>,有鉴于此,本文在 SA TS 算法的基础上作了改进,提出了一种抢先优先制的调度算法 PPS (preemptive priority scheduling),以期更好地适应实时性要求比较高的 WSN 的需求,算法如下:针对以上 3 种类型数据包使用不同队列,在簇头节点设置一个分类器,判定所收数据包类型后送至相应队列,队列调度策略采用抢先优先制。

假定 QueueTime 为数据包的排队延迟; $p$ . priority 为到达的数据包  $p$  的优先级, $p$ . type 为数据包的类型。这里假定有三类数据包,事件驱动、查询驱动和持续传输,优先级依次降低;由于最低优先级的数据包(通常是多媒体传感器网络中的媒体数据)不太重要,允许一定的丢包率,因此可以计算这类数据包的累计时延  $p$ . delay,如果大于用户定义的最终期限 deadline,可以丢弃以减少网络拥塞,进一步提高 QoS 性能。GetCurrentPriority() 为返回当前被服务数据包的优先级,GetQueueTime(int pri) 返回优先级为 pri 的排队时间。

While (TRUE)

{ If ( $p$ . priority == -1) // 数据包进行优先级分类

  If ( $p$ . type == 事件驱动)

$p$ . priority = 3;

  Elseif ( $p$ . type == 查询驱动)

$p$ . priority = 2;

  Else

$p$ . priority = 1;

  If ( $p$ . priority == 1)

$p$ . Delay += GetQueueTime(1);

    If ( $p$ . Delay > deadline)

      丢弃数据包  $p$ ;

    Else

      插入到队尾;

  Else

    Pri = GetCurrentPriority();

    If ( $p$ . priority > Pri)

      中断正在被服务数据包的服务;

$p$  插入到服务窗口服务;

      正在被服务的数据包回到优先级为 Pri 的队列头部;

    Else

$p$  插入到优先级为  $p$ . priority 的队列尾部;

}

用排队论理论对 PPS 算法进行分析。对某簇头来说,  $p_m$  为簇头达到最大容量时的概率。设  $i$  为第  $i$  级优先权数据的到达率,一共有  $N$  级数据 ( $1 \leq i \leq N$ ), 因此系统的总到达率  $\lambda = \sum_{i=1}^N \lambda_i$ , 记  $W_{qi}$ 、 $W_{si}$  分别为第  $i$  级数据在簇头中的平均排队时间和平均逗留时间。为便于分析, 设各级数据的服务时间一样, 均为  $1/\mu$ , 则  $\lambda_i = \lambda_i/\mu$  是第  $i$  级数据的业务量, 总业务量  $\lambda = \sum_{i=1}^N \lambda_i$ 。当有第 1 级优先权的数据到达时, 由于抢占优先权, 可认为系统中不存在其他级别的数据。因此

$$W_{q1} = \lambda_1(1 - p_m)W_{q1} + \lambda_1(1 - p_m)/\mu \quad (6)$$

式(6)中, 右边第一项为正在排队等待服务的第 1 级数据花费的平均服务时间, 右边第二项为正在服务的窗口出来的平均时间。由式(6)可解得

$$W_{q1} = \frac{1 - p_m}{1 - \lambda_1(1 - p_m)} \frac{1}{\mu} \quad (7)$$

进一步可得  $W_{s1}$ ,  $W_{s1} = W_{q1} + 1/\mu$ 。当第 2 级优先权数据到达时, 由于存在抢占优先权, 可认为系统中只存在第 1、2 级优先权数据, 记  $W_{s,1-2}$  为第 1 到 2 级数据在系统中时数据的平均逗留时间, 易得

$$(\lambda_1 + \lambda_2)W_{s,1-2} = \lambda_1W_{s1} + \lambda_2W_{s2} \quad (8)$$

由于较高级别数据到达时较低级别正在接受服务的数据只得中断并重新参加排队等待, 其服务时间分布不受前段已获得服务及服务多长时间的影响, 仍为负指数分布, 因此  $W_{s,1-2}$  可以按照到达率为  $\lambda_1 + \lambda_2$  时的  $M/M/1/K_i$  的排队模型公式算出, 得到

$$W_{s,1-2} = \frac{1}{\mu(1 - (\lambda_1 + \lambda_2)/\mu)} - \frac{m(2)^m}{\mu(1 - (2)^m)} \quad (9)$$

其中  $m = K_i$ ,  $(2) = (\lambda_1 + \lambda_2)/\mu$ 。

由式(8)、式(9)可解得  $W_{s2}$ , 以此类推, 归纳可得第  $k$  级数据的  $W_{sk}$  如下所示:

$$W_{sk} = \frac{1 - p_m}{1 - \lambda_1(1 - p_m)} \frac{1}{\mu} + \frac{1}{\mu}, (k = 1) \quad (10)$$

$$W_{sk} = \frac{\sum_{i=1}^k \lambda_i}{k} W_{s,1-k} - \frac{\sum_{i=1}^{k-1} \lambda_i W_{si}}{k}, (1 < k \leq N) \quad (11)$$

$$\text{其中, } W_{s,1-k} = \frac{1}{\mu(1 - (\lambda_1 + \dots + \lambda_k)/\mu)} - \frac{m(k)^m}{\mu(1 - (k)^m)},$$

$$m = K_i, (k) = (\sum_{i=1}^k \lambda_i)/\mu \quad (12)$$

作为对比, 利用在文献[8]中提出的 SATS 算法, 将其应用在本文模型中分析得到的  $W_{sk}$  如下两式所示:

$$W_{sk} = \frac{1 - p_m}{1 - \lambda_1(1 - p_m)} \frac{1}{\mu} + \frac{1}{\mu}, (k = 1) \quad (13)$$

$$W_{sk} = \frac{1 - (1 - p_m) \prod_{i=1}^{k-2} \lambda_i}{1 - (1 - p_m) \prod_{i=1}^k \lambda_i} [W_{s(k-1)} - \frac{1}{\mu}] + \frac{1}{\mu}, (1 < k \leq N) \quad (14)$$

### 4 仿真与结果

本文用 MATLAB 对算法进行了仿真和分析。系统参数设置如下: 簇的个数  $M = 5$ , 各个簇的传感器节点数  $n_1 = n_2 = \dots = n_5 = 100$ , 各个簇头的服务速率  $\mu_1 = \mu_2 = \dots = \mu_5 = 25$ , 各个簇头容量  $K_1 = K_2 = \dots = K_5 = 100$ , 各个簇的数据产生率  $g_1 = g_2 = \dots = g_5 = g$  如各图所示, 速率调整算法的调整参量经反复试验设定为  $\alpha_1 = \alpha_2 = \dots = \alpha_5 = 0.01$ ,  $g_{min} = 0.02$ 。业务 1、2 和 3 分别为事件驱动、应用驱动和持续传输, 重要性依次降低, 对 CDD 的需求依次降低。假定 3 种业务产生的概率一样。

未调整前, 集体数据丢包率 CDL 曲线如图 2 的  $CDL_1$  曲线所示, 集体数据时延 CDD 如图 3 的  $CDD_1$  曲线所示。CDL 和 CDD 都随着数据产生率的增加而增加, 并且 CDL 刚开始时 ( $g = 0.04 \sim 0.05$ ) 很小, 增幅也很小, 称此时为 CDL 的最佳区间, 而后 ( $g > 0.05$ ) 增幅较大。经过本文提出的算法调整后达到  $g_1 = g - 2 \alpha_1 = g_{min}$ ,  $g_2 = g - 2 \alpha_2 = g_{min}$ ,  $g_3 = g - 3 \alpha_3$ ,  $g_4 = g_5 = g$ , 这时的 CDL 曲线如图 2 的  $CDL_2$  曲线所示, 整条曲线往右平移, CDL

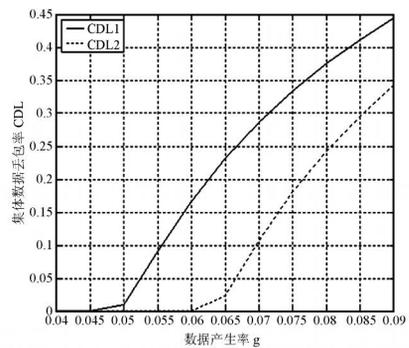


图 2 速率调整前后 CDL 曲线对比

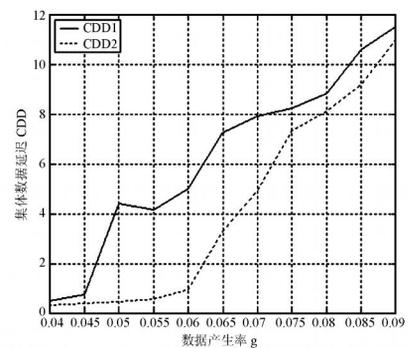


图 3 速率调整前后 CDD 曲线对比

下降了 23.0% ~ 100%, 最佳区间 (变为 0.04 ~ 0.065) 也变长了, CDL 得到极大改善; CDD 曲线如图 3 的 CDD<sub>2</sub> 曲线所示, CDD 减小, 最大减小幅度达 89.6%, 尤其在区间 (0.04 ~ 0.06) 效果最明显, CDD 也得到极大改善, 可见整个网络的 QoS 得到很大提高。

图 4 是引入本文提出的 PPS 算法的前后对比图, 引入前, 业务 1、2 和 3 都具有一样的 CDD 曲线, 如图 4 中的曲线 0 所示。引入后, 业务 1、2 和 3 的 CDD 曲线分别如图 4 中的曲线 1、2 和 3 所示。重要性高的业务 1、2 的延迟大大减少, 并且随着数据产生率增加而增加的增幅很小, 其中重要性最高的业务 1 的延迟几乎为 0, 优先级高的业务得到了很好的 QoS 保证, 而不太重要的业务 3 的 CDD 有所增加, 该算法很好地实现了“区分服务”的 QoS 机制。

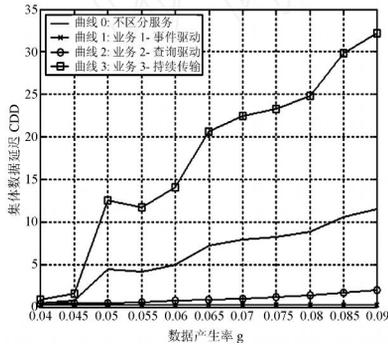


图 4 引入区分服务前后 CDD 曲线对比 (PPS)

图 5 是引入 SATS 算法的前后对比图, SATS 算法也实现了较好的区分服务。但是本文的 PPS 算法明显优于 SATS 算法: 首先, 在保障重要的数据 QoS 参数上 PPS 算法更好, 如图 6 是两种算法业务 1 数据的 CDD 曲线对比图, PPS 的 CDD 低于 SATS; 其次, 在保障不重要的数据 QoS 参数上 PPS 也优于 SATS, 如图 7 是两种算法业务 3 数据 CDD 对比图, PPS 曲线平滑, SATS 曲线则抖动较大, 业务 3 的数据一般来说是媒体数据, 媒体数据对延迟抖动非常敏感, 这样不利于媒体数据 QoS 参数保障。

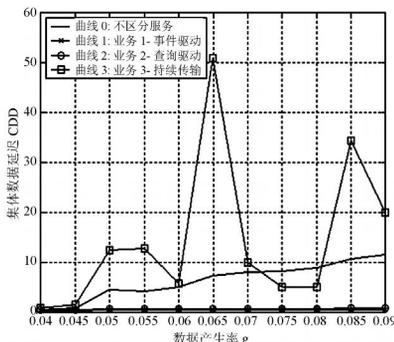


图 5 引入区分服务前后 CDD 曲线对比 (SATS)

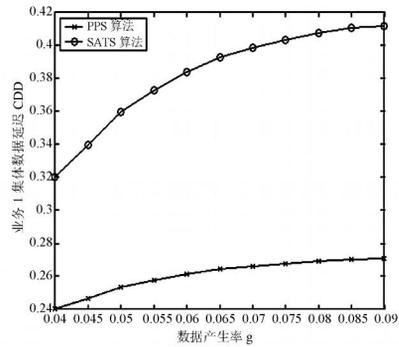


图 6 两种算法业务 1 数据 CDD 对比图

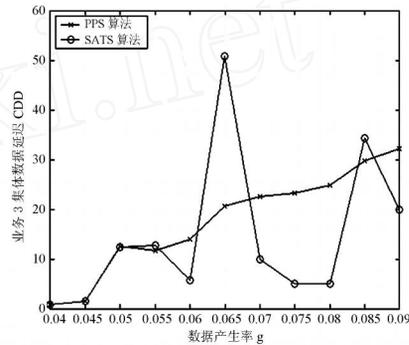


图 7 两种算法业务 3 数据 CDD 对比图

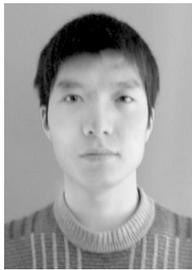
### 5 结论

WSN 是当前的研究热点, 其 QoS 保障是个极富挑战性的研究课题, 至今该领域实用性研究成果并不多。为解决该技术难题, 较为直观的解决思路是借鉴现有 IP 网络上的 QoS 保障机制, 但是 WSN 有其自身特点, 难以直接应用, 比如传统的基于端到端传输的 QoS 机制并不符合 WSN 的特点。本文针对 WSN 以数据应用为中心的、非端到端的特点, 给出了其 QoS 指标的具体公式, 并借鉴传统 IP 网络速率控制保障 QoS 的思想, 提出了一种适合传感器网络的速率调整算法。此外, 现有传感器网络研究大部分基于平等地竞争网络资源的“不区分服务”的模型, 但是随着 WSN 的发展和应用, 不同业务对 QoS 参数提出了不同的需求, 迫切要实现“区分服务”的机制。基于本文的模型和 QoS 指标, 进一步提出了 PPS 算法以实现“区分服务”。仿真和分析表明本文的算法对 WSN 的 QoS 保障是有效的, 速率调整算法从根本上改善了网络的 QoS 指标, PPS 调度算法很好地实现了 3 类应用的区分服务, 并且与 SATS 算法相比具有明显优势。今后的工作是在传感器网络硬件平台上对这些算法做深入的分析 and 实现。

### 参考文献:

[1] Jones C E, Sivalingam K M, Agrawal P, Chen J C. A Survey

- of Energy Efficient Network Protocols for Wireless Network [J]. *Wireless Networks*, 2001, 7 (4): 343-358.
- [2] Ian F. Akyildiz, Tommaso Melodia, Kaushik R. Chowdhury. A Survey on Wireless Multimedia Sensor Networks[J]. *Computer Networks* 2007, (51): 921-960.
- [3] James F. Kurose, Keith W. Ross. *Computer Networking A Top-down Approach Featuring the Internet*[M]. 北京:人民邮电出版社,2004: 184-188.
- [4] Sankar Y, Akan B, Akyildiz F. ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks[C]// *Proceedings of ACM MobiHoc '03*. Los Angeles, 2003: 177-188.
- [5] Dazhi Chen, Pramod K. Varshney. QoS Support in Wireless Sensor Networks: A Survey[C]// *Proceedings of the 2004 International Conference on Wireless Networks, LasVegas, Nevada, USA, June 21-24, 2004: 227-233*.
- [6] Younis M, Akkaya K, Eltoweissy M, et al. On handling QoS Traffic in Wireless Sensor Networks[C]// *Proceedings of the 37th Hawaii International Conference on System Sciences, Piscataway, USA, IEEE, 2004: 4653-4662*.
- [7] Akaya K, Younis M. An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks[C]// *Proceedings of 23rd International Conference on Distributed Computing Systems Workshops, May 2003: 710-715*.
- [8] 孙岩. 多媒体传感器网络中服务质量保障问题的研究[D]. 北京:北京邮电大学,2007: 49-51.
- [9] 史浩山,侯蓉晖,杨少军. 无线传感器网络 QoS 机制研究[J]. *信息与控制*,2006,35(2): 246-251.
- [10] Shensheng Tang, Wei Li. QoS Supporting and Optimal Energy Allocation for A Cluster Based Wireless Sensor Network [J]. *Computer Communications*, 2006, (29): 2569-2577.
- [11] Marco Zimmerling, Walteneus Dargie, Johnathan M. Reason. Localized Power-aware Routing in Linear Wireless Sensor Networks [C]// *Proceedings of the 2nd ACM International Conference May 2008: 24-33*.
- [12] 盛友招. 排队论及其在现代通信中的应用[M]. 北京:人民邮电出版社,2007: 193-197.
- [13] Kleinrock L. *Queueing Systems Volume 1: Theory* [M]. New York: Wiley, 1975: 103-105.
- [14] Baskiyar S, Meghanathan N. A Survey of Contemporary Real-time Operating Systems[J]. *Informatica*, 2005, (29): 233-240.



朱 畅(1985-),男,浙江金华人,硕士研究生,浙江大学信电系电子电路与信息系研究所电路与系统专业。主要从事无线传感器网络与信息处理研究, zhuchang007@gmail.com



金心宇(1958-),男,浙江大学信电系电子电路与信息系研究所教授,博士生导师。主要从事网络通信与系统级芯片、音视频通信、网络安全与智能测试的研究工作, jinxy@zju.edu.cn