

## RSPEED: 无线传感器网络中基于不确定延迟的可靠实时路由

陈权, 高宏

(哈尔滨工业大学 计算机科学与技术学院, 黑龙江 哈尔滨 150001)

**摘要:** 基于不可靠的链路, 综合考虑了链路的质量与链路的延时, 提出了一种新的转发可靠性概念, 并在此基础上, 设计了一种新的实时路由协议(RSPEED)。利用转发可靠性表示节点在给定的延迟阈值下成功将数据分组转发到邻居节点的概率, 并以此保证路由的实时性和可靠性, 提供一定的 QoS 服务。另外, 当环境中链路质量比较差时, 还能在满足实时性的基础上通过权衡传输延时与传输可靠性, 从而获得更高的传输成功率。实验结果表明, 提出的方法能够显著提高分组的传输成功率和减少超过给定延迟阈值分组的数量, 特别是当信道非常差时, 提出的方法在满足实时性上比 SPEED 高出 20%左右。

**关键词:** 无线传感器网络; 实时路由; 转发可靠性; 不可靠链路

中图分类号: TP393.01

文献标识码: A

文章编号: 1000-436X(2013)08-0110-10

## RSPEED: the reliability and real-time routing based on uncertain delay in wireless sensor network

CHEN Quan, GAO Hong

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

**Abstract:** The forwarding reliability was proposed by considering the unreliability and the delay of a link, and with which, a novel real-time routing protocol(RSPEED) was designed. The forwarding reliability denoted the probability to deliver the data successfully to the neighbor node under a given deadline, and it was used to achieve probabilistic QoS guarantee in timeliness and reliability. The protocol can also make trade-off between the transmission delay and the reliability of transmission to improve the probability of successful transmission when the environment and link quality is very poor. The experimental results show that the proposed approach can significantly improve the reliability of transmission and reduces the number of the missed deadline, and even when the channel is very poor, it can achieve 20% higher performance than SPEED.

**Key words:** wireless sensor networks; real-time routing; forwarding reliability; unreliable link

### 1 引言

随着传感器技术和无线通信技术的飞速发展, 无线传感器网络已经被广泛应用在医疗监护、环境监测、军事侦察等领域<sup>[1,2]</sup>。一个典型的传感器网络由部署在一片区域中的大量传感器节点组成。通常, 这些节点很容易失效, 导致网络的拓扑结构变化非常激烈。

在数量众多的传感器网络应用中, 尽管传感器

网络自身存在很大的不确定性, 但是支持实时的通信依然是必要的<sup>[3]</sup>。例如在军事、环境、基础设施的监测和控制系统中, 如战场上的目标检测与追踪, 医疗监测以及放射性元素检测和预防等系统中, 都需要实时通信的支持, 以便系统及时地做出反应, 否则可能会带来严重的后果。

但是, 由于传感器网络很大的不确定性, 通信的延迟是难以预测的<sup>[3]</sup>, 这给实时路由提出了严峻的挑战。特别是在周围环境的干扰和噪声很

收稿日期: 2012-07-11; 修回日期: 2013-03-01

基金项目: 国家自然科学基金资助项目(61190115, 61033015)

**Foundation Item:** The National Natural Science Foundation of China (61190115, 61033015)

大的情况下,无线信道的质量可能会激烈地波动。在这种链路不可靠的网络中,设计实时路由协议,不仅要考虑网络中的延迟,而且还应考虑链路的质量。

目前,传感器网络中的实时路由协议在满足实时性的过程中,基本上都采用了 SPEED<sup>[4]</sup>提出的转发速度的概念,即根据一跳的延时和前进的距离估计邻居节点的转发速度,从而选择转发速度大于给定阈值的节点作为下一跳的转发节点。但是,在一个链路不可靠的网络环境中,转发速度仅仅考虑了节点之间链路的延时,而没有考虑链路的质量和分组传输成功的概率,很可能导致路由协议做路径选择时,选择一条更不可靠的路径,导致分组丢失的情况更严重,如图 1 所示。

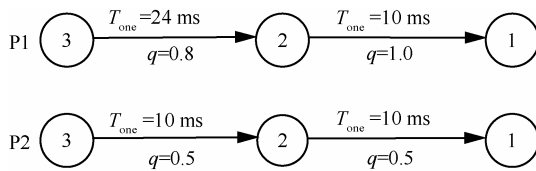


图 1 2 条期望延时相等的路径

如图 1 所示,有 2 条路径 P1 和 P2,  $q$  表示链路的质量,  $1-q$  即分组丢失的概率。 $T_{\text{one}}$  表示尝试一次传输所消耗的时间,包括数据传输的时间和 MAC 层获取无线信道的时间。通常用  $T_{\text{one}} \cdot ETX$  作为对一跳延迟的估计,其中,  $ETX$ <sup>[5]</sup>(期望传输次数)表示成功发送一个分组期望传输的次数。一般,期望传输次数  $ETX = 1/q$ ,期望传输延迟  $delay_e = T_{\text{one}}/q$ 。例如在路径 P1 中,节点 3 与节点 2 之间的链路  $L_{32}$  的期望传输延迟  $delay_{32} = 24/0.8 = 30$  ms。

可以看出,路径 P1 的期望传输延迟  $delay_{P1} = 24/0.8 + 10/1.0 = 40$  ms; P2 这条路径的期望传输延迟  $delay_{P2} = 10/0.5 + 10/0.5 = 40$  ms,路径 P2 和路径 P1 的期望传输延时是相等的。如果采用转发速度,只考虑路径的延时,将不能区分路径 P2 和路径 P1 这 2 条路径的好坏,可能就会导致选择一条更不可靠的路径。但是,设最大重传次数  $r=2$ ,则 P1 这条路径发送成功的概率为  $pdr_{P1} = [0.8 + (1-0.8) \times 0.8] \times 1.0 = 0.96$ ; 而路径 P2 发送成功的概率  $pdr_{P2} = [0.5 + (1-0.5) \times 0.5] \times [0.5 + (1-0.5) \times 0.5] = 0.5625$ 。很显然,路径 P1 的分组发送成功的概率要优于 P2。

另外,对于路径 P1,可能产生的最小延迟(每跳只传输一次)  $delay_{\min} = 24 + 10 = 34$  ms,对应的概

率为  $0.8 \times 1.0 = 0.8$ ,即大概会有 80% 的分组能够在 34 ms 内到达目的节点。对于路径 P2,可能产生的最小延迟  $delay_{\min} = 10 + 10 = 20$  ms,对应的概率为  $0.5 \times 0.5 = 0.25$ ,即大概会有 25% 的分组能够在 20 ms 内到达目的节点。这种情况下,如果给定的阈值为 20 ms,则只能选择 P2。如果采用期望延迟的估计方法,很可能就不会发现这条路径,所有阈值不大于 20 ms 的分组可能只会被丢弃。因此,仅仅采用转发速度和期望延迟的方法是不够的。

因此,本文结合链路的质量与链路的延时,提出了一种新的转发可靠性概念,并且设计了一种新的实时路由协议。本文的方法:首先,能够保证实时性和可靠性,提供一定的 QoS 服务;其次,当环境中链路质量比较差时,还能在满足实时性的基础上,通过权衡传输的延时与传输的可靠性,从而获得更高的传输成功率;另外,本文的方法还能发现那些潜在的能满足实时性的路径。

## 2 相关工作与问题定义

最近,Kumar<sup>[6]</sup>从控制的角度对无线传感器网络中的实时性做了大量的理论分析,文中主要基于多个客户端和一个接入点的模型,分析一个数据流能否满足给定的吞吐量和延迟阈值的限制。文献[7]则理论分析了无线传感器网络中事件检测的延迟,即  $n$  个检测到事件的数据分组都到达 sink 节点的延时。

文献[8]则对目前传感器网络中的实时路由协议做了一个综述。Sandhya 将目前的实时路由协议分为两类:硬实时路由协议和软实时路由协议。在硬实时系统中,端到端的延迟是不允许超出的,其主要通过物理层的调度来满足实时性,如 I-EDF<sup>[9]</sup>、dual-Mode<sup>[10]</sup>等,但是这种方法对物理条件的要求非常苛刻。而在软实时系统中,能够允许部分随机性的结果。目前,大部分的实时路由协议都是基于软实时的,其中主要有 RAP<sup>[11]</sup>、SPEED<sup>[4]</sup>、MMSPEED<sup>[12]</sup>、FT-SPEED<sup>[13]</sup>、RPAR<sup>[14]</sup>等。

根据每一跳的延时,RAP<sup>[11]</sup>提出了基于速度的路由协议 RAP,通过一个基于速度的调度模式,提高了实时数据的优先级。SPEED<sup>[4]</sup>则提出了转发速度的概念,通过保证每一跳的转发速度来保证端到端的通信延迟,但是 SPEED 没有考虑链路的质量,不能保证可靠性。MMSPEED<sup>[12]</sup>则对 SPEED 做了扩展,能够支持多种速度,并且利用多条路径冗余

的方式提供可靠性服务,但这种方式不仅要占用网络中的大量资源,而且要消耗网络中巨大的能量。同时,由于多条路径的传输引入了更多的冲突,从而导致分组的延时增加。FT-SPEED<sup>[13]</sup>则主要考虑了 SPEED 协议中容易遇到的路由空洞问题。

一些协议则是考虑通过增加电量来满足实时性<sup>[14~17]</sup>。RPAR<sup>[14]</sup>在 RAP 的基础上考虑了电量的控制。RTPA<sup>[15]</sup>将功率划分为不同的层次,综合考虑能量消耗和 QoS 服务。RTL D<sup>[16]</sup>综合考虑了速度,PRR(分组接收速率)以及剩余的能量来选择转发节点。PATH<sup>[17]</sup>则是基于两跳的邻居信息选择转发节点。

还有一些协议则是从分层的网络结构中考虑实时性<sup>[18,19]</sup>。文献[18]在每个分簇中通过调整实时数据和非实时数据所占的带宽来最大化吞吐量。文献[19]则提出了一种基于分层结构的实时数据收集方法。

最近,文献[20]提出了一种可靠实时路由 3R 协议。根据每跳传输成功的概率,计算每跳链路最大重传次数,再应用 Dijkstra 最短路径算法判断整条路径上的重传次数之和是否满足给定延迟阈值的要求,并且采用了多条路径冗余的方式满足可靠性。但是文中假设每一跳的传输时间都是相等的,并且采用的是一个集中式的方法。

但是,到目前为止,还没有一种分布式的路由协议考虑由于链路的不可靠性,导致链路的重传次数对路由的实时性和可靠性造成的影响。因为链路的重传次数不仅会影响单跳的延迟造成端到端之间延迟的不确定性,而且还能影响分组传输成功的概率。

当周围环境的干扰和噪声很严重时,链路的质量会出现比较大的变化,此时仅仅考虑实时性是不够的,因为在保证实时的过程中,可能会选择不可靠的路径,导致很多数据分组不能成功到达目的节点。例如 Liu 等在文献[21]中通过基于 GreenOrbs 上的实验,发现当网络规模为 330 个节点时,不到 60%的数据分组能够成功到达 sink 节点,而且大约 61.08%的分组丢失都是因为超过重传次数而造成的。在这种情况下设计实时路由,不仅要考虑链路的延迟,还应考虑链路的质量以及数据分组到达目的节点的传输成功率。本文研究的问题可形式化描述如下。

**问题定义** 给定一对源节点和目的节点 ( $S$ ,

$D$ ),以及源节点到目的节点的延迟  $\delta$  和一个可靠性概率  $\Delta P$ ,要求找到一条保证数据分组至少以  $\Delta P$  的概率在  $\delta$  之前到达 sink 节点(或目的节点)的路径。

### 3 路由算法设计

当周围环境的干扰和噪声很严重时,为了满足实时性和可靠性,本文考虑重传次数对延迟和可靠性造成的影响,结合链路的质量与链路的延时,提出了一种新的转发可靠性概念,然后根据提出的转发可靠性选择下一跳的转发节点。其思想如下:首先,每跳节点根据用户给定的延迟  $\delta$  计算出每跳的延迟阈值,通过选择传输时间小于该延迟阈值的邻居节点来转发以满足实时性;然后,根据链路的质量计算出在该延迟阈值下,将分组成功转发到邻居节点的概率(转发可靠性),再根据该概率选择转发节点来满足可靠性。

为了更好地适应无线传感器网络的可扩展性,本文提出的方法只基于一跳的邻居信息。本文假设每个节点都是静态的,或者节点运动的速度很慢,并且都通过 GPS 或其他位置信息服务<sup>[24]</sup>知道自己的位置信息。本文提出的协议是基于 CSMA/CA 机制的,并且 MAC 协议使用 RTS/CTS 来避免冲突。

本文提出的路由协议 RSPEED 主要分为 4 个部分:延迟估计、链路质量估计、转发策略以及邻居管理。延迟估计和链路质量估计分别是用来估计邻居节点的单跳延迟和链路质量。转发策略部分则是计算邻居节点的转发可靠性,即数据分组在满足实时性的要求下到达邻居节点的概率,再根据邻居节点的转发可靠性选择转发节点,这是本文最核心的部分。邻居管理则是管理一跳内的邻居节点信息。

#### 3.1 延迟估计

在实时路由中,一般将单跳延迟作为对邻居节点负载的一个估计。单跳延迟,即节点  $i$  将数据分组发送到邻居节点  $j$  所需要的时间,主要与 MAC 层竞争冲突的时间、分组传输的时间以及需要重传的次数  $r$  有关。对于节点  $i$  到其邻居节点  $j$  的单跳延迟  $delay_{ij}$ ,可以根据下式计算

$$delay_{ij} = (t_{trans} + t_{cont})r \quad (1)$$

其中,  $t_{trans}$  表示分组的传输时间,包括将信号编码和无线传输的时间,由分组的大小和节点的带宽决定;  $t_{cont}$  表示冲突延迟,即节点获取信道的时间,主要与周围环境和竞争信道的激烈程度有关。

本文将  $t_{\text{trans}}$  与  $t_{\text{cont}}$  之和抽象成  $T_{\text{one}}$ , 表示尝试一次传输所需要的时间。本文的延迟估计只估计尝试一次传输所需要的时间  $T_{\text{one}}$ , 继而根据  $T_{\text{one}}$  和重传次数  $r$ , 就可以计算邻居节点的单跳延迟。由于受到环境带宽的限制, 用数据分组携带的方法进行延迟估计。当节点  $i$  在 MAC 层收到一个数据分组需要发送, 并且准备获取信道时, 记录发送的时间  $t_x$ , 直到收到邻居节点  $j$  回复的 ACK 分组, 记录收到的时间  $t_r$ , 以及完成此次转发该分组所经历的重传的次数  $r$ 。根据  $t_x$ 、 $t_r$  和重传次数  $r$ , 则可以估计尝试一次传输所需要的时间  $T_{\text{one}}$ 。

$$T_{\text{one}i}^j = (t_r - t_x) / r \quad (2)$$

如果周围环境很差, 节点  $i$  在最大重传次数  $r_{\text{max}}$  下, 仍没能将数据成功地发送给邻居节点  $j$ , 这时  $i$  节点直接将分组丢弃。本文根据新发送分组的延迟和历史的延迟, 采用指数加权平均<sup>[25]</sup>的方法估计当前的延迟。另外, 为了反映邻居节点的拥塞情况, 也可以采用 ACK 分组携带的方法估计邻居节点的拥塞延迟。当邻居节点接收到一个数据分组后, 将 ACK 分组在节点中的拥塞延迟记录到 ACK 分组中。

### 3.2 链路质量估计

当周围环境的干扰和噪声很严重的情况时, 链路的质量会出现很大的变化。为了能够快速检测到链路质量的变化, 采用滑动窗口的方法来估计链路的质量。首先, 利用一个滑动的窗口  $W$  记录最近几次分组是否成功传输的情况。窗口  $W = \{s_0, s_1, \dots, s_{m-1}\}$ , 其大小设为  $m$ 。其中,  $s_0$  则表示当前分组  $p_0$  是否传输成功, 如果  $s_0=1$ , 表示传输成功, 否则表示传输失败。  $s_i$  则表示在  $p_0$  之前的第  $i$  个分组的传输成功信息。为了能够快速反映链路的变化情况, 作者为每个分组设置了一个权值, 并将最近分组的权值设置的相对高, 越久的信息, 则对应的权值越低。假设  $s_i$  表示第  $i$  个分组的发送成功情况,  $weight_i$  表示  $s_i$  的权值, 则链路质量  $q$  可通过下式计算

$$q(0, m-1) = \frac{\sum s_i \cdot weight_i}{\sum weight_i} \quad (3)$$

例如, 在本实验中, 设置窗口大小  $m=24$ 。实验证明采用滑动窗口的方法, 能够快速反映链路质量的变化情况。如图 2 所示, 本文方法在 45~60 s 和 75~80 s 时间内快速地检测到了由于引入的干扰而导致的链路质量的波动, 而如果采用普通的方法, 即采用  $\text{packet-loss} / \text{packet-total}$  的方法, 则很难快速

地反应此种变化。另外, 为了估计的准确性, 可以适当地调整窗口的大小和窗口的权值。

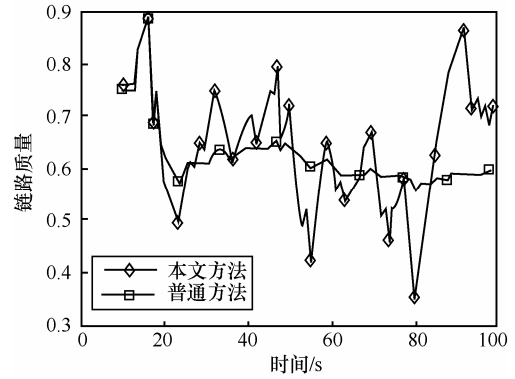


图2 链路质量估计

### 3.3 转发策略

目前, 大部分的实时路由协议都采用了 SPEED<sup>[4]</sup>中提出的转发速度的概念, 通过保证转发速度大于某个阈值来满足实时性要求。转发速度可以表示邻居节点转发数据的快慢程度。但是, 在一个不可靠的网络环境中, 不仅要考虑节点之间链路的延时, 还应该考虑链路的质量, 在保证实时性的同时, 满足传输的可靠性。因此, 在转发速度的基础上, 结合链路的质量与链路的延时, 本文提出了转发可靠性, 利用转发可靠性, 表示在满足实时性的条件下成功将数据传输到邻居节点的概率。通过保证转发可靠性, 就可以在保证实时性的同时, 满足可靠性。

首先, 将邻居节点分为两类: 比自己距离目的节点更近的节点和比自己距离目的节点更远的节点。前者称为候选转发点集合 FCS<sup>[4]</sup>。在路由选择的过程中, 为了减少传输的跳数和传输的时间, 作者只考虑将分组通过 FCS 中的节点进行转发。

其次, 为了满足实时性要求, 本文定义了转发延时。转发延时主要是用来表示下一跳节点最多允许传输的时间, 其定义如下。

**定义 1** 用  $r\_delay_i^j$  表示当前节点  $i$  到下一跳节点  $j$  的转发延时, 其计算公式为

$$r\_delay_i^j = \frac{\text{dist}(i, D) - \text{dist}(j, D)}{\text{dist}(i, D)} \cdot \text{deadline}_i \quad (4)$$

其中,  $\text{dist}(i, D)$  表示节点  $i$  到目的节点  $D$  的距离;  $\text{dist}(i, D) - \text{dist}(j, D)$  则表示节点  $i$  到下一跳节点  $j$  前进的距离; 式(4)中的  $\text{deadline}_i$  表示数据分组到达节点  $i$  后剩余的时间。

由于节点之间通信延迟的不确定性, 引入了

*deadline* 的动态调整, 节点每收到一个数据分组后, 都需要重新计算剩余的 *deadline*。定理 1 证明, 如果在每跳选择转发节点的过程中, 都能满足转发延时, 那么本文的方法和转发速度同样能满足实时性。

**定理 1** 如果  $t_{\text{deliver}_i}^j \leq r\_delay_i^j$ , 则有  $speed_i^j(D) \geq v_{\text{req}}$ 。其中,  $t_{\text{deliver}_i}^j$  表示节点  $i$  将数据转发到节点  $j$  所用的时间。  $speed_i^j(D)$  和  $v_{\text{req}}$  是 SPEED 协议中采用的保证端到端延时的方法, 其中,  $speed_i^j(D) = \frac{dist(i,D) - dist(j,D)}{t_{\text{deliver}_i}^j}$ , 表示邻居节点  $j$  的转发速度, 而  $v_{\text{req}} = \frac{dist(i,D)}{deadline_i}$ , 表示要满足实时性, 速度必须达到的一个阈值。

**证明**

$$\begin{aligned} speed_i^j(D) &= \frac{dist(i,D) - dist(j,D)}{t_{\text{deliver}_i}^j} \\ &\geq \frac{dist(i,D) - dist(j,D)}{r\_delay_i^j} \\ &\geq \frac{dist(i,D) - dist(j,D)}{dist(i,D)} \cdot \frac{dist(i,D)}{deadline_i} \\ &\geq \frac{dist(i,D)}{deadline_i} \geq v_{\text{req}} \end{aligned}$$

根据转发延时, 可以通过在每跳选择转发节点的过程中, 选择传输时间小于等于转发延时的节点作为转发节点来满足实时性。但是传输的可靠性, 即传输成功的概率依然还没有满足。接下来, 将给出转发可靠性概念, 根据转发可靠性来满足传输的可靠性。

**定义 2** 节点  $i$  到节点  $j$  的转发可靠性  $FP_{ij}$  是指链路  $L_{ij}$  将数据分组在  $r\_delay_i^j$  时间内, 转发到邻居节点  $j$  的概率。其计算公式为

$$FP_{ij} = 1 - (1 - q_{ij}) r_{ij} \quad (5)$$

其中,  $q_{ij}$  表示链路的质量;  $r_{ij}$  表示在转发延时的限制下, 最多允许重传的次数, 可以根据转发延时和  $T_{\text{one}_i}$  来计算, 即  $r_{ij} = \lfloor r\_delay_i^j / T_{\text{one}_i} \rfloor$ 。

根据转发可靠性, 就可以得到每一跳在满足实时性的条件下成功传输的概率。因此, 转发可靠性不仅能反映邻居节点满足实时性的程度, 而且还能反映邻居节点传输成功的概率。如果根据转发可靠性来选择转发节点, 就可以在满足实时性的同时, 保证一定的传输可靠性。下面将给出如何在保证实时性的条件下, 满足端到端传输成功概率的方法。

在保证实时性条件下, 源节点到目的节点传输成功的概率  $pdr(S,D)$  可以根据下式计算

$$pdr(S,D) = \prod_{i=1}^k FP_i \quad (6)$$

其中,  $k$  表示传输的跳数;  $FP_i$  表示第  $i$  跳的转发可靠性。为了满足可靠性要求, 需满足  $pdr(S,D) \geq \Delta P$ 。

由于每个节点只是基于本地的信息来选择转发节点, 无法获得全局的信息来计算  $pdr(S,D)$ , 因此本文采用通过限制传输的跳数和每跳的转发可靠性来满足可靠性要求。为此定义了一个转发可靠性阈值  $f\_AP$ ,  $f\_AP$  的计算公式为

$$f\_AP = \log_{\lceil \frac{dist(S,D)}{dist\_advance} \rceil} \Delta P \quad (7)$$

其中,  $dist\_advance$  表示每跳最少前进的距离, 可以通过设置  $dist\_advance$  来调整传输的跳数, 如果  $dist\_advance$  太小, 则转发可靠性阈值  $f\_AP$  太高, 而如果  $dist\_advance$  太大, 可能找不到足够的候选转发节点。如何设置  $dist\_advance$  的大小将在以后讨论。

**定理 2** 如果每一跳的转发可靠性  $FP_i \geq f\_AP$ ,  $1 \leq i \leq k$ , 则有  $pdr(S,D) \geq \Delta P$ 。其中,  $k$  表示传输的跳数,  $FP_i$  表示第  $i$  跳的转发可靠性。

**证明**

$$\begin{aligned} pdr(S,D) &= \prod_{i=1}^k FP_i \geq \prod_{i=1}^k f\_AP \\ &\geq \prod_{i=1}^k (\log_{\lceil \frac{dist(S,D)}{dist\_advance} \rceil} \Delta P) \\ &\geq (\log_{\lceil \frac{dist(S,D)}{dist\_advance} \rceil} \Delta P)^{\lceil \frac{dist(S,D)}{dist\_advance} \rceil} \\ &\geq \Delta P \end{aligned}$$

定理 2 说明, 如果在选择每跳转发节点的过程中, 保证最少前进的距离和下一跳节点的转发可靠性, 则能够在满足实时性的条件下保证源节点到目的节点传输成功的概率。因此, 可以通过转发可靠性, 来满足数据传输的实时性和可靠性, 提供一定的 QoS 服务。下面将给出如何根据转发可靠性来选择转发节点的方法。

**定义 3** 对于节点  $a$  和节点  $b$ ,  $b \in FCS(a)$ , 路由选择  $(a,b)$  是可行的, 当且仅当  $FP_{ab} \geq f\_AP$ 。

根据  $f\_AP$  和转发可靠性, 可以得到节点  $a$  的可行转发节点集合 FCSE

$$FCSE(a) = \{b | \text{路由选择}(a,b) \text{ 是可行的}\}$$

如果节点  $a$  的 FCSE 集合不为空, 则在节点  $a$

的 FCSE 集合中按照一定的概率 (转发概率) 选择下一跳节点。而转发概率可通过转发可靠性来计算, 如果邻居节点的转发可靠性越高, 则被选中的概率越大。

在路由协议中, 选择转发节点是最核心的部分。FCSE 算法首先根据分组中的延迟  $\delta$  和可靠性要求  $\Delta P$ , 计算当前需要满足的转发可靠性阈值, 再根据路由的目的节点计算节点的候选转发节点集合 FCS。接着, 节点再对 FCS 中的每个节点, 根据估计出的延迟和链路的质量, 计算出所允许重传的次数和转发可靠性 FP。根据转发可靠性 FP, 就可以获得节点的可行转发节点集合。

从图 3 可以看出, FCSE 算法的时间复杂度与候选转发节点集合的大小有关, 而且 4)~10) 皆可以在常数时间内完成。节点的候选转发节点集合大小与邻居节点的个数有关, 而节点的邻居节点的个数一般都是常量。因此, FCSE 算法的时间复杂度为  $o(1)$ 。

```

INPUT: 一个数据分组 P, 数据分组中包含了一个延迟  $\delta$  和  $\Delta P$ 
OUTPUT: 节点 a 的可行转发节点集合 FCSE(a)

begin procedure
1) 将节点 a 的可行转发节点集合 FCSE, 即 FCSE(a)置空;
2) 根据式 (7) 计算转发可靠性阈值  $f_{\Delta P}$ ;
3) 计算节点 a 的候选转发节点集合 FCS(a);
4) for  $\forall b \in FCS(a)$  do
5) 计算 a 到 b 之间的转发距离  $df_{ab} = dist(a,D) - dist(b,D)$ ;
6) 根据式 (4) 计算 a 到 b 转发延时  $r\_delay_a^b$ ;
7) 计算最大允许重传的次数  $r_{ab}$ ;
8) 根据式 (7) 计算 a 到 b 转发成功的概率  $FP_{ab}$ ;
9) If  $FP_{ab} \geq f_{\Delta P}$  then
10) FCSE(a)=FCSE(a)  $\cup$  {b};
11) end for
end procedure
    
```

图 3 FCSE 算法

当节点收到一个从应用层或者邻居节点传来的数据分组后, 首先节点判断是否是此次路由的目的节点。如果是, 则将数据分组返回给应用层; 如果不是, 则利用 FCSE 算法计算节点的可行候选转发节点集合, 继而根据上述的策略从转发节点集合中选择合适的节点将数据分组转发到下一跳节点, 并且根据收到的下一跳节点返回的 ACK 应答分组, 更新节点的延迟和链路质量的估计。整个路由过程

一直持续到数据分组到达目的节点。

现在, 再看图 1 中给出的例子, 如果给定的延迟  $\delta$  为 20 ms, 本文方法 RSPEED 可以发现当链路  $L_{32}$  和  $L_{21}$  允许传输的次数都为 1 时, P2 这条路径可以满足实时性, 并且此时 P2 这条路径发送成功的概率为  $pdr_{p_2} = 0.5 \times 0.5 = 0.25$ 。

另外, 本文方法还能在满足实时性的基础上通过权衡传输的延时与传输成功的概率, 从而获得更高的传输成功率。如图 4 所示, 数据分组从源节点 S 开始, 经过多跳路由之后到达节点 a。节点 a 有 2 个路由选择, 节点 b 和节点 c, 其中, 链路  $L_{ab}$  和  $L_{ac}$  尝试一次传输所需要的时间  $T_{one}$  分别为 16 ms 和 10 ms, 链路  $L_{ab}$  和  $L_{ac}$  的链路质量分别为 0.8 和 0.5。这时如果节点 a 计算出下一跳允许的延时为 50 ms, MAC 层最大允许重传的次数  $r_{max} = 3$ , 可以看出节点 b 和节点 c 都可以尝试  $r_{max} = 3$  次重传, 因为  $16 \times 3 = 48$  ms  $<$  50 ms,  $10 \times 3 = 30$  ms  $<$  50 ms。接着可以计算出节点 b 的转发可靠性  $FP_{ab} = 1 - (1 - 0.8)^3 = 0.992$ , 节点 c 的转发可靠性  $FP_{ac} = 1 - (1 - 0.5)^3 = 0.875$ 。很显然,  $FP_{ab} > FP_{ac}$ , 选择节点 b 作为转发节点, 可得

$$delay_e = \sum_{i=1}^{r_{max}} T_{one} i q (1-q)^{i-1} \quad (8)$$

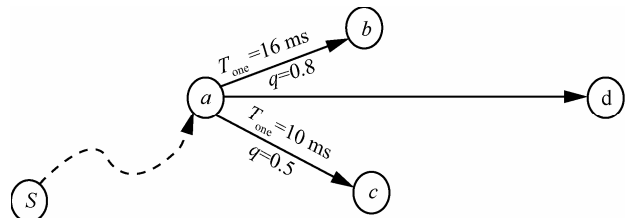


图 4 选择转发节点的例子

可以计算节点 b 的期望传输延时为 19.456 ms, 节点 c 的期望传输延时为 15.125 ms, 节点 b 的期望传输延时略高于 c 节点的期望传输延时, 但这并不影响实时性。

### 3.4 邻居管理

为了获得周围邻居节点的信息, 每个节点周期性地广播一个 beacon 分组到它所有的邻居。这种周期性的广播只是用来交换邻居节点的位置信息。每个节点都建立一个邻居表保存邻居节点的信息, 主要包括 ( $neighbor\_id$ ,  $position$ ,  $tone$ ,  $link\_quality\_window$ ,  $expiretime$ )。expiretime 用于更新一条记录, 当一个邻居节点的信息超过一个时间没有更新后, 将它从邻居表中删除。neighbor\_id 和 position

分别表示邻居节点的 *id* 和位置信息。*tone*<sub>由延迟估计的方法得到。*link\_quality\_window*是用来估计链路质量的窗口。</sub>

由于本文提出的方法在链路质量估计和延迟估计中都采用数据分组携带的方法，不会带来额外的通信开销，即主要的通信开销都在邻居管理的周期性广播中和数据分组的路由中。在数据分组的路由中，与 SPEED 一样，本文采用单条路径和通过距离目标节点更近的节点以减少传输跳数的方式进行路由，不会增加额外的通信开销。另外，邻居节点管理对许多无线传感器网络应用和路由（如 SPEED）都是必须的，而且还可以将这种周期性的广播设置得非常低，特别是当传感器网络是相对静态和移动较少的情况时。因此本文提出的方法不会增加额外的通信开销。

### 4 实验与分析

在一个基于离散事件的网络模拟器 NS2<sup>[22,23]</sup>上，实现了本文的协议 RSPEED，并且与一个 baseline 协议(SPEED<sup>[4]</sup>协议)进行了对比。SPEED 协议是目前被引用最多的实时路由协议，其最大的贡献是提出了转发速度概念，利用转发速度表示邻居节点转发数据的快慢。另外，MMSPEED<sup>[12]</sup>协议虽然也提供可靠性服务，但是主要采用多条路径冗余的方式来保证可靠性，而本文的协议则从重传的角度，考虑链路的质量提高了传输的可靠性，而且也可以采用多条路径的方式来保证可靠性。NS2 在 DARPA 支持下，经过几代的发展，现在能够提供非常丰富的网络模拟功能，包括无线传播模型、MAC 层和链路层模拟等。表 1 描述了实验环境的设置，其中，通信参数的设置主要是根据 Berkeley mote 的标准来设定。

表 1 参数设置

参数	MAC 层
	802.11 (simplified DCF)
传播模型	two-ray ground
场地	200 m×200 m
节点数	100
节点布置模型	uniform
传输半径	40 m

与 SPEED<sup>[4]</sup>中采用的设置相同，在每次模拟中，将 100 个节点均匀地布置到 200 m×200 m 的场

景中。另外，为了增加无线传感器网络中链路的不可靠性和糟糕的网络环境，在每个节点中引用了一个随机丢失分组模型 Error Model<sup>[22]</sup>。Error Model 是 NS2 中提供的一个丢失分组模型，用来模拟链路在发送数据时以一定的比率出错。在实验中，设置出错的比率(error rate)为 0.1。在传感器网络中，主要有 2 种流量模式：基站模式和点到点模式，在实验中，主要考虑的是点到点模式(peer-to-peer)。每次随机地从场景左边缘挑选节点作为源节点，再从右边缘选择节点作为目的节点。每次源节点生成一个 CBR 数据流，数据分组的大小设为 512 byte，生成的速率为 2.0 packet/s。在实验中，源节点到目的节点的平均跳数为 8~9 跳。

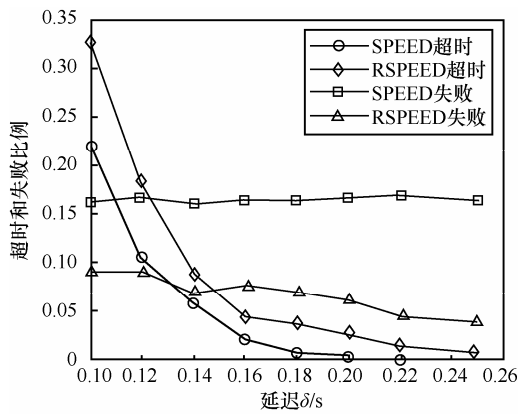
作者主要从数据分组未满足实时性要求比例 (deadline miss ratio)、超时比例(delay ratio)、传输失败比例(fail ratio)以及每个数据分组端到端之间的延时(E2E delay) 4 个方面来评价本文算法的实时性和可靠性。数据分组传输失败比例表示的是那些由于冲突或超过重传次数导致的未能到达目的节点所占的比例，主要是用来衡量路径的顽健性。数据分组超时比例则是数据分组成功到达目的节点，但是超出了给定的延迟  $\delta$  的比例。数据分组未满足实时性要求比例则包括数据分组超时比例和数据分组传输失败比例，在实时路由协议的设计中，是一个非常重要的衡量标准。

#### 4.1 实时性分析

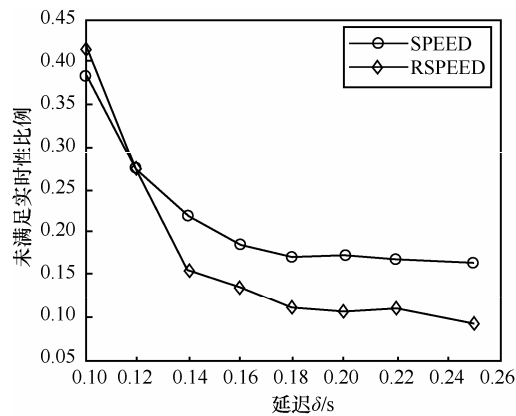
在实时路由协议的设计中，实时性的满足是一个非常重要的衡量标准。首先比较了 RSPEED 与 SPEED 在延迟  $\delta$  从 0.12~0.25 s 变化时，2 种方法在满足实时性方面的性能。图 5(a)表示的是 2 种协议在延迟  $\delta$  从 0.12~0.25 s 变化时的数据分组超时比例(实线所示)和数据分组传输失败比例(虚线所示)。从图中可以看出延迟  $\delta$  越小，数据分组超时比例越高，当延迟  $\delta$  不大于 0.16 s 后，数据分组超时比例迅速增加，此时说明，在满足实时性的前提下，每跳允许重传的次数太少，而链路的质量又无法保证。另外，随着延迟  $\delta$  越大，RSPEED 的数据分组传输失败比例也越低，数据传输成功的概率越高，但是 SPEED 传输成功的概率就没有什么变化，这是因为 SPEED 在做路由选择的过程中没有考虑链路的质量。

从图 5(a)中还可以看出本文方案数据分组传输失败比例比 SPEED 低很多，因而使得本文方案数据分组未满足实时性要求比例平均要比 SPEED 低

10%左右(如图 5(b)所示)。这是因为 SPEED 在选择转发节点的过程中选择转发速度最快的节点,而本文协议保证实时性的同时还考虑了可靠性。从图 5(b)中还可看出当延迟  $\delta$  小于 0.12 s 时,本文方案数据分组未满足实时性要求比例要略高于 SPEED。因为在这种情况下,只能选择转发速度最快的节点进行传输,如果此时再提高传输的可靠性,也不能满足可靠性要求,反而可能会导致更多的数据分组超时。如在延迟  $\delta$  为 0.1 s 时,2 种方法的数据分组未满足实时性要求比例几乎都为 40%左右,远高于设定的阈值 20%。



(a) 数据分组超时与传输失败的比较



(b) 数据分组未满足实时性的比较

图 5 不同延迟阈值  $\delta$  下的比较

图 6 和表 2 比较的是 2 种协议在延迟  $\delta$  为 0.25 s 时端到端之间的延迟。如图 6 所示,2 种方法几乎所有数据分组的延迟都小于给定的延迟  $\delta=0.25$  s,此时造成数据分组未能满足实时性要求的主要原因则是分组传输失败。如表 2 所示,可以观察到 SPEED 中有更多的丢失分组情况(E2E 时延为 0),有 42 个分组传输失败,而 RSPEED 只有 22 个分组传输失败。虽然 RSPEED 有 2 个数据分组超时,但

本文方案的数据分组未满足实时性要求的比例只有 9.41%,而 SPEED 协议则为 16.22%,几乎是本文协议的 2 倍。这是因为 SPEED 协议在选择转发节点的过程,只追求转发速度,选择传输距离远、延时小的节点作为转发节点,而没有考虑距离越远,对应的可靠性会越低。

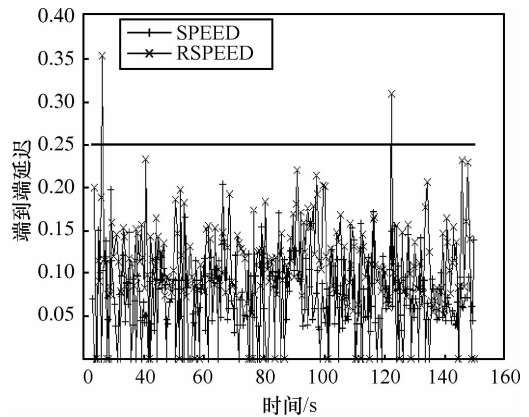


图 6 延迟  $\delta=0.25$  s 时端到端延迟的比较

表 2  $\delta$  为 0.25 s 时 2 种方法的比较

参数	协议	
	SPEED	RSPEED
超时数据分组	0	2
失败数据分组	42	22
总数据分组	259	255
未满足实时性比例	0.162 162	0.094 117
平均延迟/s	0.067 932	0.107 322

从上面的实验中可以看出本文协议在满足实时性的要求下,通过调整传输的延时与传输成功的概率,提高了数据成功转发的概率。在无线传感器网络中,应用程序给定的延迟  $\delta$  根据事件的性质不同,而要求的紧急程度不同,而且延迟  $\delta$  的紧急程度还与目的节点的距离有关。这样,路由协议在做路由选择的过程中,可以根据延迟  $\delta$  的紧急程度,在链路质量比较差时适当提高传输的可靠性。

#### 4.2 可靠性分析

图 7 对本文方法在可靠性满足方面的性能进行了分析。本次实验中,比较了在延迟  $\delta$  为 0.2 s 和 0.15 s 2 种情况下,当可靠性要求  $\Delta P$  从 0.65 变化到 0.9 时的数据分组未满足实时性要求比例。从图 7 中可看出,在延迟  $\delta$  为 0.2 s 时,随着可靠性要求越高,本文协议的数据分组未满足实时性要求比例越低,说明本文的方法选择了质量更高的链路来转发。当延迟  $\delta$  为

0.15 s 时，数据分组未满足实时性要求比例先是随着可靠性要求  $\Delta P$  的增加而减少，而当  $\Delta P$  不大于 0.8 时，此时再增加  $\Delta P$ ，未满足实时性要求比例反而越高。这是因为在延迟  $\delta$  非常小而且链路质量很差时，为了保证过高的可靠性要求，可能导致节点选择可靠性高但延迟较大的节点作为转发节点，导致数据分组超时比例以及未满足实时性要求比例的增加。此时说明链路质量无法满足可靠性要求。

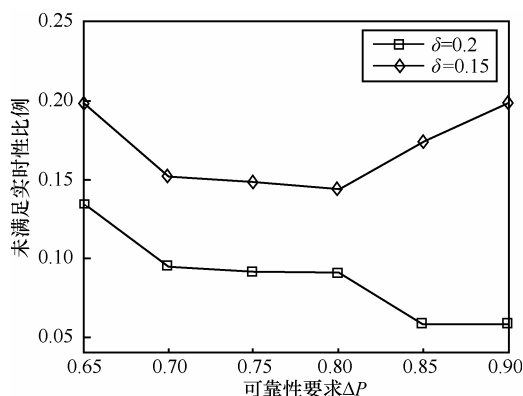


图 7 不同可靠性要求下数据分组未满足实时性的比较。

图 8 比较的是在不同的网络环境下 2 种方法的数据分组未满足实时性要求比例。图 8 中的错误率 (error rate) 表示 Error Model 中分组出错的概率，错误率越大则表示链路质量越差，网络环境越糟糕。从图 8 中可看出随着错误率越高，2 种方法的数据分组未满足实时性要求比例越高，而且本文方案的优势越明显，因为本文方法考虑了链路的可靠性，而 SPEED 只考虑了链路的延迟。

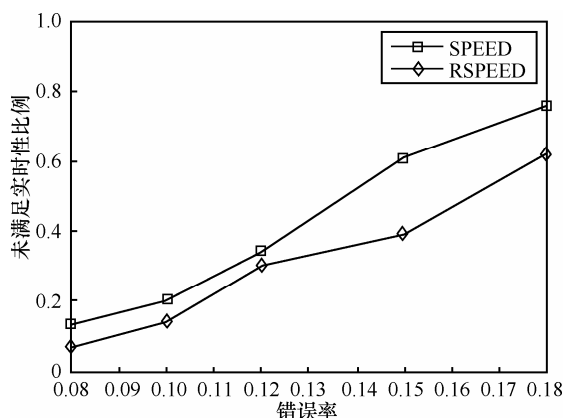


图 8 不同错误率下的数据分组未满足实时性的比较

### 4.3 不同负载下的影响

图 9 分析了本文方法在不同的跳数下数据分组未满足实时性要求比例和数据分组传输失败比例。

本次实验中，设定延迟  $\delta$  为 0.15 s，传输的跳数从 5 增加到 9。如图 9 所示，随着跳数的增加，数据分组未满足实时性要求比例和数据分组传输失败比例越高，但是作者提出的方法的数据分组未满足实时性要求比例和数据分组传输失败比例都要比 SPEED 低。图 10 分析了在不同的数据分组发送率 (packet rate) 下 2 种方法的数据分组未满足实时性要求比例。从图中可以看出随着数据分组发送率越低，数据分组未满足实时性要求比例随之减小，并且本文方法的数据分组未满足实时性要求比例要优于 SPEED。从上面的 2 组实验看出，在不同的跳数和数据分组发送率下，本文方法都要优于 SPEED。这是因为本文方法不仅考虑了实时性，还考虑了可靠性，在满足实时性的前提下，提高了数据分组发送成功的概率。

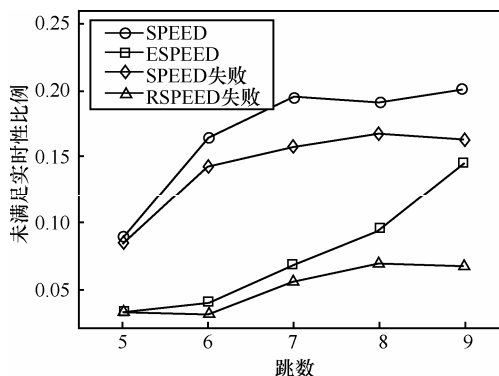


图 9 不同跳数下的数据分组未满足实时性的比较

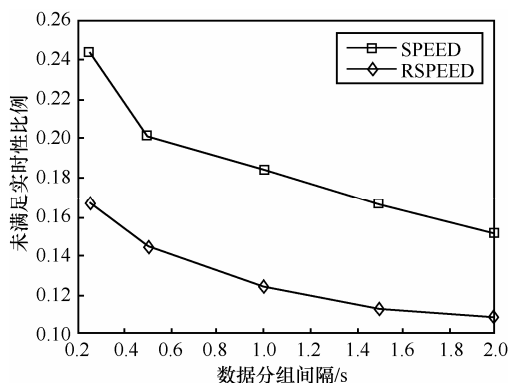


图 10 不同分组发送率下的数据分组未满足实时性的比较

## 5 结束语

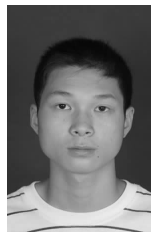
由于无线传感器网络的动态性和不可靠性，导致端到端之间的延迟是不确定的，这给实时路由的设计提出了严峻的挑战。因此，本文综合考虑了链路的质量与链路的延时，提出了一种新的转发可靠

性概念,并在此基础上,设计了一种新的实时路由协议 RSPEED。本文利用转发可靠性表示节点在给定的延迟阈值下成功将数据分组转发到邻居节点的概率,并以此保证路由的实时性和可靠性,提供 QoS 服务。当环境中链路质量比较差时,还能在满足实时的基础上通过权衡传输延时与传输可靠性,以获得更高的传输成功率。实验结果也说明了本文提出的方法的优越性。

### 参考文献:

- [1] AKYILDIZ I F, SU W, SANKARASUBRAMANIAM Y, *et al.* Wireless sensor networks: a survey[J]. *Computer Networks*, 2002, 38(4): 393-422.
- [2] 孙利民, 李建中, 陈渝等. 无线传感器网络[M]. 北京: 清华大学出版社, 2005.  
SUN L M, LI J Z, CHEN Y, *et al.* *Wireless Sensor Networks*[M]. Beijing: Tsinghua University Press, 2005.
- [3] STANKOVIC J A, ABDELZAHER T E, LU C, *et al.* Real-time communication and coordination in embedded sensor networks[A]. *Proceeding of the IEEE*[C]. 2003.1002-1022.
- [4] COUTO D, AGUAYO D, BICKET J, *et al.* A high-throughput path metric for multi-hop wireless routing[A]. *Proceedings of ACM MOBICOM* [C]. San Diego, California, USA, 2003.134-146.
- [5] COUTO D, AGUAYO D, BICKET J, *et al.* A high-throughput path metric for multi-hop wireless routing[A]. *Proceedings of ACM MobiCom*[C]. San Diego, California, USA, 2003.134-146.
- [6] HOU I, KUMAR P R. A survey of recent results on real-time wireless networking[A]. *Proceedings of Real-Time Wireless for Industrial Applications*[C]. CPS Week, Chicago, 2011.
- [7] WANG Y, VURAN M C, GODDARD S. Analysis of event detection delay in wireless sensor networks[A]. *Proceedings of IEEE INFOCOM*[C]. Shanghai, China, 2011. 1296-1304.
- [8] BANSAL S, JUNEJA D, MUKHERJEE S. An analysis of real time routing protocols for wireless sensor networks[J]. *International Journal of Engineering Science and Technology*, 2011, 3(3):1797-1801.
- [9] CACCAMO M, ZHANG L Y, SHA L, *et al.* An implicit prioritized access protocol for wireless sensor networks[A]. *Proc 23rd IEEE RTSS*[C]. Austin, TX, USA, 2002. 39-48.
- [10] WATTEYNE T, AUGEBLUM I, UBEDA S. Dual-mode real-time MAC protocol for wireless sensor networks: a validation/simulation approach[A]. *Proc InterSense*[C]. Nice, France, 2006.
- [11] LU C, BLUM B, ABDELZAHER T, *et al.* RAP: a real-time communication architecture for large-scale wireless sensor networks[A]. *Proceedings of the Eighth IEEE Real-Time and Embedded Technology and Applications Symposium*[C]. San Jose, California, USA, 2002. 55-66.
- [12] FELEMBAN E, LEE C, EKICI E. MMSPEED: multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks[J]. *IEEE Transactions on Mobile Computing*, 2006, 5(6):738-754.
- [13] LEI Z, KAN B Q, XU Y J, *et al.* FT-SPEED: a fault-tolerant, real-time routing protocol for wireless sensor networks[A]. *Proceedings of 2007 International Conference on Wireless Communications, Networking and Mobile Computing*[C]. Shanghai, China, 2007. 2531-2534.
- [14] CHIPARA O, HE Z, XIN G, *et al.* Real-time power-aware routing in sensor networks[A]. *Proceedings of the 14th IEEE International Workshop on Quality of Service* [C]. New Haven, CT, 2006.
- [15] ALI A, LATIFF L A, RAHID R A, *et al.* Real time communication with power adaptation in wireless sensor network[A]. *International Conference on Computing and Informatics*[C]. Kuala Lumpur, MALAYSIA, 2006.
- [16] ZUNIGA M, KRISHNAMACHARI B. Analyzing the transitional region in low power wireless links[A]. *IEEE SECON*[C]. Santa Clara, CA, 2004. 517-526.
- [17] REZAYAT P, MAHDAVI M, GHASEMZADEH M, *et al.* A novel real-time power aware routing protocol in wireless sensor networks[J]. *International Journal of Computer Science and Network Security*, 2010, 10(4):1-6.
- [18] AKKAYA K, YOUNIS M. An energy-aware QoS routing protocol for wireless sensor networks[A]. *Distributed Computing Systems Workshops*[C]. Rhode Island, USA, 2003. 710-715.
- [19] PENG S, LI S, PENG Y, *et al.* Real-time data delivery in wireless sensor networks: a data-aggregated, cluster-based adaptive approach[A]. *Proceedings of UIC*[C]. Hong Kong, China, 2007. 514-523.
- [20] KROGMANN M, HEIDRICH M, BICHLER D, *et al.* Reliable, real-time routing in wireless sensor and actuator networks[J]. *ISRN Communications and Networking*, 2011,2011(1): 1-8.
- [21] LIU Y, HE Y, LI M, *et al.* Does wireless sensor network scale? a measurement study on greenorbs[A]. *Proceedings of INFOCOM*[C]. Shanghai, China, 2011. 873-881.
- [22] Network simulator[EB/OL]. <http://www.isi.edu/nsnam/ns/2009>.
- [23] 徐雷鸣, 庞博, 赵耀. NS 与网络模拟[M]. 北京: 人民邮电出版社, 2003.  
XU L M, PANG B, ZHAO Y. *NS and Network Simulation*[M]. Beijing: Posts & Telecom Press, 2003.
- [24] HIGHTOWER J, BORRIELLO G. Location systems for ubiquitous computing[J]. *IEEE Computer*, 2001, 34(8): 57-66.
- [25] KUROSE J F, ROSS K W. *Computer Networking: a Top-Down Approach Featuring the Internet*[M]. Addison Wesley Longman Inc, 2004.

### 作者简介:



陈权(1989-), 男, 湖北洪湖人, 哈尔滨工业大学硕士生, 主要研究方向为无线传感器网络实时查询。



高宏(1966-), 女, 黑龙江哈尔滨人, 哈尔滨工业大学教授、博士生导师, 主要研究方向为并行数据库、并行压缩数据仓库、数据流、传感器网络数据处理等。