doi:10.11959/j.issn.1000-436x.2015213

低占空比无线传感器网络中基于动态切换的实时路由协议

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：为了实现低占空比无线传感器网络中任意端到端之间的实时数据传输，提出了一种基于动态切换的实时路由协议（DSRT）。首先针对低占空比网络中睡眠延迟太长的特点，DSRT 利用 2 跳邻居信息提出了一种可达速度的概念来帮助发现延迟更优的路径（实验证明该方法至少能够多发现 20%左右延迟更优的路径）。另外，首次发现了在低占空比网络中节点的拥塞程度不仅与缓冲队列中数据分组的个数有关，而且与数据分组的目的节点有关。然而传统基于 1 跳邻居的方法无法区分此类拥塞，因此 DSRT 利用 2 跳邻居信息结合动态切换机制提出了一种通过将缓冲队列分类的拥塞避免算法。最后，通过大量的实验证明，DSRT 比传统的路由算法在实时性和能量消耗上更高效，并且在网络发生拥塞时能够将数据分组的延迟降低 200%以上。
关键词：无线传感网络；低占空比；实时路由；动态切换
中图分类号：TP393.01 **文献标识码**：A

Dynamic switching based real-time routing in low-duty-cycle wireless sensor networks

CHEN Quan, GAO Hong
(Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

Abstract: The dynamic switching based real-time(DSRT) routing protocol was proposed to handle the arbitrary end-to-end(E2E) real-time communication in the low-duty-cycle wireless sensor networks. Firstly, the concept of available speed was designed to compensate for the big sleep latency and facilitate discovering the routes with less latency based on two-hop neighbors' information (at lease about 20% routing path with less latency was discovered by DRST in the experiments). Moreover, it was noticed that the congestion extent in the low-duty-cycle network was determined not only by the number of packets in the network output queue, but also the destination of the packets. However, the traditional method with one-hop neighbors' information cannot differentiate this kind of congestion. Therefore, combined with the dynamic switching mechanism, the DSRT proposed a congestion avoiding algorithm by classifying the packets in the queue. Through comprehensive experiments, the efficiency of routing discovering and congestion avoiding of the DSRT protocol is demonstrated, and the E2E delay is decreased by at least 200% when the traffic was high.
Key words: WSN; low-duty-cycle; real-time routing; dynamic switching

1 引言

随着无线通信、传感器技术等的发展，无线传感器网络已经被广泛地应用在各种领域中，如医疗监护、结构监测、科学探索 [1,2]等。在这些领域中，许多应用都需要提供长时间的服务，需要网络存活很长一段时间，例如几个月或者几年。然而由于受到成本和体积因素的考虑，传感器节点（如 micaz 和 telos 等）一般都是能量受限的[3]。为了解决节点的生存周期和能量之间的冲突，目前广泛采用的一种方法是让节点采用低占空比（low-duty-cycle）的工作方式，即节点只在规定的时间醒来而其他大部分时间都处于睡眠状态。另外，随着应用需求的增加和电池容量的缓慢发展，越来越多的低占空比网

收稿日期：2014-10-15；**修回日期**：2015-04-13
基金项目：国家自然科学重点基金资助项目(61190115, 61033015）
Foundation Item: The Key Program of the National Natural Science Foundation of China (61190115,61033015)

络已经被部署和实施[4]。

除了节点的能量受限问题，许多应用还需要实时的数据传输服务，即在数据分组从源节点路由到目的节点的过程中不能超过给定延迟阈值的限制[4]。例如在战场上的入侵检测、医疗救助以及污染物检测和预防等系统中，都需要保证监测数据或事故报告在给定的时间内到达基站或监控平台，以便系统能够及时做出反应，避免发生严重的后果。

基于上述考虑，本文与文献[5,6]一样考虑了一个在低占空比（例如 5%）网络中的实时路由问题。为了节省能量，在这种网络中节点大部分的时间都处于睡眠状态，只醒来很短的时间完成数据的传输和对周围环境的感知。另外，为了满足覆盖性和连通性的要求，节点都是被异步调度的，通常只有很少的节点在同一时刻能被唤醒。当节点有一个数据分组需要发送时，可能需要等待一段时间直到有邻居节点醒来后帮忙转发，其中，等待的时间即被称为睡眠延迟（sleep latency）。通常情况下，节点之间的睡眠延迟是数据传输延迟的几十倍或上百倍。因此传统的基于节点一直醒着的实时路由协议不适用于这种网络。

为了解决低占空比无线传感器网络对实时数据传输带来的挑战，文献[4～8]提出了一些实时路由协议和数据传输方法。但是文献[5,7]都是基于一种集中式的策略，需要由 sink 节点收集所有节点的信息然后再完成调度或计算。文献[4,6,8]则工作在一种 source-to-sink 的模式，需要预先建立一颗从 sink 节点到源节点的广播树。可见上面的方法都需要提前知道整个或部分网络的信息，不适合任意端到端之间的实时路由。另外，当网络流量很大导致拥塞时（例如 sink 周围附近的节点），节点之间的睡眠延迟将会显著地增加，此时还应考虑如何避免拥塞。因此，本文提出了一种完全分布式并且能够有效避免拥塞的路由协议来实现任意的 2 个节点之间的实时通信。本文的主要贡献如下。

1) 本文实现了一种在低占空比无线传感器网络中任意端到端之间的实时路由协议（DSRT）。

2) 针对低占空比网络中睡眠延迟太长的特点，DSRT 利用 2 跳邻居信息提出了一种可达速度的概念来帮助发现延迟更优的路径。

3) 本文首次发现在低占空比网络中节点的拥塞程度不仅与缓冲队列中数据分组的个数有关，而且与数据分组的目的节点有关。

4) 在 3）的基础上，DSRT 利用 2 跳邻居信息结合动态切换机制提出了一种通过将缓冲队列分类的拥塞避免算法。实验表明本文提出的方法与 SPEED[9]等方法相比能将数据分组的延迟降低 2 倍以上。

2　相关工作

无线传感器网络中的实时路由已经受到了广泛的研究。文献[9,10]根据节点的位置信息和链路的传输延迟提出了基于速度的实时路由协议来概率的保证数据分组的实时性，其中 SPEED[9]是目前被引用最多的实时路由协议。文献[11,12]则通过调整节点的传输功率来增加传输半径，从而减少数据分组的延迟。为了分析和研究大规模传感器网络中的实时性，Wang[13]在 CitySee 系统中进行了大量的数据传输实验。文献[14,15]则根据每条链路成功转发一个数据分组的时间是动态的、不确定的，对路径延迟的概率分布进行了分析。文献[16,17]则通过采用近似的方法牺牲部分的精度来满足实时性要求。但是这些方法都是基于节点一直醒着的场景，并不适合低占空比网络。

对于低占空比网络，Dousse[18]分析了在节点没有合作的情况下数据分组从源节点到 sink 节点的延迟的上下界。文献[4]通过选择转发节点的序列提出了一种优化端到端的期望延迟、能量消耗和传输成功率的方法。但是这种方法工作在一种 source-to-sink 模式，不适合任意端到端之间的通信。文献[5]则考虑通过增加节点的占空比来满足给定的延迟阈值，并且提出了一种动态规划的方法。文献[7]研究了异步占空比网络中的最短路径问题，并且提出了一种解决 all-to-one 最短路径的多项式消息和时间复杂度算法。这 2 种方法均是集中式的，不适合分布式的处理。文献[19]则提出了一种 on-demand 的最小化端到端延迟的路由算法，但是需要先通过广播的方法从源节点到目的节点来建立一条路径。文献[20]则考虑了当链路不可靠时实现低占空比网络中 100% 可靠泛洪的方法。

针对低占空比无线传感网中的实时路由，文献[6]基于期望睡眠延迟提出了一种延迟驱动的路由算法。文献[8]则在优化数据分组在 source-to-sink 模式中的期望延迟的基础上考虑了数据分组的优先级。但是这 2 种方法只适用于 source-to-sink 模式。文献[21]则考虑了低占空比网络中能量消耗和延迟之间的平衡问题，但是是一种集中式的方法。因此，

为了仅基于本地信息实现低占空比网络中任意节点之间的实时通信，并且解决网络中的冲突和拥塞问题，本文提出了一种基于 2 跳邻居信息的 DSRT 路由协议。

3 网络和延迟模型

3.1 网络模型

在一个多跳的低占空比无线传感器网络中，每个节点具有 2 种工作状态：睡眠状态和活动状态。节点只有在活动状态时才可以感知环境、发送或接收数据。在需要发送数据分组时，发送节点可以通过在邻居节点醒来时转换到活动状态来完成数据分组的发送。

由于每个节点醒来的时间不一样，整个网络的连通性也随着时间变化。用 $G(t)=(V, E(t))$表示整个网络在 t 时刻的拓扑结构，其中，$|V|=N$ 表示所有传感器节点的集合，$E(t)$表示 t 时刻边的集合。一条有向边 e_{ij} 属于 $E(t)$当且仅当：1）节点 j 处于节点 i 的通信范围内；2）节点 j 在 t 时刻处于活动状态。另外为了表示方便，用 T 表示节点的一个工作周期，每个工作周期被划分成 m 个长度相同的时间槽 τ（根据 CC2420 的标准，τ 的长度通常都是非常受限的[22]）。为了维持网络的连通性或覆盖性，每个节点都被分配了一个工作计划，用来表示节点被调度醒来的时间。对于节点 i，用一串二进制位 W_i 表示其工作计划，即

$$W_i=\{w_{i0}, w_{i1},\cdots,w_{im}\}$$

其中，$w_{ik}=1$ 表示节点在一个工作周期 T 中的第 k 个时间槽处于活动状态，而 $w_{ik}=0$ 则表示节点处于睡眠状态。很显然，节点的占空比即 W_i 中值为 1 的个数。

3.2 延时模型

在传统的无线传感器网络中，数据分组的延迟主要是由传输延迟（例如 MAC 层的延迟、排队的延迟、传播的延迟等）引起的。而在低占空比网络中，节点必须等待邻居节点醒来后才能转发数据。数据分组的延迟主要指节点之间的睡眠延迟，而传输延迟基本上是可忽略不计的。

根据节点的工作计划 W_i，节点 i 在一个工作周期 T 中可能会有多个处于活动状态的时间槽。令 $t_i=\{t_{i1},t_{i2},\cdots\}$表示节点 i 中活动时间槽的集合，下面将根据 t_i 给出睡眠延迟的数学定义。

定义 1 （睡眠延迟）给定链路 L_{ij} 以及节点 j 活动时间槽的集合 t_j，假设节点 i 在时间槽 t 时需要发送一个数据分组给 j，则节点 i 和 j 的睡眠延迟 $SL_{ij}(t)$表示 i 从时间 t 开始到最近一次 j 处于活动状态的时间间隔，即

$$SL_{ij}(t)=\begin{cases}(t_{\min}-t)\tau, \exists(t_{jp}\in t_j, t_{jp}\geqslant t)\\ (t_{j1}+T-t)\tau, \text{其他}\end{cases} \tag{1}$$

$$t_{\min}=\underset{\forall(t_{jp}\in t_j, t_{jp}\geqslant t)}{\arg\min}\{t_{jp}-t\} \tag{2}$$

其中，t_{jp} 表示节点 j 中第 p 个处于活动状态的时间槽，$t_{\min}$ 表示节点 j 在 t 时刻后第一次处于活动状态的时间槽。

图 1 给出了一个低占空比网络的例子，节点上方的括号代表每个节点的活动时间槽集合，整个网络的工作周期 T=50τ。假设节点 S 在第 3 个时间槽生成了一个数据分组需要发送到目的节点 D。此时，节点 S 没有一个邻居处于活动状态，S 需要将数据分组缓存一段时间直到某个邻居醒来帮忙转发数据。如果节点 S 选择 C 作为转发节点，则链路 SC 的睡眠延迟 $SL_{SC}(3)=(7-3)\tau=4\tau$。

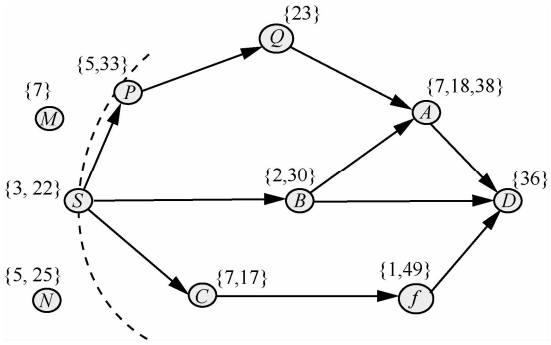


图 1 低占空比网络示例

3.3 ARQ-based 传输

在低占空比网络中，当一个节点醒来后可能会收到多个邻居的发送请求，从而导致竞争和冲突。为了保证传输的可靠性，本文在传输的过程中采用基于自动发送请求(ARQ, automatic repeat request-based)的传输机制。即当一个节点收到一个数据分组后，需要返回一个 ACK 应答分组来通知发送节点数据已被正确接收。否则，发送节点需要等待邻居节点在下一次醒来时重新发送。

另外，为了避免冲突和解决隐藏终端的问题，与文献[9]一样，本文在 MAC 协议中采用了 RTS/CTS 机制。节点在发送一个数据分组前，需要先发送一个 RTS 分组来获取信道。此时，对于一个

时间槽 τ 来说，其长度可以设置为完成一次完整的数据传输过程所包含的时间，包括成功传输：一个 RTS 分组、一个 CTS 分组、一个数据分组和一个 ACK 分组的时间。由于 RTS 分组等的大小只有几十个字节，相对于导致数据分组（1 500 byte）冲突需要重传带来的时间和能量开销来说，其能量和时间是可以忽略不计的。

4　路由算法

本节将详细介绍路由协议(DSRT)。首先假设每个节点都通过 GPS 或其他位置信息服务[23,24]知道自己的位置信息。另外，在低占空比网络中，每个节点必须与周围的邻居进行时间同步使其能够在正确的时间醒来以便与邻居进行通信。在文献[25]中，只用交换少量的数据分组就可以达到 2.24 μs 的精度，相对于一个时间槽 τ=20 000 μs 来说，这个精度无疑是足够的。本文假设通过文献[25]中的方法进行了时间同步。

本文的路由协议 DSRT 架构如图 2 所示，主要分为 4 个模块：邻居管理、转发选择、动态切换和拥塞避免。邻居管理主要负责维护节点的 2 跳邻居信息。转发选择模块根据邻居管理模块中提供的信息来计算下一跳最优的转发节点来满足实时性要求。根据计算出来的转发节点，动态切换和拥塞避免模块再对网络中的拥塞和冲突进行处理以降低数据分组的延迟。其中转发选择和拥塞避免是本文协议的主要内容。

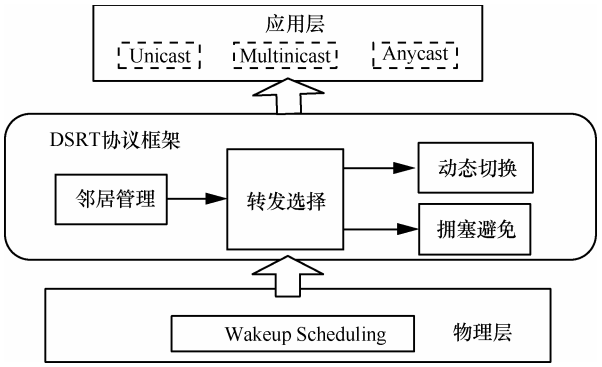


图 2　DSRT 路由协议架构

4.1　转发选择

在低占空比网络中，如果只考虑 1 跳邻居信息，可能会导致节点在路由选择时选择一个睡眠延迟较小的节点来作为转发节点，但是在下 1 跳路由时该节点到其他节点的睡眠延迟则非常大。

另外，本文发现在低占空比网络中节点的拥塞程度不仅与缓冲队列中数据分组的个数有关，而且与数据分组的目的节点有关。传统的基于 1 跳邻居的方法无法区分此类拥塞，因此本文需要根据 2 跳邻居信息来避免拥塞。另外，本文也证明了由于低占空比网络的特点，本文的拥塞避免算法并不适合利用 2 跳以上邻居信息的路由算法。下面将给出基于 2 跳邻居信息来计算转发节点的方法。

首先，令 $NB(i)$表示节点 i 可以直接进行通信的一跳邻居的集合。根据到目的节点的距离，与文献[9]一样，节点 i 到目的节点 D 的候选转发节点集合 FCS 表示距离目的节点更近的节点的集合，可以定义为

$$FCS_i(D)=\{j \mid j\in NB(i)\ \&\ dist(i,D)>dist(j,D)\} \quad (3)$$

其中，$dist(p,q)$表示节点 p 到节点 q 之间的距离。为了减少传输的跳数和传输的时间，本文只考虑将数据分组通过 FCS 集合中的节点进行转发。

接着根据上节给出的延迟模型和睡眠延迟的定义，给出计算 2 跳延迟的方法。

定义 2　（2 跳延迟）给定节点 i、j、k，其中 $j\in NB(i)$，$k\in NB(j)$，2 跳延迟 $delay_{ijk}(t)$是指数据分组从时间槽 t 开始先后经过链路 L_{ij} 和链路 L_{jk} 后到达节点 k 的时间，即

$$delay_{ijk}(t)=SL_{ij}(t)+SL_{jk}(t')+2\tau \quad (4)$$

$$t'=t+SL_{ij}(t)+\tau \quad (5)$$

其中，t'表示数据分组经过链路 L_{ij} 后到达节点 j 的时间。对于 2 跳延迟 $delay_{ijk}(t)$，其不仅包括 2 条链路的睡眠延迟，还包括传输数据分组所占用的 2 个时间槽。为了保证实时性，本文利用转发速度来表示节点转发数据的快慢。根据 2 跳延迟，2 跳转发速度的定义如下。

定义 3　（2 跳转发速度）给定节点 i、j、k，其中 $j\in NB(i)$，$k\in NB(j)$，2 跳转发速度 $speed_{ijk}(t)$表示数据分组从时间槽 t 开始先后经过链路 L_{ij} 和链路 L_{jk} 后到达节点 k 的速度，即

$$speed_{ijk}(t)=\frac{dist(i,D)-dist(k,D)}{delay_{ijk}(t)} \quad (6)$$

其中，$dist(i,D)-dist(k,D)$表示数据分组从节点 i 到 k 所前进的距离。

很显然，对于节点 j 的每一个邻居节点 k，均可以利用式（6）获得一个 2 跳转发速度 $speed_{ijk}(t)$。此时，可以利用 j 邻居节点中的最大 2 跳转发速度

来表示节点 j 的转发速度，因此有下面的定义。

定义 4 （可达速度）给定节点 i、j、k，其中 $j\in NB(i)$，$k\in NB(j)$，对于链路 L_{ij}，其可达速度 $aspeed_{ij}(t)$可定义为

$$aspeed_{ij}(t)=\max\left\{speed_{ijk}(t),\forall k\in NB(j)\right\} \quad (7)$$

根据上面的定义可以看出，对于链路 L_{ij}，其可达速度表示节点 i 通过节点 j 可达到的最大速度。如果节点 i 根据可达速度来选择转发节点，即使发现链路 L_{ij} 的睡眠延迟很大导致其转发速度很慢，可以通过 j 的邻居节点来弥补，但不会忽略 j 仍可以作为转发节点的可能性。例如在图 1 中，如果 S 需要知道通过 B 可达到的最大速度，则可通过计算 S 到 B 的邻居节点 A 和 D 的 2 跳转发速度来获得，即 $\max\{speed_{SBA}(t),speed_{SBD}(t)\}$。可以发现虽然链路 L_{SB} 的睡眼延迟最大为 27τ，但是其可达速度最大。

当数据分组给定延迟阈值 $deadline=\delta$ 后，根据可达速度的定义，可以通过选择可达速度大于给定阈值的节点作为转发节点来满足实时性。

定义 5 （可选转发节点集合 FFS）给定节点 i 及 $FCS_i(D)$，其可选转发节点集合 $FFS_i(D)$可定义为

$$FFS_i(D)=\{j\,|\,j\in FCS_i(D)\ \&\ aspeed_{ij}(t)\geqslant req\_speed\} \quad (8)$$

$$req\_speed=\frac{dist(i,D)}{deadline_i} \quad (9)$$

其中，$deadline_i$ 表示数据分组到达节点 i 后剩余的时间。为了保证数据分组的实时性，引入了 $deadline$ 的动态更新策略，即每当节点 i 收到一个数据分组后，需要对剩余的时间 $deadline$ 进行重新计算。

如果节点 i 的候选转发节点集合不为空，则节点 i 可以根据一定的概率从集合 $FFS_i(D)$中选择转发节点。如果 $FFS_i(D)$集合为空，此时将直接选择可达速度最快的节点作为转发节点。具体算法描述如下。

算法 1 FFS 集合计算

输入：节点 i 的邻居集合 $NB(i)$，目的节点 D

输出：节点 i 的可行转发节点集合 $FFS_i(D)$

begin procedure

1) 根据式(9)计算 req_speed;

2) 根据式(3)计算节点 i 的候选转发节点集合 $FCS_i(D)$;

3) for $\forall j\in FCS_i(D)$ do

4) 根据式(1)计算链路 L_{ij} 的睡眠延迟 $SL_{ij}(t)$;

5) for $\forall k\in NB(j)$ do

6) 根据式(4)和式(5)计算 2 跳延迟 $delay_{ijk}(t)$;

7) 根据式(6)计算 2 跳转发速度 $speed_{ijk}(t)$;

8) if $speed_{ijk}(t)\geqslant req\_speed$ then

9) $FFS_i(D)=FFS_i(D))\cup\{j\}$;

10) break;

11) end if;

12) end for;

13) end for;

14) return $FFS_i(D)$

end procedure

可以看出，算法 1 的时间复杂度与节点 i 的候选转发节点集合 $FCS_i(D)$和节点 j 的邻居集合 $NB(j)$ 大小有关。候选转发节点集合大小与邻居节点的个数有关，而节点的邻居个数一般都是常量，而且步骤 6)~步骤 11)皆可以在常数时间内完成。因此，算法 1 的时间复杂度为 $O(1)$。

4.2 动态切换

在低占空比无线传感器网络中，尽管节点可以采用基于优先级 MAC 的方法，通过提高数据分组的优先级来提高节点获取信道的概率，节点还是会因为竞争失败而失去发送的机会，或者由于链路不可靠而导致发送失败。此时，节点必须等到转发节点下一次醒来的时间来重新进行调度发送，这无疑会急剧地增加数据分组的延迟。因此，本文提出了一种动态切换的方法。

对于节点 i 的可选转发节点集合 $FFS_i(D)$，首先根据节点醒来的时间对 $FFS_i(D)$进行排序。然后，节点 i 依次选择排序后的 $FFS_i(D)$集合中的节点作为转发节点来进行传输。设其中最早醒来的节点是 j，则节点 i 首先选择 j 作为其转发节点，并且尝试将数据分组转发给节点 j。如果节点 j 在接收数据分组的过程中发生冲突或者传输失败（并没有返回 ACK 分组），节点 i 则先根据 j 下一次醒来的时间重新计算其可达速度。如果仍然满足式（8），则重新将节点 j 插入排序后的 $FFS_i(D)$中。如果不满足，节点 i 则将 j 从 $FFS_i(D)$中删除，并且重新从 $FFS_i(D)$中选择下一个醒来的节点重复上述的过程。整个过程一直迭代进行，直到节点 i 成功将数据分组转发

到下一跳的节点中。

由于本文只根据集合 $FFS_i(D)$中的节点进行转发，因此本文的方法在经历多次动态调整的过程后依然能满足数据分组的实时性要求。如果 $FFS_i(D)$集合为空或者对 $FFS_i(D)$集合中的每个节点均传输失败，此时可以通过动态更新 *deadline* 的方法，在后面的路由中进行弥补。

4.3　拥塞避免

在低占空比网络中，当网络发生拥塞时，与传统的无线传感器网络不同的是数据分组的排队延迟不仅仅与缓冲队列中数据分组的个数有关。如图 3 所示，节点 *s* 在路由选择时发现其邻居节点 *j* 中的缓冲队列塞满了数据分组（其中灰色和白色方块分别表示要发送到节点 *a* 和 *b* 的数据分组）。但是在 *j* 的缓冲队列中，并没有需要发送到节点 *k* 的数据分组。此时对于节点 *k* 来说，*j* 并没有产生拥塞，仍可以作为 *s* 的转发节点。

可见，低占空比网络中节点的拥塞程度不仅与缓冲队列中数据分组的个数有关，而且与数据分组的目的节点（例如 *k*）有关。此时，基于传统 1 跳邻居的方法由于无法发现节点 *k* 的存在，则无法发现节点 *j* 仍可以作为其转发节点。因此，本文需要利用 2 跳邻居信息来避免此类拥塞。下面将介绍本文的拥塞避免算法。

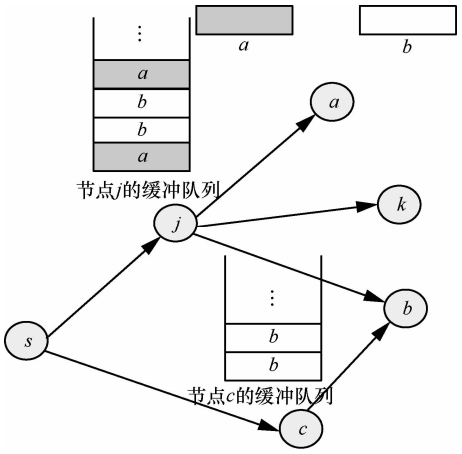


图 3　网络发生拥塞时

假设每个节点都有一个缓冲队列 *Q*，并且队列 *Q* 均按先入先出(FIFO, first in first out)的方式组织。首先，利用转发节点将数据分组进行分类。对于每个数据分组，利用队列信息表来记录数据分组的编号、数据分组的转发节点和其在对应转发节点中的序号以及该数据分组在缓冲队列中的位置。令 $m(j,k)$表示节点 *j* 的缓冲队列中需要发送到邻居 *k* 的数据分组的个数。此时，节点 *j* 的队列信息表 Q_j 如表 1 所示。

表 1　　**节点 *j* 的队列信息表 Q_j**

数据分组编号	转发节点	转发节点中的序号	位置
1	*a*	1	P_{a1}
2	*a*	2	P_{a2}
3	*k*	1	P_{k1}
4	*k*	2	P_{k2}
5	*b*	1	P_{b1}
⋮	⋮	⋮	⋮
n	*a*	$m(j,a)$	$P_{am(j,a)}$

其中，*a*，*b* 表示节点 *j* 的邻居节点，$P_{am(j,a)}$表示需要传输到邻居 *a* 的第 $m(j,a)$个数据分组所在的位置。

另外，可以发现数据分组在低占空比网络中的排队延迟也不是可以简单地通过缓冲队列中数据分组个数的累加来获得。下面以 *j* 的邻居节点 *k* 为例，给出计算 *j* 中数据分组排队延迟的方法。很显然，节点 *j* 中发送到节点 *k* 的第一个数据分组的延迟可以根据式(1)和式(2)计算其睡眠延迟来获得。由于本文假设每个时间槽只能发送一个数据分组，第二个数据分组只能在第一个数据分组发送后才能进行下一次调度。令 $queue_{jk}^{n}(t)$表示链路 L_{jk} 上第 *n* 个数据分组的排队延迟，则有

$$queue_{jk}^{1}(t) = SL_{jk}(t) + \tau \tag{10}$$

$$queue_{jk}^{2}(t) = SL_{jk}(t + queue_{jk}^{1}(t)) + \tau \tag{11}$$

对于第 $n(n>1)$个数据分组的排队延迟，则可以利用式（12）迭代计算

$$queue_{jk}^{n}(t) = SL_{jk}(t + queue_{jk}^{n-1}(t)) + \tau \tag{12}$$

从式(12)可以看出，为了计算数据分组从节点 *j* 到达 *k* 的排队延迟，需要获取节点 *j* 的队列信息表。但是由于 *j* 的队列信息表过大，如果直接传输需要用一个或多个数据分组来进行传输，造成延迟急剧的增加。

下面将介绍一种节点 *i* 获取邻居节点 *j* 缓冲队列信息的简化方法。可以发现，在计算排队延迟的过程中，只需要知道 *j* 的缓冲队列中需要传输到每个邻居节点 *k* 的数据分组的个数 $m(j,k)$，而不需要

获取整个队列信息表。因此可以只用获取下面的队列信息 *QUEUE*

$$QUEUE_j = \{(k, m(j,k)), \forall k \in NB(j)\} \quad (13)$$

对于每个邻居 k，k 和 $m(j,k)$分别只需要 1 byte 来表示，整个队列信息的长度将不超过 $2|NB(j)|$。因此，可以将队列信息 *QUEUE* 嵌入在 CTS 分组中返回给 i。

本文拥塞避免算法具体工作过程如下：根据上一节的介绍，先将集合 $FFS_i(D)$中的节点按照醒来的时间进行排序。假设第一个醒来的节点是 j，则节点 i 首先向 j 发送一个 QUERY 请求，请求节点 j 把缓冲队列中的信息 *QUEUE* 返回给 i。当 j 收到 QUERY 请求后，则根据式（13）对队列信息 *QUEUE* 进行统计并将其返回给 i。当 i 收到节点 j 的队列信息 *QUEUE* 后，根据式（12）计算数据分组的排队延迟，并且重新计算 j 的可达速度。如果仍然满足实时性要求，则直接将数据分组转发给节点 j。反之，如果此时发现无法满足实时性要求，则说明节点 j 中发生了拥塞，节点 i 需要主动进行一次动态切换的过程，重新选择下一个转发节点。由于整个动态调整的过程都是在 $FFS_i(D)$中进行，所以仍然能保证数据分组的实时性。具体过程如算法 2 所示。

算法 2 拥塞避免

输入：节点 i 的可选转发节点集合 $FFS_i(D)$

输出：节点 i 的转发节点

begin procedure

1) 根据节点的醒来时间将 $FFS_i(D)$进行排序；
2) for l=0; l<|$FFS_i(D)$|; l++ do
3) 令 j= $FFS_i(D)[l]$；
4) 节点 i 向节点 j 发送一个 QUERY 请求；
5) If “transmission fail” then
6) 计算节点 j 的下一个醒来的时间槽；
7) 重新计算其可达速度；
8) 如果满足式（8），则将其重新插入到排序好的 $FFS_i(D)$集合中；
9) continue;
10) end if;
11) else
12) 节点 j 返回队列信息 *QUEUE*；
13) for $\forall k \in NB(j)$ do
14) 节点 i 根据式（12）计算排队延迟；
15) 节点 i 重新计算 2 跳延迟；
16) 节点 i 在根据式（6）重新计算 2 跳速度 $speed_{ijk}(t)$;
17) If $speed_{ijk}(t)$>req_speed then
18) return j;
19) end if;
20) end for;
21) end else;
22) end for;

end procedure

在本文路由协议 DSRT 的设计中，QUERY 和 *QUEUE* 信息都可以包含在 RTS 和 CTS 数据分组中进行传输，因此不会增加额外的能量和延迟开销。另外，定理 1 也证明了由于低占空比网络的特点，本文的拥塞避免算法并不适合利用 2 跳以上邻居信息的路由算法。

定理 1 本文的拥塞避免算法并不适合利用 2 跳以上邻居信息的路由算法。

证明 由上可知本文拥塞避免算法的核心是先获取邻居节点的队列信息，再根据其队列信息来计算排队延迟和判断拥塞。对于 2 跳邻居信息来说，本文需要获取其 1 跳邻居的队列信息，可以通过动态切换的方法，在节点醒来的时间槽利用 RTS 和 CTS 分组来获取。而对于 3 跳邻居信息来说，需要获取其 2 跳邻居节点的队列信息，则需要保持其 1 跳邻居和 2 跳邻居同时醒来，而这对于低占空比网络来说是很难保证的。

4.4 邻居管理

邻居节点管理对于许多无线传感器网路中的应用和路由都是必须的。本文的邻居管理主要负责获取和维持周围 2 跳的邻居节点信息。

首先，可以通过低占空比网络中邻居发现的方法[26]或者可以通过在初始化阶段互相广播的方法来获得其一跳邻居的信息。然后，在获得自己的一跳邻居后，节点再将自己一跳邻居的信息以 BEACON 分组广播通知给其他节点。在接收到所有邻居的 BEACON 分组后，节点就可以建立其 2 跳邻居的信息表。由于本文只需要获得邻居节点的位置信息和工作计划，因此 BEACON 数据分组中只需要包含 3 个域（neighborId_, position_, wakeSlot_），其中 neighborId_表示邻居节点的 id，position_表示节点的位置，wakeSlot_表示节点的工作计划 W_i。数据分组的结构如图 4 所示。

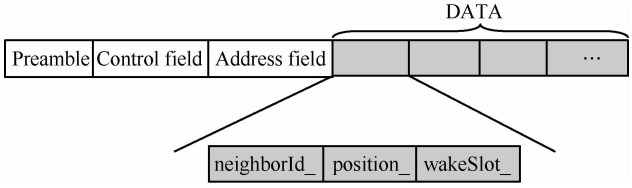


图 4　BEACON 数据分组的格式

在本文 DSRT 路由协议中，neighborId_占 1 byte，position_占 8 byte，wakeSlot_占 16 byte。因此，一个邻居的信息大小总共为 25 byte。对于有 10 个邻居的节点说，只需要用 250 byte 就可以表示整个邻居表的信息。这对大小为 1 500 byte 的数据分组来说，一个数据分组是完全足够的。

5　实验

5.1　实验设置

为了证明本文方法的有效性，在 ns-2[27]模拟器上实现了本文的路由协议 DSRT，并且与其他几种 Baseline 方法进行了对比。在实验中，100~300 个节点被随机部署在一个 200 m×200 m 的区域中。节点的传输半径设置为 40 m，无线电的参数则根据 CC2420[22]硬件标准来设置。节点工作的占空比设置为从 1%变化到 5%。在实验模拟的过程中，每次在目标区域的两边随机选择 2 个节点作为源节点和目的节点。每组实验均重复了 100 遍。

由于目前针对低占空比网络下的实时路由协议[4~8]都不是完全分布式的，因此本文主要与下面 3 种分布式的 Baseline 方法进行了对比。

1) 地理位置路由(GPRS)：每个节点根据自己的位置信息选择转发节点，距离目的节点最近的节点将被选为转发节点。由于考虑了距离信息，这种方法可能找到具有最小跳数的路径。

2) 最小延迟路由(MLF)：在这种方法中，具有最小睡眠延迟的节点将被选为转发节点。另外，为了保证数据分组最后能够到达目的节点，只考虑通过候选转发节点集合中的节点进行转发。这种方法虽然考虑了延迟，但是可能会在源节点和目的节点之间生成一条较大弯路的路径。

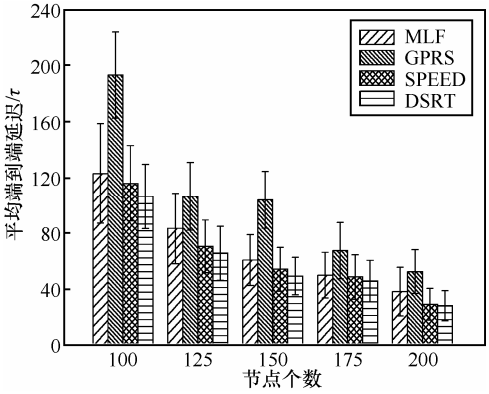
3) 速度路由(SPEED)[9]：每个节点首先计算自己到一跳邻居节点的转发速度（前进的距离除以延迟），其中具有最快转发速度的节点将被作为转发节点。这种方法是目前实时路由中引用最多的一种方法。

在实验中，首先比较了几种算法的平均端到端延迟（E2E delay, end to end delay），即数据分组从源节点到目的节点所经历的延迟。为了对实时性进行分析，对数据分组未满足实时性的比例（deadline miss ratio）进行了比较。在实时路由中，数据分组未满足实时性的比例是一个非常重要的衡量标准。另外，还比较了 3 种方法在不同情况下所产生路径的跳数，显然跳数越多则传输次数和消耗的能量也越多。

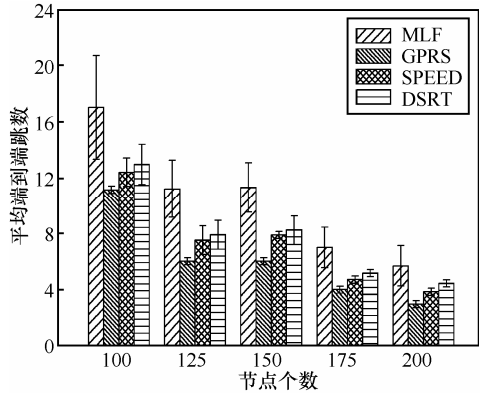
5.2　不同的网络密度

首先，本文对几种方法在不同网络密度下的性能进行了比较。在图 5 中，节点的占空比设为 3%，而节点的个数则从 100 增加到 200。从图 5(a)中可以看出，随着节点密度的增加，4 种方法的平均端到端延迟显著降低，其中本文方法 DSRT 的平均端到端延迟最小。另外，可以注意到 GPRS 方法的平均端到端延迟最大，这是因为 GPRS 每次仅选择距离最远的节点作为转发节点而没有考虑延迟。在图 5(b)中，MLF 方法产生的端到端之间的跳数最高，导致消耗的能量也最多，这是由于其仅考虑了延迟而没有考虑距离，导致数据分组在路由过程中每跳前进的距离很短。另外本文方法 DSRT 产生的跳数要略高于 SPEED 和 GPRS，这是因为 DSRT 在路由的过程中利用 2 跳邻居信息通过增加额外的跳数来减小了延迟。另外，从图 5(c)中可以看出本文方法的数据分组未满足实时性的比例也最低，比 MLF 和 SPEED 方法平均要低 10%左右。这说明本文的方法能更好地满足实时性要求。

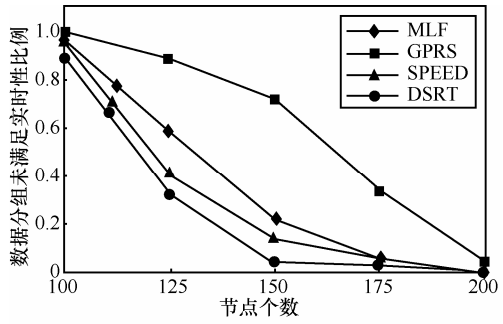
另外，为了进一步说明 DSRT 能够找到延迟更优的路径，本文随机生成了 100 对源节点和目的节点，并将其他 3 种方法在每次路由中所产生路径的延迟与本文方法进行了对比。在图 6 中，实线部分(-GE)表示 3 种 Baseline 方法在 100 次路由中所产生的路径中延迟大于或等于本文方法的部分，而虚线部分(-G)则表示所产生的路径中延迟大于本文方法的部分。从图 6 中可以看出，3 种方法所产生的路径延迟中 90%都要高于或等于本文的方法，而且在其中 20%到 30%的路径中，本方法产生的延迟更小，说明本文利用 2 跳邻居信息的方法能更有效地发现延迟更优的路径。另外，可以看出部分情况下几种方法产生的路径延迟相同，这是因为在不考虑拥塞的情况下，在点对点的路由过程中几种方法可能找到了相同的路径。



(a) 平均端到端延迟



(b) 平均端到端跳数



(c) 数据分组未满足实时性比例

图 5　不同的网络密度下的性能比较

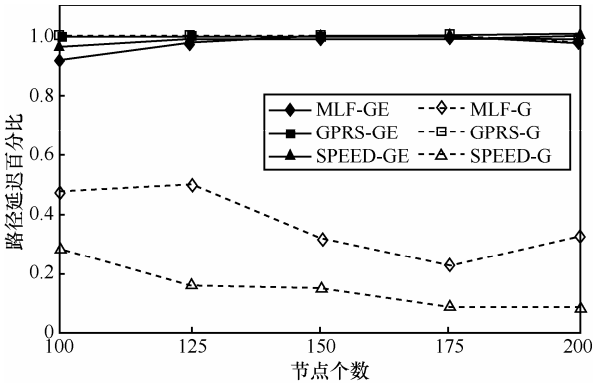


图 6　路径延迟百分比性能比较

5.3　不同的占空比

在图 7 中，节点的个数设为 150，节点的占空比则从 1%变化到 5%。从图 7(a)中可以看出，随着占空比的增加，几种方法的平均端到端延迟均显著地降低，而且本文方法产生的平均端到端延迟最小。在图 7(b)中 MLF 方法所产生的跳数仍然最高，耗费的能量也最大。另外，从图 7(a)和 7(b)中可以看出，虽然 GPRS 方法产生的跳数最小，但是其产生的延迟最高，几乎是其他方法的一倍。在图 7(c)中，本文方法产生的数据分组未满足实时性要求的比例最低，平均要比 SPEED 等方法减少 10%左右。而 GPRS 方法所产生的数据分组未满足实时性要求的比例基本上为 100%。很显然，GPRS 方法并不适合低占空比网络中的实时路由。

从图 7(d)中可以看出，本文的方法所产生的路径中，其中 20%左右要比 MLF 和 SPEED 方法更优，说明本文的方法能更有效地发现网络中延迟较小的路径。另外，还可以从图 5 和图 7 中看出，数据分组的延迟随着节点的密度和占空比的减小而指数下降。

5.4　拥塞的性能

在本次实验中，本文对几种方法在网络发生拥塞时的性能进行了分析。从图 8 中可以看出当在 100 s 引入拥塞后，4 种方法产生的端到端延迟均显著增加。其中 GPRS 方法产生的延迟增加了 8 倍，MLF 和 SPEED 方法分别增加了 3 倍和 5 倍。这是因为当低占空比网络发生拥塞后，仅基于 1 跳邻居的方法无法有效地避免拥塞，导致节点等待的时间显著增加。另外可以发现，当发生拥塞后 MLF 方法产生的延迟要远远高于 SPEED，这是因为 MLF 方法产生的跳数更多，遇到拥塞的可能更大。与其他方法相比，本文方法在面对拥塞时增加的延迟最少，只有 80%左右，而且相比最好的 SPEED 方法则减少了 2 倍左右，说明本文提出的动态切换和拥塞避免算法非常有效地避免了拥塞。

另外，为了证明本文利用 2 跳邻居信息避免拥塞的性能，本文将 DSRT 方法与不利用 2 跳邻居信息进行拥塞避免的方法(DSRT-N)进行了比较。从图 9 中可以看出，利用 2 跳邻居信息进行拥塞避免的方法将数据分组的延迟减少了一倍左右。

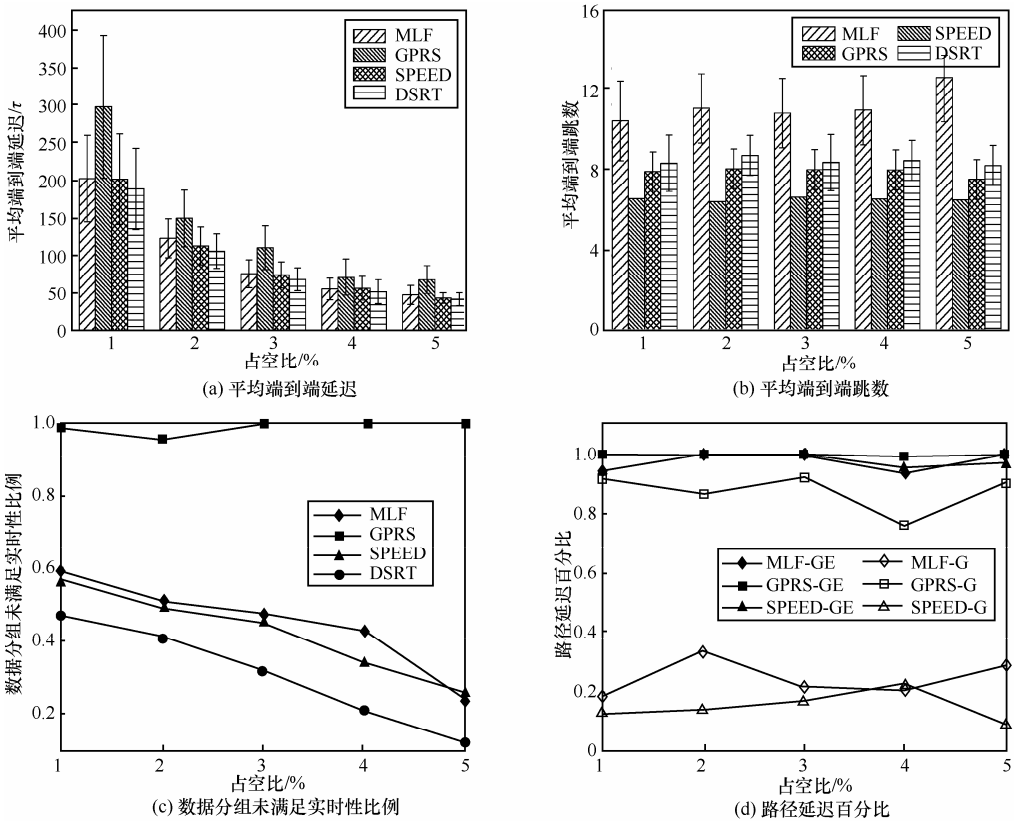


图 7　不同的占空比下的性能比较

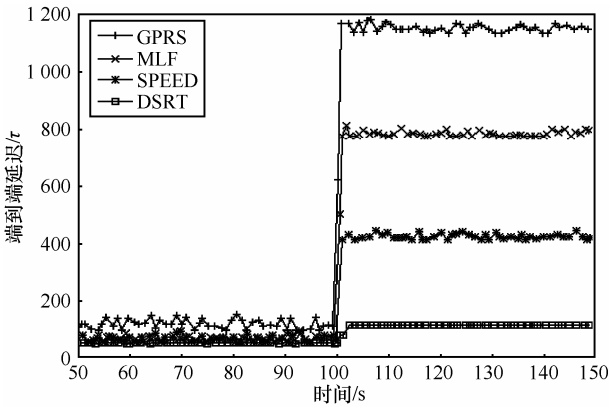


图 8　拥塞时的性能比较

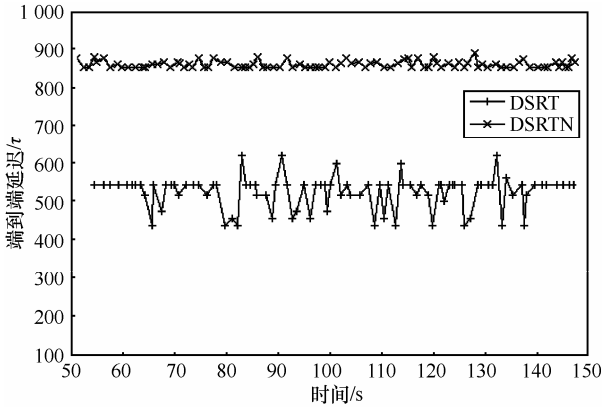


图 9　利用 2 跳邻居信息避免拥塞的性能比较

6　结束语

本文提出了一种基于动态切换的实时路由协议框架(DSRT)来实现任意端到端之间的实时数据传输。首先 DSRT 利用 2 跳邻居信息提出了一种可达速度的概念，来弥补仅仅基于 1 跳邻居做出路由选择的不足。另外，当网络发生冲突和网络拥塞之后，提出了一种基于动态切换的拥塞避免算法。最后，大量的实验结果表明本文设计的路由协议 DSRT 在实时性和能量方面更优越。

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] BOUKERCHE A, CHENG X, LINUS J. Energy-aware data-centric routing in microsensor networks[A]. Proceedings of the 6th ACM International Workshop on MSWiM[C]. San Diego, USA, 2003.42-49.

[4] GU Y, HE T. Dynamic switching-based data forwarding for low-duty-cycle wireless sensor networks[J]. IEEE Transactions on Mobile Computing, 2011, 10(12): 1741-1754.

[5] GU Y, HE T, LIN M, *et al.* Spatiotemporal delay control for low-duty-cycle sensor networks[A]. Proceedings of IEEE RTSS[C]. Washington, USA, 2009.

[6] FAN Z. Delay-driven routing for low-duty-cycle sensor networks[J]. International Journal of Distributed Sensor Networks, 2013, 62(2): 178-179.

[7] LAI S, RAVINDRAN B. On distributed time-dependent shortest paths over duty-cycled wireless sensor networks[A]. Proceedings of the IEEE INFOCOM[C]. San Diego, USA, 2010.1-9.

[8] SUN G D, BIN X. Dynamic routing algorithm for priority guarantee in low duty-cycled wireless sensor networks[A]. Proceedings of WASA[C]. Beijing, China, 2010.146-156.

[9] HE T, STANKOVIC J, LU C, *et al.* SPEED: a stateless protocol for real-time communication in sensor networks[A]. Proceedings of International Conference on Distributed Computing Systems[C]. Providence, USA, 2003.

[10] FELEMBANE, LEE C, EKICIE. 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.

[11] 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.

[12] ALI A, LATIFF L A, RAHID R A, *et al.* Real time communication with power adaptation in wireless sensor network[A]. Proceedings of International Conference on Computing and Informatics[C]. Kuala Lumpur, Malaysia, 2006.1-7.

[13] WANG J, DONG W, CAO Z, *et al.* On the delay performance analysis in a large-scale wireless sensor network[A]. Proceedings of IEEE RTSS[C]. San Juan, Puerto Rico, 2012.

[14] LIU X, ZHANG H, XIANG Q, *et al.* Taming uncertainties in real-time routing for wireless networked sensing and control[J]. IEEE Transactions on Smart Grid, 2013, 4(1): 288-301.

[15] 陈权,高宏.无线传感器网络中基于链路质量的路径延时分析[J].通信学报,2014,35(6):100-109.
CHEN Q, GAO H. Link quality based path delay analysis in wireless sensor networks[J]. Journal on Communication, 2014, 35(6):100-109.

[16] CHENG S, LI J, CAI Z. $O(\varepsilon)$ -approximation to physical world by sensor networks[A]. Proceedings of IEEE INFOCOM[C]. Turin, Italy, 2013.3084-3092.

[17] CAI Z, LIN G, XUE G. Improved approximation algorithms for the capacitated multicast routing problem[A]. Proceedings of COCOON[C]. Kunming, China, 2005.136-145.

[18] DOUSSE O, MANNERSALO P, THIRAN P. Latency of wireless sensor networks with uncoordinated power saving mechanisms[A]. Proceedings of ACM MobiHoc[C]. Tokyo, Japan, 2004.109-120.

[19] SU L, LIU C, SONG H, *et al*. Routing in intermittently connected sensor networks[A]. Proceedings of IEEE ICNP[C]. Orlando, USA, 2008.278-287.

[20] CHENG L, GU Y, HE T, *et al.* Dynamic switching-based reliable flooding in low-duty-cycle wireless sensor networks[A]. Proceedings of INFOCOM[C]. Turin, Italy, 2013. 1393-1401.

[21] GU Y, HE T. Bounding communication delay in energy harvesting sensor networks[A]. Proceedings of ICDCS[C]. Genoa, Italy, 2010. 837-847.

[22] CC2420 Datasheet[EB/OL]. http://www.ti.com, 2004.

[23] CHENG X, THAELER A, XUE G, *et al.* TPS: a time-based positioning scheme for outdoor wireless sensor networks[A]. Proceedings of INFOCOM[C]. Hong Kong, China, 2004. 2685-2696.

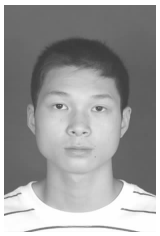
[24] GAO Y, NIU J, ZHOU R, *et al.* ZiFind: exploiting cross-technology interference signatures for energy-efficient indoor localization[A]. Proceedings of INFOCOM[C]. Turin, Italy, 2013. 2940-2948.

[25] FERRARI F, ZIMMERLING M, THIELE L, *et al.* Efficient network flooding and time synchronization with glossy[A]. Proceedings of IPSN[C]. Chicago, USA, 2011. 73–84.

[26] YOU L, YUAN Z, YANG P, *et al.* ALOHA-like neighbor discovery in low-duty-cycle wireless sensor networks [A]. Proceedings of IEEE WCNC[C]. Cancun, Quintana Roo, 2011. 749-754.

[27] Network simulator[EB/OL]. http://www.isi.edu/nsnam/ns/, 2009.

作者简介：



陈权（1989-），男，湖北洪湖人，哈尔滨工业大学博士生，主要研究方向为无线传感器网络中实时路由和实时数据收集。

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。