

# Flow-Based Real-Time Communication in Multi-Channel Wireless Sensor Networks

Xiaodong Wang<sup>1</sup>, Xiaorui Wang<sup>1</sup>, Xing Fu<sup>1</sup>, Guoliang Xing<sup>2</sup>, and Nitish Jha<sup>1</sup>

<sup>1</sup> Department of EECS, University of Tennessee, Knoxville, TN 37996

<sup>2</sup> Department of CSE, Michigan State University, MI 48824

{xwang33,xwang,xfu1,njha}@utk.edu, glxing@cse.msu.edu

**Abstract.** As many radio chips used in today's sensor mote hardware can work at different frequencies, several multi-channel communication protocols have recently been proposed to improve network throughput and reduce packet loss for wireless sensor networks. However, existing work cannot utilize multiple channels to provide explicit guarantees for application-specified end-to-end communication delays, which are critical to many real-time applications such as surveillance and disaster response. In this paper, we propose MCRT, a multi-channel real-time communication protocol that features a flow-based channel allocation strategy. Because of the small number of orthogonal channels available in current mote hardware, MCRT allocates channels to network partitions formed based on many-to-one data flows. To achieve bounded end-to-end communication delay for every data flow, the channel allocation problem has been formulated as a constrained optimization problem and proved to be NP-complete. We then present the design of MCRT, which includes a channel allocation algorithm and a real-time packet forwarding strategy. Extensive simulation results based on a realistic radio model demonstrate that MCRT can effectively utilize multiple channels to reduce the number of deadlines missed in end-to-end communications. Our results also show that MCRT outperforms a state-of-the-art real-time protocol and two baseline multi-channel communication schemes.

## 1 Introduction

Many wireless sensor networks (WSN) applications heavily rely on information being transmitted in a timely manner. For example, a WSN-based disaster warning system must report detected events within a specified real-time deadline. Likewise, a surveillance system needs to notify authorities promptly upon the detection of any intruders. In WSNs, due to the lossy nature of wireless links, real-time communication protocols are commonly designed to provide only soft probabilistic real-time guarantees. There are many factors that may affect the end-to-end delay of a packet from the source to the destination (*e.g.*, a base station). Among them, a major factor is the retransmission caused by unreliable wireless links and channel contention [1].

A common way to improve link quality is to increase transmission power [2]. Transmission power can be used as a knob to reduce end-to-end delays due to

several advantages. First, it is supported by current sensor mote hardware. For example, the CC2420 radio chip [3] used in many mote hardware platforms has 31 different transmission power levels. Second, higher transmission power can lead to higher signal to noise ratio, hence reduce the number of retransmissions for a packet to be delivered [2]. Third, it may also increase the area range of high packet reception rate (*i.e.*, boundary of the gray area) of each node [1], and thus may lead to reduced number of hops needed to reach the destination. Previous work [4] has also shown that desired delays can be achieved by adapting transmission power of each node along an end-to-end path. However, a well-known drawback of increasing power for shorter delays is that high transmission power may significantly increase interferences and channel contention. As a result, the network capacity may be reduced [5]. This has greatly limited the feasibility of using transmission power to provide real-time guarantees.

Recently, multi-channel communication protocols have been proposed for WSNs to improve the performance of traditional single-channel protocols commonly used in WSNs. For example, a multi-channel protocol has been designed in [6] to improve network throughput and reduce packet loss for WSNs. Multi-channel MAC protocols [7][8] have also been proposed to improve network throughput for WSNs. Their simulation results show that those multi-channel protocols outperform their corresponding single-channel protocols. Multi-channel communications are promising because many radio chips used in today's sensor mote hardware can work at multiple frequencies. For example, the CC2420 radio chip provides 16 non-overlapping channels with radio frequency from 2,400 to 2,483MHz. However, existing multi-channel protocols do not provide explicit guarantees for application-specified end-to-end communication delays. On the other hand, as demonstrated in [9], multiple channels can significantly increase network capacity and thus greatly alleviate the drawback of using transmission power as a knob to achieve desired communication delays.

In this paper, we propose MCRT, a Multi-Channel Real-Time communication protocol that utilizes both multiple channels and transmission power adaptation for real-time WSNs. MCRT features a flow-based channel allocation strategy that is designed based on the multi-channel realities identified in a recent empirical study [6]. In particular, MCRT uses only a small number of orthogonal channels and avoids costly time synchronization. In MCRT, channels are allocated to network partitions formed based on many-to-one data flows in a WSN. To achieve bounded end-to-end communication delay for every data flow, we formulate the channel allocation problem as a constrained optimization problem and reduce it to the Maximum Length-Bounded Disjoint Paths problem [10], which is known as NP-complete. We then present an allocation algorithm designed based on well-established heuristics [11] to maximize the number of disjoint paths in the network that can meet the specified end-to-end communication delay. MCRT allocates a different channel to each data flow to minimize the channel contention among different flows. After the network partitions are established, transmission power adaptation is used to achieve energy-efficiency while forwarding each packet with real-time guarantees. We compare MCRT

against three baselines. The first baseline is a simple flow-based channel allocation solution. The second baseline has a node-level channel allocation strategy. The third baseline is a recently published work [4] that uses only transmission power to achieve real-time performance on a single channel. Extensive simulation results based on a realistic radio model demonstrate that MCRT outperforms all the three baselines and can effectively utilize multiple channels to reduce the number of deadlines missed in end-to-end communications. Specifically, the contributions of this paper are four-fold.

- We formulate the flow-based channel allocation problem in multi-channel real-time communications as a constrained optimization problem.
- We prove that the channel allocation problem is NP-complete and present a novel allocation strategy based on well-designed heuristics.
- We combine our channel allocation strategy with a power-efficient real-time packet forwarding protocol to achieve bounded communication delay on each channel.
- We evaluate the performance of our protocol against three baselines using extensive simulations in ns-2.

The rest of paper is organized as follows. Section 2 highlights the distinction of our work by discussing the related work. Section 3 introduces the formulation, proof, and algorithm of our flow-based channel allocation strategy. Section 4 discusses power-efficient real-time packet forwarding. Section 5 provides the design of the baselines used in our experiments. In Section 6, we evaluate the performance of our protocol using simulations. Section 7 concludes the paper.

## 2 Related Work

Many real-time communication protocols have been proposed for wireless sensor and ad hoc networks. A comprehensive review of real-time communication in WSNs is presented in [12]. At the MAC layer, Implicit EDF [13] is a collision-free real-time scheduling scheme by exploiting the periodicity of WSN traffic. RAP [14] uses a novel velocity monotonic scheduling scheme to prioritize real-time traffic based on a packet's deadline and distance to the destination. At higher layers, SPEED [15] achieves desired end-to-end communication delays by enforcing a uniform communication speed throughout the network. MMSPEED [16] can provide QoS differentiation to meet both reliability and timeliness requirements. SWAN [17] also proposes stateless control algorithms for differentiated services. Karenos et al. [18] have also presented a flow-based real-time traffic management mechanism. However, none of the existing real-time protocols takes advantage of the capability of multi-channel communications available in today's mote hardware. Our proposed MCRT protocol is specially designed for real-time communications in multi-channel WSNs.

Recently, several multi-channel MAC protocols have been proposed for WSNs [7][8]. In these protocols, channels are assigned to different nodes locally to minimize interferences. This strategy is referred to as node-based channel assignment.

In node-based protocols, a node usually has a different channel from its downstream node and upstream node in a data flow. Therefore, each pair of nodes must switch to the same channel for communication, which may require precise time synchronization and lead to non-trivial overhead. In addition, some node-based strategies may require a large number of orthogonal channels, which may not be practical for existing mote hardware, as discussed in [6]. Nonetheless, simulation results demonstrate that these protocols can improve communication performance such as network throughput for WSNs. In this paper, one of our baselines also uses node-based channel assignment but requires neither time synchronization nor a large number of orthogonal channels. In contrast to the above related work, MCRT is designed to achieve application-specified end-to-end delays by using only a small number of orthogonal channels.

Some recent work [6][19] proposes coarse-grained channel assignment policies, which allocate channels to disjoint trees or partitions. By minimizing the interferences between different trees or partitions, parallel transmissions can be exploited. In addition, experiments on Micaz hardware are also presented in [6] to investigate multi-channel realities. Two important realities have been reported. First, the number of orthogonal channels is actually small such that a practical multi-channel protocol should rely on only a small number of non-adjacent channels. Second, time synchronization protocols in WSNs could be expensive, in terms of bandwidth and power consumption. Hence, frequent re-synchronization should be avoided in protocols design. Our proposed MCRT protocol organizes the network into different partitions based on data flows, such that the interferences among different flows can be minimized. In addition, MCRT is designed to achieve desired end-to-end communication delays and reduce power consumption at the same time, which are not addressed in existing multi-channel work.

Transmission power control for energy efficiency has been studied extensively in the context of wireless ad hoc networks. The previous work can be roughly classified into two categories: topology control and power-aware routing. Topology control preserves the desirable property of a wireless network (e.g., connectivity) by reducing transmission power to the maximum degree. A survey on existing topology control schemes can be founded in [20] and several representative projects are [21][22][23][24]. The goal of power-aware routing is to find energy-efficient routes by varying transmission power, as presented in [25][26][27][28][29]. Although the above studies demonstrate the effectiveness of transmission power control in reducing energy consumption, none of them deals with real-time requirements in multi-channel WSNs. In our work, we propose multi-channel protocol that uses transmission power adaptation to meet packet deadlines.

Different from all the aforementioned work that handles real-time guarantees, multi-channel communications, and energy-efficiency in isolation, our MCRT communication protocol utilizes the realistic capabilities of existing sensor mote hardware to support power-efficient multi-channel communications for real-time WSNs.

### 3 Flow-Based Channel Allocation

Our MCRT protocol features a flow-based channel allocation policy, which is mainly motivated by two observations. First, we conducted hardware experiments to motivate this work. As shown in our empirical results presented in an extended version of this paper [30], multiple data flows in a WSN compete for the shared wireless channel and thus result in degraded real-time performance. Hence, it is preferable that each data flow uses a different channel. Second, dynamic channel switching at the node level incurs overhead in terms of switching delay and energy consumption. Therefore, it is also preferable that nodes do not need to switch channel too frequently for data transmissions in a data flow.

In our flow-based channel allocation policy, we try to allocate a different channel to each data flow in the network such that the interferences among different data flows can be reduced. A data flow is composed of a source node and the destination, as well as the intermediate nodes in the network that can be used to forward packets from the source node to the destination. Data packets are periodically sent from the source node to the destination in a data flow, but each data packet may take a different path in the flow under different network conditions. Since each data flow is using a different channel, any node in the network (except the destination) can only be allocated to at most one data flow. To establish data flows, we partition the network by searching for a set of disjoint paths from source nodes to the destination. In order for each data flow to meet the specified deadline, the worst-case end-to-end communication delay of each path needs to be smaller than the deadline.

In this section, we first formulate the problem of finding disjoint paths with bounded delay as a constrained optimization problem. We then prove that the problem is NP-complete and propose a search algorithm based on well-established heuristics [11] to find the required number of disjoint paths in the network.

#### 3.1 Problem Formulation

As discussed in [31], any two nodes A and B in a WSN may have three types of communication relationship. First, if B can reliably receive packets transmitted by A, we say that there exists a communication link from A to B. Second, if B cannot reliably receive packets from A, but A's transmission interferes with any transmission intended for B, we say that there exists an interference link from A to B. Last, if A and B cannot communicate or interfere with each other, there is no link between A and B. In this paper, we use *Packet Reception Rate (PRR)* to determine the communication relationship between two nodes in the WSN. Based on the empirical studies presented in previous work [24], we set a link with  $PRR \geq 90\%$  to be a communication link and a link with  $PRR \geq 10\%$  to be an interference link. The PRR value of each link can be measured using an online link quality estimator (*e.g.*, [32]).

Based on the definitions of communication and interference links, a WSN can be represented as a directional graph  $G = (V, E)$ , where V is a set of nodes

and  $E$  is a set of directional links. In the graph, each link is assumed to be uni-directional. Our assumption is reasonable in our applications because we try to meet the end-to-end deadline for the data flows from the source nodes to the destination. In our real-time forwarding algorithm presented in Section 4, we adopt a greedy forwarding policy, *i.e.*, every node forwards packets to neighbors that are closer to the destination. As a result, based on the locations of the nodes, all the communication links in the network can be treated uni-directional. In this paper, we assume that each node is stationary and knows its location via GPS or other localization services. This assumption is reasonable because localization is a basic service essential to many WSN applications that need to know the physical location of sensor readings.

In order for each data flow to meet the given end-to-end communication deadline, we first consider the one-hop delay for a node to successfully receive a packet from another node along a data flow. Without accounting for interferences, the expected number of re-transmissions needed for the node to successfully receive a data packet from the other node can be calculated as the inverse of the PRR value of the communication link between the two nodes. The retransmission number is initially used as the weight of the communication link. For example, the weight of the link from  $E$  to  $B$  in Figure 1(a) can be calculated as 4, which corresponds to a PRR of 25%. We then consider the interferences that may increase the one-hop delay from  $E$  to  $B$ . Because interferences occur at the receiver, the communication link  $FB$  and the interference link  $AB$  in Figure 1 may interfere with  $EB$ . The longest delay that  $EB$  could experience is when all the transmissions happen to occur at the same time, *i.e.*,  $E$  is sending a packet to  $B$ ;  $F$  is also sending a packet to  $B$ ;  $A$  is interfering with  $B$  by sending a packet to another node (*e.g.*,  $C$ ). The worst-case for  $EB$  is that  $E$  has to wait for both  $F$  and  $A$  to finish their transmissions before starting its own transmission. Therefore, the worst-case delay for  $EB$  is the aggregate weight of all the communication and interference links directed to  $B$ . As a result, the one-hop delay of  $EB$  is estimated as 12 (retransmissions). The weight of the interference link  $AB$  can be estimated as the maximum weight of the outgoing communication link from  $A$ , because  $EB$  needs to wait for the longest time when  $A$  is sending a packet to  $C$ . By doing this calculation for every communication link, we have a new graph where the weight of each link is its longest one-hop delay, as shown in Figure 1(b).

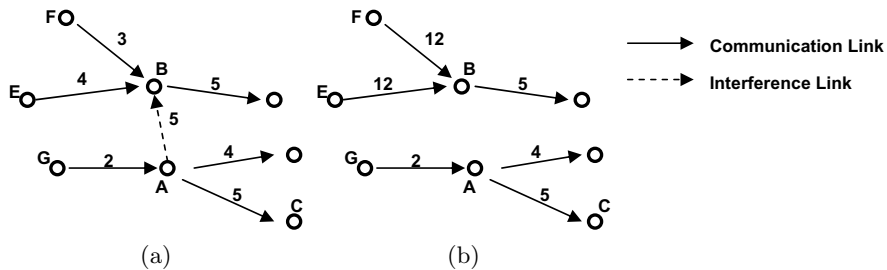


Fig. 1. An example of estimating the worst-case one-hop delay

With the one-hop delay of each link, we now need to find a set of disjoint paths from the source nodes to the destination such that the end-to-end delay of each path is smaller than the deadline. The delay of a path is calculated as the sum of the weights of all the links in the end-to-end path. Those paths are used to partition the network to form data flows with bounded communication delays. Therefore, the problem of finding Disjoint Paths with Bounded Delay (DPBD) can be formulated as a constrained optimization problem as follows. Given a directed graph  $G = (V, E)$  with  $k$  source vertices as  $s_1, \dots, s_k$ , a destination vertex  $t$ , and a set of edges with various weights, we need to find  $k$  or more mutually vertex-disjoint (except the sources and destination) paths from  $s_i, (1 \leq i \leq k)$  to  $t$ . The optimization problem is subject to the constraint that the weight (*i.e.*, delay) of each path needs to be smaller than the deadline  $W$ . If the number of vertex-disjoint paths is greater than  $k$ , some data flows can have more than one path. If we cannot find a path from a source to the destination, the end-to-end delay of that data flow cannot be guaranteed to be smaller than the deadline.

### 3.2 Proof of NP-Completeness

We now prove the DPBD problem to be NP-complete by reducing it to a well-known NP-complete problem, the Maximum Length-Bounded Disjoint Paths (MLBDP) problem [10], which is stated as follows. Given a graph  $G = (V, E)$  with specified vertices  $s, t$  and positive integers  $k, W' \leq \|V\|$ , does  $G$  contain  $k$  or more mutually vertex-disjoint paths from  $s$  to  $t$ , none involving more than  $W'$  edges?

**Theorem 1.** *The DPBD problem is NP-Complete.*

*Proof.* First, it is clear that our problem belongs to NP problem because given a set of  $k$  disjoint paths, we can check if the weight of each path is bounded by the value  $W$ . This check can be performed in polynomial time.

We now reduce our problem to the MLBDP problem. There are two differences between our DPBD problem and the MLBDP problem. First, the edges (*i.e.*, links) in our graph have various weights while the edges in the MLBDP problem have a uniform weight of 1. Second, we need to find one or more paths from each of the  $k$  source nodes to the same destination. However, the MLBDP problem aims to find  $k$  or more paths between the same source  $s$  and the same destination  $t$ . We use two steps to reduce our problem to the MLBDP problem.

In the first step, as the weight of each edge is a rational number, we can always find the greatest common denominator for all the edge weights in the graph, which is denoted as  $c$ . Thus, the weight of each edge can be expressed as  $I \times 1/c$ , where  $I$  is an integer. We then replace this edge with a chain composed of  $I$  new edges (with a weight of  $1/c$ ) and  $I - 1$  new intermediate vertices, as shown in Figure 2(a). As a result, the total weight between the two vertices of the original edge is still  $I \times 1/c$  while each edge now has a uniform weight of  $1/c$ . All the edges in the graph can be replaced in the same way, which leads to a new graph where all the edges have the same weight. In the second step, we

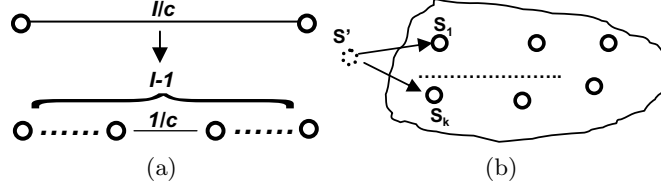


Fig. 2. Graph transformation of the DPBD problem

first add an auxiliary vertex, denoted as  $s'$  to the graph  $G$ . We then link  $s'$  to each of the  $k$  source vertices with an edge whose weight is the uniform value, as shown in Figure 2(b). If we can find  $k$  disjoint paths from  $s'$  to  $t$ , each source node will have one path to the destination with bounded delay.

After the two steps, we have transformed our graph to a new graph  $G' = (V', E')$  with specified vertices  $s', t$  and positive integers  $k, W' = W \times c + 1$ . The DPBD problem is reduced to a new problem stated as follows. Given the new graph  $G'$ , does  $G'$  contain  $k$  mutually vertex-disjoint paths from  $s'$  to  $t$ , none involving more than  $W'$  edges? The new problem is exactly the MLBDP problem. Therefore, our problem is NP-complete.  $\square$

### 3.3 Disjoint Paths Search Algorithm

In this section, we propose a search algorithm designed based on well-established heuristics [11] to find the required number of disjoint paths in the new graph  $G'$  in two steps.

In the first step, the algorithm adopts the Dijkstra's algorithm to find the shortest path from  $s'$  to the destination  $t$  in the network. If the shortest path is not bounded by  $W'$ , it is impossible to find  $k$  disjoint weight-bounded paths. In that case, the search algorithm fails. If the shortest path is bounded, it is added to the solution set  $T$ .

In the second step, based on the shortest path found in the first step, the algorithm iteratively search for the rest  $k - 1$  bounded disjoint paths. Every iteration of the algorithm finds a new path, whose length is bounded by  $W'$ , and guarantees that all the paths found so far are disjoint. Note that each iteration may modify the paths found in previous iterations to maximize the number of bounded paths. Specifically, each iteration works as follows.

Starting from  $s'$ , the algorithm adopts the Depth-First-Search (DFS) method to search for a new path toward  $t$  whose length is bounded by  $W'$ . Suppose that the search has reached node  $n$  and is looking for the next hop, as shown in Figure 3. In order to guarantee that the path found by DFS is disjoint from the existing paths in  $T$ , the algorithm first tries to pick the next-hop node of the new path from the neighbors of  $n$  that do not belong to any existing paths (referred to as free neighbors). If such a neighbor is available and the total length of the path after adding this neighbor is still smaller than the bound, the neighbor is picked by DFS as the next hop in the new path.



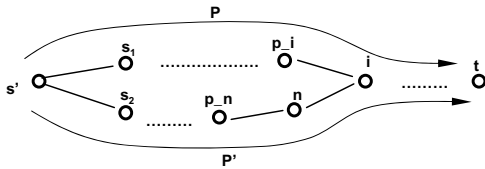


Fig. 3. Example of the Matching Procedure

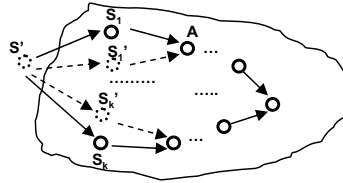


Fig. 4. Extended DPBD problem

If such a neighbor is unavailable, the algorithm starts an augmentation procedure called *matching*. The procedure checks if  $n$  has any neighbor, which belongs to a path in  $T$ , can provide a  $W'$  bounded path toward  $t$ . For example, suppose  $i$  is such a neighbor and  $i$  belongs to an existing path  $P$  in  $T$ . The procedure forms a new path  $P'$ , which includes the current search path from  $s'$  to  $n$ , the link between  $n$  and  $i$ , and the part of path  $P$  from  $i$  to  $t$ . If the length of the new path is bounded by  $W'$ ,  $P$  is deleted from the solution set  $T$  and  $P'$  is added to  $T$ . The procedure then uses  $i$ 's predecessor,  $p_i$ , in path  $P$  as the current node. After the matching procedure, the algorithm starts DFS again from node  $p_i$ .

Since DFS may fail to find the next hop and need to back off, the search may go back to node  $s'$ . In that case, if all the neighbors of  $s'$  have already been visited, it indicates that the last matching procedure was not successful. The algorithm then adds path  $P$ , which was deleted in the last matching procedure, back to  $T$ , and then removes the new path  $P'$  established in last matching from  $T$ . The algorithm then rolls back to continue DFS from node  $p_n$ , which is the predecessor of the current node  $n$  in the last matching procedure.

The whole algorithm terminates under two conditions. First, if the destination  $t$  is reached, the algorithm has successfully found a new disjoint length-bounded path. Second, if the search goes back to  $s'$  with no more neighbor to visit and all the matching procedures conducted before have been rolled back, the algorithm fails to find a new disjoint length-bounded path. The number of paths in  $T$  is the maximum number of disjoint paths with bounded length that the algorithm can find. The detailed algorithm of finding a new disjoint path in the second step is presented in the pseudo code (Algorithm 1).

Based on the analysis in [11], the time complexity for DFS to find a new path is  $O(W' \|E\|)$ . The time complexity of the matching procedure in the algorithm is  $O(W'^2 \|V\| \|E\|)$ . Therefore, the time complexity of finding a new disjoint path with bounded delay is  $O(W'^2 \|V\| \|E\|)$ . The algorithm is currently a centralized procedure but will be extended to a distributed procedure in our future work.

In a real WSN, it is preferable to have multiple paths in each data flow for two reasons. First, a data flow composed of only a single path is not fault-tolerant as node failures may disconnect the flow. Second, with more nodes in a data flow, each node can have more choices to pick the most power-efficient next hop for packet forwarding while meeting the deadline requirement. Power-efficient real-time forwarding is discussed in detail in Section 4. Therefore, in our real implementation of the algorithm, we extend the DPBD problem to allow a

---

**Algorithm 1.** Finding One New Disjoint  $W'$  Bounded Path

---

```

Assume we have a solution set  $T$  that contains  $l \leq k$  disjoint paths;
 $n \leftarrow s'$ ; Matching Stack Height  $\leftarrow 0$ ;
while  $n \neq t$  do
  Use DFS to find next free neighbor  $n + 1$  that provides  $W'$  length bounded path
  to  $t$ ;
  if no free neighbor available then
    Find a neighbor  $i$  in path  $P \subset T$ , which can provide a  $W'$  length bounded path
    to  $t$ ; Establish path  $P'$  through  $n$  and  $i$ ; Push  $n$  into the stack;  $T = T - P + P'$ ;
     $n \leftarrow p_i$ ;
    Continue;
  end if
  if no non-free neighbor available then
     $n \leftarrow p_n$ ;
    if  $n = s'$  then
      if matching stack height = 0 then
        Return failure.
      else
         $n \leftarrow$  stack pop out;  $T = T - P' + P$ ;
      end if
    end if
  end if
end while

```

---

source node to have multiple (*e.g.*,  $m$ ) paths. To this end, we further transform the new graph  $G' = (V', E')$  in the proof of Theorem 1 to make  $m - 1$  copies of each source node  $s_i$ , as shown in Figure 4. Each copy of a source node has the same edges that the source node has. For example,  $s_1$ 's copy,  $s'_1$ , also has edges to  $s'$  and  $A$ . We then run the search algorithm to find the maximum number of disjoint paths for the new graph. As a result, some of the data flows may have multiple paths to the destination. After all the disjoint paths are found, the channel allocation algorithm merges all the copies for each source node in the graph. All the paths that share the same source node belong to the same data flow and are thus allocated a different channel. As a result, interferences among different data flows can be minimized.

## 4 Power-Efficient Real-Time Routing

Based on the theoretical analysis presented in Section 3, all data flows are guaranteed to have bounded end-to-end delay even when every node experiences the worst-case one-hop delay. In a real WSN, end-to-end deadline can be set to be shorter than the theoretical bound. In this section, we present the power-efficient real-time routing strategy adopted by MCRT, which is the second step of MCRT after the channel allocation step. Since each data flow is allocated a different channel, we consider the communication of one data flow in this section.

The design principle of our power-efficient real-time routing strategy is to use adaptive transmission power control to achieve required one-hop delay. Empirical results [2] demonstrate that higher transmission power may lead to improved link quality due to the increased packet reception rate. High reception rate will in turn reduce the number of retransmissions needed to deliver a packet, and thus reduce the transmission delay. Another advantage of power adaptation is energy efficiency. An unnecessary high power level may lead to excessive power consumption. In addition, high transmission power may cause increased interferences and channel contention, and hence reduce the network capacity. In this paper, we implement power adaptation to use just enough power for desired transmission delays. Please note that though we only control transmission power in this paper, our protocol can be integrated with energy-efficient WSN MAC protocols with periodic sleeping (*e.g.*, B-MAC [33]) for further energy saving at the cost of longer communication delays. The integration is our future work.

#### 4.1 Real-Time Forwarding

Based on the single-channel real-time routing algorithm presented in [4], we adopt a dynamic velocity assignment policy and a forwarding policy based on delay estimation. We assume that a data flow periodically sends a packet to the destination. The end-to-end deadline of the data flow is embedded in the packet. Each node that receives the data packet needs to forward the packet to a neighbor based on whether the neighbor can meet the delay requirement of the packet at the *minimum cost of energy consumption*. In this paper, we use two metrics: *required velocity* and *provided velocity* to map a packet's end-to-end deadline to a set of local deadlines for each node to meet. Specifically, when a node needs to forward a packet, it calculates its local deadline, *i.e.*, the required velocity to be achieved for the current hop based on the following equation:

$$velocity_{required}(s, d) = \frac{dis(s, d)}{slack} \quad (1)$$

where  $dis(s, d)$  is the Euclidean distance from the current node  $s$  to the destination node  $d$ .  $slack$  is the amount of time left before the deadline. Note that with this deadline assignment policy, if a packet can meet its required velocity at every hop, it can guarantee to meet its end-to-end deadline. The required velocity is recomputed at each hop. The slack is initially set to be the end-to-end deadline at the source node. At each hop, the slack is decremented to account for queuing, contention and transmission delays based on the estimation methods introduced in [4].

To meet the velocity requirement, the velocity that can be provided by each forwarding choice (*i.e.*, a neighbor node with a certain power level) in the neighborhood table is computed. In the case when node  $s$  forwards a packet to destination  $d$  using a forwarding choice  $(n, p, c)$ , which means node  $n$  is selected as the next hop,  $p$  is the transmission power, and  $c$  is  $n$ 's channel, the velocity provided by the forwarding choice is:

$$velocity_{provided}(n, p, c) = \frac{dis(s, d) - dis(n, d)}{delay(n, p, c)} \quad (2)$$

The one-hop delay  $delay(n, p, c)$  is estimated based on the methods described in [4].  $dis(s, d) - dis(n, d)$  is the progress made toward the destination by forwarding the packet to node  $n$ . To ensure that the neighbor receives the packet, the node has to receive the MAC-layer ACK from the neighbor. If the number of needed retransmissions is larger than 5, the data packet is dropped. This multi-channel forwarding policy eliminates the need of costly time synchronization used in previous node-based multi-channel work (*e.g.*, [7][8]).

## 4.2 Neighborhood Management

We adopt the reliable routing framework proposed in [32] to deal with the dynamic and lossy nature of WSNs. First, link quality and status need to be measured dynamically through a link estimator. Second, measured link quality must be maintained in a neighborhood table for making reliable routing decisions in dynamic environments. In our protocols, we measure the one-hop delay between the node and its each neighbor using data packets to avoid the overhead of probing packets. The delay information of each neighbor is stored in a neighbor table and used to make reliable routing decisions in our protocol. Specifically, we maintain a neighbor table for each node to record the provided velocity of each neighbor. When a node receives a data packet, it searches the table to find a neighbor that can provide the requested velocity and has the lowest transmission power. In that way, we use just enough power for the desired velocity and thus can achieve power-efficiency.

If no neighbor can provide the requested velocity, the node will select some neighbors to conduct power adaptation [2]. The neighbor node used in the last successful packet delivery to the same destination will be considered first, because its link status is most up-to-date. If the last used node is not eligible, the second last used node will be considered. We only consider those neighbor nodes that have a non-zero retransmission number as there is space for power adaptation to reduce their delays. If the neighbor's corresponding transmission power is not the highest power level yet, we use a policy similar to the well-known Multiple Increase Linear Decrease (MILD) backoff algorithm to adjust the power level used to transmit a packet to the neighbor. Specifically, the power level will be multiplied by 1.5 for timely delivery of the current data packet. For example, if the current power level is 10, a power level of 15 will be used to transmit the data packet. This policy is used because timeliness is regarded more important than energy-efficiency in this work. After the packet is successfully transmitted, the power level will be decreased by 1 and will continue to decrease upon every successful packet transmission to this neighbor.

If a node cannot find a neighbor eligible for power adaptation, it sends out a *Routing Request* (RR) packet to find new neighbors that can provide the required velocity. The RR packet contains the required velocity and neighborhood table information, and is broadcast using the highest power level. When neighbors

that are not currently in the neighbor table receive the RR packet, they check whether they can provide the required velocity. If a neighbor can provide required velocity, it replies to the RR packet after a random delay. When other neighbors overhear the reply, they stop sending replies to the current node to reduce the chance of network congestion caused by a large number of replies.

## 5 Design of Baseline Algorithms

In all our experiments, we compare our MCRT protocol against three well-designed baselines: a simple flow-based multi-channel real-time protocol, a node-based multi-channel real-time protocol, and a single-channel power-aware real-time protocol [4].

The first baseline we use to compare with MCRT is a simple flow-based multi-channel real-time protocol called SIMPLE, which is designed in the same way as MCRT except that it uses a simple heuristic to find disjoint paths for channel allocation. During the initialization phase of the network, the source node of each flow broadcasts an explorer packet on the common channel with the distance from the source to the destination attached. Nodes that receive this packet check its own distance to the destination. If its distance is shorter than that in the packet, it waits for a random time and then replies to the source node. Other nodes that overhear the reply will stop sending reply message to avoid network congestion. The packet is then forwarded to the first replying node. The process continues until the explorer packet arrives at the destination. A path from the source to the destination is then established. A multi-hop ACK packet is then sent from the destination back to the source. Every node on the path switches to the new channel immediately after successfully receiving the MAC-layer ACK. In our experiments, two explore packets are used to find two paths for a data flow.

The second baseline is a node-based multi-channel real-time protocol. In this protocol, instead of allocating channels to data flows, every node has its own default channel and needs to dynamically switch channel in order to communicate with another node. In the initialization phase of a WSN, every node claims its own default channel in a way to have approximately even distribution of the channels. During the data transmission phase, if the current node wants to forward a packet to a neighbor on a different channel, the node needs to switch to that channel to send the packet. To ensure that the neighbor receives the packet, the node has to receive the MAC-layer ACK from the neighbor before switching back to its own channel. Note that our node-based baseline eliminates the need of costly time synchronization used in previous node-based multi-channel work (*e.g.*, [7][8]). Different from the single-channel work such as [4], RR packets are broadcast on different channels. In our baseline, a node first broadcasts on its own channel and then switches to other channels to broadcast the RR packet. After that the node switches back to its own channel. If a qualified node needs to send a RR reply to the current node, it switches to the current node's channel to do so.

The last baseline is a single-channel real-time routing protocol called RPAR [4]. We compare MCRT against RPAR to show that multiple channels can be effectively utilized to reduce packet drop ratio, and thus reduce the number of needed retransmissions and communication delays, especially when the specified deadlines are tight. Note that RPAR outperforms several existing real-time and energy-efficient protocols (including one similar to SPEED [15]), by achieving a smaller deadline miss ratio and less energy consumption, as demonstrated in [4]. Therefore, by having better real-time performance and less energy consumption than RPAR, our MCRT protocol also outperforms the baseline protocols used by RPAR.

## 6 Performance Evaluation

In this section, we first introduce our simulation setup. We then present the simulation results to compare our MCRT protocol against the three baselines, under different transmission deadlines, data rates, number of data flows, and network densities.

### 6.1 Simulation Setup

We implement the MCRT protocol in the ns-2.29.3 release of the ns-2 network simulator [34]. We configure ns-2 based on the characteristics of Mica2 sensor nodes. Each node has 31 transmission power levels, from -20 dBm to 10 dBm. The bandwidth is set as 40Kbps for the experiments. The probabilistic radio model in [35] has been implemented in ns-2 to model lossy links. The ns-2 simulator is also modified to support multiple channels and to allow dynamic channel switching. The MAC protocol used in our simulations is a simple CSMA scheme similar to B-MAC [33], the default MAC protocol in TinyOS.

The network topology used in the experiments includes 130 nodes distributed in a  $150\text{m} \times 150\text{m}$  area. The area is divided into  $13 \times 10$  grids, each of which is roughly  $13\text{m} \times 10\text{m}$ . Each grid is configured to have a node randomly deployed in it. In Section 6.4, the network topology consists of 361 nodes distributed in the same area with a higher density. We use the common *many-to-one* traffic pattern in our simulations. In each experiment, the first source node is selected to be the node in the center of the left-most grid column in the network. Other source nodes are randomly selected from the left-most grids with a certain distance from each other. We assume the destination node (*i.e.*, the base station) is a special node that is equipped with multiple radio transceivers, such that it can receive packets on multiple channels simultaneously. We assume the destination locates just outside the right-side boundary of the network. The destination can directly talk, on different channels, to several adjacent nodes located in the center of the right-most grids. As long as a packet can be delivered from a source node to one of those nodes, the packet is assumed to be successfully delivered to the destination.

We use a traffic generator that varies the interval between two data packets based on the sum of a constant (300ms) and a random number generated by

an exponential distribution. The following setup is used in the experiments if not otherwise indicated. The network is configured to have 3 data flows from 3 source nodes to the destination. Each source node generates a new packet every 4 seconds. The end-to-end transmission deadline is 300ms. Three channels are used in our simulations due to the limited availability of orthogonal channels, as reported in [6]. All the nodes start with no neighbor information and thus have an empty neighborhood table.

We use two performance metrics to evaluate the performance of the four protocols: the MCRT protocol and the three baselines. The first metric is *deadline miss ratio* which is the fraction of data packets that miss their deadlines during end-to-end transmissions. This metric examines the real-time performance required in many real-time WSN applications. The second metric we use is *energy consumption per data packet*, which is the ratio between the total energy consumed in transmissions and the number of packets that successfully meet their deadlines. This metric evaluates the energy efficiency of the proposed protocols. Each data point in all the figures is the average of five different runs. The 90% confidence interval of each data point is also plotted.

### 6.2 Different Transmission Deadlines

The first set of experiments evaluates the performance of the four protocols under different end-to-end transmission deadlines. Figure 5 shows the deadline miss ratios when the deadline varies from 150 ms to 350 ms. MCRT has the lowest miss ratio among all the four protocols. MCRT has better performance than RPAR because it can utilize multiple channels for reduced communication delays. MCRT outperforms the node-based scheme because the node-based scheme needs to broadcast RR packet in multiple channels when it fails to find a neighbor that can provide the required velocity, which contributes to longer delay. The performance of MCRT is slightly better than that of SIMPLE because the data flows in MCRT are formed to be bounded even when every node has the worst-case one-hop delay. In contrast, as introduced in Section 5, SIMPLE randomly picks nodes to form data flows without considering interferences and one-hop delay of each node.

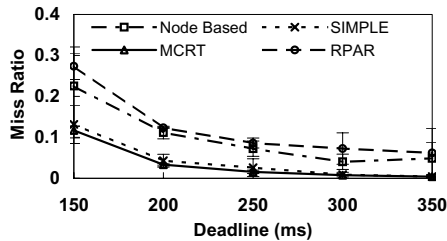


Fig. 5. Miss ratio when deadline is varied

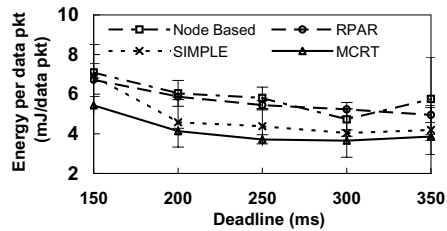
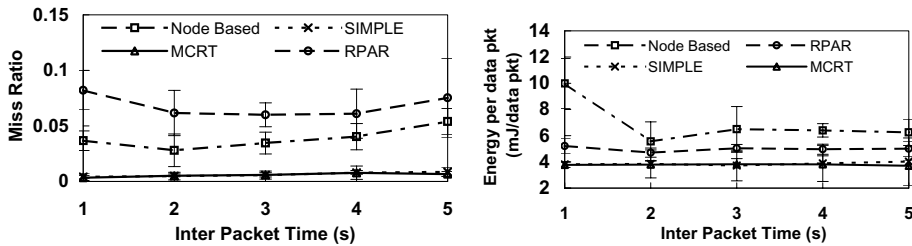


Fig. 6. Energy consumption when deadline is varied

Figure 6 shows the energy consumption of the four protocols. MCRT has the lowest energy consumption for all the deadlines. The reason is that MCRT has the smallest number of retransmissions, which greatly reduces the energy consumption. In addition, MCRT also has a much lower deadline miss ratio as shown in Figure 5. As a result, more packets successfully meet their deadlines, which leads to improved energy efficiency.

### 6.3 Different Data Rates

This set of experiments studies the performance of the four protocols when the data rate of the three source nodes is increased from one packet per 5 seconds to one packet every second. Figure 7 shows that there is no clear evidence that data rate may significantly affect the miss ratio for the four protocols. MCRT and SIMPLE have better real-time performance because they divide the network into partitions, with a different channel used in each partition. As a result, the interferences between different data flows can be reduced.



**Fig. 7.** Miss ratio when data rate is varied **Fig. 8.** Energy consumption when data rate is varied

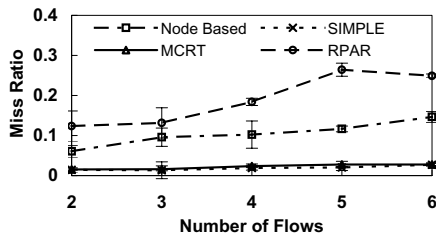
Figure 8 shows the energy consumption. Both MCRT and SIMPLE have lower energy consumptions, compared with the other two protocols. This is because they have much fewer retransmissions caused by the channel contention among different data flows. The node-based scheme consumes the most energy because it needs to send out RR packets on multiple channels. In addition, it is easier for the node-based scheme to have some long-distance neighbors that may need higher power to successfully transmit packets. Those long-distance neighbors are due to the fact that the node-based scheme switches between multiple channels to broadcast RR packets, and thus has a smaller chance of finding nearby neighbors to provide the required velocity.

### 6.4 Different Number of Data Flows

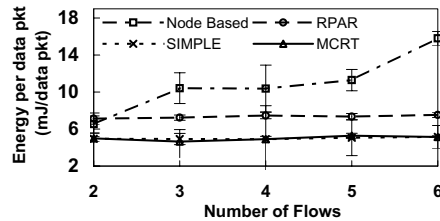
In this experiment, we vary the number of data flows in the network from 2 to 6 and 361 nodes are deployed in the network. When the number of flows is greater than the number of channels, we try to evenly distribute the flows to each channel for the MCRT and SIMPLE protocols. For example, with four



flows and three channels, two of the flows will share a single channel. Figure 9 shows that the deadline miss ratios of MCRT and SIMPLE remain the same when the number of flows increases from 2 to 3. This is because each flow can transmit on a separate channel when the number of flows is smaller than or equal to 3. The miss ratios of MCRT and SIMPLE increase slightly when the number of flows increases from 3 to 6, because two flows need to share one channel, leading to slightly increased channel contention and deadline miss ratios. The increased number of flows has the biggest impact on RPAR, raising its deadline miss ratio to almost 30%. The results show that single-channel protocols are more vulnerable to the increasing number of competing data flows. On the other side, multi-channel protocols can utilize multiple channels to effectively reduce packet drop ratio, and so mitigate the impact of increased data flows.



**Fig. 9.** Miss ratio when number of data flows is varied

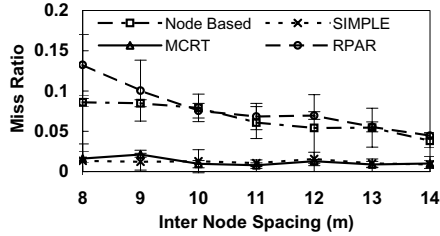


**Fig. 10.** Energy consumption when number of data flows is varied

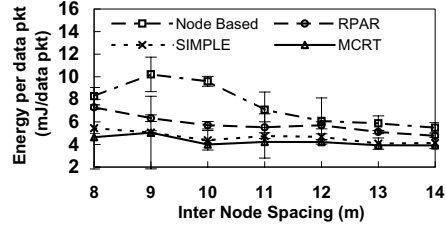
As shown in Figure 10, MCRT and SIMPLE have lower energy consumption. The reason is that MCRT and SIMPLE have fewer retransmissions caused by the channel contention among different flows, as they have fewer flows in each network partition. MCRT is slightly better than SIMPLE (only except for 5 flows) because the delays of the data flows in MCRT are bounded, which results in fewer nodes in each flow and hence smaller number of transmissions to reach the destination. The node-based protocol has the highest energy consumption because each node is more likely to have long-distance neighbors, which require higher power for successful transmissions. In addition, the node-based protocol broadcasts RR packets in multiple channels, which contributes significantly to its energy consumption because more RR packets need to be sent when more data flows are competing for the channel.

## 6.5 Different Network Densities

In this set of experiments, we vary the network density by changing the spacing between every two nodes from 14m to 8m, which in turn changes the total number of nodes from 121 to 361. Figures 11 and 12 show the miss ratio and energy consumption for all the four protocols, respectively. The miss ratios of RPAR and the node-based protocol increase when the network density increases. This is because the neighborhood table of each node is filled up with some short-distance neighbors in a high density network. As a result, the required number of hops for



**Fig. 11.** Miss ratio when network density is varied



**Fig. 12.** Energy consumption when network density is varied

a packet to reach the destination is increased, making it hard to meet the deadline. MCRT and SIMPLE are not significantly impacted by the varying network density because partitioning the network leads to fewer short-distance neighbors in each data flow. The energy consumption decreases for all the four protocols when the network density decreases. This is because more packets are successfully transmitted to the destination due to smaller miss ratios and the number of hops to deliver a packet becomes smaller when the density is lower. MCRT has the lowest energy consumption because it has fewer retransmissions and a lower deadline miss ratio. The node-based protocol has the highest energy consumption because it uses more RR packets than other protocols.

## 7 Conclusion

In this paper, we have presented MCRT, a multi-channel real-time communication protocol that utilizes both multiple channels and transmission power adaptation to achieve real-time communications in WSNs. MCRT features a flow-based channel allocation strategy, which is designed based on the multi-channel realities identified in previous work to use only a small number of orthogonal channels. To achieve bounded end-to-end communication delay for every data flow, the channel allocation problem has been formulated as a constrained optimization problem and proved to be NP-complete. The design of MCRT includes a channel allocation algorithm designed based on well-established heuristics and a real-time packet forwarding strategy. Extensive simulation results demonstrate that MCRT can effectively utilize multiple channels to reduce the number of deadlines missed in end-to-end communications. Our results also show that MCRT outperforms a state-of-the-art real-time protocol and two baseline multi-channel communication schemes.

## References

1. Zhao, J., Govindan, R.: Understanding packet delivery performance in dense wireless sensor networks. In: *SenSys* (2003)
2. Lin, S., He, T., Zhang, J., Zhou, G., Gu, L., Stankovic, J.A.: ATPC: Adaptive transmission power control for wireless sensor. In: *SenSys* (2005)

3. CC2420 2.4 GHz IEEE 802.15.4/ZigBee-ready RF Transceiver, <http://www.chipcon.com>
4. Chipara, O., He, Z., Xing, G., Chen, Q., Wang, X., Lu, C., Stankovic, J., Abdelzaker, T.: Real-time power-aware routing in sensor networks. In: IWQoS (2006)
5. Gupta, P., Kumar, P.R.: The capacity of wireless networks. *IEEE Transactions on Information Theory* 46(2) (2000)
6. Wu, Y., Stankovic, J., He, T., Lin, S.: Realistic and efficient multi-channel communications in dense sensor networks. In: INFOCOM (2008)
7. Zhang, J., Zhou, G., Huang, C., Son, S.H., Stankovic, J.A.: TMMAC: An energy efficient multi-channel mac protocol for ad hoc networks. In: IEEE ICC (2007)
8. Zhou, G., Huang, C., Yan, T., He, T., Stankovic, J.A., Abdelzaker, T.F.: MMSN: Multi-frequency media access control for wireless sensor networks. In: INFOCOM (April 2006)
9. Kyasanur, P., Vaidya, N.H.: Capacity of multi-channel wireless networks: impact of number of channels and interfaces. In: MobiCom (2005)
10. Garey, M.R., Johnson, D.S.: *Computer and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, New York (1979)
11. Ronen, D., Perl, Y.: Heuristics for finding a maximum number of disjoint bounded paths. *Networks* 14, 531–544 (1984)
12. Stankovic, J.A., Abdelzaker, T., Lu, C., Sha, L., Hou, J.: Real-time communication and coordination in embedded sensor networks. *Proceedings of the IEEE* 91(7) (2003)
13. Caccamo, M., Zhang, L.Y., Sha, L.: An implicit prioritized access protocol for wireless sensor networks. In: RTSS (2002)
14. Lu, C., Blum, B.M., Abdelzaker, T.F., Stankovic, J.A., He, T.: RAP: A real-time communication architecture for large-scale wireless sensor networks. In: RTAS (2002)
15. He, T., Stankovic, J., Lu, C., Abdelzaker, T.: SPEED: A stateless protocol for real-time communication in sensor networks. In: ICDCS (2003)
16. Lee, E.F.C.-G., Ekici, E.: MMSPEED: Multi-path multi-speed protocol for qos guarantee of reliability and timeliness in wireless sensor networks. *IEEE Transactions on Mobile Computing* 5(6), 738–754 (2006)
17. Ahn, G.-S., Sun, L.-H., Veres, A., Campbell, A.T.: SWAN: Service differentiation in stateless wireless ad hoc networks. In: INFOCOM (2002)
18. Karenos, K., Kalogeraki, V.: Real-time traffic management in sensor networks. In: RTSS (2006)
19. Vedantham, R., Kakumanu, S., Lakshmanan, S., Sivakumar, R.: Component based channel assignment in single radio, multi-channel ad hoc networks. In: MobiCom (2006)
20. Santi, P.: *Topology control in wireless ad hoc and sensor networks*. Istituto di Informatica e Telematica, Pisa - Italy, Tech. Rep. IIT-TR-04 (2003)
21. Ramanathan, R., Hain, R.: Topology control of multihop wireless networks using transmit power adjustment. In: INFOCOM (2000)
22. Li, L., Halpern, J.Y., Bahl, P., Wang, Y.-M., Wattenhofer, R.: Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In: PODC (2001)
23. Li, N., Hou, J.C., Sha, L.: Design and analysis of an mst-based topology control algorithm. In: INFOCOM (2003)
24. Son, D., Krishnamachari, B., Heidemann, J.: Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks. In: SECON (2004)

25. Singh, S., Woo, M., Raghavendra, C.S.: Power-aware routing in mobile ad hoc networks. In: *MobiCom* (1998)
26. Li, Q., Aslam, J., Rus, D.: Online power-aware routing in wireless ad-hoc networks. In: *MobiCom* (2001)
27. Chang, J.-H., Tassiulas, L.: Energy conserving routing in wireless ad-hoc networks. In: *INFOCOM* (2000)
28. Sankar, A., Liu, Z.: Maximum lifetime routing in wireless ad-hoc networks. In: *INFOCOM* (2004)
29. Doshi, S., Bhandare, S., Brown, T.X.: An on-demand minimum energy routing protocol for a wireless ad hoc network. *SIGMOBILE Mob. Comput. Commun. Rev.* 6(3) (2002)
30. Wang, X., Wang, X., Fu, X., Xing, G., Jha, N.: Flow-based real-time communication in multi-channel wireless sensor networks, Tech Report, University of Tennessee, Knoxville, TN (2008), <http://www.ece.utk.edu/~xwang/papers/mcrt.pdf>
31. Chipara, O., Lu, C., Stankovic, J.: Dynamic conflict-free query scheduling for wireless sensor networks. In: *ICNP* (2006)
32. Woo, A., Tong, T., Culler, D.: Taming the underlying challenges of reliable multi-hop routing in sensor networks. In: *SenSys* (2003)
33. Polastre, J., Hill, J., Culler, D.: Versatile low power media access for wireless sensor networks. In: *SenSys* (2004)
34. ns-2 Network Simulator, [http://nsnam.isi.edu/nsnam/index.php/Main\\_Page](http://nsnam.isi.edu/nsnam/index.php/Main_Page)
35. Zuniga, M., Krishnamachari, B.: Analyzing the transitional region in low power wireless links. In: *SECON* (2004)