MCRT: Multi-Channel Real-Time Communications in Wireless Sensor Networks

XIAORUI WANG, XIAODONG WANG, and XING FU
University of Tennessee, Knoxville
and
GUOLIANG XING
Michigan State University

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 proven 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 and empirical results on a real hardware testbed of Tmote nodes both 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.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communication*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*General*; C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and Embedded Systems

General Terms: Algorithms, Design, Performance

Additional Key Words and Phrases: Wireless sensor networks, multi-channel communications, end-to-end delay, energy efficiency

---

This is a significantly extended version of a conference paper [Wang et al. 2009] published in the 6th European Conference on Wireless Sensor Networks (EWSN 2009), Cork, Ireland, February 2009.

Author's address: Xiaorui Wang, Xiaodong Wang, and Xing Fu, Department of Electrical Engineering and Computer Science, University of Tennessee. Ferris Hall, 1508 Middle Drive, Knoxville, TN 37996-2100. Email: {xwang, xwang33, xfu1}@utk.edu.
Guoliang Xing, Department of Computer Science and Engineering, Michigan State University. 3115 Engineering Building, East Lansing, MI 48824-1226. Email: glxing@cse.msu.edu.

1. INTRODUCTION

Many wireless sensor network (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, such that the average communication delays can be bounded with minimized energy consumption. 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 [Zhao and Govindan 2003].

A common way to improve link quality is to increase transmission power [Lin et al. 2006]. 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 [Chipcon] 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 [Lin et al. 2006]. Third, it may also increase the area range of high packet reception rate (*i.e.*, boundary of the gray area) of each node [Zhao and Govindan 2003], and thus may lead to a reduced number of hops needed to reach the destination. Previous work [Chipara et al. 2006] 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 [Gupta and Kumar 2000]. 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 [Wu et al. 2008] to improve network throughput and reduce packet loss for WSNs. Multi-channel MAC protocols [Zhang et al. 2007][Zhou et al. 2006] 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 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 [Kyasanur and Vaidya 2005], 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 to achieve soft real-time guarantees in WSNs. MCRT features a flow-based channel allocation strategy that is designed based on the multi-channel realities identified in a recent empirical study [Wu et al. 2008]. 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. We first analyze the worst-case one-hop delays with the considerations of retransmissions and interferences. 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 [Garey and Johnson 1979], which is known as NP-complete. We then present an allocation algorithm designed based on well-established heuristics [Ronen and Perl 1984] to maximize the number of disjoint paths in the network that can meet the specified end-to-end communication delay. MCRT then merges multiple paths as data flows for improved fault-tolerance and energy efficiency and 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 [Chipara et al. 2006] that uses only transmission power to achieve real-time performance on a single channel. Extensive simulation results based on a realistic radio model and empirical results on a real hardware testbed of Tmote nodes both 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 both extensive simulations in ns-2 and hardware experiments.

The rest of this paper is organized as follows. Section 2 highlights the distinction of our work by discussing the related work. Section 3 presents hardware results to motivate this work. Section 4 introduces the formulation, proof, and algorithm of our flow-based channel allocation strategy. Section 5 discusses power-efficient real-time packet forwarding. Section 6 provides the design of the baselines used in our experiments. In Section 7, we evaluate the performance of our protocol using simulations and hardware experiments. Section 8 concludes the paper.

2. RELATED WORK

Some 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 [Stankovic et al. 2003]. At the MAC layer, Implicit EDF [Caccamo et al. 2002] is a collision-free real-time scheduling scheme exploiting the periodicity of WSN traffic. RAP [Lu et al. 2002] 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 [He et al. 2003] achieves desired end-to-end communication delays by enforcing a uniform communication speed throughout the network. MMSPEED [Felemban et al. 2006] can provide QoS differentiation to meet both reliability and timeliness requirements. SWAN [Ahn et al. 2002] also proposes stateless control algorithms for differentiated services. Jurcik et al. [Jurcik et al. 2010] propose closed-form recurrent expressions to compute worst-case end-to-end delays. Karenos et al. [Karenos and Kalogeraki 2006] present 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 [Zhang et al. 2007][Zhou et al. 2006][Kim et al. 2008]. 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 [Wu et al. 2008]. Nonetheless, simulation results demonstrate that these protocols can improve communication performance such as network throughput for WSNs. A control theory approach has also been proposed to assign channels to nodes for load balancing [Le et al. 2007]. 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 [Wu et al. 2008][Vedantham et al. 2006] 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 [Wu et al. 2008] 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 protocol design. Our proposed MCRT protocol organizes the network into different partitions based on data flows, such that

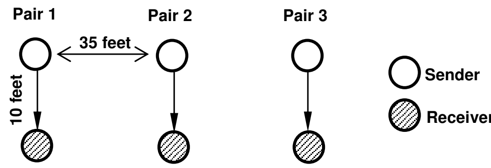


Fig. 1. Experimental setup of empirical studies on real hardware for motivation

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. A recent study [Wang et al. 2010b] assumes that the network is already partitioned and assigns channels to different partitions for minimum transmission energy consumption under delay constraints. In contrast, one of the key contributions of MCRT is its novel allocation strategy that partitions the network into different data flows. Le et al. [Le et al. 2008] recently propose a practical multi-channel MAC protocol that partitions the network based on a K-way cut heuristic to minimize the overhead of channel switching for cross-channel communication. Our work is different because we partition the network for real-time communication.

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 minimum level possible while being able to communicate. A survey on existing topology control schemes can be founded in [Santi 2003] and several representative projects are [Ramanathan and Rosales-Hain 2000][Li et al. 2001][Li et al. 2003][Son et al. 2004]. The goal of power-aware routing is to find energy-efficient routes by varying transmission power, as presented in [Singh et al. 1998][Li et al. 2001][Chang and Tassiulas 2000][Sankar and Liu 2004][Doshi et al. 2002]. 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 requirements, 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. MOTIVATION BASED ON EMPIRICAL STUDIES

In this section, we conduct experiments on real hardware to motivate the design of our multi-channel real-time communication protocol. In particular, our experimental results show that increasing transmission power for one node may cause other nodes in the neighborhood to have long delays due to their increased number of retransmissions caused by high packet drop ratio. This problem can be greatly alleviated by having parallel data transmissions on multiple channels.

Our experiments are performed with 6 Tmote Invent motes in an indoor environment. Each mote is equipped with a CC2420 radio [Chipcon] whose bandwidth

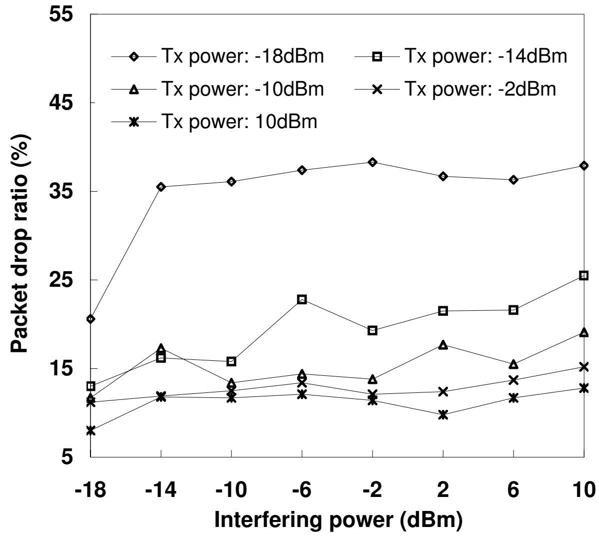


Fig. 2. Packet drop ratio measured in hardware experiments with a single channel.

specification is 250 kbps. We organize the 6 motes in three pairs. Each pair is configured to have a one-to-one communication as shown in Figure 1. The sender of each pair periodically sends out packets with a packet size of 18 bytes and at a rate of 50 packets per second. The three senders are synchronized to transmit packets at the same time for increased chance of interferences at the receivers. We measure the packet drop ratio of the middle pair, *i.e.*, Pair 2, to evaluate the interferences caused by the other two parallel pairs. We vary the transmission power of the senders of Pairs 1 and 3, which is referred to as *interfering power*, to examine the impact of increasing the transmission power on other data flows. We also vary the transmission power of the sender of Pair 2, which is referred to as *transmission power*, to see the impact of transmission power adaptation.

In the first experiment, all the 6 nodes are using the same channel. From Figure 2, we can see that the packet drop ratio increases when the interfering power increases. For example, with a transmission power of -14dBm at the sender of Pair 2, the drop ratio at its receiver increases from 12% to 26% when the interfering power increases from -18dBm to 10dBm. This result shows that increasing transmission power has a negative impact on other data flows in the network. The resultant high drop ratio will cause more retransmissions and thus longer delays for neighbor nodes.

We can also observe that the packet drop ratio is higher when the sender of Pair 2 is using a lower transmission power. That means a node is more likely to experience a long delay when it is using low power to transmit while its neighbors are using high power. In a real-time communication protocol that relies on transmission power adaptation to achieve short delays such as [Chipara et al. 2006], the long delay may in turn cause the node to switch to high transmission power. Consequently, all the nodes may eventually switch to high transmission power to compete for the shared channel, which will lead to excessive power consumption and degraded real-time performance.

In the second experiment, we configure each pair to transmit on a different chan-

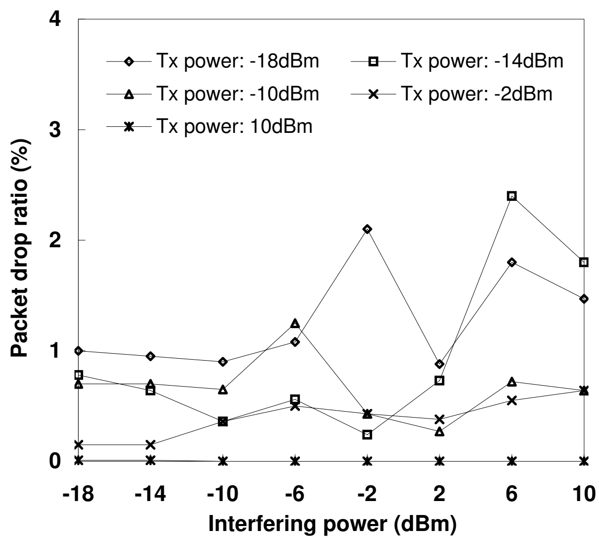


Fig. 3. Packet drop ratio measured in hardware experiments with multiple channels.

nel. Based on the multi-channel realities presented in [Wu et al. 2008], we use three non-adjacent channels as 2,405MHz, 2,420MHz and 2,435MHz in our experiments. Figure 3 shows that the packet drop ratio with multiple channels in the same scenario is much smaller than that with a single channel. Note that the y-axis scale of Figure 3 is about *10 times* smaller than that of Figure 2.

The two experiments lead to three observations: 1) Increasing transmission power results in increased interferences to other data flows on the same channel. 2) Using transmission power adaptation to achieve short delays in a single-channel network may cause excessive power consumption and degraded real-time performance. 3) Multiple channels can be effectively utilized to reduce packet drop ratio, and thus reduce the number of retransmissions and communication delays.

4. FLOW-BASED CHANNEL ALLOCATION

Our MCRT protocol features a flow-based channel allocation policy, which is mainly motivated by two observations. First, as discussed in Section 3, 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. We then merge multiple paths to form a data flow so that the data flow can be resilient to node failures and can still achieve the desired real-time performance even when the network conditions (*e.g.*, interference) change at runtime. We discuss this in detail in Section 4.4.

Similar to other coarse-grained channel assignment policies (*e.g.*, [Wu et al. 2008][Vedantham et al. 2006]), MCRT is more practical for existing mote hardware and can overcome the limitations of node-based channel assignment policies, as discussed in Section 2. It is important to note that most coarse-grained channel assignment policies require that the network be partitioned to disjoint subtrees [Wu et al. 2008] or components [Vedantham et al. 2006], so that different subtrees or components can work on different channels for reduced interference. Similarly, in MCRT, the network is partitioned to disjoint data flows. However, the requirement of using disjoint data flows can be relaxed by allowing selected nodes to work in a node-based manner. For example, if there are not enough nodes around the base station to form the required number of disjoint data flows, we can allow selected nodes to switch channel at runtime to serve multiple flows (like the root of a subtree) at the cost of increased delays. The integration with node-based channel assignment policies is our future work.

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 [Ronen and Perl 1984] to find the required number of disjoint paths in the network. Finally, we discuss how to merge multiple paths as data flows.

4.1 Problem Formulation

As discussed in [Chipara et al. 2006], 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 [Son et al. 2004], 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 at runtime using an online link quality estimator (*e.g.*, [Woo et al. 2003]).

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

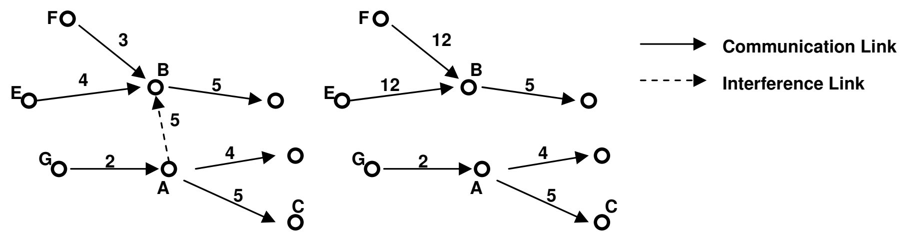


Fig. 4. An example of estimating the worst-case one-hop delay

forwarding algorithm presented in Section 5, 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 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 considering interferences, the expected number of retransmissions 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 4(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 4 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 detects the transmissions of F and A (*e.g.*, using CSMA-like MAC protocols) and then has to back off and 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 (*i.e.*, AC), 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 4(b). Note that the retransmission numbers used in this analysis are based on average PRR values because the worst-case PRR value can be 0 (*i.e.*, infinite number of retransmissions), making it infeasible to analyze the one-hop delays.

In the above one-hop delay analysis, we mainly rely on metrics that are specific to a network topology, such as PRR and interferences from neighbors, which are largely independent of traffic patterns or packet scheduling policies. As a result, this model predicts the average probability that each link would have interferences and hence needs more time for retransmissions at runtime, no matter what traffic patterns or scheduling policies are used. Clearly, traffic pattern and scheduling policy can be used to improve the accuracy of our delay model. For example, if we could know the traffic pattern of each node in a neighborhood (*e.g.*, inter-arrival times of packets) and the transmission schedule of all the neighbors (*e.g.*, when a neighbor will start to transmit a packet), the delay of each packet caused by interference can be precisely calculated. However, there are several challenges in analyzing one-hop delays in such a fine-grained way. First, we need to make assumptions on the underlying MAC protocol. In particular, we need to know either the contention-free transmission schedules from a TDMA-like protocol or the detailed backoff algorithm of a CSMA protocol. The resultant delay analysis would be highly dependent on the MAC protocol implementation. Second, even when the packet scheduling policy is given, unpredictable runtime network dynamics, such as environmental noises, may significantly affect the number of needed retransmissions for a link and thus affect the transmission schedule of the neighbors. As a result, the offline delay analysis based on a certain scheduling policy may become inaccurate. Third, while we may assume a certain traffic pattern at each source node, end-to-end forwarding and dynamic routing can make the traffic pattern highly unpredictable for each relaying node on a multi-hop path. Therefore, although our analysis may lead to conservative estimation of delay, it allows our real-time protocol to be more general and also more resilient to unpredictable network dynamics.

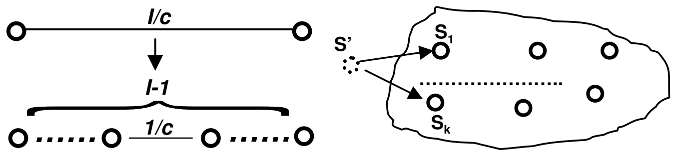
With the one-hop delay of each link, we now need to find a set of disjoint paths from the source nodes to the destination such that the end-to-end delay of each path is smaller than the deadline. The delay of a path is calculated as the sum of the weights of all the links in the end-to-end path. Those 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 $s_1, \ldots, s_k$, a destination vertex t, and a set of edges with various weights, we need to find the maximum number of (k or more) mutually vertex-disjoint (except the sources and destination) paths from $s_i, (1 \le i \le k)$ 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.

4.2 NP-Completeness Proof of DPBD Problem

We now prove that the DPBD problem is NP-Complete by reducing it to a well-known NP-complete problem, the Maximum Length-Bounded Disjoint Paths (MLBDP) problem [Garey and Johnson 1979], which is stated as follows. Given a graph G' = (V', E') with specified source s, sink t and positive integers k, $W' \le \|V\|$, does G' contain k or more mutually vertex-disjoint paths from s to t, none involving more than W' edges?

THEOREM 4.2.1. *The DPBD problem is NP-Complete.*

PROOF. We first show that DPBD $\in$ NP. Suppose that the solution to the DPBD problem results in k disjoint paths whose lengths are bounded by W. We can verify



(a) (b)

Fig. 5. Graph transformation of the DPBD problem

this solution with a complexity of $O(kW)$, which is in polynomial time. Therefore, DPBD $\in$ NP.

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 5(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 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 5(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, the DPBD problem is NP-Complete. $\square$

4.3 Disjoint Paths Search Algorithm

In this section, we propose a search algorithm designed based on well-established heuristics [Ronen and Perl 1984] 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 length of 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 length of the shortest path is bounded by W', it is added to the solution set T.

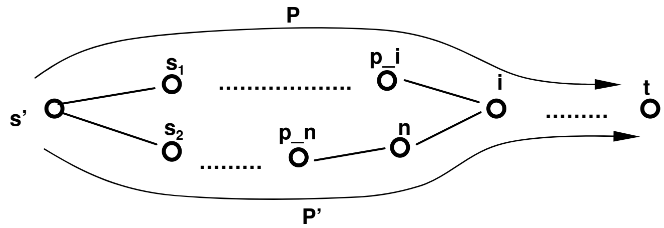


Fig. 6. Example of the matching procedure

In the second step, based on the shortest path found in the first step, the algorithm iteratively searches for the $k-1$ other length-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 length-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 6. 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.

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

Algorithm 1 Finding One New Disjoint W' Bounded Path

```
Assume we have a solution set T that contains l ≤ k disjoint paths;
n ⇐ s'; Matching Stack Height ⇐ 0;
while n ≠ t do
  if Use DFS to find next free neighbor n + 1 that provides W' length bounded
  path to t then
    n ⇐ n + 1;
  else
    if Find a neighbor i in path P ⊆ T, which can provide a W' length bounded
    path to t then
      Establish path P' through n and i; Push n into the stack; T ⇐ T − P + P';
      n ⇐ p_i;
      Continue;
    else
      if n = s' then
        if Matching Stack Hight = 0 then
          Return failure.
        else
          n ⇐ stack pop out; T ⇐ T − P' + P;
        end if
      else
        n ⇐ p_n;
      end if
    end if
  end if
end while
```

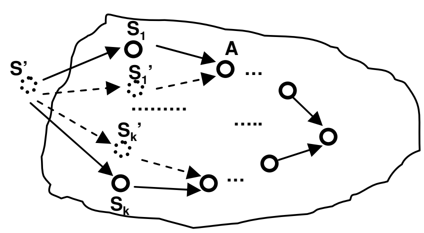


Fig. 7. Extended DPBD problem.

presented in the pseudo code (Algorithm 1).

Based on the analysis in [Ronen and Perl 1984], 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 the disjoint path search problem can be solved in a distributed way with slightly worse solution quality [Sidhu et al. 1991][Cheng et al. 1990]. The distributed algorithm includes two steps. In the first step, the sink node sends out a packet to establish a shortest-path tree rooted at the sink in a distributed way. In the second step, the sink sends out messages to explore more paths to the source. Each node individually checks whether a new path is found and notifies the sink to merge the new path into the solution set if the bound requirement is met. This algorithm continues until the desired number of paths is found. Similarly, our solution can also be extended to run on the sensor nodes in the network in a distributed way. The detailed extension is beyond the scope of this paper. In addition, note that many real-world WSN applications (*e.g.*, [Selavo et al. 2007][Jeong et al. 2005]) adopt many-to-one communication for data collection, in which the sink is usually a sensor mote connected to the base station, such as a computer. The base station is commonly used to make centralized decisions for these applications. Therefore, our algorithm can also run on the base station computer periodically or in an on-demand manner with PRR values measured at runtime to handle varying network conditions.

4.4 Discussion

In a real WSN, it is preferable to have multiple paths in each data flow for several reasons. First, a data flow composed of only a single path is not fault-tolerant as node failures may disconnect the flow. Second, a data flow with more nodes from multiple paths can better achieve the desired real-time performance when the network conditions (e.g., interferences) change at runtime. For example, when a path in the data flow is under interferences from concurrent transmissions on the same channel between nodes external to the data flow (or even the network), having more nodes can provide alternative paths to avoid or mitigate the external interferences for improved real-time performance. Note that although the link quality (PRR) values used in our problem formulation are measured online and our algorithm can also be executed at the base station periodically or in an on-demand manner to handle varying network conditions, having more nodes in a data flow can allow the network to be more adaptive to short-term variations in network conditions. Finally, 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 5.

Therefore, in our real implementation of the algorithm, we extend the DPBD problem to allow a 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 7. 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 of 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 the same channel. As a result, interferences among different data flows can be minimized. Note that although with multiple paths merged in a flow, packets may not strictly follow the disjoint paths for forwarding, a key advantage of using the DPBD problem as a guideline to form data flows is that we can analytically avoid possible hot areas in the network, by having several paths whose delays are bounded by the desired communication deadlines.

In the rest of this paper, by MCRT, we refer to our solution with multiple paths in a data flow resulting from the extended DPBD problem. The solution with only a single path in each flow based on the original DPBD problem is referred to as MCRT-simple. In Section 7, we demonstrate that although both MCRT and MCRT-simple outperform the baselines introduced in Section 6, MCRT can achieve an even lower network energy consumption and a smaller deadline miss ratio than MCRT-simple, especially when the data flows in the network are under the interferences caused by concurrent transmissions on the same channels.

5. POWER-EFFICIENT REAL-TIME ROUTING

Based on the theoretical analysis presented in Section 4, 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 [Lin et al. 2006] 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 [Polastre et al. 2004] and DutyCon [Wang et al. 2010a]) for further energy saving at the cost of longer communication delays. The integration is our future work.

5.1 Real-Time Forwarding

Based on the single-channel real-time routing algorithm presented in [Chipara et al. 2006], 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:

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

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. 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. 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 [Chipara et al. 2006].

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 from the partition of the s's flow, p is the transmission power, and c is n's channel, the velocity provided by the forwarding choice is:

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

The one-hop delay $delay(n, p, c)$ is estimated based on the methods described in [Chipara et al. 2006]. $dis(s, d) - dis(n, d)$ 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.*, [Zhang et al. 2007][Zhou et al. 2006]).

5.2 Neighborhood Management

We adopt the reliable routing framework proposed in [Woo et al. 2003] 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 [Lin et al. 2006]. 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.

Similar to the neighborhood management schemes used in [Woo et al. 2003] and [Chipara et al. 2006], we use the FREQUENCY algorithm to manage the neighbor table, such that the total table size is bounded to maintain only the most frequently used neighbors. As a result, it is practical for sensor motes with severe memory constraints to run MCRT. Specifically, the FREQUENCY algorithm maintains a frequency counter for each forwarding choice. When a forwarding choice is used for routing, its frequency counter is incremented. When a new forwarding choice is inserted and the table size reaches the bound, this new choice will replace the forwarding choice with the smallest frequency count. Consequently, only the frequently used forwarding choices are kept in the table with bounded size.

6. 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 [Chipara et al. 2006].

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. Each node that receives this packet checks 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.*, [Zhang et al. 2007][Zhou et al. 2006]). Different from the single-channel work such as [Chipara et al. 2006], 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 an RR reply to the current node, it switches to the current node's channel to do so.

The third baseline is a state-of-the-art single-channel real-time routing protocol called RPAR [Chipara et al. 2006]. 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 [He et al. 2003]), by achieving a smaller deadline miss ratio and less energy consumption, as demonstrated in [Chipara et al. 2006]. Therefore, by having better real-time performance and less energy consumption than RPAR, our MCRT protocol also outperforms the baseline protocols used by RPAR.

In our experiments, we also compare MCRT with the MCRT-simple scheme introduced in Section 4.3. In MCRT-simple, we only have a single path for each data flow in the network. In contrast, in MCRT, we use the extended DPBD problem to search for the maximum number of disjoint paths in the network and then merge them as data flows. As discussed in Section 4.4, MCRT has several advantages over MCRT-simple. First, MCRT is more fault-tolerant to node failures. Second, MCRT is more resilient to interferences. Finally, in MCRT, each node can have more choices to pick the most power-efficient next hop for packet forwarding while meeting the deadline requirement. In Section 7, we demonstrate that although both MCRT and MCRT-simple outperform the baselines, MCRT achieves an even lower network energy consumption and a smaller deadline miss ratio than MCRT-simple, especially when the data flows in the network are under the interferences caused by concurrent transmissions on the same channels.

7. 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 and MCRT-simple, under different transmission deadlines, data rates, number of data flows, network densities, and number of available channels. Finally, we discuss our empirical results conducted on a hardware testbed of Tmote nodes.

7.1 Simulation Setup

We implement the MCRT protocol in the ns-2.29.3 release of the ns-2 network simulator [ns2]. We configure ns-2 based on the characteristics of Mica2 sensor motes. 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 [Zuniga and Krishnamachari 2004] 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 [Polastre et al. 2004], the default MAC protocol in TinyOS.

The network topology used in the experiments includes 130 nodes distributed in a 150m $\times$ 150m area. The area is divided into 13 $\times$ 10 grids, each of which is roughly 13m $\times$ 10m. Each grid is configured to have a node randomly deployed in it. In Section 7.4, 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 all the simulation experiments by default 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 in today's mote hardware, as reported in [Wu et al. 2008]. 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 five protocols: the MCRT protocol, MCRT-simple, and the three baselines. The first metric is *deadline miss ratio*, which is the fraction of data packets that miss their deadlines

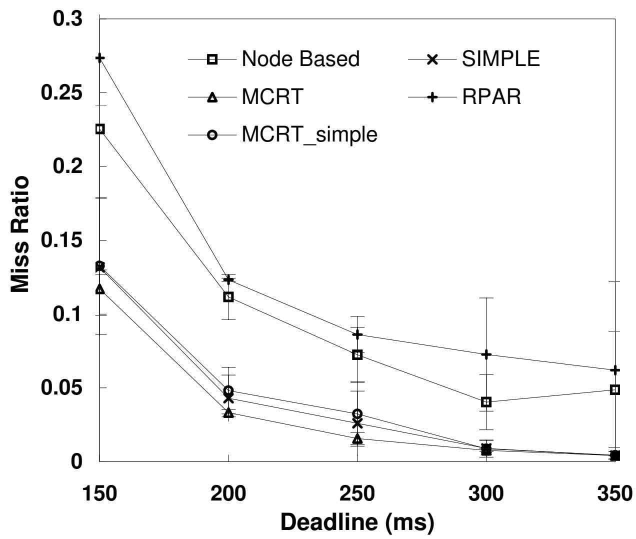


Fig. 8. Miss ratio when deadline is varied

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. To calculate the total transmission energy consumption, we use the typical power consumption data (of different power levels) from the data sheet of the CC2420 radio chip [Chipcon] used on Mica2, Tmote Sky, and other sensor motes. The transmission time of each packet and the corresponding transmission and receiving power consumption values are recorded in the simulation trace file and used to calculate the total transmission energy after each simulation. 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.

7.2 Different Transmission Deadlines

The first set of experiments evaluates the performance of the five protocols under different end-to-end transmission deadlines. Figure 8 shows the deadline miss ratios when the deadline varies from 150 ms to 350 ms. MCRT has the lowest miss ratio among all the 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 on 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 6, SIMPLE randomly picks nodes to form data flows without considering interferences and one-hop delay of each node. Compared with MCRT-simple, MCRT also has slightly better performance, because each node in

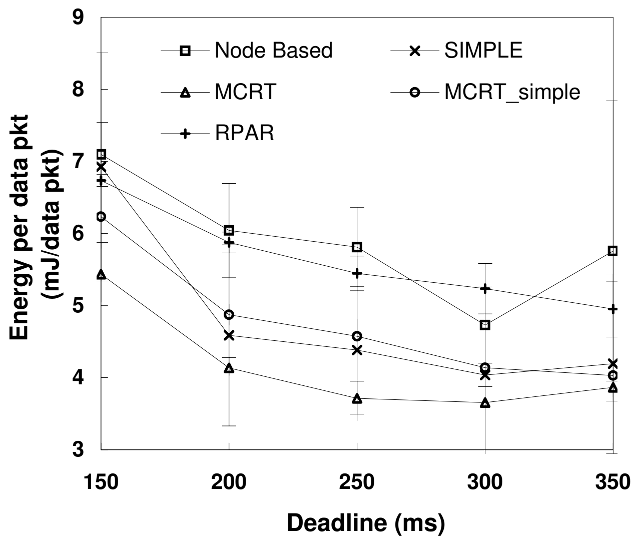


Fig. 9. Energy consumption when deadline is varied

MCRT has more neighbor nodes to choose for real-time forwarding, which leads to better real-time performance. MCRT-simple is slightly worse than SIMPLE because SIMPLE has two paths merged as a flow and thus can handle varying network conditions (*e.g.*, interferences) better than MCRT-simple, which relies on a single path. This demonstrates the importance of merging multiple paths in a flow.

Figure 9 shows the energy consumption of the five 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 8. As a result, more packets successfully meet their deadlines, which leads to improved energy efficiency.

7.3 Different Data Rates

This set of experiments studies the performance of the five protocols when the data rate of the three source nodes is increased from one packet per 5 seconds to one packet every second. Figure 10 shows that there is no clear evidence that data rate may significantly affect the miss ratios of the protocols. MCRT, SIMPLE and MCRT-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.

Figure 11 shows the energy consumption. MCRT, MCRT-simple, and SIMPLE have lower energy consumption, 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

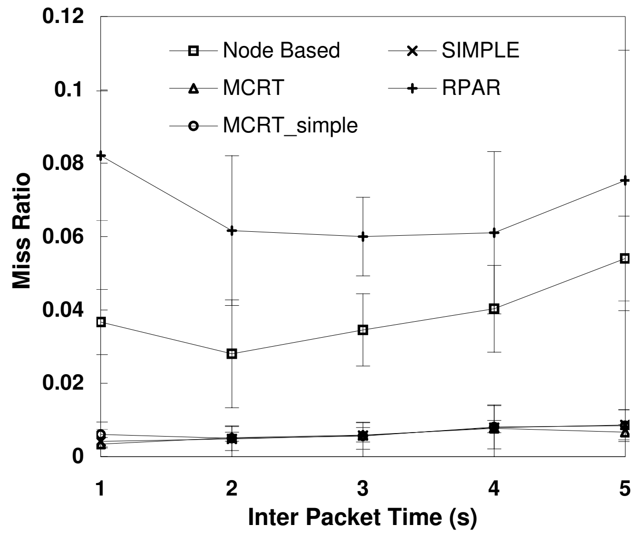


Fig. 10. Miss ratio when data rate is varied

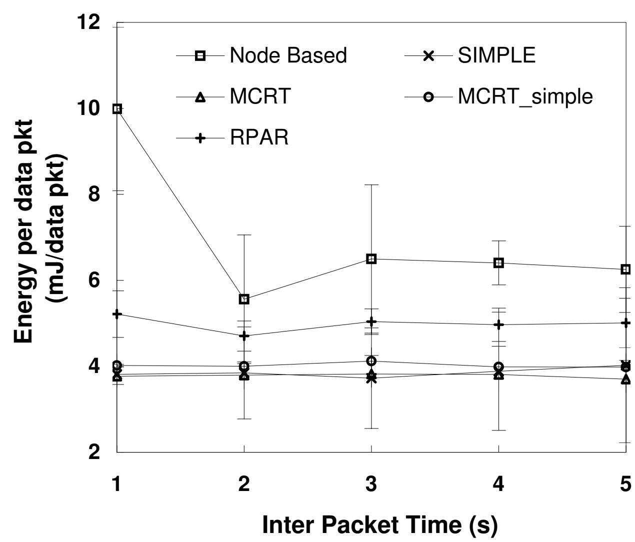


Fig. 11. Energy consumption when data rate is varied

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. MCRT-simple has slightly higher energy consumption than MCRT and SIMPLE, since it has a single path in each flow and thus has fewer choices for real-time forwarding, which results in higher transmission power.

7.4 Different Number of Data Flows

In this experiment, we vary the number of data flows in the network from 2 to 6. 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

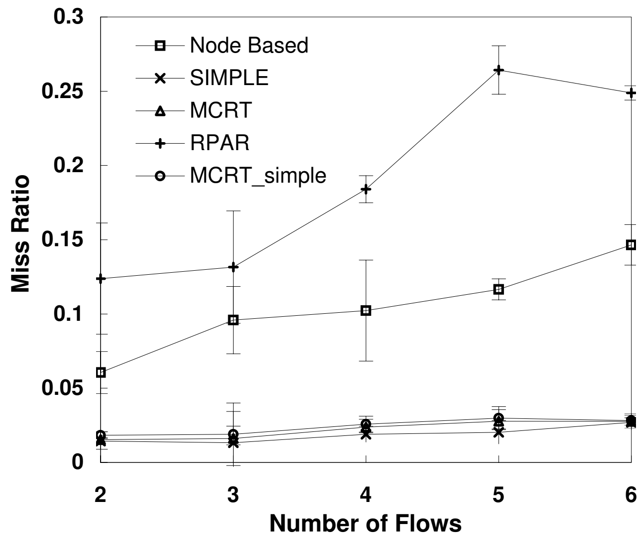


Fig. 12. Miss ratio when flow number is varied

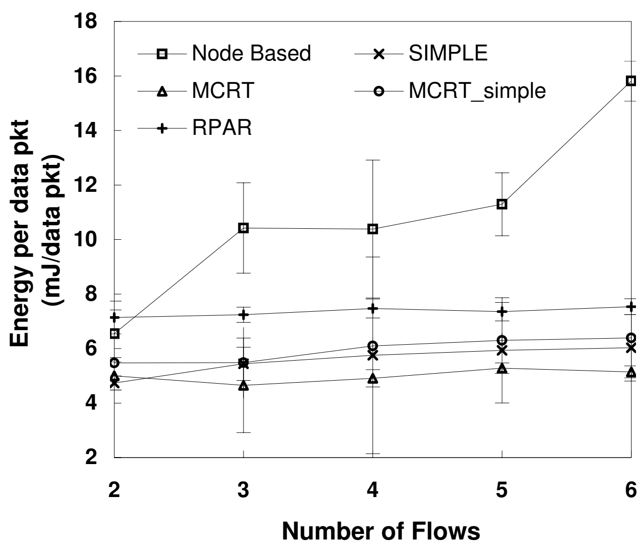


Fig. 13. Energy consumption when flow number is varied

the MCRT, SIMPLE and MCRT-simple protocols. For example, with four flows and three channels, two of the flows will share the same channel. Figure 12 shows that the deadline miss ratios of MCRT, SIMPLE, and MCRT-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, SIMPLE, and MCRT-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

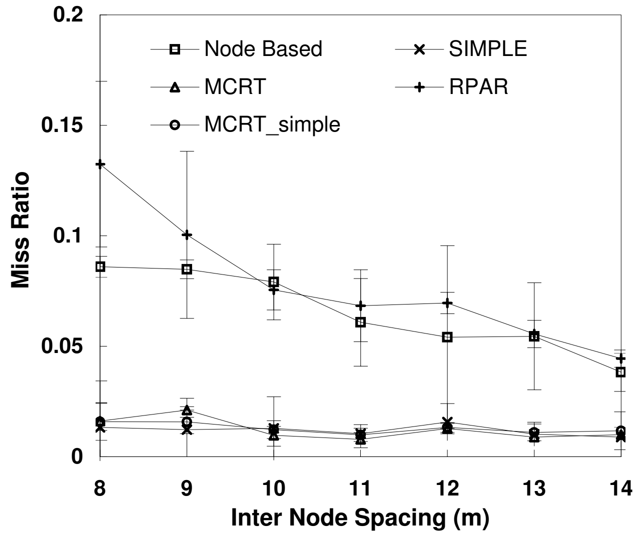


Fig. 14. Miss ratio when network density is varied

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.

As shown in Figure 13, MCRT, SIMPLE, and MCRT-simple have lower energy consumption. The reason is that these three protocols 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. MCRT also performs better than the MCRT-simple, because during the real-time forwarding phase of MCRT, there are more choices for the next hop, which leads to lower transmission power consumption. 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.

7.5 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 14 and 15 show the miss ratio and energy consumption for all the five 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 a packet to reach the destination is increased, making it hard to meet the deadline. MCRT, SIMPLE, and MCRT-simple are not significantly impacted by the varying network

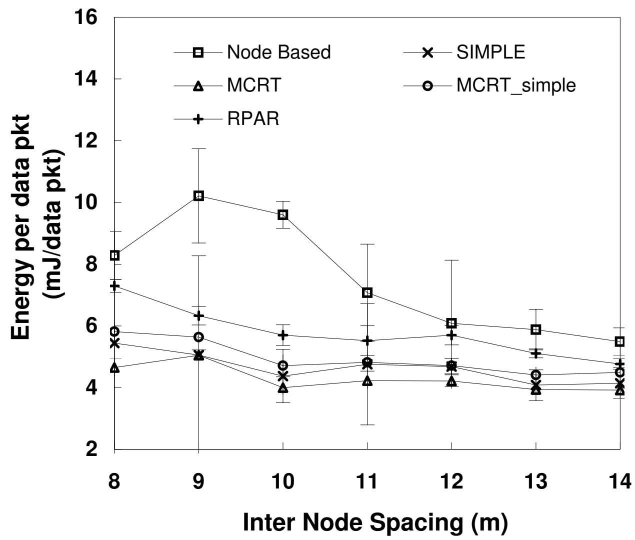


Fig. 15. Energy consumption when network density is varied

density because partitioning the network leads to fewer short-distance neighbors in each data flow. The energy consumption decreases for all the 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. MCRT-simple has higher energy consumption than MCRT, especially when the density is low. This is because with fewer nodes to choose for real-time forwarding, MCRT-simple has to use a higher transmission power level in order to meet the real-time performance requirement. The node-based protocol has the highest energy consumption because it uses more RR packets than other protocols.

7.6 Different Number of Channels

In this set of experiments, we vary the number of channels from 1 to 5, as the available number of orthogonal channels is limited on today's mote hardware. Since RPAR is a single-channel protocol, we do not evaluate RPAR in this set of experiments. Figure 16 shows the miss ratios of the other four protocols. MCRT, SIMPLE, and MCRT-simple can benefit from the increased number of channels by having reduced numbers of deadline misses, due to less channel contention and thus fewer packet retransmissions. When the number of channels is greater than 3, *i.e.*, the number of data flows in the network, these three protocols have no more performance improvements because those extra channels are not being used. The node-based protocol also benefits when the number of channels increases from 1 to 4. However, with 5 channels, the miss ratio starts to increase, because each node in the network has fewer neighbors working on the same channel at a certain time when more channels are available. When a node needs to broadcast an RR packet to find new neighbors, the node has to wait for a longer time to receive replies from qualified neighbors. Therefore, in the node-based protocol, when the number

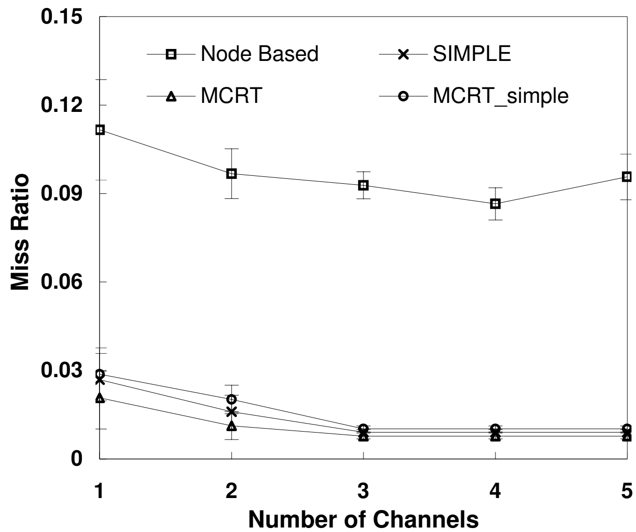


Fig. 16. Miss ratio when the number of channels is varied

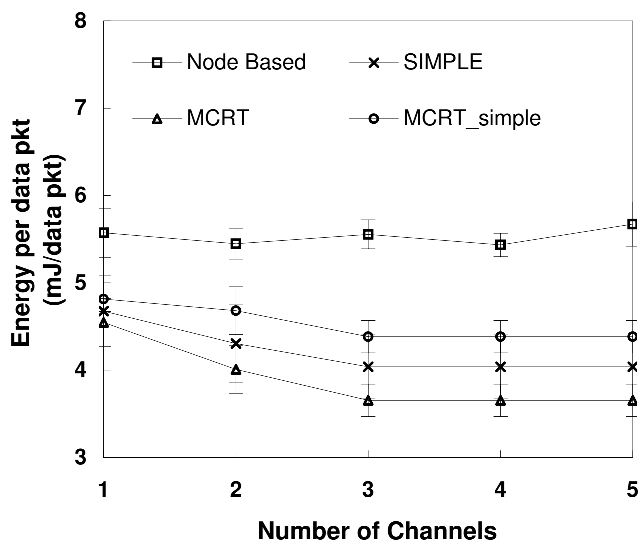
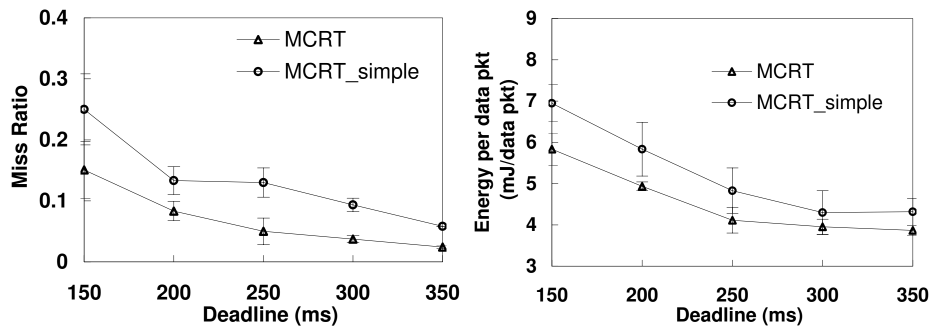


Fig. 17. Energy consumption when the number of channels is varied

of channels increases to a certain degree, the overhead caused by channel switching outweighs the benefits of having multiple channels, resulting higher end-to-end delays.

Figure 17 shows the energy consumption under different number of channels. MCRT, SIMPLE, and MCRT-simple achieve reduced energy consumption when the number of channels increases, due to fewer retransmissions caused by the channel contention among different data flows. The varying number of channels has no clear impact on the energy consumption of the node-based protocol, because the node-based protocol has more RR packets and channel switching with more channels, which compensate for the energy savings received from less channel contention.



(a) Miss ratio (b) Energy consumption

Fig. 18. Miss ratio and energy consumption of MCRT and MCRT-simple under interferences

This experiment demonstrates that MCRT outperforms the other three protocols in terms of real-time performance and energy consumption under different number of available channels.

7.7 MCRT vs. MCRT-simple under Interference

In the previous subsections, we have shown that MCRT has slightly lower energy consumption than MCRT-simple. This is because MCRT has more choices to find more power-efficient next-hop neighbors for real-time forwarding. In this subsection, we conduct another experiment that compares MCRT and MCRT-simple to examine their performance when the network has interfering concurrent transmissions. We use the 130-node topology to do this experiment. The network is configured to have 3 data flows to transmit on 3 different channels. As we introduce in Section 4.3, MCRT-simple only has a single path for each data flow in the network and is thus vulnerable to interferences. To evaluate its performance under interferences, we add 3 pairs of interfering nodes to the network, with each pair transmitting on each channel at a rate of 1 packet per second. To generate interference, we put each interfering pair close to the path used by MCRT-simple on the same channel. As a result, all the data flows in MCRT-simple are under interferences. We then perform the same experiment with MCRT. Since MCRT merges 3 paths in each data flow, it uses 6 more paths, 2 in each flow, in addition to the 3 paths used by MCRT-simple. Since the 6 additional flows are not as close to the interfering nodes, MCRT has alternative paths to avoid interferences.

Figures 18(a)(b) show that MCRT has more significant improvement than MCRT-simple, in terms of both miss ratio and energy consumption, when the network has interferences. On average, MCRT-simple's miss ratio is almost twice that of MCRT. MCRT-simple also consumes about 20% more energy. The reason is that the alternative paths used by MCRT provide more next-hop choices for real-time and power-efficient forwarding. As a result, even when a path is under interferences, each node can choose another path for packet forwarding. Without alternative paths, MCRT-simple can only transmit packets under interferences, which leads to more deadline misses and higher energy consumption.

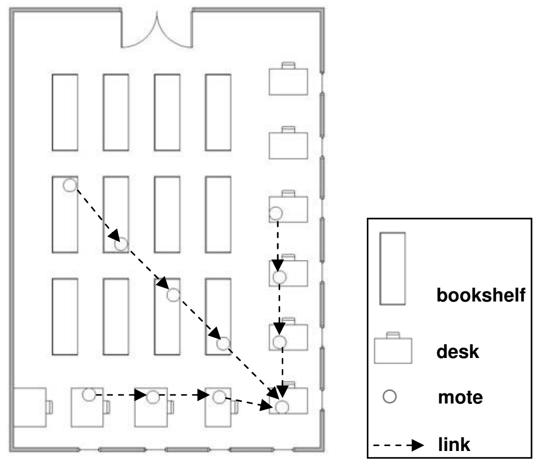
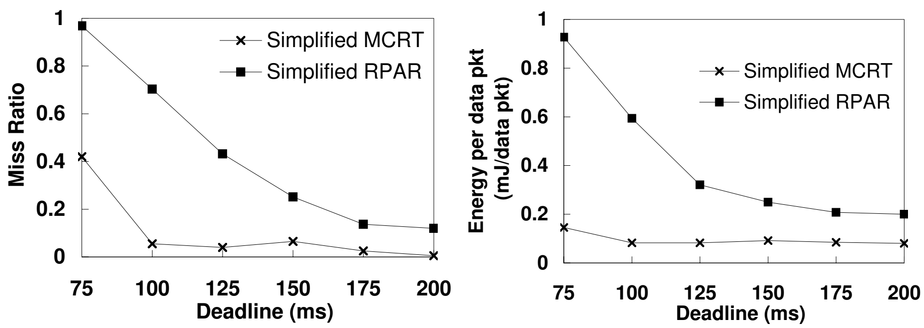


Fig. 19. Network topology used in our hardware experiments conducted in a university library.

7.8 Evaluation on a Hardware Testbed

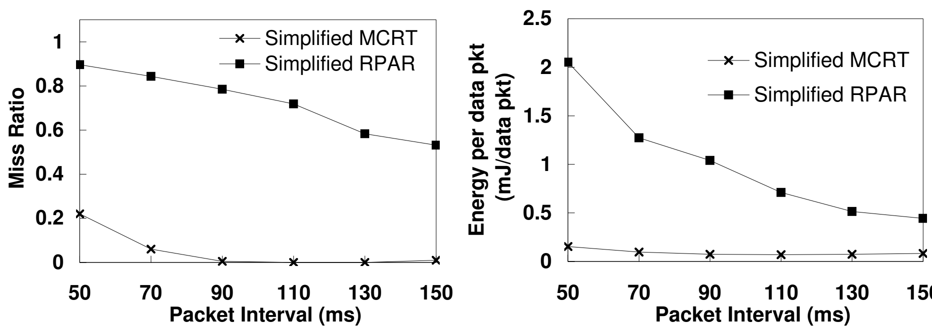
In this section, we present our empirical results on a 12-mote hardware testbed to demonstrate that utilizing multiple channels can lead to improved real-time performance and reduced energy consumption. Our testbed includes 12 Tmote Invent motes, forming 3 flows. Each flow has 4 motes. The data rate of each flow is one packet every 4 seconds. We establish the testbed in a university library using the topology shown in Figure 19. The destination of each flow is placed at the same place (at the bottom right corner) to emulate a base station with 3 radios. The distance between every two adjacent nodes in the two flows at the edge of the topology is 2m. Due to limited hardware devices and the space limitation of the experimental environment, we slightly modify both MCRT and RPAR such that each flow is configured to have a fixed routing path. We conduct transmission power adaptation as discussed in Section 5.2 to achieve energy efficiency and real-time guarantees for all the flows. This modification allows us to highlight the key difference between MCRT and RPAR, *i.e.*, the utilization of multiple channels. The modified MCRT and RPAR are referred to as *Simplified MCRT* and *Simplified RPAR*, respectively. In Simplified MCRT, three orthogonal channels are used for the 3 flows, while only one channel is used for all the 3 flows in Simplified RPAR.

In our experiments, we first evaluate Simplified MCRT and Simplified RPAR under different end-to-end deadlines. Figure 20(a) shows the packet miss ratios when the deadline is varied from 75ms to 200ms. Simplified MCRT outperforms Simplified RPAR significantly by effectively utilizing multiple channels to reduce channel contention and interferences among different data flows. When the desired deadline is 100ms, the miss ratio of Simplified MCRT is only 8%, while Simplified RPAR misses the deadline for more than 70% of packets. When the deadline is 75ms, almost 100% of packets miss the deadline under Simplified RPAR, while only 40% of packets have deadline misses under Simplified MCRT. The experiments demonstrate that utilizing multiple channels available on today's mote hardware can allow a WSN to achieve siginificantly improved real-time performance. Figure



(a) Miss ratio (b) Energy consumption

Fig. 20. Miss ratio and energy consumption under different deadlines on the hardware testbed



(a) Miss ratio (b) Energy consumption

Fig. 21. Miss ratio and energy consumption under different packet rates on the hardware testbed

20(b) shows the results of energy consumption under the two schemes. When the deadeline is 75ms, the energy consumption of Simplified RPAR is more than 5 times that of Simplifed MCRT. When the deadline increases from 75ms to 200ms, both the two schemes achieve more energy savings. However, even when the deadline is 200ms, Simplified RPAR still consumes more than twice energy than Simplified MCRT does. This figure shows that utilizing multiple channels also helps to achieve less energy consumption in addition to better real-time performance.

We then test Simplified MCRT and Simplified RPAR under different packet rates, when the packet interval time increases from 50ms to 150ms. The deadline used for this set of experiments is 200ms. Figure 21(a) shows that Simplified MCRT has significantly better real-time performance than Simplified RPAR. When the packet interval is longer than 90ms, almost 100% of packets can meet the deadline under Simplified MCRT, while Simplified RPAR still has a miss ratio of 58% to 70%. Figure 21(b) shows the results of energy consumption of Simplified MCRT is much lower than that of Simplified RPAR. For example, Simplified RPAR consumes almost ten times higher energy than Simplified MCRT when the packet interval is 50ms. Those results confirm what we have observed from Figure 20, *i.e.*, utilizing multiple channels can achieve improved real-time performance and energy efficiency in WSNs.

8. CONCLUSION

Multi-channel communications have recently shown great promise to improve network throughput and reduce packet loss for wireless sensor networks. However, existing research does not 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 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 proven 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 and empirical results on real hardware testbed of Tmote nodes both 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

ns-2 Network Simulator. http://nsnam.isi.edu/nsnam/index.php/Main_Page.

Ahn, G.-S., Campbell, A., Veres, A., and Sun, L.-H. 2002. Swan: service differentiation in stateless wireless ad hoc networks. In *INFOCOM '02: Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol. 2. 457–466 vol.2.

Caccamo, M., Zhang, L., Sha, L., and Buttazzo, G. 2002. An implicit prioritized access protocol for wireless sensor networks. In *RTSS '02: Proceedings of the 23rd IEEE International Real-Time Systems Symposium*. 39–48.

Chang, J.-H. and Tassiulas, L. 2000. Energy conserving routing in wireless ad-hoc networks. In *INFOCOM '00: Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol. 1. 22–31 vol.1.

Cheng, C., Kumar, S. P. R., and Garcia-Luna-Aceves, J. J. 1990. A distributed algorithm for finding k disjoint paths of minimum total length. In *Allerton '90: Proceedings of the Annual Allerton Conference on Communication, Control, and Computing*.

Chipara, O., He, Z., Xing, G., Chen, Q., Wang, X., Lu, C., Stankovic, J., and Abdelzaher, T. 2006. Real-time power-aware routing in sensor networks. In *IWQoS '06: Proceedings of the 14th IEEE International Workshop on Quality of Service*. 83–92.

Chipara, O., Lu, C., and Stankovic, J. 2006. Dynamic conflict-free query scheduling for wireless sensor networks. In *ICNP '06: Proceedings of the 14th IEEE International Conference on Network Protocols*. 321–331.

Chipcon. CC2420 2.4 GHz IEEE 802.15.4/ZigBee-ready RF Transceiver. http://www.chipcon.com.

Doshi, S., Bhandare, S., and Brown, T. X. 2002. An on-demand minimum energy routing protocol for a wireless ad hoc network. *SIGMOBILE Mob. Comput. Commun. Rev. 6,* 3.

Felemban, E., Lee, C.-G., and Ekici, E. 2006. Mmspeed: multipath multi-speed protocol for qos guarantee of reliability and. timeliness in wireless sensor networks. *IEEE Transactions on Mobile Computing 5,* 6 (June), 738–754.

Garey, M. R. and Johnson, D. S. 1979. *Computer and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

Gupta, P. and Kumar, P. R. 2000. The capacity of wireless networks. *IEEE Transactions on on Information Theory 46,* 2.

He, T., Stankovic, J., Lu, C., and Abdelzaher, T. 2003. Speed: a stateless protocol for real-time communication in sensor networks. In *ICDCS '03: Proceedings of the 23rd International Conference on Distributed Computing Systems*. 46–55.

Jeong, J., Culler, D. E., and Oh, J.-H. 2005. Empirical analysis of transmission power control algorithms for wireless sensor networks. Tech. Rep. UCB/EECS-2005-16, EECS Department, University of California, Berkeley. Nov.

Jurcik, P., Severino, R., Koubaa, A., Alves, M., and Tovar, E. 2010. Real-time communications over cluster-tree sensor networks with mobile sink behaviour. In *RTCSA '08: Proceedings of the 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*.

Karenos, K. and Kalogeraki, V. 2006. Real-time traffic management in sensor networks. In *RTSS '06: Proceedings of the 27th IEEE International Real-Time Systems Symposium*. 422–434.

Kim, Y., Shin, H., and Cha, H. 2008. Y-MAC: An energy-efficient multi-channel mac protocol for dense wireless sensor networks. In *IPSN '08: Proceedings of the 7th international conference on Information processing in sensor networks*.

Kyasanur, P. and Vaidya, N. H. 2005. Capacity of multi-channel wireless networks: impact of number of channels and interfaces. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*. ACM, New York, NY, USA, 43–57.

Le, H. K., Henriksson, D., and Abdelzaher, T. 2007. A control theory approach to throughput optimization in multi-channel collection sensor networks. In *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*.

Le, H. K., Henriksson, D., and Abdelzaher, T. 2008. A practical multi-channel media access control protocol for wireless sensor networks. In *IPSN '08: Proceedings of the 7th international conference on Information processing in sensor networks*.

Li, L., Halpern, J. Y., Bahl, P., Wang, Y.-M., and Wattenhofer, R. 2001. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *PODC '01: Proceedings of the 20th annual ACM symposium on Principles of distributed computing*. ACM, New York, NY, USA, 264–273.

Li, N., Hou, J., and Sha, L. 2003. Design and analysis of an mst-based topology control algorithm. In *INFOCOM 2003. Procceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol. 3. 1702–1712.

Li, Q., Aslam, J., and Rus, D. 2001. Online power-aware routing in wireless ad-hoc networks. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*. ACM, New York, NY, USA, 97–107.

Lin, S., Zhang, J., Zhou, G., Gu, L., Stankovic, J. A., and He, T. 2006. Atpc: adaptive transmission power control for wireless sensor networks. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM, New York, NY, USA, 223–236.

Lu, C., Blum, B., Abdelzaher, T., Stankovic, J., and He, T. 2002. Rap: a real-time communication architecture for large-scale wireless sensor networks. In *RTAS '02: Procceedings of the 18th IEEE Real-Time and Embedded Technology and Applications Symposium*. 55–66.

Polastre, J., Hill, J., and Culler, D. 2004. Versatile low power media access for wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, New York, NY, USA, 95–107.

Ramanathan, R. and Rosales-Hain, R. 2000. Topology control of multihop wireless networks using transmit power adjustment. In *INFOCOM '00: Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol. 2. 404–413 vol.2.

Ronen, D. and Perl, Y. 1984. Heuristics for finding a maximum number of disjoint bounded paths. *Networks 14*, 531–544.

Sankar, A. and Liu, Z. 2004. Maximum lifetime routing in wireless ad-hoc networks. In *INFOCOM '04: Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol. 2. 1089–1097 vol.2.

Santi, P. 2003. Topology control in wireless ad hoc and sensor networks. Tech. Rep. IIT-TR-04, Istituto di Informatica e Telematica, Pisa - Italy.

Selavo, L., Wood, A. D., Cao, Q., Sookoor, T., Liu, H., Srinivasan, A., Wu, Y., Kang, W., Stankovic, J. A., Young, D., and Porter, J. 2007. Luster: wireless sensor network for environmental research. In *SenSys*.

Sidhu, D., Nair, R., and Abdallah, S. 1991. Finding disjoint paths in networks. *SIGCOMM Comput. Commun. Rev. 21,* 4, 43–51.

Singh, S., Woo, M., and Raghavendra, C. S. 1998. Power-aware routing in mobile ad hoc networks. In *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*. ACM, New York, NY, USA, 181–190.

Son, D., Krishnamachari, B., and Heidemann, J. 2004. Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks. In *SECON '04: Proceedings of the 1st IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*. 289–298.

Stankovic, J., Abdelzaher, T., Lu, C., Sha, L., and Hou, J. 2003. Real-time communication and coordination in embedded sensor networks. *Proceedings of the IEEE 91,* 7 (July), 1002–1022.

Vedantham, R., Kakumanu, S., Lakshmanan, S., and Sivakumar, R. 2006. Component based channel assignment in single radio, multi-channel ad hoc networks. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*. ACM, New York, NY, USA, 378–389.

Wang, X., Wang, X., Fu, X., Xing, G., and Jha, N. 2009. Flow-based real-time communication in multi-channel wireless sensor networks. In *EWSN '09: Proceedings of the 6th European Conference on Wireless Sensor Networks*.

Wang, X., Wang, X., Xing, G., and Yao, Y. 2010a. Dynamic duty cycle control for end-to-end delay guarantees in wireless sensor networks. In *IWQoS '10: Proceedings of the 18th IEEE International Workshop on Quality of Service*.

Wang, X., Wang, X., Xing, G., and Yao, Y. 2010b. Exploiting overlapping channels for minimum power configuration in real-time sensor networks. In *EWSN '10: Proceedings of the 7th European Conference on Wireless Sensor Networks*.

Woo, A., Tong, T., and Culler, D. 2003. Taming the underlying challenges of reliable multihop routing in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, New York, NY, USA, 14–27.

Wu, Y., Stankovic, J., He, T., and Lin, S. 2008. Realistic and efficient multi-channel communications in wireless sensor networks. In *INFOCOM '08: Proceedings of the 27th Annual Joint Conference of the IEEE Computer and Communications Societies*. 1193–1201.

Zhang, J., Zhou, G., Huang, C., Son, S., and Stankovic, J. 2007. Tmmac: An energy efficient multi-channel mac protocol for ad hoc networks. In *ICC '07: Proceedings of the IEEE International Conference on Communications*. 3554–3561.

Zhao, J. and Govindan, R. 2003. Understanding packet delivery performance in dense wireless sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, New York, NY, USA, 1–13.

Zhou, G., Huang, C., Yan, T., He, T., Stankovic, J. A., and Abdelzaher, T. F. 2006. Mmsn: Multi-frequency media access control for wireless sensor networks. In *INFOCOM '06: Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications*.

Zuniga, M. and Krishnamachari, B. 2004. Analyzing the transitional region in low power wireless links. In *SECON '04: Proceedings of the 1st IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*.