Efficient Multi-Channel Communications in Wireless Sensor Networks

YAFENG WU, University of Virginia
KIN SUM LIU, Stony Brook University
JOHN A. STANKOVIC, University of Virginia
TIAN HE, University of Minnesota
SHAN LIN, Stony Brook University

This paper demonstrates how to use multiple channels to improve communication performance in Wireless Sensor Networks (WSNs). We first investigate multi-channel realities in WSNs through intensive empirical experiments with Micaz motes. Our study shows that current multi-channel protocols are not suitable for WSNs, because of the small number of available channels and unavoidable time errors found in real networks. With these observations, we propose a novel tree-based multi-channel scheme for data collection applications, which allocates channels to disjoint trees and exploits parallel transmissions among trees. In order to minimize interference within trees, we define a new channel assignment problem which is proven NP-complete. Then we propose a greedy channel allocation algorithm which outperforms other schemes in dense networks with a small number of channels. We implement our protocol, called TMCP, in a real testbed. To adjust to networks with link quality heterogeneity, an extension of TMCP is also proposed. Through both simulation and real experiments, we show that TMCP can significantly improve network throughput and reduce packet losses. More importantly, evaluation results show that TMCP better accommodates multi-channel realities found in WSNs than other multi-channel protocols.

CCS Concepts: •**Networks** → *Network protocol design;*

General Terms: Algorithms, Design, Performance

Additional Key Words and Phrases: Wireless sensor networks, multi-channel, channel allocation, interference

ACM Reference Format:
Yafeng Wu, Kin Sum Liu, John A. Stankovic, Tian He, and Shan Lin, 2015. Efficient Multi-Channel Communications in Wireless Sensor Networks. *ACM Trans. Sensor Netw.* 9, 4, Article 39 (March 2016), 24 pages.
DOI: 0000001.0000001

1. INTRODUCTION

As an emerging technology, Wireless Sensor Networks (WSNs) have a wide range of potential applications, including environmental monitoring, smart buildings, medical care, and many other industry and military applications. A large number of protocols have been proposed for the MAC, routing and transport layers. However with a single channel, WSNs cannot provide reliable and timely communication with high data rate requirements because of radio collisions and limited bandwidth. For example, in the Ears on the ground project [Zhang et al. 2006], the network cannot transmit multiple

This work is supported by the National Science Foundation, under grant IIS-1460370, grant CNS-1463722, grant CNS-1536086, grant CNS-1444021, grant CNS-1239483 and grant CNS-1138963.
Author's addresses: Y. Wu and J. A. Stankovic, Computer Science Department, University of Virginia; K.S. Liu, Computer Science Department, Stony Brook University; T. He, Computer Science Department, University of Minnesota; S. Lin, Electrical & Computer Engineering Department, Stony Brook University.

DOI: 0000001.0000001

acoustic streams to the sink. On the other hand, current WSN hardware, such as Micaz and Telos that use the CC2420 radio, already provide multiple frequencies. So it is imperative to design multi-channel based communication protocols in WSNs to improve network throughput and provide reliable and timely communication services.

Some MAC layer multi-channel protocols are proposed to improve network performance in WSNs. These protocols typically assign different channels to two-hop neighbors to avoid potential interferences, and also design sophisticated MAC schemes to coordinate channel switching and transmissions among nodes. For example, MMSN[Zhou et al. 2006], TMMAC[Zhang et al. 2007] and MCMAC[Chen et al. 2006] are such protocols designed for WSNs. Simulation results show that they can significantly improve network throughputs over MAC protocols using a single channel. In this paper, we focus on how to efficiently use multiple channels in WSNs to improve communication performance. Different from previous works, we first investigate multichannel realities found in WSNs through a set of empirical experiments. Next, we propose a Tree-based Multi-Channel Protocol (TMCP) for data collection applications, and study a new channel assignment problem. The main contributions of this work are:

— This paper presents an empirical study of multi-channel realities through intensive experiments, and analyzes the practical issues of current multi-channel protocols. We show that these protocols are not suitable for general WSNs because of the small number of available channels and unavoidable time synchronization errors found in practice.
— TMCP partitions the whole network into multiple subtrees, allocates different channels to each subtree, and then forwards each data flow only along its corresponding subtree. This scheme can work well with a small number of channels and has a very simple transmission scheme without the need for synchronization at nodes, which makes it suitable for practical WSNs.
— We define and solve a new problem of how to partition networks into subtrees with minimizing the intra-tree interferences. We prove the complexity of the problem and propose a greedy solution algorithm. Evaluation results show that it reduces interference in dense networks over other schemes.
— We implement TMCP in a real test bed and evaluate its performance through both simulation and real experiments. It is shown that TMCP can greatly improve network throughput, while maintaining high packet delivery ratios and low delivery latency. Furthermore, we show that it outperforms other multi-channel protocols.
— We extend the network model to incorporate the link quality and removal of the edges. A new pruning algorithm is proposed to meet the reliability requirement of data collection. The evaluation indicates that the pruning procedure is important to maintains high reliability in networks with diverse link qualities.

The rest of paper is organized as follows. In section 2, we explain related work. In section 3, we present empirical results from experiments that investigate multi-channel realities found in WSNs. The design of TMCP is presented in section 4. In section 5, we describe the related channel assign problem and present a greedy algorithm with its evaluation. In section 6, we evaluate the performance of TMCP with simulation and real experiments. In section 7, we present the extended TMCP model. Finally, in section 8, we present conclusions.

2. RELATED WORK

In the general wireless network, a significant number of multi-channel protocols have been proposed, such as multi-channel MAC protocols [So and Vaidya 2004] [Li et al.

2003] [Tzamaloukas and Garcia-Luna-Aceves 2001] [Bahl et al. 2004]. These protocols either require multiple radio transceivers at each node, or need certain kinds of control messages for channel negotiation. However, they are not suitable for WSN applications. First, each sensor device is usually equipped with a single radio transceiver, which cannot function on different frequencies simultaneously. Second, the network bandwidth in WSNs is very limited and the data packet size is very small. Therefore, channel negotiation packets can not be ignored as small overhead.

In the literature, MMSN [Zhou et al. 2006], TMMAC [Zhang et al. 2007] and MC-MAC [Chen et al. 2006] are three multi-channel MAC protocols designed especially for WSNs. They all try to assign different channels to nodes in a two-hop neighborhood to avoid potential interferences. We call these node-based multi-channel protocols. Simulation results show that they improve performance in WSNs compared with single channel protocols. However, with node-based channel assignment schemes, a node typically has a different channel from its downstream and upstream nodes. Within a multi-hop flow, nodes have to switch channels to receive and forward packets which can cause very frequent channel switching and potential packet losses. In order to avoid such packet losses, node-based protocols use some negotiation or scheduling schemes to coordinate channel switching and transmissions among nodes with different channels. For instance, all three protocols mentioned above use time slots to coordinate transmissions. They face practical issues in real WSNs, including: 1) a large number of orthogonal channels are needed for channel assignment in dense networks; 2) they require precise time synchronization at nodes, 3) channel switching delay and scheduling overhead cannot be ignored because of frequent channel switching, especially for high data rate traffic, and 4) these protocols are typically complex, which require more resources at motes. Our paper studies these practical issues through empirical experiments with Micaz motes and shows that node-based protocols may not be suitable for WSNs in practice. Therefore, two different channel assignment methods are proposed. A component-based protocol is presented in [Vedantham et al. 2006] which assigns channels to connected components in wireless ad hoc network, and in [Le et al. 2007], nodes dynamically select channels based on a control theory approach to achieve load balance among channels. While these solutions have a similar favor in channel assignment to us, our scheme focuses on how to use multi-channels to construct the optimal topology with low interferences and optimize throughputs in practical WSNs. To address these issues, a preliminary version of our work was published in IEEE INFOCOM 2008 [Wu et al. 2008].

Recently, there are more works on solving different scenarios in multi-channel WSN. To reduce the power consumption in WSNs, [Xing et al. 2009] [Gong et al. 2010] propose different strategies. The former attempts to use partially overlapping channels in a low-power wireless networks. The latter makes use of cooperative multiple-input multiple-output technique to enhance energy efficiency. [Luo et al. 2011] focus on finding a maximum lifetime tree for nodes with different energy constraint as a semi-matching problem. To solve control channel saturation and triple hidden terminal problems, [Li et al. 2010] proposes utilizing an receiver-initiated MAC protocol with duty cycling. Besides, the channel assignment problem also attracts and leverages different disciplines. For example, a Game Theory formulation of the channel assignment problem is presented in [Yu et al. 2010] and the Nash Equilibrium of the game is analyzed.

There is also a rich literature about system and technology in WSNs. From an application perspective, the application specific goals cannot be achieved without an understanding of characteristics in the underlying WSNs,. So the realities of the physical environment and dynamic nature of the physical world requires the application to work with a high level of adaptivity. [Lin et al. 2006], and [Liu et al. 2014] both take

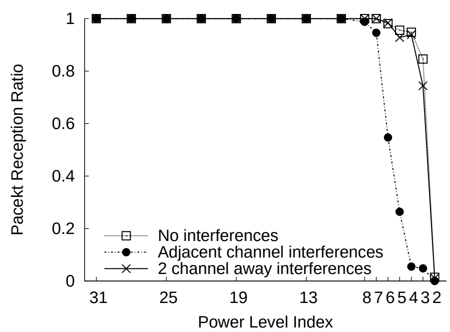


Fig. 1. Transmission power level index vs. Packet reception ratio

an account of an unreliable network in terms of connectivity and capability to maintain quality of service. This robustness is often critical in the life-saving and medical domain in which WSNs plays an increasingly larger role [Zhou et al. 2008], and [Asare et al. 2012].

3. EXPERIMENTS ON MULTI-CHANNEL REALITY

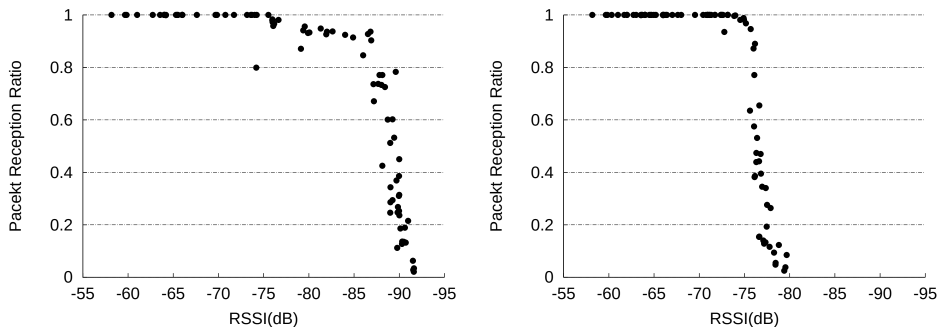
In order to design good protocols, we need to better understand multi-channel realities in WSNs. In this section, we first conduct a set of empirical experiments to investigate multi-channel interference properties of Micaz hardware, including adjacent channel interferences and interferences with 802.11 networks. These properties are well studied in wireless ad hoc networks [Zhou et al. 2006] [Petrova et al. 2007] [Mishra et al. 2006], but there is a lack of empirical studies in WSNs. Then we measure the performance of node-based multi-channel schemes on a single path and investigate the impact of time synchronization errors. With these experimental results, our analysis shows that current node-based schemes are not suitable for dense and large WSNs, as well as for applications with high data rates.

3.1. Number of Available Orthogonal Channels

An important parameter for multi-channel designs is the number of channels which can actually be used in WSNs. The CC2420 radio chip [Instruments 2006] used in Micaz provides 16 non-overlapping channels, with 5MHz spacing. However, not all channels can be used in a single sensor network to provide parallel transmissions because of close channel interferences and interferences caused by 802.11 networks.

3.1.1. Non-orthogonal Channel Interferences. Non-orthogonal channel interferences are well studied in general wireless networks [Mishra et al. 2006]. For WSNs hardware, the CC2420 chip specification [Instruments 2006] indicates that the adjacent channel rejection is 45/30 dB. In the following, we present experiments to study its real impact on the performance of multi-channel WSNs.

In the first experiment, we place three Micaz motes in a line, with one transmitter, one receiver and one jammer. The jammers transmission is synchronized with the transmitter to generate interferences. Both the transmitter and the receiver use channel 11. While the transmitter changes its transmission power, we measure packet reception ratios of the receiver in three cases, without the jammer interfering, with the jammer interfering at channel 12 (the adjacent channel interference) and at channel 13 (2 channels away). The results of this experiment are illustrated in Figure 1. We can see that without interferences, the receiver can maintain an above 90% packet reception ratio until the transmitter uses power levels lower than 3. However, with



(a) RSSI vs. Reception ratio without interferences

(b) RSSI vs. Reception ratio with adjacent channel interferences

Fig. 2. RSSI and Packet reception ratios under two types of interference

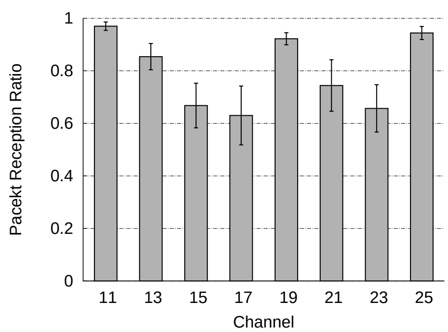


Fig. 3. Packet reception ratios of different channels

adjacent channel interferences, the packet reception ratio decreases to 50% when the transmission power level is below 7, which clearly shows that adjacent channel interferences greatly impact radio reception and they are not negligible. On the other hand, the curve of the two channel away interferences is very close to the one without interferences, which implies that the impact of two channel away interferences is small. We run the same experiments with other channels, and they show the similar result.

In order to further investigate the impact of adjacent channel interferences, another set of experiments is conducted to determine the relation of Received Signal Strength Indication (RSSI) threshold and different channel interferences. In these experiments, we fix the positions of the receiver and the jammer, which are 2 feet apart, and the transmitter moves along the line in different places. Experiments are run in two cases, with and without adjacent channel interferences. Results are shown in Figure 2(a) and (b), where each data point presents a pair of RSSI and packet reception ratios. We can see that the RSSI threshold for above 90% packet reception ratio is around -87dB without interferences, while that threshold increases to -77dB with adjacent channel interferences. Transmission links with RSSI between -77dB and -87dB become unreliable when adjacent channel interference occurs. The existence of adjacent channel interference can cause unexpected collisions and packet losses, and the safe way is to only use non-adjacent channels in multi-channel protocols.

3.1.2. Interferences with 802.11 networks. Another factor that affects the number of available channels is the interference with 802.11 networks. 802.15.4 specification shows

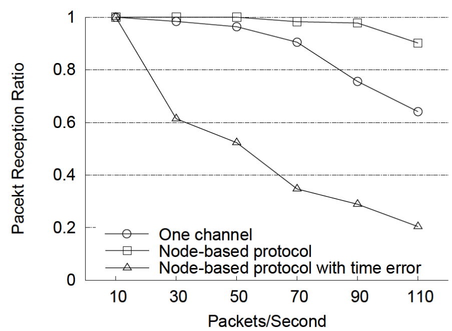


Fig. 4. Packet reception ratio vs. Source Data rate

that one 802.11 channel can potentially collide with four 802.15.4 channels. This problem is also studied in [Zhou et al. 2006] [Petrova et al. 2007]. Here, we also present a simple experiment to show how 802.11 networks impact channels in WSNs. We put 8 pairs of Micaz nodes closely together in a department office, where multiple 802.11 networks exist. Each pair uses one unique channel to transmit packets within the pair. All 8 channels are orthogonal with each other. We run the experiment several times and measure the average packet reception ratios. Results are shown in Figure 3, with the standard deviation of each data. We can see only 3 channels (11,19,25) have good link qualities (reception ratios above 90%), and link qualities of the other 4 channels are poor (reception ratios around 60%) and unstable (large standard deviations). This experiment shows that multi-channel protocols must have capabilities to work well with a small number of available channels. Otherwise their performance may greatly degrade in such indoor scenarios.

3.2. Impact of Time Synchronization Errors

Another crucial factor which can greatly impact the performance of current node-based protocols is time synchronization error. As mentioned before, current node-based schemes need precise time synchronization at each node to coordinate transmissions and channel switching. But, low-power Micaz motes cannot provide very high time accuracy. The clock drift of a Micaz is known to be 40ppm (part-per-million), which means that the clock drift can be $40\mu s$ after 1 second. In order to investigate the impact of time errors, we conduct a set of experiments on Micaz motes. We put 5 Micaz motes on a line. The first node transmits packets to the final node one-by-one hop. Each node is assigned a unique channel. At the beginning, all nodes are synchronized.

First, we use a simple time-slot based scheme as a prototype of node-based protocols. In this scheme, a time period of 10ms is divided into two time slots. In the first time slot, nodes in odd positions switch their channel and send packets to their next nodes, while nodes in even positions stay at their own channels and receive packets, and vice versa in the second slot. With different data rates at the source, we measure the end-to-end performance in terms of packet reception ratios. After these experiments, we wait for 10 minutes, do the same experiment again without re-synchronization and measure the second set of results, which present the performance of the node-based protocol with time errors. Finally, we modify all nodes to use a single channel, and employ the standard CSMA protocol to transmit packets. These results are illustrated in Figure 4. It can be seen that without time errors, the node-based scheme always has higher packet reception ratios than the single channel scheme. The saturated packet rate (packet reception ratio is above 90%) of the two schemes are around 90msg/sec and

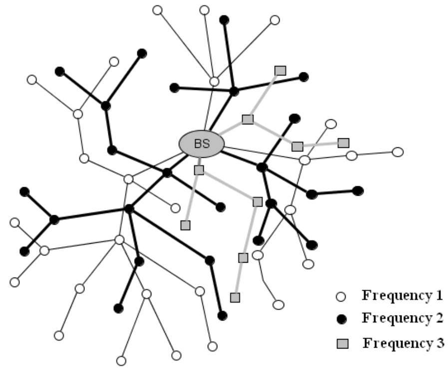


Fig. 5. The conceptual design of TMCP

50msg/sec, respectively. On the other hand, with time errors, the node-based protocol has very low packet reception ratios. The saturated packet rate is around 10msg/sec, which means that the protocol can only support a low data rate for end-to-end traffic (around 3kb/s) without synchronization. This experiment confirms that node-based protocols can improve communication performance, but have large performance degradation with time errors. Furthermore, this degradation can be amplified in large and dense networks, with longer paths and more complex coordination schemes. It also shows that node-based protocols can not provide reliable and stable communication services for high data rate traffic. One possible solution is to perform the time synchronization operation periodically. For the above experiments, nodes need to be synchronized more frequently than every 10 minutes to guarantee the performance. However, time synchronization protocols in WSNs can be costly, consuming extra bandwidth and power, which makes frequently re-synchronizing impractical, especially for high data rate applications or for dense and larger networks.

4. A TREE BASED MULTI-CHANNEL PROTOCOL

Every multi-channel protocol for WSNs has two main components, channel assignment and transmission coordination. As shown in section 3, the multi-channel realities of WSNs affect current node-based multi-channel protocols in both components. The small number of available orthogonal channels cannot satisfy the requirement of node-based channel assignment, especially for dense networks. Unavoidable time errors impact transmission coordination among nodes with different channels, especially for applications with high data rates. In order to overcome these two problems in practical networks, we believe that new multi-channel schemes should first use a coarse-grained channel assignment strategy, instead of node-level assignment, and secondly, it should try to avoid complex coordination schemes by reducing channel switching and communication among nodes with different channels.

On the other side, we also notice that sensor networks have a dominant traffic pattern, the data collection traffic, where multiple information flows generated at sensor nodes converge to the base-station. Currently, most data collection schemes build some tree structure connecting the base station and nodes, and then forward packets along the tree. However, with a single channel, transmission collisions within the tree and flow congestions at nodes greatly decrease the network performance.

Based on above observations, we propose a Tree-based Multi-Channel Protocol (TMCP) for data collection applications in WSNs. The idea of using multi-channel is to firstly partition the whole network into multiple vertex-disjoint subtrees all rooted at the base station and allocate different channels to each subtree, and then forward each flow only along its corresponding subtree, shown in Figure 5. The superiority of TMCP is two-fold. First, for practical concerns, with a coarse-grained channel assignment, it requires much fewer channels than node-based protocols. Also since every flow is forwarded in one subtree with one channel, we do not need a sophisticated channel coordination scheme, which implies that TMCP can work without the need for time synchronization. Secondly, for performance concerns, because it assigns different channels among subtrees, it can increase network throughput and reduce packet losses by eliminating inter-tree interferences and exploiting spatial reuses of parallel transmissions among subtrees.

TMCP has three components, Channel Detection (CD), Channel Assignment (CA), and Data Communication (DC). The CD module finds available orthogonal channels which can be used in the current environment. To do this, two motes are used to sample the link quality of each channel by transmitting packets to each other, and then among all channels with good link qualities, non-adjacent channels are selected. At this point we have k channels.

Given k orthogonal channels, the CA module partitions the whole network into k subtrees and assigns one unique channel to each subtree. This is the key part of TMCP. The goal of partitioning is to decrease potential interference as much as possible. We can see that after partitioning, interferences in the original network can be divided into two categories, one is the interference among different trees, called intertree interference, which is eliminated by assigning different orthogonal channels to each subtree, and the other is the potential interference among nodes within a tree, called the intra-tree interference. Because we assign the same channel to all nodes of one subtree, the intra-tree interference can not be avoided in our scheme and becomes the main performance bottleneck. So, the goal of partitioning is to divide networks into subtrees, each of which has lower intra-tree interferences. In next section, we will further study this problem.

After assigning channels, the DC component manages the data collection through each subtree. When a node wants to send information to the base station, it just uploads packets along the subtree it belongs to. Here, we assume that the base station is equipped with multiple radio transceivers, each of which works on one different channel. We can see that because of the tree-based channel assignment strategy, DC is very simple without the need of time synchronization. Also, the base station can use this network structure to perform data dissemination. When the base station wants to send commands or update the code, it can send out packets through all transceivers, and then packets will go through every subtree and reach all nodes in networks.

5. MINIMUM INTERFERENCE CHANNEL ASSIGNMENT PROBLEM

TMCP uses a new tree-based channel assignment scheme. As mentioned earlier, the goal of this assignment scheme is to minimize intra-tree interferences. In this section, we formally define this problem, study its complexity, and present a greedy algorithm, and evaluate its performance by simulation experiments.

5.1. Model and Problem Definition

We assume that a sensor network is a static graph $G = (V, E)$, where V is the set of all nodes in the network, and E is the set of edges between two nodes who can talk to each other in one hop. Here, we only consider the data collection traffic in networks. Next, we define the interference value of a node in a tree. Reference [Burkhart et al. 2004]

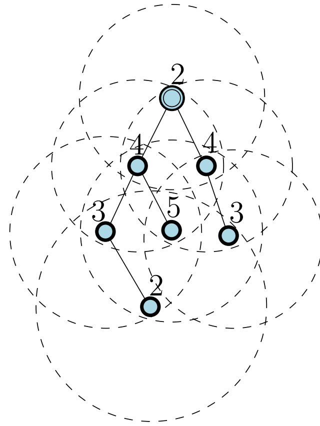


Fig. 6. A tree with 7 nodes. Each node is labeled with the interference value. The intra-tree interference value of the tree is 4

introduces an explicit definition of the interference value, based on the number of other nodes potentially disturbed by transmission of this node. In other words, interference is considered to be an issue at the sender instead of at the receiver. Because of the fact that the interference is actually a problem occurring at the receiver, we use a receiver-centric interference definition. The interference value of a node A is the number of other nodes by which the reception at A can be disturbed.

Definition 5.1. The interference set of a node u should be defined as $INT(u) = \{v|u \in D(v, I_v)\}$, where $D(v, I_v)$ is the interference disk with node v in its center and radius I_v, and the interference value of a node u is defined as $int(u) = |INT(u)|$.

Here, we assume that when a node is transmitting, all nodes within the transmitter's interference disk will be disturbed. We note that this assumption may not always be true in real networks because the interference region is not spherical observed in [Zhou et al. 2005], and interference sets of nodes may change during the time. But we can use a larger interference disk to cover the actual interference region, and compute a conservative interference set for each node. We use the interference range I_v instead of the communication range R_v to describe the interference region. By the observations of [Zhou et al. 2005], they are different in real networks. Furthermore, we use the assumption from the protocol interference model [Kyasanur and Vaidya 2005], where $I_v = (1 + \alpha) \times R_v$, and $\alpha > 0$ implies that all of u's neighbors belongs to $INT(u)$.

Next, we define the intra-tree interference value of a tree. There are two concerns. First, we should use the maximum interference value I_{max} as the interference value of the tree. Given the bandwidth B at each node, it can be proved that the theoretical lower bound of the single-flow capacity in this tree is B/I_{max}. Thus, I_{max} decides the lowest data rate of a single flow through the tree, which is important for applications. Secondly, since our interference model is receiver-centric and leaf nodes are not receivers in data collection traffic, the interference of a tree is the maximum interference value among all *non-leaf* nodes.

Definition 5.2. The intra-tree interference value of a tree T is defined as $int(T) = \max\{int(u) : u \text{ is a non-leaf of } T\}$

As an example, the intra-tree interference value of the tree in Figure 6 is 4, in spite of the fact that there is a leaf node with the interference value of 5. Here, we want to emphasize that dealing with the non-leaf condition is not trivia. In fact, it implies that if a node has a large interference value, we can set it as a leaf and then it is not needed to receive packets from other nodes in the data collection traffic. By doing this, we can indeed reduce the interference in the tree.

Now, we can define the partition and channel assignment problem. Given k available orthogonal channels, the problem is to Partition a sensor network into k vertex-disjoint trees with Minimizing the maximum intra-tree Interference value of all Trees, called the PMIT problem. Next, we study its complexity.

5.2. The Complexity of the PMIT problem

The PMIT problem is similar to graph coloring problem, but different at that this problem requires that nodes with the same color construct a tree with minimum interference, while the graph coloring problem seeks independent sets for each color. Unfortunately, we find that this problem is also NP-complete, like the graph coloring problem. First, we restate this optimization problem as a decision problem. Given an integer d, the decision PMIT problem $< G, k, d >$ is to determine whether a graph G can be partitioned into k node-disjoint trees and the interference value of every tree is no more than d. The following theorem shows that this problem is NP-complete.

THEOREM 5.3. *The PMIT problem is NP-Complete.*

PROOF. First, it is clear that PMIT belongs to NP problem, because given a partition we can calculate the interference value of each non-leaf node of trees, and get the interference value of each tree. This verification can be performed straightforwardly in polynomial time.

Next, we prove that the PMIT problem is NP-hard by reducing the k-coloring problem to PMIT. Given a graph $G = (V, E)$ and k colors, the k-coloring problem is to determine whether each node can be assigned one color such that adjacent nodes must have different colors. Before the reduction, we first calculate the maximum degree Δ among all vertices of G. Then we define a structure $\Delta + 1$-star, where a $\Delta + 1$-star consists of a vertex as the core and $\Delta + 1$ other vertices which are all adjacent to the core but have no edges to each other. The reduction algorithm takes as input an instance $< G, k >$ of k-coloring problem. It modifies the graph G into a new graph $G' = (V', E')$ as follows. First, we add a new vertex r as the root, and directly connect the root to every vertex in G. Then, for each edge $(u, v) \in E$, delete this edge, add a new $\Delta + 1$-star, called S_{uv}, into the graph and connects u and v to the core of this star respectively. At last, we calculate the proper interference range of each vertex, such that the interference disk of one vertex only covers its neighbors; in other words, the interference number of a vertex equals to its degree. After these operations, we get a new instance of the decision PMIT problem $< G', k, \Delta + 2 >$. Obviously, this reduction can be done in polynomial time. Figure 7 illustrates an example of the reduction. In the original graph, $\Delta + 1 = 2$. We first add the root, and then add 3-stars to replace every edge of the original graph.

Now, we show that the graph G can be k-colored if and only if the PMIT problem $< G', k, \Delta + 2 >$ can be satisfied. First, suppose that graph G' can be partitioned into a set G of trees $T_1, T_2, \ldots, T_k$, which satisfy PMIT problem. Then we compute a collection Γ of sets of vertices $C_1, C_2, \ldots, C_k$, where $C_i = T_i \cap V$. We claim that Γ is a proper k coloring for graph G. First, it is clear that every vertex of G is contained in one set of Γ. Secondly, for an arbitrary edge (u, v) in G, if both u and v are in the same tree T after partitioning G', the star S_{uv} must also belong to T, because the star only connects to u and v. In T, the core of S_{uv} is not a leaf and the interference value of the core is $\Delta + 3$, since the core has $\Delta + 3$ neighbors. Then the interference value of T must be at least $\Delta + 3$, which contradicts the constraint of the PMIT problem $< G', k, \Delta + 2 >$. So, we prove that if any two vertices u and v are adjacent in G, they are not in the same tree

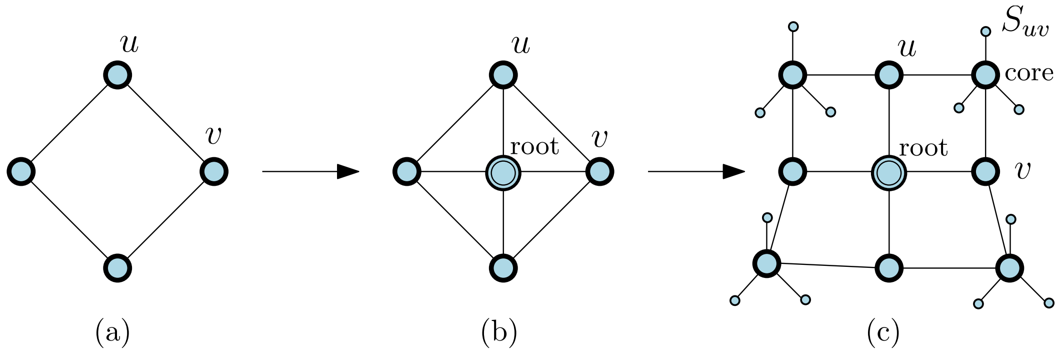


Fig. 7. Reducing k-coloring to PMIT. (a) original graph. (b) after adding a root. (c) after adding stars.

of G, which means that they are not in the same set of Γ. Therefore, Γ is a proper k coloring for graph G.

On the other hand, if there is a proper k coloring Λ for graph G, where $\Lambda = \{C_1, C_2, \ldots, C_k\}$, we can construct a set of trees in G' satisfying the PMIT problem $< G', k, \Delta + 2 >$. First, we add the root r into every set C_i of Λ, and then for every star S_{uv}, because u and v are adjacent, they must be in two different sets C_u and C_v. We arbitrarily put the star S_{uv} into one of two sets. Suppose it is the set C_u, and then C_u induces a tree in G', in which original vertex u of G connects directly to the root r, and the star S_{uv} connects to u. Next, consider the interference value of T. Since the maximum degree of u is Δ in G, then there are at most Δ stars connecting to u. The interference number of u is at most $\Delta + 1$, plus the root. For the core of the star S_{uv}, the interference number is $\Delta + 1$, because it lies in a $\Delta + 1$-star, and only u is in the same tree. Thus the interference value of T is at most $\Delta + 2$. Therefore, we can finally find a set of trees satisfying PMIT.

Above all, we prove that the reduction is legal. Since k-coloring problem is NP-hard, the PMIT is also NP-hard. □

In the light of NP-completeness, there is no polynomial time exact algorithm which can always find the optimal partition. In next subsection, we introduce a greedy heuristic for the PMIT problem.

5.3. The PMIT Algorithm

In this algorithm, we assume that the interference sets of all nodes are already known. For a node u, let c_u denote u's channel, and p_u denote u's parent.

This algorithm firstly applies a Breadth-First search algorithm to compute a fat tree rooted at the base station. There are two important properties of the fat tree. First, nodes keep its height and have multiple parents on the fat tree. Secondly, the fat tree is actually a shortest path tree, where branches from the base station to each node are paths with the least hop count, because we use BFS strategy to build the tree.

Next, we execute the channel allocation one-by-one level from top to bottom on the fat tree. At each level, we always process nodes with fewer parents first, because they are less free to choose channels. For each node, we choose an optimal channel, in other words select an optimal tree to add the node in. The criteria is that the tree must connect to the node, and adding the node brings the least interference to this tree. If multiple trees tie, the tree with fewer nodes is chosen. After a node joining a tree, it selects a parent which has the least interference value among all possible parents within the tree selected. It is clear that the algorithm covers all nodes of graphs, and when a node gets a channel, the algorithm ensures it connects to one tree rooted at

ALGORITHM 1: Greedy PMIT

Input: k channels, a graph $G = (V, E)$, a root r and the interference set of every node.
Output: For each node u, c_u and p_u
Use BFS-Fat-tree algorithm to construct a fat-tree with rooted at r.;
for *each channel* i **do**
    $T_i = r$;
end
for *each node* u **do**
    $c_u = 0$; $p_u = null$;
end
$level = 1$;
repeat
    $node\_list = \{u|height(u) == level; c_u == 0\}$
    sort $node\_list$ in ascending order by the number of node's parents.
    for *each node* u *in* $node\_list$ **do**
        find T_i which keep connected and has the least interference after adding u.
        $T_i = T_i \cup \{u\}$; $c_u = i$; $p_u = v$, which connects to u and has the least interference among all nodes in T_i.
        update the interference value of T_i.
    end
    $level++$;
until $level >$ *the maximum height of the fat tree*;

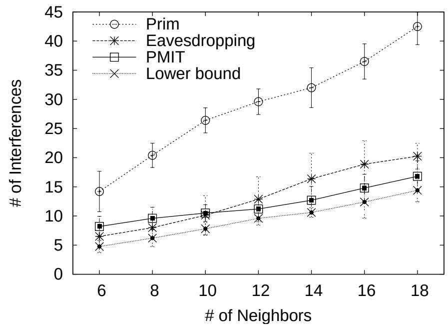
ALGORITHM 2: Breadth-First-Search Fat tree

Input: a graph $G = (V, E)$ and a root r.
Output: For every node u, its parent set $parent(u)$ and its height in the tree $height(u)$.
for *each node* u *in* G **do**
    $height(u) = max - integer$; $parent(u) = null$;
end
$S = r$; $height(r) = 0$;
for *each node* u *in* S **do**
    for *each node* u*'s neighbor* v **do**
        if $height(v) > height(u)$ **then**
            $height(v) = height(u) + 1$;
            $parent(v) = parent(v) \cup \{u\}$;
            $S = S \cup \{v\}$;
        end
    end
end

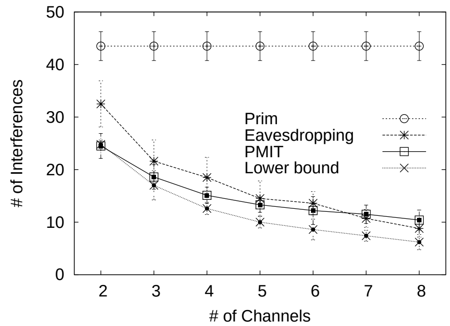
r, which demonstrates the correctness of the algorithm. The following theorem states the time complexity of the algorithm.

THEOREM 5.4. *The time complexity of the Greedy PMIT algorithm is $O(d \times k \times n^2)$, where d is the diameter of the graph, n is the number of nodes, and k is the number of channels.*

PROOF. The time complexity of construing a FAT tree is $O(d \times \Delta \times n)$, where Δ is the maximum degree in the graph. In PMIT algorithm, Step 12 takes $O(k \times n)$ in the worst case, and the loop beginning at Step 11 may run at most n time. Thus, the procedure within the repeat loop takes $O(k \times n^2)$, and the repeat loop iterates at most d times, because the tree height never exceeds the diameter of the graphs. The time complexity is $O(d \times k \times n^2)$ in the worst case. □



(a) Interference vs. Node density



(b) Interference vs. Channels

Fig. 8. Performance Evaluation of the PMIT algorithm

A good property of this algorithm is that every node keeps the shortest path to the base-station. This property comes from the fact that the algorithm processes nodes one-by-one level from top to bottom of this fat tree. Therefore, this partitioning algorithm does not require extra transmissions and does not reduce the end-to-end delivery ratios, neither increase energy consumption during data collection.

This algorithm can be easily modified to a distributed algorithm because it only needs a local search at each node. First, nodes can construct a fat tree by broadcasting messages. During channel allocation, nodes make their own decision based on message from their parents, and notify their children. Also, since the network is static, we can run the centralized algorithm once at the beginning, or very infrequently, which is still practical even for large WSNs.

5.4. Evaluation of the Greedy Algorithm

As mentioned earlier, the network partition and channel assignment are very crucial to network performance improvement. In this subsection, we evaluate the performance of our greedy algorithm. We develop a graph simulator in JAVA, which can randomly generate a graph, and apply different schemes to do the channel allocation. In all experiments, we simulate a $200m \times 200m$ field, 250 nodes are uniformly distributed in the field, and the communication range is 10∼35m and interference range is always 1.5 times as the communication range. Since we are the first to study the PMIT problem, there are no other PMIT algorithms we can compare against. We use three alternative schemes as comparisons. One is to apply the Prim's algorithm to construct a minimum spanning tree as the data collection tree. This scheme is referred to as a base scheme with a single channel. Secondly, we implement the Eavesdropping channel assignment method proposed in [Zhou et al. 2006]. We refer to it as a typical method used by node-based protocols. Note that this scheme does not ensure the connectivity among nodes in each channel. Next, we find the maximum interference value ρ among all nodes, and use ρ/k as the lower bound of the interference value after allocating k channels. Finally, we run the greedy algorithm and measure the maximum interference value among all trees after partitioning. In all experiments, each data point comes from the average result of 50 repeated experiments. For each data point, we also give its 90% confidence interval.

In the first set of experiment, we use 3 channels and vary the number of neighbors by adjusting the communication range. The result is shown in Figure 8(a). We can see that the greedy algorithm can always get around 1/3 interferences of the Prim's algorithm with a single channel, which shows that our algorithm efficiently utilize 3 channels to decrease interferences. Comparing with Eavesdropping algorithm, we see that when the density is low, the Eavesdropping has less interferences than ours, mainly because it does not ensure the connectivity, but when the density becomes larger, the greedy algorithm outperforms the random scheme, for example when the density is 18, it gets 17% less interferences than Eavesdropping scheme. The reason is two-fold. First, when the density is large, there are no enough channels for nodes in two hop neighbors, so Eavesdropping have to randomly choose channels among nodes, which makes the maximum interference relatively large. But our algorithm always tries to find the local optimal, which can achieve more stable performance. Secondly, when the density is large, our greedy algorithm has more chances to set nodes with large interferences as leaves, which can further reduce interferences of subtrees. Finally, when comparing with the lower bound, the result of our algorithm is close to the lower bound of the interference value, and more importantly, the gap does not scale up with the density increasing, which suggests that our greedy algorithm has a good scalability with different densities.

In the second experiment, the radio range is 35m and we change the number of available channels. Results are illustrated in Figure 8(b). It is clear that with the small number of channels, our PMIT algorithm compute less interferences than Eavesdropping scheme, especially, when only 2 channels can be used, our algorithm has 24% less interferences than Eavesdropping scheme, and 51% less than a single channel. With more channels, performance of two schemes become closer. When there are 8 channels, Eavesdropping scheme has 18% less interference than our greedy algorithms. Comparing with the lower bound, we can see that with the small number of channels, our algorithm computes almost the same number of interferences as the lower bound.

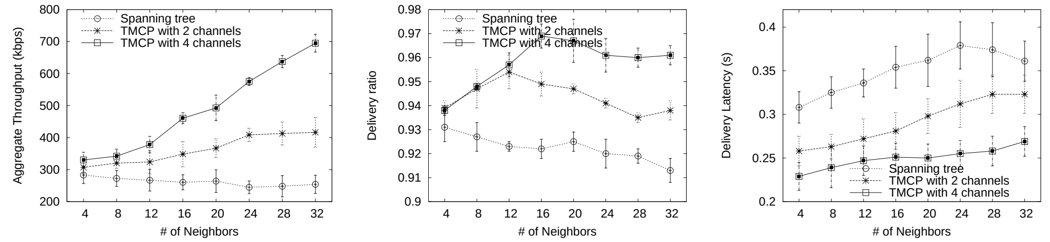
6. PERFORMANCE EVALUATION OF TMCP

TMCP uses the greedy algorithm in the channel assignment component. In this section, we evaluate the communication performance of TMCP, by simulation and by experiments in a real testbed.

6.1. Simulation Evaluation

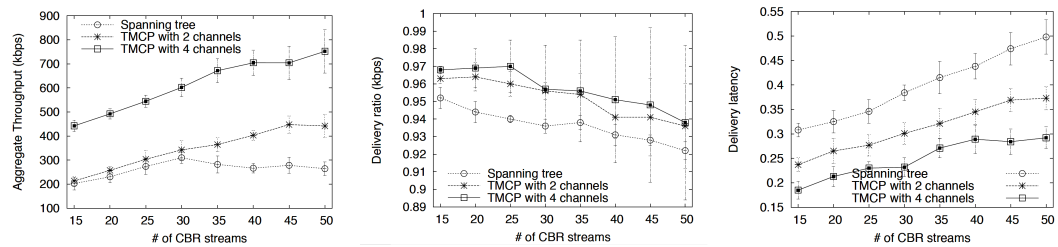
First, we evaluate the performance of TMCP through simulation experiments. We implement TMCP in GloMoSim. We use the same setting as simulations in section 5.4, where the communication range is 10$\sim$40m and the interference range is always 1.5 times as the communication range. This communication model is typically used to simulate the RF model of the CC2420 radio. Also, in the MAC layer, we use CSMA with the ACK-retransmission mechanism, which ensures that most packets can be received.

We conduct three sets of experiments. In the first two experiments, we compare TMCP with 2 and 4 channels and a spanning tree routing protocol with a single channel. First, we measure network performance with different node density. In this experiment, there are 50 Many-to-one CBR streams in the network, and the rate of each CBR is 40 packets per second. Results are shown in Figure 9, with the 90% confidence interval of each data point. According to the results, TMCP outperforms the original protocol in the following aspects. 1) By using the sophisticated network partition and frequency assignment algorithm, TMCP with 2 and 4 channels can decrease potential transmission collisions, which leads to an average 1.6 and 2.7 times higher aggregate throughput than the spanning tree algorithm. 2) By splitting traffic into different subtrees, TMCP decreases radio collisions as well as traffic congestion, which leads to higher packet delivery ratios and lower latency. 3) When the node density is increasing, TMCP shows good scalability. For example, in Figure 9(a) TMCP with 4 channels results in an increasing throughput as the number of neighbors increases, because with more nodes, TMCP more evenly partitions and channel allocation, which



(a) Throughput vs. Node density (b) Delivery ratio vs. Node density (c) Delivery latency vs. Node density

Fig. 9. Performance with different node density



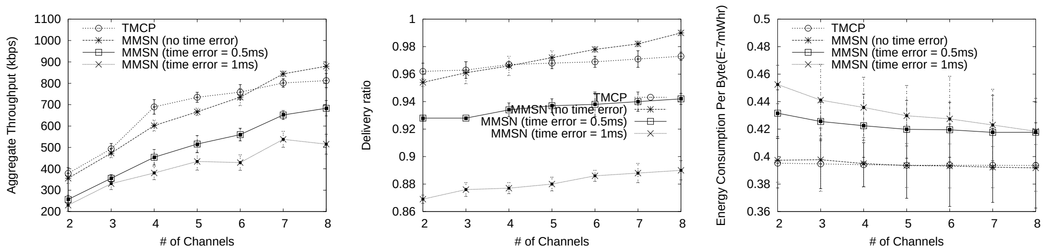
(a) Throughput vs. # of CBR streams (b) Delivery ratio vs. # of CBR streams (c) Delivery latency vs. # of CBR streams

Fig. 10. Performance with different network workloads

leads to better spatial reuse of concurrent transmissions. TMCP with 2 channels also shows this trend, but stops increasing the throughput when nodes have more than 20 neighbors, because the number of interferences exceeds the capacity of 2 channels.

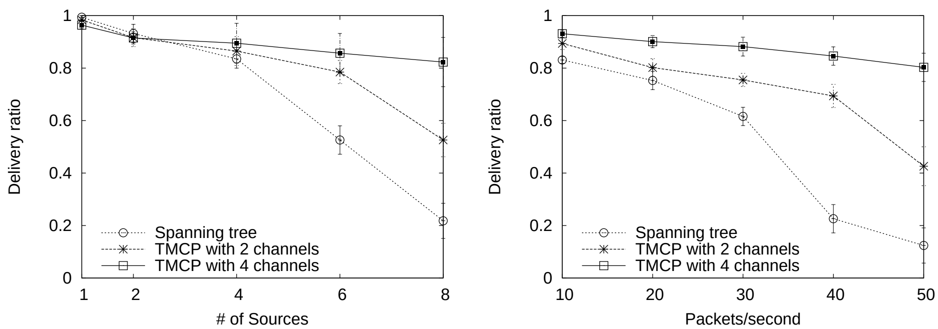
Second, we measure the performance with different network workloads. In Figure 10, we see that TMCP always exhibits better performance than the spanning tree protocol, especially in heavy workloads. For example, with 50 CBR streams TMCP with 4 channels achieves 2.8 times aggregated throughput and 42% lower delivery latency than the spanning tree. Also, the spanning tree protocol has a decreased packet delivery ratio from 95.2% to 92.1% in Figure 10(b), while TMCP has a much smaller decrease. This is because TMCP splits heavy workloads into different trees and is more tolerant to system load variation than the spanning tree algorithm. However, we also find the performance of TMCP is unstable. For example, in Figure 10(b), when the workload increases, the variation of delivery ratios of TMCP becomes larger. This is because these CBR streams are not evenly distributed among subtrees, and flow congestion can occurs on subtrees at which many CBR streams cluster.

Last, we compare TMCP with MMSN [Zhou et al. 2006], a typical node-based multi-channel protocol. In this group of experiments, 50 CBR streams are used and the node density is set to 38, by configuring the radio range to 40m. As mentioned in section 3, time synchronization errors may impact the performance of multi-channel protocols. Here with the number of channels changing, we compare TMCP and MMSN with different time errors. All results are presented in Figure 11. Here, we compare throughput, delivery ratio and energy consumption. Overall, the performance of TMCP and MMSN is very close. More precisely, when the number of channels is small, TMCP has a little better performance than MMSN. For example, in Figure 11(a), TMCP achieves a 10% higher throughput on average than MMSN with less than 5 channels. But when the number of channels increases, MMSN outperforms TMCP. This agrees with the



(a) Throughput (b) Delivery ratio (c) Energy Consumption

Fig. 11. Performance comparison of TMCP and MMSN



(a) Delivery ratio vs. Number of Sources (b) Delivery ratio vs. Data rate

Fig. 12. Evaluation in a test bed

evaluation results in section 5.4, where our channel assignment algorithm works better than other channel assignment schemes with a small number of channels. Also Figure 11(c) shows that the power consumption of TMCP and MMSN are close. However, here we only consider the power consumption of data communication. As discussed in section 5.3, the channel assignment is executed infrequently, and that power consumption can be amortized during the time. On the other hand, time synchronization errors cause a great performance degradation for MMSN, but without any impact on TMCP. Considering multi-channel realities, TMCP is more suitable for practical WSNs than node-based multi-channel protocols.

6.2. Evaluation in a Real Testbed

Besides simulation evaluations, we also implement TMCP in a real testbed with Micaz motes. The testbed consisted of 20 Micaz motes, and four motes are laid closely together to act as a base station with four transceivers. Before the experiment, we first use the channel detection technique described in section 4 to find available orthogonal channels, and then run the channel assignment algorithm on a PC. After computing the assignment, the results are sent out to all motes. During the experiments, some nodes are selected as sources to transmit packets to the base station. We conduct two sets of experiment, and compare a normal spanning tree protocol with a single channel and TMCP with 2 and 4 channels. All experiments are repeated several times and averaged.

In the first set of experiments, while changing the number of sources, we measure the packet reception ratios. Here, all sources send packets with the data rate of 20 packets per second. The results are shown in Figure 12(a). We see that when the number of sources is above 4, the spanning tree protocol has low reception ratios below 60%, while TMCP with 2 channels can get high reception ratio until there are 8 sources and TMCP with 4 channels always maintains a high reception ratio. Performance gains of TMCP come from the fact that it effectively reduces interferences and mitigates congestion at nodes.

In the second set of experiments, we use 4 sources in the networks, and change the data rate of the sources. We also measure packet reception ratios at the base station. The result is shown at Figure 12(b). We see that the saturated data rate (reception ratio above 80%) of the sources is 20 packets per seconds for the spanning tree protocol. For TMCP with 2 channels, the saturated rate is 30 packets per seconds, and TMCP with 4 channels can support 50 packets per second. These experiments show that TMCP works well in real sensor networks.

Above all, our evaluation results show that TMCP better accommodate multi-channel realities found in WSNs than other multi-channel protocols, and more suitable to provide reliable, low- latency and robust communication for high-rate and dense WSNs for mission-critical applications.

7. PRUNING TMCP: RELIABLE TREE-BASED MULTI-CHANNEL PROTOCOL

In the last section, we present TMCP, a novel Tree-based Multi-Channel Protocols, which significantly improves network performance, in terms of throughput, delivery ratio and latency. Especially, TMCP outperforms existing protocols in dense networks with high volume traffic, which makes it a better solution for communication requirements of mission-critical WSN applications. TMCP is designed with the assumption that all links have the same quality, and low-level MAC protocols provide reliable point-to-point transmissions. However, this assumption may not hold for many mission-critical applications, where links in networks exhibit the great heterogeneity in link qualities. In [Lin et al. 2009], Shan et al. observe different packet reception ratios of links and identify two types (stable and unstable) of links in an indoor WSN system. This quality diversity are caused by multi-path fading of signals and shadowing effects of humans and obstacles. In [Sexton et al. 2005], Sexton et al. measure the link characteristics in several industrial facilities, which consist of rooms with lots of metal surfaces and rotating machinery. Their results show that links in this environment have a wide quality range and vary substantially in stabilities. From all these observations, it is clear that multi-channel protocol design should consider the link diversity in WSNs, and address potential delivery failures caused by poor links.

Unfortunately, the original TMCP cannot guarantee reliable end-to-end deliveries in the presence of the link diversity. TMCP builds routing sub-trees to minimizing interferences. It is possible that routing trees consist of low-quality links, which leads to the delivery failures. In this section, we focus on how to extend TMCP to minimize interferences, as well as to meet end-to-end reliability requirements.

7.1. Model and Reliable Channel Assignment Problem

We extend the basic network model in section 4 with the link quality metric. For every link (edge) e, $p(e)$ stands for the probability of successful transmission for a single attempt on this link. In realities, this probability can be obtained by link estimation methods in [Fonseca et al. 2007] [Woo et al. 2003]. Also, the function $pdr(e, x)$ denotes the probability of successful transmission for x attempts (retransmissions), and we have $pdr(e, x) = 1 - (1 - p(e))^x$. Next, we assume that there is only one base station in the network. For node u, $E2EPDR(u)$ stands for the probability of successful transmission from node u to the base station. Suppose that packets are relayed along a k-hop path, and then we have $E2EPDR(u) = \prod_{i=1}^{k} pdr(e_i, x)$, where e_i is the i-th link along the path. Finally, we use RR to denote the minimum acceptable probability of successful transmission from nodes to the base station. Given a RR, the communication reliability requirement is: $\forall u \in G, E2EPDR(u) \geq RR$.

With this model, we extend the PMIT problem in section 5 to a new reliable channel assignment problem. Given k available orthogonal channels and the reliability requirement RR, the problem is to partition a sensor network into k vertex-disjoint trees, such that (1) the partition minimizes the maximum intra-tree interference value of all trees; (2) $\forall u \in G, E2EPDR(u) \geq RR$. We call this problem as RPMIT problem. As an extension of the PMIT problem, the RPMIT problem is also NP-Complete.

7.2. The pruning algorithm

As mentioned before, TMCP effectively reduces interferences by building optimal routing subtrees, but it cannot meet the reliability requirements, because subtrees may consist of low-quality links. In order to address this issue, we propose a two-step solution. First, we use a pruning algorithm to remove all links which cannot meet the reliability requirement, and then run TMCP on the remaining fat tree to minimize interferences.

In order to prune the fat tree, we firstly use a downward pruning. During the iteration, we first compute the $E2EPDR$ of parent nodes, and then remove links that connect to child nodes but make children's $E2EPDR$ lower than RR. Since a node may have multiple path to the root on the fat tree, one question is which $E2EPDR$ should be used to compute the $E2EPDR$ of its children. The first option is to use the minimum $E2EPDR$ which always guarantees that all remaining paths can meet the reliability requirement. But this over strict pruning removes links that actually can be used in a quality path, and may make it impossible for low-level nodes to find a path. It also makes the remaining fat tree too sparse, thus reduce the choices for the channel assignment. On the contrary, the second option is to use the maximum $E2EPDR$, which only removes links whose qualities are too poor and cannot be used in any path. But it cannot guarantee that all remaining paths are qualified.

We choose to use the maximum $E2EPDR$ during the downward pruning. After that, we further run an upward pruning from leaf nodes to the root. The upward pruning aims to remove low-quality links to guarantee that all remaining paths are qualified. We introduce a new node property, the required end-to-end delivery ratio, denoted as $RE2EPDR$. For a node u, $RE2EPDR(u)$ is defined as the minimum $E2EPDR$ that allows u's descendants to meet the reliability requirements. $RE2EPDR(u)$ can be computed by:

$$RE2EPDR(u) = \max_{v \in u\text{'s children}} \left(\frac{RE2EPDR(v)}{pdr(e_{u,v}, x)}\right) \tag{1}$$

During the upward pruning, we compute the $RE2EPDR(u)$ for any node u, and then remove u's upstreaming link $e_{u,p}$, if $pdr(e_{u,p}, x) < RE2EPDR(u)$. Here, the upstreaming link e is the link connecting u and u's parent p. This upward pruning guarantees that all remaining paths meet the reliability requirement, because it only keeps links that satisfy all descendants' $RE2EPDR$ requirements. The detailed algorithm is presented in the following.

Our pruning algorithm have two good properties. First,it keeps the connectivity of the graph. More specifically, if there exist a qualified path connecting a node u to the root in the original graph,it is guaranteed that u still connects to the root with a qualified path after pruning. Second, the algorithm guarantees that the remaining network consists of only qualified path. This property results from the upward pruning which

ALGORITHM 3: The Pruning Algorithm

Input: a graph $G = (V, E)$, a root r, a link quality profile $p(e)$ for every $e \in E$, the maximum number of retransmissions n and the end-to-end reliability requirement RR.
Output: a new graph $G' = (V, E')$

```
G' ← G;
Use BFS-Fat-tree algorithm to construct a fat-tree with rooted at r.;
h ← 0 ;
/* do a downward pruning                                                    */
while h <= the fat-tree's max_height do
    for each node u in the tree level h do
        max_E2EPDR ← 0 ;
        for each parent p in parent(u) do
            if E2EPDR(p) * pdr(e_{u,p}, n) < RR then
                remove e_{u,p} from G'
            end
            else
                if E2EPDR(p) * pdr(e_{u,p}, n) > max_E2EPDR then
                    max_E2EPDR ← E2EPDR(p) * pdr(e_{u,p}, n)
                end
            end
        end
        E2EPDR(u) ← max_E2EPDR
    end
    h ← h + 1
end
/* compute the requirement E2EPDR for leaf nodes                            */
for each leaf node u do
    RE2EPDR(u) ← RR;
    for each parent p in parent(u) do
        if pdr(e_{u,p}, x) < RE2EPDR(u) then
            remove e_{u,p} from G'
        end
    end
end
h ← the fat-tree's max_height − 1;
/* do an upward pruning                                                     */
while h > 0 do
    for each node u in the tree level h do
        max_RE2EPDR ← RR;
        /* compute the RE2EPDR for this node                                */
        for each child node c in children(u) do
            if RE2EPDR(c) / pdr(e_{u,c}, x) > max_RE2EPDR then
                max_RE2EPDR ← RE2EPDR(c) / pdr(e_{u,c}, x)
            end
        end
        RE2EPDR(u) ← max_RE2EPDR;
        /* remove unqualified links                                         */
        for each parent p in parent(u) do
            if pdr(e_{u,p}, x) < RE2EPDR(u) then
                remove e_{u,p} from G'
            end
        end
    end
    h ← h − 1
end
```

remove all links that cannot meet the minimum reliability requirement. The following theorem states the time complexity of the algorithm.

THEOREM 7.1. *The time complexity of the pruning algorithm is $O(d \times \Delta \times n)$, where d is the diameter of the graph, Δ is the maximum degree in the graph and n is the number of nodes.*

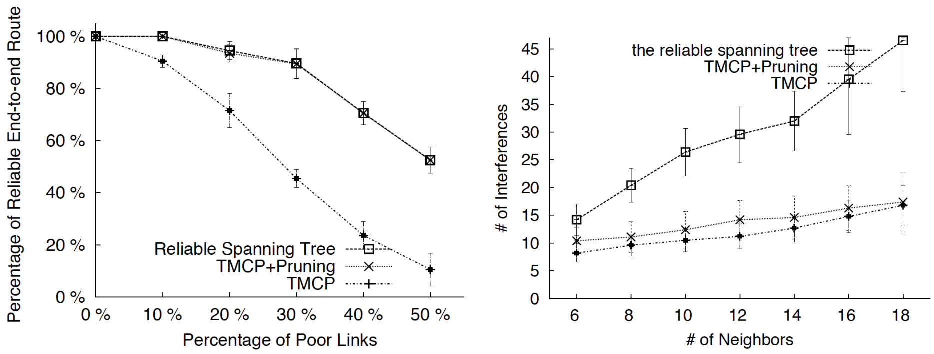
PROOF. The time complexity of the pruning algorithm is equal to the time complexity of construing a FAT tree, which is $O(d \times \Delta \times n)$ □

As well as TMCP, this pruning algorithm can be changed to a distributed algorithm because every node only need to collect information from its neighbors (parents and children). In real deployment, we can first measure link qualities in networks, and run the pruning algorithm and TMCP at the beginning. During run-time, the pruning algorithm and TMCP can be triggered periodically to ensure the communication reliability.

7.3. Evaluation of the Pruning TMCP

In this subsection, we evaluate the performance of the new TMCP in two steps. First, we analyze the performance of the original and new TMCP by observing their impact on two network properties. The first property is the percentage of reliable end-to-end routes, defined as the percentage of nodes which have a qualified route to the base station over all nodes. The latter metric is the number of potential interferences. We develop a graph simulator in JAVA, which can randomly generate a graph, and apply different schemes to do the channel allocation. In all experiments, we simulate a $200m \times 200m$ field, 250 nodes are uniformly distributed in the field, and the communication range is 10∼35m and interference range is always 1.5 times as the communication range. In order to simulate the link diversity in real systems, we follow the link quality model in [Lin et al. 2009], where links are categorized as poor links and good links. In simulations, the packet reception ratio of one good link is randomly selected in the range of [0.9, 1], while that of poor links in [0.5, 0.8]. We also assume that the maximum number of retransmission is 2, the end-to-end reliability requirement RR is 80%, and there are 3 available channels to use. Besides the original and Pruning TMCP, we also run the reliable spanning tree algorithm, which use Dijkstra's algorithm to compute a spanning tree on which each node connects to the base station through the maximum reliability (weight) route. This approach can always achieve the maximum percentage of reliable end-to-end routes, but cannot reduce the number of potential collisions. We refer it as the best single-channel tree-base routing scheme. In all experiments, each data point comes from the average result of 50 repeated experiments. For each data point, we also give its 90% confidence interval.

In the first set of experiments, we study the performance of all three approaches with different level link diversities, by varying the ratio of poor links in networks. All results are shown in Figure 13(a). All three algorithms compute less reliable routes when poor links increase in networks, but the reliable spanning tree and the pruning TMCP compute much more reliable routes than TMCP. For instance, with 30% poor links, TMCP makes only 42% nodes with reliable routes, but the pruning TMCP makes over 85% nodes with reliable routes. This reliability improvement comes from the fact that the pruning algorithm removes poor links which lead to unreliable routes. Furthermore, it is clear that the Pruning TMCP has the almost same result with the reliable spanning tree scheme. As aforementioned, the reliable spanning tree can always find a reliable route if existed. This result demonstrates that the pruning algorithm prunes sufficient links and ensures that the remaining network consists of only reliable routes.



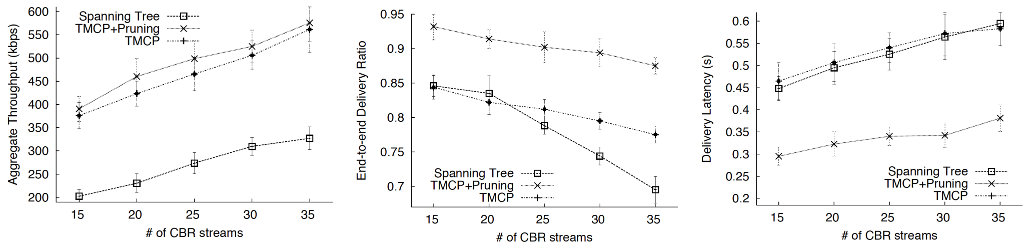
(a) Reliable end-to-end routes with different-level link diversity

(b) Numbers of potential collisions with different densities

Fig. 13. Performance Evaluation of the Pruning TMCP algorithm

In the second set of experiments, we observe the performance of three approaches under different node densities, by varying the number of neighbors. In these experiments, networks consists of 20% poor links. As shown in Figure 13(b), both TMCP schemes get much less potential interferences of the reliable spanning tree scheme, which demonstrates that the channel assignment scheme efficiently utilizes 3 channels to reduce interferences. Comparing with the original TMCP, this new pruning TMCP has more interferences. For example, with 10 neighbors, the pruning TMCP gets around 12 potential interferences, while the original TMCP has around 10 interferences. This performance degradation is expected because the pruning algorithm removes a number of links, make the network sparser, and then reduce channel selection options. Overall,the pruning TMCP significantly improves the end-to-end reliability with the cost of slightly more potential interferences, which makes it a better solution for reliability-sensitive applications.

As the second step of our evaluation, we measure the general performance of the pruning TMCP with network traffic. The performance metrics include throughput, end-to-end delivery ratio, and latency. We implement both TMCP in GloMoSim. In this experiment, the percentage of poor links in networks is 20%, the average number of neighbors is 10, and the number of available channels is 4. Other settings are the same with previous experiments. Since mission-critical applications trigger burst and crowed traffic, our evaluation focuses on studying the performance with different network workloads, by varying the number of CBR streams in networks. Results are shown in Figure 14. We can see that the Pruning TMCP always exhibits better performance than the spanning tree and the original TMCP, especially with higher reliability and lower latency. For example, with 35 CBR streams, the pruning TMCP improves end-to-end delivery ratio by 12% and reduces the latency by 35% over TMCP. This performance improvement is because the pruning TMCP removes poor links, and tends to choose better links to build the routing tree, which increase the likelihood of successful transmissions and reduce the number of retransmissions. Furthermore, the pruning TMCP is more tolerant to workload increase than TMCP. Finally, when the number of CBR streams increases, all three schemes have a reliability decrease and a latency increase. But the pruning TMCP has the least performance degradation among all three.



(a) Throughput vs. # of CBR streams (b) Delivery ratio vs. # of CBR streams (c) Delivery latency vs. # of CBR streams

Fig. 14. Performance with different network workloads

8. CONCLUSION

This paper studies how to efficiently use multiple channels to improve network performance in WSNs. First, we study multi-channel realities in WSNs through a set of empirical experiments. It is shown that current node-based multi-channel protocols are not suitable for real WSNs because of the small number of available channels and unavoidable time synchronization errors. In light of this observation, we propose a tree-based multi-channel protocol called TMCP. By assigning channels to several trees instead of nodes, TMCP works with a small number of channels and without the need for time synchronization. By using a greedy algorithm, TMCP effectively decrease potential radio interference. Finally, we implement TMCP in a real testbed and evaluate its performance through simulations and testbed experiments. Results show that TMCP can greatly improve the throughput of networks, while maintaining high packet delivery ratios and low delivery latency in sensor networks. Furthermore, an extended scheme is proposed to maintain performance in networks with high link diversity.

REFERENCES

Philip Asare, Danyang Cong, Santosh G Vattam, BaekGyu Kim, Andrew King, Oleg Sokolsky, Insup Lee, Shan Lin, and Margaret Mullen-Fortino. 2012. The medical device dongle: an open-source standards-based platform for interoperable medical device connectivity. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*. ACM, 667–672.

Paramvir Bahl, Ranveer Chandra, and John Dunagan. 2004. SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM, 216–230.

Martin Burkhart, Pascal Von Rickenbach, Rogert Wattenhofer, and Aaron Zollinger. 2004. Does topology control reduce interference?. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 9–19.

Xun Chen, Peng Han, Qiu-Sheng He, Shi-liang Tu, and Zhang-Long Chen. 2006. A multi-channel MAC protocol for wireless sensor networks. In *Computer and Information Technology, 2006. CIT'06. The Sixth IEEE International Conference on*. IEEE, 224–224.

Rodrigo Fonseca, Omprakash Gnawali, Kyle Jamieson, and Philip Levis. 2007. Four-Bit Wireless Link Estimation.. In *HotNets*.

Dawei Gong, Miao Zhao, and Yuanyuan Yang. 2010. A multi-channel cooperative MIMO MAC protocol for wireless sensor networks. In *Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on*. IEEE, 11–20.

Texas Instruments. 2006. CC2420: 2.4 GHz IEEE 802.15. 4/ZigBee-ready RF Transceiver. *Available at Available at http://www. ti. com/lit/gpn/cc2420* (2006), 53.

Pradeep Kyasanur and Nitin H Vaidya. 2005. Capacity of multi-channel wireless networks: impact of number of channels and interfaces. In *Proceedings of the 11th annual international conference on Mobile computing and networking*. ACM, 43–57.

Hieu Khac Le, Dan Henriksson, and Tarek Abdelzaher. 2007. A control theory approach to throughput optimization in multi-channel collection sensor networks. In *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 31–40.

Jiandong Li, Zygmunt J Haas, Min Sheng, and Yanhui Chen. 2003. Performance evaluation of modified IEEE 802.11 MAC for multi-channel multi-hop ad hoc networks. *Journal of Interconnection Networks* 4, 03 (2003), 345–359.

Jinbao Li, Desheng Zhang, Longjiang Guo, Shouling Ji, and Yingshu Li. 2010. ARM: An asynchronous receiver-initiated multichannel MAC protocol with duty cycling for WSNs. In *Performance Computing and Communications Conference (IPCCC), 2010 IEEE 29th International*. IEEE, 114–121.

Shan Lin, Jingbin Zhang, Gang Zhou, Lin Gu, John A Stankovic, and Tian He. 2006. ATPC: adaptive transmission power control for wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM, 223–236.

Shan Lin, Gang Zhou, Kamin Whitehouse, Yafeng Wu, John A Stankovic, and Tian He. 2009. Towards stable network performance in wireless sensor networks. In *Real-Time Systems Symposium, 2009, RTSS 2009. 30th IEEE*. IEEE, 227–237.

Hengchang Liu, Zhiheng Xie, Jingyuan Li, Shan Lin, David J Siu, Pan Hui, Kamin Whitehouse, and John A Stankovic. 2014. An Automatic, Robust, and EfficientMulti-User Breadcrumb Systemfor Emergency Response Applications. *Mobile Computing, IEEE Transactions on* 13, 4 (2014), 723–736.

Dijun Luo, Xiaojun Zhu, Xiaobing Wu, and Guihai Chen. 2011. Maximizing lifetime for the shortest path aggregation tree in wireless sensor networks. *INFOCOM, 2011 Proceedings IEEE* (2011), 1566–1574.

Arunesh Mishra, Vivek Shrivastava, Suman Banerjee, and William Arbaugh. 2006. Partially overlapped channels not considered harmful. *ACM SIGMETRICS Performance Evaluation Review* 34, 1 (2006), 63–74.

Marina Petrova, Lili Wu, Petri Mahonen, and Janne Riihijarvi. 2007. Interference measurements on performance degradation between colocated ieee 802.11 g/n and ieee 802.15. 4 networks. In *Networking, 2007. ICN'07. Sixth International Conference on*. IEEE, 93–93.

Daniel Sexton, Michael Mahony, Michael Lapinski, and Jay Werb. 2005. Radio channel quality in industrial wireless sensor networks. In *Sensors for Industry Conference, 2005*. IEEE, 88–94.

Jungmin So and Nitin H Vaidya. 2004. Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 222–233.

Asimakis Tzamaloukas and JJ Garcia-Luna-Aceves. 2001. A receiver-initiated collision-avoidance protocol for multi-channel networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Vol. 1. IEEE, 189–198.

Ramanuja Vedantham, Sandeep Kakumanu, Sriram Lakshmanan, and Raghupathy Sivakumar. 2006. Component based channel assignment in single radio, multi-channel ad hoc networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking*. ACM, 378–389.

Alec Woo, Terence Tong, and David Culler. 2003. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, 14–27.

Yafeng Wu, John A Stankovic, Tian He, and Shan Lin. 2008. Realistic and efficient multi-channel communications in wireless sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE.

Guoliang Xing, Mo Sha, Jun Huang, Gang Zhou, Xiaorui Wang, and Shucheng Liu. 2009. Multi-channel interference measurement and modeling in low-power wireless networks. In *Real-Time Systems Symposium, 2009, RTSS 2009. 30th IEEE*. IEEE, 248–257.

Qing Yu, Jiming Chen, Yanfei Fan, Xuemin Shen, and Youxian Sun. 2010. Multi-channel assignment in wireless sensor networks: A game theoretic approach. *INFOCOM, 2010 Proceedings IEEE* (2010), 1–9.

Jingbin Zhang, Gang Zhou, Chengdu Huang, Sang Hyuk Son, and John A Stankovic. 2007. TMMAC: An energy efficient multi-channel mac protocol for ad hoc networks. In *Communications, 2007. ICC'07. IEEE International Conference on*. IEEE, 3554–3561.

Jingbin Zhang, Gang Zhou, Sang H Son, and John A Stankovic. 2006. Ears on the ground: An acoustic streaming service in wireless sensor networks. *IEEE/ACM IPSN Demo Abstract* (2006).

Gang Zhou, Tian He, John A Stankovic, and Tarek Abdelzaher. 2005. RID: radio interference detection in wireless sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Vol. 2. IEEE, 891–901.

Gang Zhou, Chengdu Huang, Ting Yan, Tian He, John A Stankovic, and Tarek F Abdelzaher. 2006. MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks.. In *Infocom*, Vol. 6. 1–13.

Gang Zhou, Jian Lu, Chieh-Yih Wan, Mark D Yarvis, and John A Stankovic. 2008. Bodyqos: Adaptive and radio-agnostic qos for body sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE.

Gang Zhou, John A Stankovic, and Sang H Son. 2006. Crowded spectrum in wireless sensor networks. *IEEE EmNets* 6 (2006).