Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks

Yafeng Wu, John A. Stankovic, Tian He[†], Jiakang Lu, and Shan Lin
Computer Science Department, University of Virginia, {*yw5s,stankovic,jl9jb,sl8yc*}@*cs.virginia.edu*
[†]Department of Computer Science and Engineering, University of Minnesota, *tianhe@cs.umn.edu*

Abstract—**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. 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.**

I. 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 [1], the network cannot transmit multiple 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.

Recently some MAC layer multi-channel protocols have been 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[2], TMMAC[3] and MCMAC[4] are such protocols designed for WSNs. Simulation results show that they can significantly improve network throughput 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 multi-channel 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 sub-trees, 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 analyze 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 testbed 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.

The rest of paper is organized as follows. In section II, we explain related work. In section III, we present empirical results from experiments that investigate multi-channel realities found in WSNs. The design of TMCP is presented in section IV. In section V, we describe the related channel assignment problem and present a greedy algorithm with its evaluation. In section VI, we evaluate the performance of TMCP with simulation and real experiments. Finally, in section VII, we present conclusions and future work.
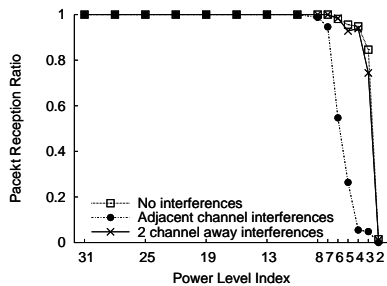
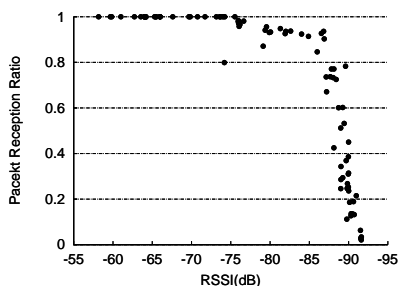Fig. 1. Transmission power level index vs. Packet reception ratio

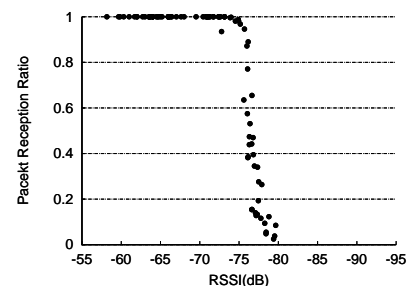Fig. 2. RSSI vs. Reception ratio without interferences

Fig. 3. RSSI vs. Reception ratio with adjacent channel interferences

II. RELATED WORK

In the state-of-the-art research, a significant number of multi-channel protocols have been proposed for WSNs in general, such as multi-channel MAC protocols [5] [6] [7] [8]. 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.

Recently, MMSN [2], TMMAC [3] and MCMAC [4] are three new 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.

More recently, two different channel assignment methods are proposed. A component-based protocol is presented in [9] which assigns channels to connected components in wireless ad hoc networks, and in [10], 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 ours, our scheme focuses on how to use multi-channels to construct the optimal topology with low interferences and optimize throughputs in practical WSNs.

III. EXPERIMENTS ON MULTI-CHANNEL REALITIES

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 [11] [12] [13], 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.

A. 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 [14] used in Micaz motes 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.

1) Non-orthogonal Channel Interferences: Non-orthogonal channel interferences are well studied in general wireless networks [13]. For WSNs hardware, the CC2420 chip specification [14] indicates that the adjacent channel rejection is 45/30 dB, but few works study its real impact on the performance of multi-channel WSNs. In the following we present experiments to study this phenomenon.

In the first experiment, we place three Micaz motes in a line, with one transmitter, one receiver and one jammer. The jammer's transmission is synchronized with the transmitter to generate interferences. Both the transmitter and the receiver
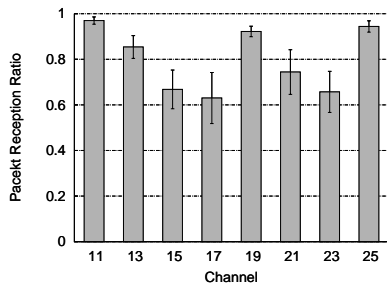
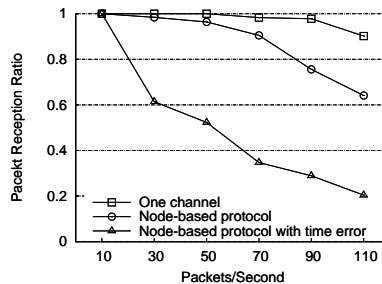Fig. 4. Packet reception ratios of different channels

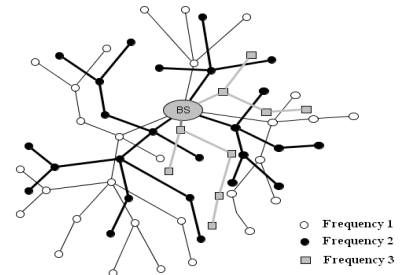Fig. 5. Packet reception ratio vs. Source Data rate

Fig. 6. The conceptual design of TMCP

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), 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 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 results.

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 and Figure 3, 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.

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 that one 802.11 channel can potentially collide with four 802.15.4 channels. This problem is also studied in [11] [12]. 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 4, 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.

B. 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 5. 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 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.

IV. 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 III, 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 congestion 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 6. 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 inter-tree 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.

V. 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.

A. 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 which 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 [15] 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 1: The interference set of a node u is defined as $INT(u) = \{v | v \in D(v, I_v)\}$, 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 as observed in [16], and interference sets of nodes may change during 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 [16], they are different in real networks. Furthermore, we use the assumption from the protocol interference model [17], 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. Second, since our interference model is receiver-centric and leaf nodes are not receivers for data collection traffic, the interference of a tree is the maximum interference value among all *non-leaf* nodes.

Definition 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 7 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 trivial. In fact, it implies that if a node has a large interference value, we can
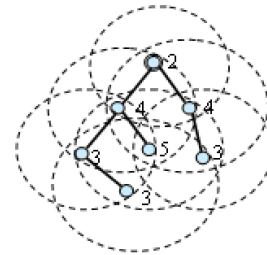

Fig. 7. A tree with 7 nodes. Each node is labeled with the interference value. The intra-tree interference value of the tree is 4

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.

Theorem 1: The PMIT problem is NP-Complete.

We prove that the PMIT problem is NP-hard by reducing the k-coloring problem to it. Readers can see the proof in [18].

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.

B. 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 first applies a Breadth-First search algorithm to compute a fat tree rooted at the base station [18]. There are two important properties of the fat tree. First, nodes keep their height and have multiple parents on the fat tree. Second, 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 a 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 joins 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 r, which demonstrates the correctness of the algorithm. The following theorem states the time complexity of the algorithm.

Theorem 2: The time complexity of the Greedy PMIT algorithm is $O(d \times k \times n^2)$, where d is the diameter of the

---

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

  for each node u **do**

    $c_u = 0$; $p_u = null$;

  end for

  $level = 1$;

  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 for

    $level + +$;

  until $level >$ the maximum height of the fat tree

(a) Interference vs. Node density

(b) Interference vs. Channels

Fig. 8. Performance Evaluation of the PMIT algorithm

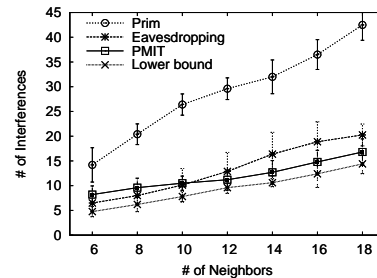graph, n is the number of nodes, and k is the number of channels.

Proof: The time complexity of constructing a FAT tree is $O(d \times \Delta \times n)$, where Δ is the maximum degree in the graph. In the 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 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 particular fat tree. Therefore, this partitioning algorithm does not require extra transmissions and does not increase energy consumption during data collection.
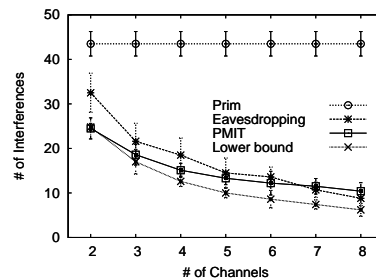
This algorithm can be easily modified to a distributed algorithm because it only needs a local search at each node. First, nodes can construct a fat tree by broadcasting messages. During channel allocation, nodes make their own decision based on message from their parents, and notify their children. 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.

C. 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 $10 \sim 35m$ and interference range is always 1.5 times 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 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. Second, we implement the Eavesdropping channel assignment method proposed in [2]. 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 experiments, 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 always gets around 1/3 the interference of the Prim's algorithm with a single channel, which shows that our algorithm efficiently utilizes 3 channels to decrease interferences. Comparing with the 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
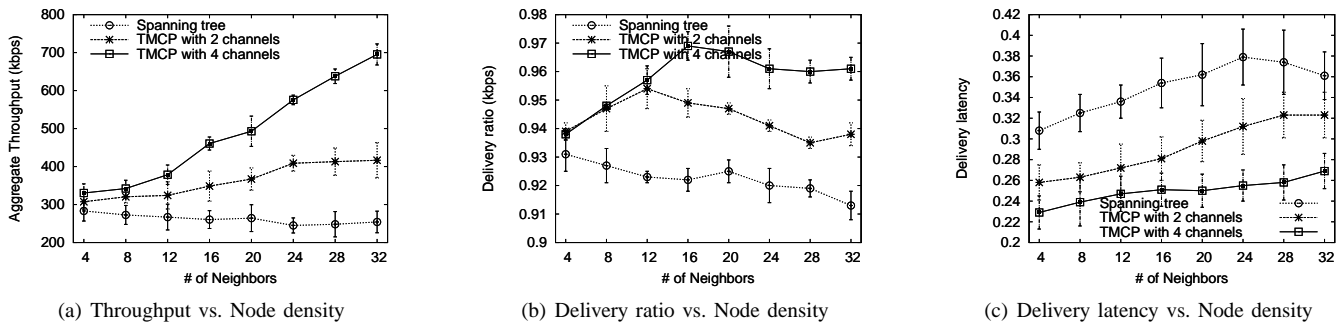
Fig. 9. Performance with different node density

greedy algorithm outperforms the Eavesdropping scheme, for example when the density is 18, it gets 17% less interferences than the Eavesdropping scheme. The reason is two-fold. First, when the density is large, there are not enough channels for nodes within two hop neighbors, so Eavesdropping has 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. Second, 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 computes less interferences than the Eavesdropping scheme, especially, when only 2 channels can be used, our algorithm has 24% less interferences than the Eavesdropping scheme, and 51% less than a single channel. With more channels, performance of the two schemes become closer. When there are 8 channels, the Eavesdropping scheme has 18% less interference than our greedy algorithm. Comparing with the lower bound, we see that with a small number of channels, our algorithm computes almost the same number of interferences as the lower bound.

VI. PERFORMANCE EVALUATION OF TMCP

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

A. 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 V-C, where the communication range is 10~40m 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 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
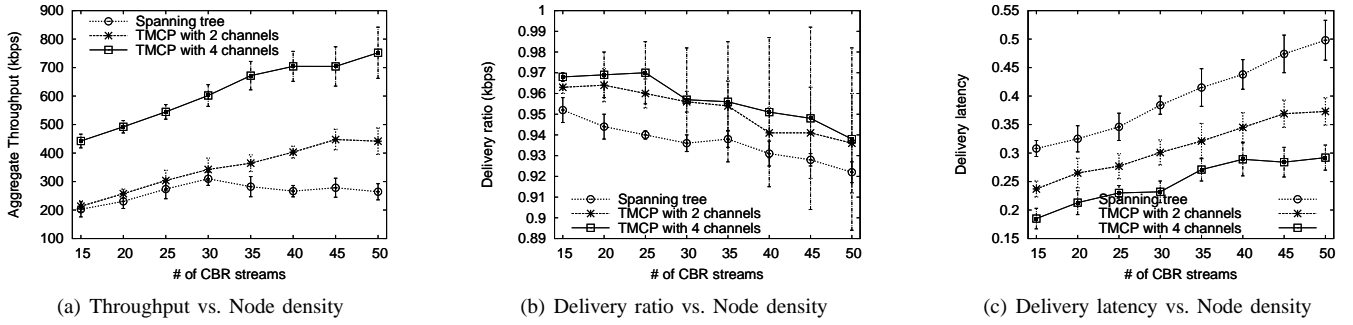
(a) Throughput vs. Node density   (b) Delivery ratio vs. Node density   (c) Delivery latency vs. Node density

Fig. 10.　Performance with different network workloads

(a) Throughput   (b) Delivery ratio   (c) Energy Consumption
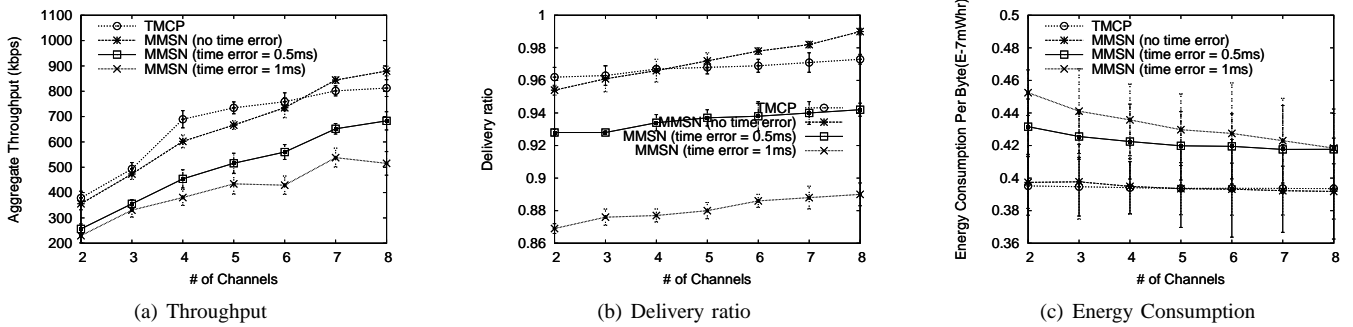
Fig. 11.　Performance comparison of TMCP and MMSN

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.

Lastly, we compare TMCP with MMSN [2], 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 III, 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 evaluation results in section V-C, 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 V-B, 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.

B. Evaluation in a Real Test Bed
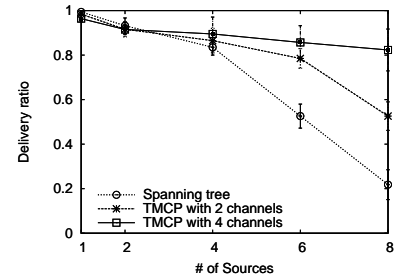
(a) Delivery ratio vs. Number of Sources
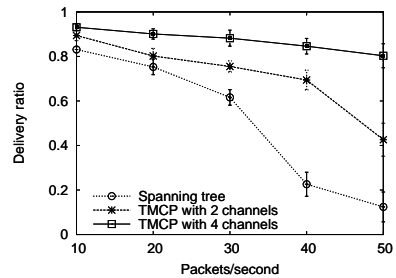
(b) Delivery ratio vs. Data rate

Fig. 12.　Evaluation in a test bed

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 IV 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 a real sensor network.

VII. CONCLUSION AND FUTURE WORK

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.

In the future, we plan to extend TMCP in two directions. First, we want to design a new channel assignment algorithm which not only considers interference conditions, but also traffic distributions in networks; second, currently TMCP cannot adapt to dynamic conditions such as interference changes and nodes dying. We want to develop mechanisms to dynamically maintain the optimal network topology without globally re-running the whole algorithm. These extension can make TMCP more valuable for large and dense WSNs in practice.

REFERENCES

[1] J.Zhang, G. Zhou, S. Son, and J. A. Stankovic, "Ears on the Ground: An Acoustic Streaming Service in Wireless Sensor Networks," in *IEEE/ACM IPSN Demo Abstract*, 2006.
[2] G. Zhou, C. Huang, T.Yan, T. He, and J. A. Stankovic, "MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks," in *IEEE Infocom*, 2006.
[3] J. Zhang, G. Zhou, C. Huang, S. Son, and J. A. Stankovic, "TMMAC: An Energy Efficient Multi-Channel MAC Protocol for Ad Hoc Networks," in *IEEE ICC*, 2007.
[4] X. Chen, P. Han, Q. He, S. Tu, and Z. Chen, "A Multi-Channel MAC Protocol for Wireless Sensor Networks," in *IEEE CIT*, 2006.
[5] J. So and N. Vaidya, "Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver," in *ACM MobiHoc*, 2004.
[6] J. Li, Z. J. Haas, M. Sheng, and Y. Chen, "Performance Evaluation of Modified IEEE 802.11 MAC for Multi-Channel Multi-Hop Ad Hoc Network," in *IEEE AINA*, 2003.
[7] A. Tzamaloukas and J.J. Garcia-Luna-Aceves, "A Receiver-Initiated Collision-Avoidance Protocol for Multi-Channel Networks," in *IEEE InfoCom*, 2001.
[8] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks," in *ACM MobiCom*, 2004.
[9] R. Vedantham, S. Kakumanu, S. Lakshmanan, and R. Sivakumar, "Component based channel assignment in single radio, multi-channel ad hoc networks," in *ACM MobiCom*, 2006.
[10] H. K. Le, D. Henriksson, and T. F. Abdelzaher, "A control theory approach to throughput optimization in multi-channel collection sensor networks," in *ACM/IEEE IPSN*, 2007.
[11] G. Zhou, J. A. Stankovic, and S. Son, "Crowded Spectrum in Wireless Sensor Networks," in *IEEE EmNets*, 2006.
[12] M. Petrova, L. Wu, P. Mahonen, and J. Riihijarvi, "Interference Measurements on Performance Degradation between Colocated IEEE 802.11g/n and IEEE 802.15.4 Networks," in *IEEE ICN*, 2007.
[13] A. Mishra, V. Shrivastava, S. Banerjee, and W. Arbaugh, "Partially Overlapped Channels Not Considered Harmful," in *ACM SigMetrics*, 2006.
[14] "CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver," http://www.chipcon.com.
[15] M. Burkhart, P. V. Rickenbach, R. Wattenhofer, and A. Zollinger, "Does topology control reduce interference," in *ACM MobiCom*, 2004.
[16] G. Zhou, T. He, J. A. Stankovic, and T. F. Abdelzaher, "Radio Interference Detection in Wireless Sensor Networks," in *IEEE INFOCOM*, 2005.
[17] P. Kyasanur and N. Vaidya, "Capacity of multi-channel wireless networks: Impact of number of channels and interfaces," in *ACM MobiCom*, 2005.
[18] Y. Wu and J. A. Stankovic, "Exploiting multi-channels in Wireless Sensor Networks," Tech. Rep., UVA, 2007, http://www.cs.virginia.edu/ yw5s/multi-channel.pdf.