Distributed Channel Allocation Protocols for Wireless Sensor Networks

Abusayeed Saifullah, You Xu, Chenyang Lu, and Yixin Chen

Abstract—Interference between concurrent transmissions can cause severe performance degradation in wireless sensor networks (WSNs). While multiple channels available in WSN technology such as IEEE 802.15.4 can be exploited to mitigate interference, channel allocation can have a significant impact on the performance of multi-channel communication. This paper proposes a set of distributed protocols for channel allocation in WSNs with theoretical bounds. We first consider the problem of minimizing the number of channels needed to remove interference in a WSN, and propose both receiver-based and link-based distributed channel allocation protocols. Then, for WSNs with an insufficient number of channels, we formulate a fair channel allocation problem whose objective is to minimize the maximum interference (MinMax) experienced by any transmission link in the network. We prove that MinMax channel allocation is NP-hard, and propose a distributed link-based MinMax channel allocation protocol. Finally, we propose a distributed protocol for link scheduling based on MinMax channel allocation that creates a conflict-free schedule for transmissions. The proposed decentralized protocols are efficient, scalable, and adaptive to channel condition and network dynamics. Simulations based on the topologies and data traces collected from a WSN testbed of 74 TelosB motes have shown that our channel allocation protocols significantly outperform a state-of-the-art channel allocation protocol.

Index Terms—Wireless sensor network, multi-channel, channel allocation, distributed algorithm.

◆

1 INTRODUCTION

Interference between concurrent transmissions can cause severe performance degradation in wireless sensor networks (WSNs). Multi-channel communication is an attractive approach to reducing interference and enhancing spatial reuse. Since channels are a scarce resource in a WSN, channel allocation significantly influences the performance of multi-channel WSNs. It is particularly important in Time Division Multiple Access (TDMA) based WSNs where interfering transmissions scheduled at the same time slot must be assigned different channels. It is, therefore, important to allocate the channels to reduce interference and increase the number of concurrent transmissions.

Channel allocation has been widely studied for wireless ad-hoc networks. These protocols are not applicable to WSNs since the applications, routing, node resources, and network structure in WSNs are quite different from traditional ad-hoc networks. While channel allocation has also been studied for WSNs, most of them focus on simple heuristics [1]–[3] without any performance guarantee for channel hopping or just focus on centralized solutions [4], [5].

In this paper, we formulate optimal channel allocation as constrained optimization problems, and propose a set of distributed channel allocation protocols with theoretical bounds for WSNs. We first consider the problems of minimizing the number of channels needed to remove interference in a WSN for both receiver-based and link-based channel allocation. A receiver-based channel allocation is suitable for both CSMA/CA and TDMA protocols. A link-based channel allocation allows better spatial reuse due to the flexibility in assigning different channels to different senders, but it is more suitable for TDMA protocols under which the receiver can switch to channels according to the expected sender scheduled in each time slot. We present distributed protocols for both receiver-based and link-based channel allocation.

WSNs usually have a moderate number of channels (e.g., 16 channels specified IEEE 802.15.4), and noisy environments may further reduce the number of available channels due to blacklisting [6]. Therefore, there may not exist enough channels to remove all interference. Existing works on channel allocation with an insufficient number of channels usually consider receiver-based allocation and propose centralized heuristics [4], [5], [7]. A recently proposed distributed protocol for channel allocation in WSNs has addressed receiver-based allocation to minimize total interference suffered by all receivers [8]. In contrast, we formulate a link-based fair channel allocation problem whose objective is to minimize the maximum interference (MinMax) experienced by any transmission link in a WSN. The key advantage of the MinMax objective is that it can mitigate bottlenecks in a WSN where a node or link experiences excessive interference. We prove that MinMax channel allocation is NP-hard, and propose a distributed MinMax channel allocation protocol. Furthermore, since channel allocation cannot always resolve all transmission

---

- *The authors are with the Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130. E-mail: {saifullaha, yx2, lu, chen}@cse.wustl.edu*

conflicts due to an insufficient number of channels, it is complemented by a time slot assignment algorithm to create a conflict-free schedule. We propose a distributed protocol for link scheduling based on MinMax channel allocation. Our contributions are:

- We present distributed protocols for both receiver-based and link-based interference-free minimum channel allocation.
- We formulate a link-based fair channel allocation problem, called MinMax channel allocation, whose objective is to minimize the maximum interference experienced by any transmission link in a WSN, and prove it to be NP-hard.
- We propose a distributed protocol for MinMax channel allocation in WSN.
- We propose a distributed protocol for link scheduling based on MinMax channel allocation.

The proposed algorithms are efficient, scalable, and adaptive to channel condition and network dynamics. We provide the time complexity and performance bound of each algorithm. Simulations using the real topologies and data traces collected from a WSN testbed have shown that our protocols significantly outperform a state-of-the-art protocol [8].

2 RELATED WORK

Multi-channel MAC protocols have been extensively studied for wireless ad-hoc network [9]–[39]. However, there are some key differences between these existing protocols for traditional wireless ad-hoc network and the channel allocation protocols proposed in this paper for WSN as detailed below.

First, the protocols in [11], [15]–[24], [28], [32], [34]–[39] assume that the hardware is able to listen to multiple channels simultaneously. But each sensor device is usually equipped with a single radio transceiver (e.g., TelosB mote [40] with Chipcon CC2420 radio) that cannot transmit and receive at the same time, and cannot operate on different channels simultaneously. Second, the protocols in [12], [19], [20], [29], [30], [32], [37] involve heavy centralized computation such as linear programming [19], [37], mixed integer linear programming [12], and subgradient method [29], [30]. But a WSN has limited bandwidth (e.g. 250kbps in 802.15.4 network), and each sensor device has limited memory (e.g. 10KB in TelosB motes [40]) and limited processing power (8MHz MSP430 microcontroller in TelosB motes), making a WSN unsuitable for such heavy-weight computations. Third, the protocols in [9], [10], [13], [14] use RTS/CTS for channel negotiation. But, due to limited bandwidth in WSNs, the MAC layer packet size in WSNs is much smaller (typically 30$\sim$50 bytes) than that of general ad hoc networks (typically 512+ bytes). Hence, RTS/CTS control packets result in significant overhead for WSN, thereby making these protocols unsuitable for WSN.

Graph theory based multi-channel protocols for wireless ad-hoc networks studied in [18], [24]–[27], [31], [33], [38], [39] are most related to our work. The protocols in [18], [24], [31], [33], [38], [39] are distributed but assume that each node can listen to multiple channels simultaneously (as already discussed), while those in [25], [26] are based on single radio in each node but consider centralized solutions. The work in [27] that uses a distributed approach considering single radio in each node, and the work in [24] that uses a game theoretic approach are particularly related to our work. But, both works focus on maximizing link data rate instead of interference-free minimum channel allocation or minimizing the maximum interference, which are our focus.

In summary, none of the above protocols is applicable for WSNs since the applications, routing, and network structure in WSNs are quite different from traditional ad-hoc networks. For example, in contrast to traditional ad hoc networks designed to support general communication patterns and routes, WSNs are typically involved in monitoring applications requiring data collection with unique communication patterns and routing structures. Sensor nodes are prone to failures, and the network topology changes more frequently. Besides, sensor nodes mainly use broadcast communication paradigms whereas most traditional ad-hoc networks are based on point-to-point communications. In a WSN, nodes are usually densely deployed, and the number of nodes can be several orders of magnitude higher than that in a traditional wireless ad-hoc network.

Channel allocation has also been studied for WSN in recent years. MMSN [41] is an early multi-channel protocol proposed for WSN. MMSN ignores routing information for channel allocation. In contrast, we propose routing-aware channel allocation protocols that do not assign channels to the links not involved in traffic. Tree-based Multi-Channel Protocol (TMCP) proposed in [42] uses the distance-based interference model which does not hold in practice as shown by recent empirical studies [7]. TMCP has been extended in [7] to employ inter-channel RSS models for interference assessment in channel allocation [7]. All these protocols are *centralized*, and lack any performance bound. The protocols proposed in [1]–[3] use simple heuristics for channel hopping. These protocols do not address interference-free minimum channel allocation or minimizing the maximum interference, which are the focus of our work in this paper.

Interference-aware channel allocation based on graph-theory has been studied in [4], [5], [43] for WSN. But the work in [43] is designed for unit-disk graph. The work in [4] assigns a channel to each flow. The work in [5] shows that minimizing schedule length for multi-channel arbitrary network is NP-hard, and presents a constant factor approximation algorithm for unit-disk graph [5]. These algorithms are *centralized*. Due to frequent topology changes, distributed protocols are more suitable for WSNs.

A distributed game theory based protocol has been proposed in [8] for channel allocation in WSN. It addresses only receiver-based allocation, and minimizes total interference suffered by all receivers.

In contrast to existing channel allocation protocols for WSNs, we present distributed protocols for both receiver-based and link-based interference-free minimum channel allocation. The key novelty of our work lies in formulating a link-based fair channel allocation problem, called MinMax channel allocation, whose objective is to minimize the maximum interference experienced by any transmission link in a WSN. In addition, we prove that an optimal MinMax channel allocation is NP-hard. Furthermore, we propose a distributed protocol for MinMax channel allocation based on heuristic. We also propose a distributed protocol for link scheduling based on MinMax channel allocation. The key advantage of the MinMax objective is that it can mitigate bottlenecks in a WSN where a node or link experiences excessive interference.

3 NETWORK MODEL

A WSN consists of a set of sensor nodes. A node, called the *base station*, serves as the sink of the network. A *communication link* $e = (u, v)$ indicates that the packets transmitted by node u may be received by v. We assume that every communications link is symmetric. This assumption holds for WSNs relying on acknowledgement for reliable communication (e.g., WirelessHART networks [6] based on IEEE 802.15.4). An *interference link* $e = (u, v)$ indicates that u's transmission interferes with any transmission intended for v even though u's transmission may not be successfully received by v. Thus, any two concurrent transmissions that happen on the same channel are *conflicting* if there is an interference link from one's sender to the other's receiver. Several practical protocols [44], [45] exist that model interference in WSNs using Signal-to-Noise plus Interference Ratio (SNIR). A set of transmissions on the same channel is conflict-free if the SNIR of all receivers exceeds a threshold. For example, RID [46] is a distributed protocol for determining interference links in a WSN based on Received Signal Strength (RSS) measurements.

We model a WSN as an *Interference-Communication* (IC) graph, a notion introduced in [47]. In the IC graph $G = (V, E)$, V is the set of sensor nodes (including the sink s); E is the set of communication or interference links between the nodes. A subset of the communication links forms the routing tree that is used for data collection at the sink. Let $E_T \subseteq E$ denote the set of links in the routing tree. Any link $e = (u, v)$ in E_T indicates that v is the parent of u. For any node u, we use p_u to denote its parent in the routing tree. Since the transmissions along non-tree links do not aim at the receiver, every non-tree link (that is not a part of the routing tree) is an *interference link*. $E_I = E - E_T$ is the set of all interfering links in the IC graph. Any link $e = (u, v)$ in E_I indicates an interference link from u to v. A node cannot both send and receive at the same time, nor can it receive from more than one sender at the same time. The set of channels available in the WSN is denoted by M. We use m to denote $|M|$ i.e. the total number of channels. The channels are numbered through 1 to m. In this work, we particularly focus on TDMA based WSN.

4 PROBLEM FORMULATION

In *receiver-based channel allocation*, each sensor node is assigned a fixed channel to receive message; the neighbors which have messages to deliver to it should use this channel to send. In this allocation, the leaves (i.e., nodes without children) in the routing tree do not receive any message, and hence are not assigned any channel. Let the nodes that receive message (i.e., the nodes other than leaves) be denoted by $R \subset V$. Therefore, the *receiver-based channel allocation* is a function $f : R \mapsto M$, where M is the set of channels.

In *link-based channel allocation*, every link $e \in E_T$ is assigned a channel so that every transmission along that link happens on that channel. In contrast to receiver-based channel allocation, here for the same receiver, different senders can use different channels, thereby providing more flexibility in avoiding interference. Any *link-based assignment* is a function $f : E_T \mapsto M$. Since every node has unique sending link, a link-based channel assignment function can also be defined as $f : V - \{s\} \mapsto M$, where s is the root (i.e., the sink) of the routing tree and it does not send to anyone. Thus every sender in the network is assigned a channel. For reception, the receiver uses the same channel that the sender uses to transmit.

Interference caused by siblings (in the routing tree) to each other cannot be resolved by channel assignment because the shared parent cannot receive from more than one of them at the same time. This can be resolved through a time-slot assignment. Therefore, for channel allocation purpose, we are concerned only about interference through non-tree links E_I (that are not parts of the routing tree), and simply use the term 'conflict' to denote the interference through these links. In the worst-case, the maximum number of transmissions that can be conflicting through interference links with a transmission along link (u, v) is equal to the total number of incoming interference links of v and outgoing interference links of u. Thus, we define *conflict of transmission link* (u, v) or *conflict of node* u as the maximum number of transmissions that can be conflicting through interference links with a transmission of node u. For a node u, in a channel assignment f, we use $C(u, f)$ to denote its conflict, and define as follows (where p_u is the parent of u):

$$C(u, f) = \big|\{z \big| \big((z, p_u) \in E_I \vee (u, p_z) \in E_I\big) \wedge f(z) = f(u)\}\big|$$

That is, $C(u, f)$ counts the total number of nodes that use the same channel as u's and that has either an outgoing interfering link to the parent of u or an incoming interfering link to its parent from u. The interference a node receives is not only decided by the number of interference sources, but also by the strengths of interfering signals. However, to develop an efficient and distributed approach, we consider the metric $C(u, f)$ as the signals can be determined based on a threshold on signal strength. $C(u, f)$ is an effective metric and can be used for effective channel allocation because the total number of interfering signals has strong correlations with transmission failures and retries. The higher the value of $C(u, f)$, the more transmissions that u's transmission may conflict with. Namely, the more interfering links a receiver hears, the more retries a message needs to be successfully received by that receiver, thereby incurring longer delay. For example, in [8], the total number of interfering links a receiver hears was shown to be approximately linear with the total number of retries a message needs to be successfully received by that receiver.

Problem 1: Receiver-based interference-free channel allocation. The number of channels is usually fixed and limited in practice. Our first objective is to minimize the total number of channels to remove all interferences in the IC graph $G = (V, E)$. Let $f(R)$ denote the range of function $f : R \mapsto M$, i.e., the set of channels used in f. In *receiver-based interference-free channel allocation*, our objective is to determine a channel assignment $f : R \mapsto M$ so as to

$$\begin{aligned} &\text{Minimize} \quad |f(R)| \\ &\text{subject to} \quad C(u, f) = 0, \ \forall (u, v) \in E_T \end{aligned}$$

Problem 2: Link-based Interference-free channel allocation. While receiver-based channel allocation is simple in the sense that a receiver can avoid switching to different channels for different senders, it can end up with extra interference for some transmission link, thereby limiting the communication possibilities for some nodes. Such a limitation of receiver-based channel allocation can be significantly overcome by adopting link-based allocation. In *link-based interference-free channel allocation*, our objective is to determine a channel assignment $f : V - \{s\} \mapsto M$ to

$$\begin{aligned} &\text{Minimize} \quad |f(V - \{s\})| \\ &\text{subject to} \quad C(u, f) = 0, \ \forall u \in V - \{s\} \end{aligned}$$

Problem 3: Minimizing Maximum interference (MinMax) channel allocation. The number of channels required to remove all interference may be greater than the total available channels. Therefore, when the available channels are not sufficient to remove all interference, a fair channel allocation is the one that minimizes the maximum interference experienced by any transmission link in G. Since link-based channel allocation allows better spatial reuse of channels, we use link-based allocation for MinMax objective. In MinMax channel allocation, our objective is to determine a link-based channel assignment $f : V - \{s\} \mapsto M$ so as to

$$\begin{aligned} &\text{Minimize} \quad \max\{C(u, f) | u \in V - \{s\}\} \\ &\text{subject to} \quad f(u) \in M, \ \forall u \in V - \{s\} \end{aligned}$$

Problem 4: Link scheduling. After MinMax channel allocation, a conflict-free schedule is required to avoid transmission conflicts through both tree (transmission) links and the residual interference links. This needs to be resolved through time slot assignment. That is, after channel allocation in phase 1, we consider the link scheduling in phase 2. While it may be possible to combine two phases into one, such an approach complicates the optimization problem as the solution space becomes larger. Instead, decoupling it into two phases simplifies the optimization problem for conflict resolution. Hence, in our solution approach, channel allocation is done in the first phase, which is followed by a time slot assignment in the second phase. In TDMA, a transmission needs one time slot, and a sequence of time slots forms a *frame*. The frame is repeated continuously. Every link is assigned a relative time slot within a frame and it is activated at that slot of the frame. Therefore, here our objective is to schedule all links to minimize the frame length. Thus, for link scheduling, after MinMax channel allocation, our objective is to determine a time slot assignment $g : E_T \mapsto \{1, 2, 3, \cdots\}$ so as to

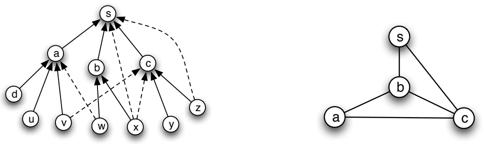
$$\text{Minimize} \quad |g(E_T)|$$

5 INTERFERENCE-FREE CHANNEL ALLOCATION

5.1 Receiver-based Channel Allocation

We first consider receiver-based channel allocation to minimize the number of channels to eliminate all interference. This problem has been proven to be NP-hard in [5]. In the following, we provide a distributed algorithm based on *vertex-coloring* for this problem.

Two receivers are called *interfering* if the transmission of some child of one receiver is interfered by the transmission of some child of the other receiver. In order to eliminate all interference, every receiver must be assigned a channel that is different from all of its interfering receivers' channels. Therefore, for the given IC graph $G = (V, E)$, we can assume a *receiver-based conflict-graph*, denoted by $G_R = (R, E_R)$, that consists of all receivers R as nodes, and an edge (in E_R) between every interfering receiver pair. For example, Figure 1(b) shows the receiver-based conflict-graph of the IC graph of Figure 1(a). In an IC graph, we use dotted lines and solid lines to indicate interference links and transmission links, respectively. Considering every *channel* as a *color*, vertex-coloring of G_R provides the solution for receiver-based interference-free channel allocation in G to minimize the number of



(a) IC graph G (b) Conflict graph G_R

Fig. 1. IC graph and receiver-based conflict graph

channels (colors). To construct the conflict graph, each node needs to know the channel conditions between itself and other nodes. This requirement can be met in practice, even in scenarios with fast channel fading or network dynamics, since the fast-fading and fluctuating channels can be blacklisted at deployment time and infrequent maintenance cycles in wireless sensor networks, e.g., as specified in the WirelessHART standard for industrial wireless sensor networks [6]. Our approach can leverage existing algorithms specifically designed to detect conflict graphs efficiently in wireless sensor networks [45], [46]. These methods use RSS measurements and determine and store conflict graphs in a distributed fashion: a node only knows its incoming/outgoing communication and interference edges based on SNIR. Hence, conflict graph construction method is distributed and efficient in practice.

For vertex-coloring in distributed manner, the best known deterministic algorithm [48] employs $D^{1+O(1)}$ colors, where D is the maximum degree of the given graph. The distributed methods for vertex-coloring available in the literature of theoretical computer science [48] involve multiple phases. A phase starts only after its previous phase converges. Since the WSN devices are characterized by low power and resources, these algorithms are too heavy-weight for WSNs. Here we present a simple and deterministic distributed protocol suitable for WSNs, which can employ at most $\Delta_R + 1$ channels, with Δ_R being the degree of the receiver-based conflict graph.

Let $N_R(u)$ denote the neighbors of node u in G_R. In our distributed method, every node $u \in R$ has to communicate with its neighbors $N_R(u)$ in G_R. Note that two neighbors u and v in G_R may not be one-hop away from one another in IC graph G. In such cases, u and v communicate with one another by increasing their transmission power like what is done in [46], [47]. If this is not possible, communication between u and v is done through the end-to-end route between u and v. Channel allocation is done iteratively and every round consists of communication between the neighbors in G_R. In every communication round, all nodes use the same channel. Once the algorithm converges, every node uses the channel determined by the algorithm for subsequent communication. The distributed receiver-based interference-free channel allocation protocol consists of the following steps comprising a procedure that is invoked iteratively:

1) In the beginning, every node $u \in R$ is assigned channel 1 (the smallest numbered channel). In every round, each node $u \in R$ broadcasts a message containing its ID and chosen channel to its neighbors $N_R(u)$.
2) Considering the current channel allocation among neighbors $N_R(u)$, every node u repeatedly switches to the smallest channel not used by any of its neighbors. Two neighbors cannot switch channels simultaneously. If two neighbors in G_R want to switch at the same time, the node with the smallest ID wins (as a local agreement among neighbors) and switches channel.
3) After choosing the channel, each node u broadcasts its chosen channel in a message (that also contains its ID) to its neighbors $N_R(u)$ in G_R.
4) The procedure is repeated until every node has chosen a channel different from its neighbors in G_R and cannot choose a channel that is smaller than its current channel.

The procedure in each node is shown as Algorithm 1 in Appendix. As stated before, two neighbors in G_R do not execute the procedure simultaneously.

The above algorithm converges when every node in G_R has chosen a channel different from those of its neighbors $N_R(u)$, and cannot switch to a smaller channel. In every round, the total number of messages that are sent or received by a node u is $O(|N_R(u)|)$. Theorem 1 proves the convergence of the algorithm. Theorem 2 shows that the algorithm requires at most $\Delta + 1$ channels, where Δ is the maximum degree of G. For the network shown in Figure 1(a), the channels selected by the nodes in different rounds (up to the convergence) of Receiver-based channel assignment are shown in Table 1 in Appendix for any $m > 3$.

Theorem 1: Receiver-based interference-free channel allocation algorithm converges in $|E_I|$ rounds, where $|E_I|$ is the total number of interfering links in G.

Proof: See Appendix. □

Theorem 2: Receiver-based interference-free channel allocation algorithm requires at most $\Delta + 1$ channels, where Δ is the maximum degree in G.

Proof: See Appendix. □

5.2 Link-based Channel Allocation

Receiver-based allocation can end up with extra interference for some transmission link, thereby limiting the communication possibilities for some nodes. As a result, when all transmission conflicts are completely resolved through a time slot assignment phase, the schedule length becomes longer if a receiver-based allocation is adopted. This limitation can be significantly overcome by adopting a link-based allocation since it allows better spatial reuse. This is illustrated in Figure 2 through a simple example considering m= 2. The number in the rectangle beside every receiver shows its assigned channel. Under this receiver-based allocation, every time node w transmits, none of a's children should transmit. This problem can be avoided using a link-based channel allocation instead (as shown beside the links) by assigning channel 1 to node w, and channel 2 to node x.

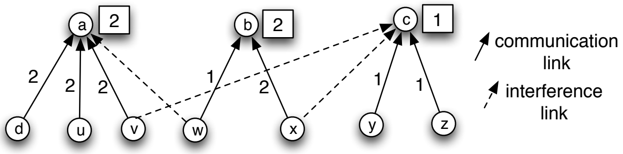


Fig. 2. Link-based channel allocation

A reduction similar to the one used in [5] (that proves that receiver-based interference-free channel assignment is NP-hard) can also be used to prove that link-based interference-free channel allocation is NP-hard as shown in Theorem 3.

Theorem 3: Given a routing tree T on an IC graph $G = (V, E)$, and a total of m channels, it is NP-complete to decide whether there exists some channel allocation f to the links in T such that G becomes interference-free.

Proof: See Appendix. □

Now we present a distributed algorithm for link-based channel allocation to minimize the number of channels in order to eliminate all interfering links. This approach is also similar to the distributed vertex-coloring adopted for receiver-based allocation in the previous subsection.

Two senders in G are called *interfering* if one's transmission is interfered by the other. In order to eliminate all interference, every sender's transmission link must be assigned a channel that is different from those of its interfering senders. Therefore, for the IC graph $G = (V, E)$, we can assume a *link-based conflict-graph*, denoted by $G_L = (V - \{s\}, E_L)$, that consists of all senders $V - \{s\}$ as nodes, and an edge (in E_L) between every interfering sender pair. For example, Figure 3 shows the link-based conflict-graph of the IC graph of Figure 1(a). Considering every *channel* as a *color*, vertex-coloring of G_L provides the solution for link-based interference-free channel allocation in G to minimize the number of channels (colors).

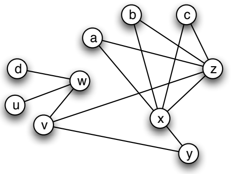


Fig. 3. Link-based conflict graph G_L of G

Using the same distributed algorithm as the one used for receiver-based channel allocation in the preceding subsection, we now vertex color graph G_L. The procedure in each node is shown as Algorithm 2 in Appendix. As stated before, two neighbors in G_L do not execute Algorithm 2 simultaneously. If two neighbors in G_L want to execute simultaneously, the node with the smallest ID wins (as a local agreement among neighbors) and executes to switch its channel. When the entire distributed algorithm converges, every sender (i.e., every sender's transmission link) is assigned a channel that is different from any interfering sender's channel. This algorithm converges within $|E_I|$ rounds, and employs at most $\Delta_L + 1$ channels, where Δ_L is the maximum degree in G_L (proofs are similar to those of Theorems 1 and 2). For the network shown in Figure 1(a), the channels selected by the nodes in different rounds (up to the convergence) of Link-based channel assignment are shown in Table 2 in Appendix for any $m > 3$.

6 MINMAX CHANNEL ALLOCATION

Note that WSNs usually have a moderate number of channels (e.g., 16 channels for WSNs based on IEEE 802.15.4), and noisy environments may further reduce the number of available channels due to blacklisting [6]. Therefore, there may not exist enough channels to remove all interference using the algorithms presented in the previous section. In such a situation, we adopt MinMax channel allocation whose objective is to minimize the maximum interference experienced by any transmission link across the network. Since receiver-based allocation may not minimize the maximum interference experienced by a transmission link (Subsection 5.2), we follow a link-based approach for MinMax channel allocation.

We first prove that MinMax allocation is NP-hard by showing that its decision version is NP-complete.

Theorem 4: Given a routing tree T on an IC graph $G = (V, E)$, m channels, and an integer k, it is NP-complete to decide if there exists a channel allocation f to the links in T such that the maximum conflict in G is at most k.

Proof: See Appendix. □

Now we present a distributed algorithm for MinMax channel allocation. In the protocol, every node needs to communicate with its neighbors in link-based conflict graph G_L (see Subsection 5.2 and Figure 3 for G_L) to compute its conflict. For any node u, the set of its neighbors in G_L is denoted by $N_L(u)$. Communication in the neighborhood in G_L is done based on the same approach presented in the previous section. Distributed MinMax algorithm consists of the following procedure that is invoked iteratively:

1) Before the invocation of the procedure, every node $u \in V - \{s\}$ is assigned a random channel in the range between 1 and m. Every node $u \in V - \{s\}$ broadcasts a message containing its ID and channel to its neighbors $N_L(u)$ in G_L.
2) Considering the current channel allocation among the neighbors in G_L, every node calculates its conflict $C(u, f)$ and broadcasts again to the neighbors $N_L(u)$.
3) For each node u, once it receives the message containing $C(v, f)$ from each neighbor v in G_L, node u calculates its conflict $C(u, f)$ on every channel. Any channel used by a neighbor v with $C(v, f) > C(u, f)$ is considered unavailable at u. That is, node u excludes all channels used by the neighbors with higher conflicts in the current round. This is done because switching to such a channel increases the neighbor's conflict which may increase the maximum conflict in the

network. Among the available channels, node u switches to the channel that results in the smallest $C(u, f)$, breaking ties arbitrarily. Two neighbors cannot switch channels simultaneously. If two neighbors want to switch at the same time, the node with the smallest ID wins.

4) After choosing the channel, every node broadcasts its chosen channel to its neighbors in G_L.
5) The procedure repeats as long as some node u can decrease $C(u, f)$ using its available channels.

The procedure in each node is shown as Algorithm 3 in Appendix. As stated before, two neighbors in G_L do not execute Algorithm 3 simultaneously. If two neighbors in G_L want to execute simultaneously, the node with the smallest ID wins (as a local agreement among neighbors) and executes to switch its channel.

In each communication round, all nodes use the same channel for communication. Once the algorithm converges, every node uses the channel determined by the algorithm for subsequent communication. Each node u needs to send or receive $O(|N_L(u)|)$ messages in a round. The algorithm converges when no node can decrease its conflict using its available channels. Theorem 5 proves its convergence. For the network shown in Figure 1(a), the channels selected by the nodes in different rounds (up to the convergence) of Link-based channel assignment are shown in Table 3 in Appendix considering $m = 2$.

Theorem 5: MinMax Channel Allocation converges in $|E_I|$ rounds, where $|E_I|$ is the total number of interfering links in G.

Proof: See Appendix. □

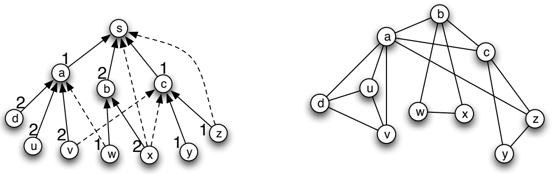
Theorem 6: Upon MinMax Channel Allocation, the maximum conflict in G is at most $\lfloor \frac{C_{\max}}{m} \rfloor$, where $C_{\max}$ is the maximum conflict in G under single channel.

Proof: See Appendix. □

The key advantage of the MinMax objective is that it can mitigate bottlenecks in a WSN where a node or link experiences excessive interference. The simulation results (presented in Section 8) indicate that the MinMax objective is more effective than minimizing the total interference in the network in terms of critical network metrics such as latency.

7 DISTRIBUTED LINK SCHEDULING

Note that channel allocation cannot resolve all transmission conflicts in a WSN due to two reasons. First, the number of available channels is limited and may not suffice to remove all interference. Second, each WSN device is equipped with a half-duplex radio that prevents a node from both transmitting and receiving at the same time, and also prevents reception from two senders simultaneously. Therefore, a channel allocation is complemented by a time slot assignment. Namely, any two conflicting transmissions are assigned different time slots. While this can be achieved through a joint channel allocation and time slot assignment, performing channel allocation and time slot assignment in two different phases simplifies this optimization problem. In this section, we present a distributed algorithm for time slot assignment after MinMax channel allocation. Namely, we first perform MinMax channel allocation. Then, we perform a time slot assignment that avoids transmission conflicts through both tree links and the residual interference links to create a conflict-free schedule.



(a) G (b) Schedule conflict graph G_S

Fig. 4. IC graph and schedule conflict graph

In the time slot assignment algorithm, every link is assigned a relative time slot in a frame, and the link is activated at that slot of the frame. The frame is repeated continuously. Note that, after MinMax channel allocation, the network can still be considered as a new IC graph with reduced interference. Therefore, a proof similar to Theorem 3 implies that scheduling all links to minimize the frame length is NP-hard. We provide a distributed method for time slot assignment that minimizes the frame length.

To resolve the conflict through both tree links and residual interference links after MinMax channel allocation, we determine a schedule conflict graph G_S of IC graph G as follows:

- Ignore all interfering links that are removed by MinMax channel allocation.
- Add links between siblings. The links between parent and children remain unchanged.
- For every interfering link (u, v) from u to v that still exists after channel allocation f, add a link from u to every child z of v with $f(z) = f(u)$.

For the IC graph G shown in Figure 1(a), let Figure 4(a) shows the channel allocation, where the number beside a sender shows its assigned channel. Then Figure 4(b) shows its schedule conflict graph G_S. In a TDMA schedule, any two nodes that are neighbors in G_S must be scheduled on different time slots. We use the same distributed algorithm as the one used for interference-free channel allocation. We run the algorithm considering schedule conflict graph G_S. Now, instead of channel, we allocate a time slot to every node in G_S. Every node starts with slot 1. In each round, the nodes switch to the smallest slot not assigned to any neighbor in G_S. The maximum time slot assigned to a node indicates the length of the frame, since the frame will repeat after this slot.

Theorem 7: The frame length determined by the distributed link scheduling algorithm is at most $\lfloor \frac{C_{\max}}{m} \rfloor + \Delta_T + 1$, where $C_{\max}$ is the maximum conflict in G under single channel, Δ_T is the maximum degree of the routing tree.

Proof: See Appendix. □

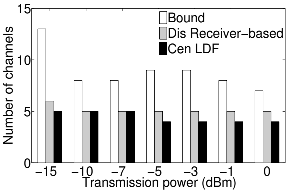
8 EVALUATION

We evaluate our channel allocation protocols through simulations based on the topologies of a WSN testbed [49] spread over two buildings of Washington University. The testbed consists of 74 TelosB motes, each equipped with a Chipcon CC2420 radio compliant with IEEE 802.15.4. We have developed a discrete-event simulator that operates based on interference data traces collected from the testbed. Figure 11 in Appendix shows the interference and communication edges on the testbed when every node's Tx power is set to -5dBm. We have collected 7 sets of data traces at 7 transmission (Tx) power levels: -15, -10, -7, -5, -3, -1, 0, all nodes operating on channel 26. Interference links are determined based on SNIR using RID protocol [46] (detailed in Appendix). The routing tree is constructed based on high quality links.

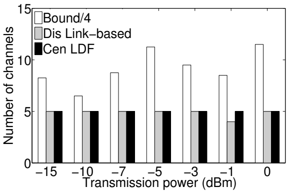
We also evaluate scalability of our protocols using random topologies. A random network is generated with an edge-density of 50%, i.e. with $n(n-1)50/200$ edges for a network with n nodes. Packet reception rate (PRR) along a link is assigned randomly in a range $[0.60, 1.0]$. A node with the highest degree is selected as the sink. A subset of links forms the routing tree. All other links are interference links.

8.1 Interference-free Channel Allocation

Figure 5 shows the number of channels required in our interference-free channel allocation (1 run) on testbed topologies at different Tx power. For receiver-based channel allocation (Figure 5(a)), our protocol requires no more than 6 channels (marked as 'Dis Receiver-based' in the figure) in every topology, and these values are less than the theoretical upper bound. We compare the results against a well-known centralized heuristic, called *Largest Degree First* (LDF) [5] (where a node is assigned the first available frequency in non-increasing order of degrees). While LDF is inherently more effective at the cost of centralized behavior, the figure indicates that the numbers of channels required by the centralized LDF and that by our distributed protocol vary at most by 1. For the link-based allocation (Figure 5(b)), the number of channels required by our protocol is much less than its theoretical bound.



(a) Receiver-based allocation



(b) Link-based allocation

Fig. 5. Channel allocation on testbed topologies to remove all interferences

8.2 MinMax Channel Allocation

Now we evaluate the MinMax algorithm. We plot the maximum conflict among all transmission links and the average conflict per transmission link after channel allocation. Each data point is the average of 5 runs. We compare the results with that of GBCA [8], the only known distributed protocol that minimizes the total interferences in the network in a receiver-based allocation. We also compare the performance with a *centralized greedy approach* that works as follows. Every time it determines the link that experiences the maximum conflict. If there exists a link such that switching its channel to a different one decreases the maximum conflict, then it switches to that channel. Any sender that does not affect the maximum conflict switches to the channel that results in maximum decrease in its own conflict.

Figure 6 shows the performance of MinMax protocol on the testbed topology with -5 dBm Tx power under varying number of channels. It shows that the maximum conflict in GBCA using 2 channels is 27 while that in MinMax is only 13. The average conflict per link is 4.55 in GBCA, and 2.87 in MinMax. The centralized greedy heuristic results in a maximum conflict of 11, and an average of 2.85 per link. Both maximum and average conflict in GBCA are higher than those in MinMax allocation since GBCA does not aim to minimize the maximum conflict.

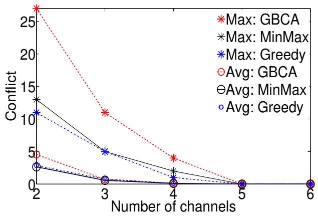
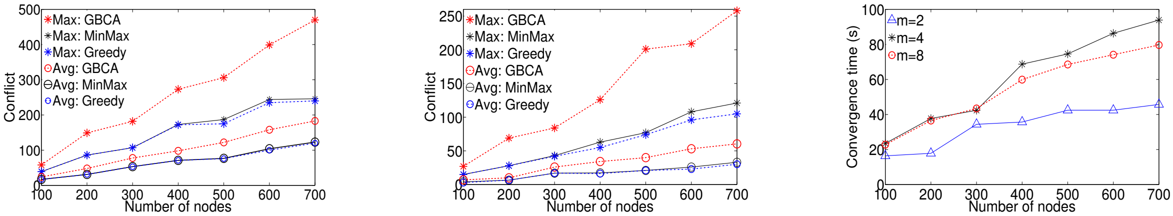


Fig. 6. MinMax channel allocation on testbed topology with -5 dBm Tx power

Figure 7 shows the performance of MinMax protocol on random topologies with different number of nodes. Figure 7(a) shows that the performance gap between GBCA and MinMax increases with the increase of network size. In a 700-node network with 2 channels, the maximum conflicts in GBCA, MinMax, and centralized greedy heuristic are 470, 246, and 240, respectively; the average conflicts per link in GBCA, MinMax, and centralized greedy heuristic are 183, 123, and 120, respectively. Figures 7(b) shows the similar results using 4 channels. The results show that MinMax protocol is highly effective in minimizing the maximum interference. It also results in less (compared to GBCA) average conflict which is very close to that of the centralized greedy algorithm. The MinMax protocol converges in 39s when the number of nodes is no greater than 300 (Figures 7(c)). For a 700-node network with 4 channels, it converges in 87s.

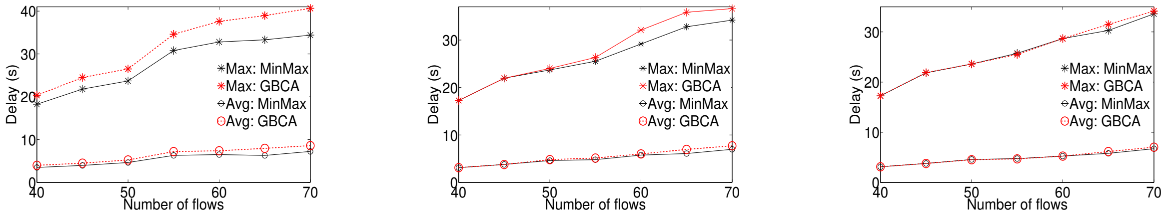
8.3 Latency under MinMax Channel Allocation

Here we implement our distributed link scheduling protocol after both MinMax and GBCA channel allocation. We consider TDMA with each time slot of 10ms (similar to WirelessHART [6] based on 802.15.4). For scheduling, each node periodically generates a packet resulting in a flow to the sink. All node have the same period. We record the maximum packet



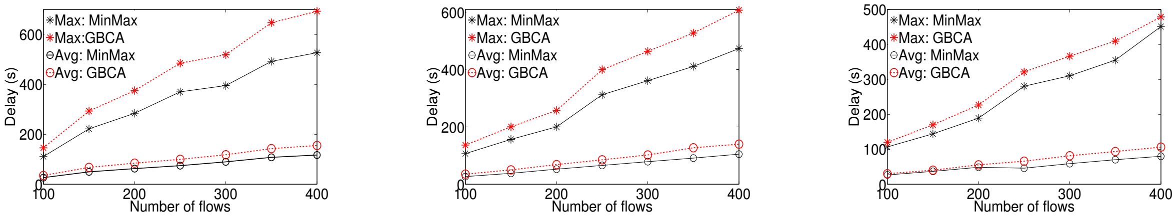
(a) Conflict using 2 channels (b) Conflict using 4 channels (c) Convergence time

Fig. 7. MinMax channel allocation on random topologies



(a) Delay using 2 channels (b) Delay using 3 channels (c) Delay using 4 channels

Fig. 8. Network performance on testbed topology at -5 dBm



(a) Delay using 2 channels (b) Delay using 4 channels (c) Delay using 8 channels

Fig. 9. Network performance on random topology of 400 sensor nodes

delay and the average packet delay in both protocols. The delay of a packet is counted as the difference between the time when it is delivered to the sink and the time when it was released at its source. In every run, a set of source nodes is selected randomly. Each data point is the average of 5 runs.

Figure 8 shows the delays under different number of flows on the testbed topology at -5 dBm Tx power. Figure 8(a) shows that the maximum delay among 70 flows under GBCA using 2 channels is 40.65s while that under MinMax allocation is only 34.40s. The average delay per packet is 8.60s under GBCA, and 7.24s under MinMax. In every setup, the 95% confidence interval remains within $\pm 1.7s$ for maximum delay, and within $\pm 0.43s$ for average delay for each protocol. The performance difference between GBCA and MinMax increases in larger networks as shown for random topologies of 400 nodes in Figure 9. For 400 flows and 2 channels (Figure 9(a)), the maximum delay is 692.61s under GBCA, and 526.68s under MinMax; the average delay per packet is 155.18s under GBCA, and 117.04s under MinMax. In every setup, the 95% confidence interval remains within $\pm 16.7s$ for maximum delay, and within $\pm 4.65s$ for average delay for each protocol. The results indicate that MinMax allocation is more effective in terms of packet latency.

8.4 Channel Allocation Message Overhead

Figure 10 shows the total number of messages used in MinMax channel allocation and link scheduling along with the total number of data transmissions. We used a setup similar to the preceding experiment. We consider networks with 101, 201, 301, and 401 nodes where in each case 1 node serves as the sink while all the other nodes are the sources of data. Every source node periodically generates a packet, all nodes having the same period. We compare the message overhead with the number of data messages in *one* cycle of data collection. Note that realistically channel allocation will be needed after *multiple* rounds of data collection. This result shows that even when compared to *one* cycle of data collection, channel allocation and link scheduling have lower message overhead. For example, the proportion of the total data transmissions to the total messages needed for channel allocation and link scheduling is 0.7 for 1 cycle of data collection in a network of 400 nodes. For c cycles of data collection, this fraction becomes $\frac{0.7}{c}$. Usually, upon channel allocation once, a multi-channel application (such as data collection) can run continuously based on that allocation until some network condition changes. For example, the message overhead will be below 3% of the data load if data allocation and scheduling are performed once every 25 rounds of data collection. The message overhead is therefore acceptable in many deployment scenarios.

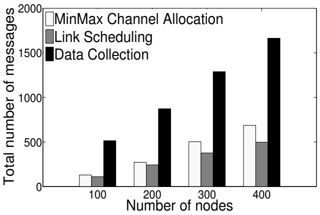


Fig. 10. Comparison of message cost for channel allocation and one round of data collection

9 CONCLUSION

We have proposed a set of distributed protocols for channel allocation in WSNs. For WSNs with an insufficient number of channels, we have proposed a fair channel allocation protocol that minimizes the maximum interference experienced by any transmission link. In the future, we plan to design traffic-aware protocol, and implement the results on testbeds.

REFERENCES

[1] H. Le, Henriksson, and T. Abdelzaher, "A practical multi-channel media access control protocol for wireless sensor networks," in *IPSN '08*.

[2] Y. Kim, H. Shin, and H. Cha, "Y-MAC: An energy-efficient multi-channel MAC protocol for dense wireless sensor networks," in *IPSN '08*.

[3] H. K. Le, D. Henriksson, and T. Abdelzaher, "A control theory approach to throughput optimization in multi-channel collection sensor networks," in *IPSN '07*.

[4] X. Wang, X. Wang, X. Fu, G. Xing, and N. Jha, "Flow-based real-time communication in multi-channel wireless sensor networks," in *EWSN '09*.

[5] A. Ghosh, O. Durmaz Incel, V. Anil Kumar, and B. Krishnamachari, "Multi-channel scheduling for fast aggregated convergecast in wireless sensor networks," in *MASS '09*.

[6] "WirelessHART," 2007, http://www.hartcomm.org.

[7] G. Xing, M. Sha, J. Huang, G. Zhou, X. Wang, and S. Liu, "Multi-channel interference measurement and modeling in low-power wireless networks," in *RTSS '09*.

[8] Q. Yu, J. Chen, Y. Fan, X. Shen, and Y. Sun, "Multi-channel assignment in wireless sensor networks: A game theoretic approach," in *INFOCOM '10*.

[9] J. So and N. H. Vaidya, "Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," in *MobiHoc '04*.

[10] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks," in *MobiCom '04*.

[11] S. Pediaditaki, P. Arrieta, and M. Marina, "A learning-based approach for distributed multi-radio channel allocation in wireless mesh networks," in *ICNP '09*.

[12] M. Alicherry, R. Bhatia, and L. E. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *MobiCom '05*.

[13] A. Raniwala and T. Chiueh, "Architecture and algorithm for an ieee 802.11-based multi-channel wireless mesh network," in *INFOCOM '05*.

[14] A. Tzamaloukas and J. G.-Luna-Aceves, "A receiver-initiated collision-avoidance protocol for multi-channel networks," in *INFOCOM '01*.

[15] K. Xing, X. Cheng, L. Ma, and Q. Liang, "Superimposed code based channel assignment in multi-radio multi-channel wireless mesh networks," in *MobiCom '07*.

[16] X. Lin and S. Rasool, "A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad-hoc wireless networks," in *INFOCOM '07*.

[17] P. Kyasanur, N. Vaidya, and M. Zorzi, "Capacity of multi-channel wireless networks: Impact of number of channels and interfaces," in *MobiCom '05*.

[18] A. Dhananjay, H. Zhang, J. Li, and L. Subramanian, "Practical, distributed channel assignment and routing in dual-radio mesh networks," in *SIGCOMM '09*.

[19] M. Alicherry, R. Bhatia, and L. E. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *MobiCom '05*.

[20] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *INFOCOM '06*.

[21] D. Wu and P. Mohapatra, "From theory to practice: Evaluating static channel assignments on a wireless mesh network," in *INFOCOM '10*.

[22] W. Cheng, X. Cheng, T. Znati, X. Lu, and Z. Lu, "The complexity of channel scheduling in multi-radio multi-channel wireless networks," in *INFOCOM '09*.

[23] J. Wang, Y. Fang, and D. Wu, "A power-saving multi-radio multi-channel mac protocol for wireless local area networks," in *INFOCOM '06*.

[24] L. Gao and X. Wang, "A game approach for multi-channel allocation in multi-hop wireless networks," in *MobiHoc '08*.

[25] V. Bhandari and N. Vaidya, "Connectivity and capacity of multi-channel wireless networks with channel switching constraints," in *INFOCOM '07*.

[26] ——, "Capacity of multi-channel wireless networks with random (c, f) assignment," in *MobiHoc '07*.

[27] R. Vedantham, S. Kakumanu, S. Lakshmanan, and R. Sivakumar, "Component based channel assignment in single radio, multi-channel ad hoc networks," in *MobiCom '06*.

[28] B. Raman, "Channel allocation in 802.11-based mesh networks," in *INFOCOM '06*.

[29] S. Merlin, N. Vaidya, and M. Zorzi, "Resource allocation in multi-radio multi-channel multi-hop wireless networks," in *INFOCOM '08*.

[30] M. Hajiaghayi, M. Dong, and B. Liang, "Optimal channel assignment and power allocation for dual-hop multi-channel multi-user relaying," in *INFOCOM '11*.

[31] A. Mishra, V. Shrivastava, D. Agrawal, S. Banerjee, and S. Ganguly, "Distributed channel management in uncoordinated wireless environments," in *MobiCom '06*.

[32] A. Raniwala, K. Gopalan, and T.-c. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *Mob. Comput. Commun. Rev.*, 2004.

[33] A. Raniwala and T.-c. Chiueh, "Architecture and algorithms for an ieee 802.11-based multi-radio wireless mesh networks," in *INFOCOM '05*.

[34] M. Shin, S. Lee, and Y. ah Kim, "Distributed channel assignment for multi-radio wireless networks," in *MASS '06*.

[35] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multi-radio unification protocol for ieee 802.11 wireless networks," in *BroadNets '04*.

[36] A. K. Das, H. M. K. Alazemi, R. Vijayakumar, and S. Roy, "Optimization models for fixed channel assignment in wireless mesh networks with multiple radios," in *SECON '05*.

[37] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *MobiCom '05*.

[38] D. Yang, X. Fang, and G. Xue, "Channel allocation in non-cooperative multi-radio multi-channel wireless networks," in *INFOCOM '12*.

[39] A. Naveed, S. S. Kanhere, and S. K. Jha, "Topology control and channel assignment in multi-radio multi-channel wireless mesh networks," in *MASS '07*.

[40] "TelosB," http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf.

[41] G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, and T. F. Abdelzaher, "MMSN: Multi-frequency media access control for wireless sensor networks," in *INFOCOM '06*.

[42] Y. Wu, J. Stankovic, He, and Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," in *INFOCOM '08*.

[43] K. Chowdhury, P. Chanda, D. Agrawal, and Q.-A. Zeng, "Dca- a distributed channel allocation scheme for wireless sensor networks," in *PIMRC '05*, vol. 2, sept. 2005, pp. 1297 –1301.

[44] R. Maheshwari, S. Jain, and S. R. Das, "A measurement study of interference modeling and scheduling in low-power wireless networks," in *SenSys '08*.

[45] S. Liu, G. Xing, H. Zhang, J. Wang, J. Huang, M. Sha, and L. Huang, "Passive interference measurement in wireless sensor networks," in *ICNP'10*.

[46] G. Zhou, T. He, J. A. Stankovic, and T. F. Abdelzaher, "RID: radio interference detection in wireless sensor networks," in *INFOCOM '05*.

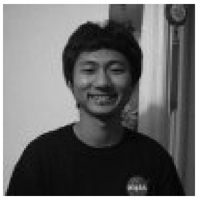
[47] O. Chipara, C. Lu, and J. Stankovic, "Dynamic conflict-free query scheduling for wireless sensor networks," in *ICNP '06*.

[48] L. Barenboim and M. Elkin, "Deterministic distributed vertex coloring in polylogarithmic time," in *PODC '10*.

[49] Testbed, http://mobilab.wustl.edu/testbed.

[50] K. Srinivasan and P. Levis, "RSSI is under appreciated," in *EmNets '06*.

Abusayeed Saifullah is a Ph.D. candidate in the Department of Computer Science and Engineering at Washington University in St Louis. Advised by Chenyang Lu, he is a member of the Cyber-Physical Systems Laboratory at Washington University. Abu's research focuses on real-time wireless sensor-actuator networks used in emerging cyber-physical systems, and spans a broad range of topics in wireless sensor networks, embedded systems, real-time systems, and multi-core parallel computing. He has received the best student paper awards at the 32nd IEEE Real-Time Systems Symposium (RTSS 2011) and at the 5th International Symposium on Parallel and Distributed Processing and Applications (ISPA 2007), and best paper nomination at the 18th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2012).



You Xu is a Ph.D. candidate of Computer Science at the Washington University in St. Louis. He received the M.Sc degree in computer science from the Washington University in St. Louis in 2009. His research interests include nonlinear optimization, constrained search, planning and scheduling. He is now working for Google.

Chenyang Lu is a Professor of Computer Science and Engineering at Washington University in St. Louis. Professor Lu is Editor-in-Chief of ACM Transactions on Sensor Networks and Associate Editor of Real-Time Systems. He has also served as Program Chair of IEEE Real-Time Systems Symposium (RTSS 2012) and ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS 2012). Professor Lu is the author and co-author of over 100 research papers with over 9000 citations and an h-index of 44. He received the Ph.D. degree from University of Virginia in 2001, the M.S. degree from Chinese Academy of Sciences in 1997, and the B.S. degree from University of Science and Technology of China in 1995, all in computer science. His research interests include real-time systems, wireless sensor networks and cyber-physical systems.

Yixin Chen is an Associate Professor of Computer Science at the Washington University in St Louis. His research interests include data mining, machine learning, artificial intelligence, optimization, and cyber-physical systems. He received a Ph.D. in Computing Science from University of Illinois at Urbana-Champaign in 2005. He received the Best Paper Award at the AAAI Conference on Artificial Intelligence (2010) and International Conference on Tools for AI (2005), and best paper nomination at the ACM KDD Conference (2009). His work on planning has won First Prizes in the International Planning Competitions (2004 & 2006). He has received an Early Career Principal Investigator Award from the Department of Energy (2006) and a Microsoft Research New Faculty Fellowship (2007). He is an Associate Editor for ACM Transactions of Intelligent Systems and Technology and IEEE Transactions on Knowledge and Data Engineering, and serves on the Editorial Board of Journal of Artificial Intelligence Research.

APPENDIX

SUPPLEMENTAL MATERIALS

Distributed Channel Allocation Protocols for Wireless Sensor Networks

Abusayeed Saifullah, You Xu, Chenyang Lu, Yixin Chen

Proof of Theorem 1

Until the algorithm converges, in every round at least one node switches its channel that is different from its neighbors in the receiver-based conflict graph G_R. If a node u switches to a channel that is different from its neighbors' channels, the interference links between u and its neighbors $N_R(u)$ are removed. Since no two neighbors in G_R switch channels in the same round, at least one interfering link in G is removed in every round. Since the total interfering links in G is $|E_I|$, the algorithm converges in at most $|E_I|$ rounds.

Proof of Theorem 2

Let Δ_R be the maximum degree of G_R. The channels are numbered 1,2, $\cdots$ in increasing order. Every node initially has channel 1. Every time a node switches channel, it switches to the smallest channel not used by the neighbors. Hence, the largest possible channel to which a node can switch is $\Delta_R + 1$, which happens if all first Δ_R channels are chosen by its neighbors in G_R. Hence, the algorithm employs at most $\Delta_R + 1$ channels. Since $\Delta \geq \Delta_R$, the theorem follows.

Proof of Theorem 3

The problem is in NP since, given an instance of the problem, we can verify in $O(|E|)$ time whether the network is interference-free. Following reduction from vertex-coloring implies NP-hardness. Given any instance $\langle \mathbb{G}, k \rangle$ of the vertex-coloring problem in graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, we create a sink node s as the parent of every $u \in \mathbb{V}$, and create a child for every u. Now for every edge $(u, v) \in \mathbb{E}$, we create an interfering link between u's child and v (or between v's child and u). This constructs an IC graph $G = (V, E)$. A channel allocation f uses the color of $u \in \mathbb{V}$ as the channel of u's child, and uses any channel c, $1 \leq c \leq k$, for u in G. Thus, $\mathbb{G}$ can be vertex-colored with k colors if and only if f can remove all interference links from G using k channels.

Proof of Theorem 4

When $k = 0$, the decision problem of the theorem represents a decision version of the link-based interference-free channel allocation (Problem 2) that has been proved to be NP-complete in Theorem 3. Thus, this decision problem is a generalization of that of Problem 2 and, hence, is NP-complete.

Proof of Theorem 5

Since MinMax algorithm is repeated as long as some node u can decrease its $C(u, f)$ using its available channels, in every round at least one node switches its channel. Assuming the neighbors of u in G_L keep their channels unchanged, changing the channel of u that decreases $C(u, f)$ implies that the total number of interference links between u and its neighbors decreases. Since no two neighbors in G_L switch channel simultaneously, at least one interfering link in G is removed in every round. Hence, similar to Theorem 1, the algorithm converges in at most $|E_I|$ rounds.

Proof of Theorem 6

Let $d(u)$ denote the degree of node u in link-based conflict graph G_L of G. The value $d(u)$ is equal to the conflict of u under single channel. We first prove that, when MinMax Algorithm converges, at most $\lfloor \frac{d(u)}{m} \rfloor$ neighbors in G_L can have the same channel as the one assigned to u, for any node u. Suppose to the contrary, after the algorithm converges, there exists some node v such that $\lfloor \frac{d(v)}{m} \rfloor + 1$ of its neighbors in G_L have the same channel as the one assigned to v. Let c be the channel assigned to v, and $Z \subseteq N_L(v)$ be the neighbors of v in G_L that have been assigned channel c. Now according to *the pigeon-hole principle*, there must be at least one channel $c' \neq c$ such that at most $\lfloor \frac{d(v)}{m} \rfloor$ neighbors of v have been assigned channel c'. If $\exists z \in Z$ such that $C(v, f) \leq C(z, f)$, then z will switch to channel c' since it can decrease its $C(z, f)$. If $C(z, f) \leq C(v, f)$, then v will switch to channel c' since it can decrease its $C(v, f)$. Both cases contradict with the hypothesis that the algorithm has converged. Therefore, when MinMax Algorithm converges, at most $\lfloor \frac{d(u)}{m} \rfloor$ neighbors in G_L can have the same channel as the one assigned to u, for any node u. Since $C_{\max}$ is equal to the maximum degree in G_L, the theorem follows.

Proof of Theorem 7

According to Theorem 2, the total time slots used in the frame is at most $\Delta_S + 1$, where Δ_S is the maximum degree in G_S. After MinMax channel allocation, $\Delta_S \leq \lfloor \frac{C_{\max}}{m} \rfloor + \Delta_T$. Hence, the bound follows.

Fig. 11. Testbed topology at -5 dBm Tx power (solid green lines are communication link; dotted red lines are interference link; the solid black node is the sink)

Algorithm 1: Receiver-based channel assignment at receiver node u

```
input: channel ids $1, 2, \cdots, m$;
output: a channel different from the channels chosen by the neighbors in $G_R$;
$f(u) \leftarrow 1$ /* first assign the smallest channel */
Broadcast the message $\langle u, f(u) \rangle$ to its neighbors $N_R(u)$;
if the message from each node in $N_R(u)$ has been received then
    $ch \leftarrow \min\{c | 1 \le c \le m$ and $c \ne f(z) \forall z \in N_R(u)\}$;
    if $f(u) \ne ch$ then $f(u) \leftarrow ch$ /* switch channel */
end
Broadcast the message $\langle u, f(u) \rangle$ to its neighbors $N_R(u)$;
```

node time	a	b	c	s
Round 1	1	1	1	1
Round 2	2	1	1	2
Round 3	2	3	1	2

TABLE 1
Channels selected in different rounds by the receiver nodes in Receiver-based channel assignment

Algorithm 2: Link-based channel assignment at sender node u

```
input: channel ids $1, 2, \cdots, m$;
output: a channel different from the channels chosen by the neighbors in $G_L$;
$f(u) \leftarrow 1$ /* first assign the smallest channel */
Broadcast message $\langle u, f(u) \rangle$ to its neighbors $N_L(u)$ in $G_L$;
if the message from each node in $N_L(u)$ has been received then
    $ch \leftarrow \min\{c | 1 \le c \le m$ and $c \ne f(z) \forall z \in N_L(u)\}$;
    if $f(u) \ne ch$ then $f(u) \leftarrow ch$ /* switch channel */
end
Broadcast the message $\langle u, f(u) \rangle$ to its neighbors $N_L(u)$;
```

node time	a	b	c	d	u	v	w	x	y	z
Round 1	1	1	1	1	1	1	1	1	1	1
Round 2	2	2	2	2	2	2	1	1	1	1
Round 3	2	2	2	2	2	2	1	3	1	1

TABLE 2
Channels selected in different rounds by the sender nodes in Link-based channel assignment

Algorithm 3: MinMax channel assignment at node u

```
$f(u) \leftarrow rand(1, m)$; /* assign a random channel */
Broadcast message $\langle u, f(u) \rangle$ to its neighbors $N_L(u)$ in $G_L$;
if message $\langle v, f(v) \rangle$ from each $v \in N_L(u)$ has been received then
    Determine $C(u, f)$ considering current assignment;
    Broadcast message $\langle u, C(u, f) \rangle$ to its neighbors $N_L(u)$;
end
if message $\langle v, C(v, f) \rangle$ from each $v \in N_L(u)$ is received then
    $A \leftarrow \{1 \le c \le m | \nexists v \in N_L(u)$ s.t. $C(v, f) > C(u, f)\}$;
    /* $A$ indicates the set of channels from which node $u$ will choose a channel */
    $ch \leftarrow$ the channel in $A$ that causes the minimum $C(u, f)$;
    if $f(u) \ne ch$ then $f(u) \leftarrow ch$ /* switch channel */
end
Broadcast the message $\langle u, f(u) \rangle$ to its neighbors $N_L(u)$;
```

node time	a	b	c	d	u	v	w	x	y	z
Round 1(random)	2	1	2	1	1	1	2	1	1	1
Round 2	2	2	2	1	1	2	2	1	1	1

TABLE 3
Channels selected in different rounds by the sender nodes in MinMax channel assignment when $m = 2$

GENERATING WSN TOPOLOGY WITH INTERFERENCE LINKS FOR EVALUATION

We evaluate our channel allocation and link scheduling protocols on the topologies of an indoor WSN testbed [49] spread over two buildings (Bryan Hall and Jolley Hall) of Washington University in St louis. The testbed consists of 74 TelosB motes each equipped with a Chipcon CC2420 radio compliant with IEEE 802.15.4. We have developed a discrete-event simulator that operates based on interference data traces collected from the testbed. The traces were obtained by having each node in the testbed take turns broadcasting a sequence of 50 packets. All nodes operated on channel 26 of IEEE 802.15.4. While the application transmits packets as soon as possible, the MAC layer applied for each transmission a randomized back-off uniformly distributed in the interval [10ms, 170ms]. The batch of 50 packets takes 4.5s on average to transmit. The remainder of the nodes recorded the Received Signal Strength (RSS) of the packets they receive. The short delay between the transmissions of packet pertaining to the same batch allows us to capture the short-term variability of RSS. We have collected 7 sets of data traces at 7 transmission (Tx) power levels: -15, -10, -7, -5, -3, -1, 0 dBm. Collecting the data traces over three consecutive days captured the long-term variability. RSS traces collected from the 74-node testbed are used to configure the simulations. Figure 11 shows the interference and communication edges on the testbed when every node's Tx power is set to -5dBm. The topology shown in Figure 11 is embedded on the floor plan of two buildings.

The network topologies used in the simulations are based on RSS traces collected from the testbed. We determine the communication and interference links between nodes as follows. A node A may communicate with a node B if node B's RSS average during A's transmissions exceeds a threshold of -85 dBm. Prior empirical studies have shown that links with RSS above this threshold typically have high packet reception rate (PRR) [50]. Interference links are determined based on RID protocol [46]. RID models interference as a graph that is constructed as follows. To determine whether the transmissions of other nodes can interfere with a communication link (A, B), RID calculates the Signal to Noise Plus Interference Ratio (SNIR) at node B for each set of k senders (k = 3 in our setup) assuming they transmit simultaneously as A transmits to B. For each set of senders $S(B)$, RID computes the SNIR at B when A and the set of senders $S(B)$ transmit simultaneously. The RSS of a link is computed as the average of the four 50 packet batches collected from the testbed. The RSS of missing packets is overestimated to equal the receiver sensibility of CC2420 (-90 dBm). If the computed SNIR is below a threshold a link from each node in $S(B)$ to B is added as an interference link. The SNIR threshold was set to 5 dB consistent with empirical studies that showed that meeting this threshold is usually sufficient for correctly decoding packets in the presence of interference [7], [46]. The routing tree on a topology is constructed based on high quality links.