Multi-channel TDMA Scheduling in Wireless Sensor Networks

Ozge Uyanik and Ibrahim Korpeoglu

Bilkent University
Department of Computer Engineering
06800, Ankara, Turkey
{ouyanik,korpe}@cs.bilkent.edu.tr

Abstract

In wireless sensor networks (WSNs) that use TDMA-based scheduled channel access, spatial re-use of time-slots is possible among a non-conflicting set of nodes. In this way, data gathering delays can be reduced and aggregate network throughput can be increased. Besides spatial re-use, available multiple channels, which is already an available feature in some sensor node platforms, can be utilized to increase concurrency and minimize the number of time-slots required for a round of communication. In this paper, we propose TDMA-based scheduling algorithms for multi-channel wireless sensor networks. By redefining the conflicts in a multi-channel environment, we extend two existing single-channel TDMA scheduling algorithms into multi-channel structure. We also present two channel assignment schemes (called NCA and LCA) appropriate to use with the extended multi-channel scheduling algorithms. We evaluate our proposed schemes by extensive simulation experiments and compare them with other single-channel and multi-channel algorithms from literature. The results show that in large networks our proposed algorithms can provide better performance, more concurrency, and up to 50% less delay compared to other methods.

Keywords: wireless sensor networks, multi-channel, TDMA, scheduling, channel assignment

---

1. Introduction

Sensor nodes are devices that are low-cost, low-power, and have short communication range. Each sensor node senses and produces data signal to be transported to a central location, so called base station or sink. Depending on the application, a monitoring activity may require a wireless sensor network to collect data from sensor nodes to the sink node as quickly as possible. It is also important that the data is carried without losses and errors. Errors and losses can happen due to collisions and interference. A proper scheduling method can prevent them.

Because of the short transmission range of sensor nodes [2], which is approximately 10 to 100 m depending on power output and environmental characteristics, covering a large area of interest requires the deployment of a large number of sensor nodes. These nodes need to cooperate for transmission of packets to the center by multi-hop communication through the sink.

Since wireless sensor nodes usually have limited source of energy, the lifetime of a sensor network, besides many other things, is affected by medium access control (MAC) protocol used. MAC protocols using time division multiple access (TDMA) are very successful in avoiding collisions compared to contention-based protocols [3]. Besides, efficiency in power is obtained more easily in TDMA-based MAC protocols, since nodes can remain silent and only get activated at their scheduled time-slots, whereas idle-listening and collisions cause energy waste in contention-based protocols. Moreover, TDMA-based protocols can create a schedule for transmissions with some QoS guarantees in terms of delay, jitter and throughput. It is very difficult for contention-based protocols to provide such guarantees.

Main objective in a TDMA scheduling scheme is to assign time-slots to nodes for accessing a channel, considering network topology and interference. The schedule produced by a TDMA protocol in a wireless sensor network enables the data packets of all sensor nodes to reach to the sink in a collision free manner. The number of slots used for a round of data gathering from all sensor nodes to the sink node is defined to be the *schedule length*. Shortening the schedule length with an intelligent scheduling algorithm makes the network accomplish the same data gathering task faster, hence reduces delay and increases throughput. An appropriate scheduling mechanism is required in order to arrange transmission order of sensor nodes to prevent collisions and to carry data to sink as fast as possible. An effective factor in arrangement of scheduling is interference. When operating on the same frequency, nodes that are spatially close to each other can interfere and cause incorrect decoding of packets at the receiver side, unless a method to completely eliminate or reduce the interference is applied.

Some sensor node platforms have multiple channels that can be used for transmission. For example, IEEE 802.15.4 [2] standard specifies medium access control and physical layer for low-rate wireless personal area networks (LR-WPANs) and can operate in one of the three different unlicensed bands, supporting 1 channel in 868 MHz band, 10 channels in 902 MHz band, and 16 channels in 2.4 GHz ISM (industrial, scientific and medical) band. Upper layers of the standard are not defined and can be specified in various standards, such as ZigBee [4] and WirelessHART [5]. Contention based or TDMA based channel access method can be applied over the base MAC layer of the sensor nodes using IEEE 802.15.4 standard. Since multiple channels are available, sensor nodes can apply intelligent channel assignment and channel access scheduling algorithms.

Another short-range wireless technology that supports multiple channels and also uses 2.4 GHz ISM band is the ubiquitous IEEE 802.11 [6] standard, also known as Wi-Fi. Although there are 14 channels defined in the standard, availability of the channels depends on band regulations of countries. On 2.4 GHz, channels of 802.11 [6] overlap with channels of 802.15.4 and can interfere with each other. Therefore, coexistence of multiple networks in an environment can have a negative effect on each other even though the networks use different wireless technologies. Coexistence issues of 802.11 and 802.15.4 as well as other wireless technologies operating on the ISM band are investigated in [7]. In a multi-network environment, the number of available channels for successful transmission in a WSN can be even less, as other wireless technologies such as 802.11 co-exists.

In this paper we propose channel assignment and scheduling algorithms for multi-channel wireless sensor networks that use time-slotted channel access. Our scheduling methods enable multiple transmissions in a time-slot by both considering spatial re-use and multiple available channels. Our work is inspired from the work of Ergen et al. [8] which propose TDMA-based single-channel scheduling algorithms for static WSNs. We extend their work by also considering and utilizing multiple channels available by some wireless communication technologies to further reduce schedule length, decrease packet latency and increase network throughput. When multiple channels are available and utilized, nodes in a WSN will have more chance to concurrently access the channel in a collision free manner, since they will have more freedom for preventing interference: they can either choose non-conflicting time-slots or they can choose different radio channels.

Our contributions in this paper are three-fold: First, we evaluate and redefine conflicts types in order to distinguish the ones that can be resolved by utilizing multi-channels. Second, we extend single-channel scheduling algorithms of [8] into a multi-channel structure. Hence, for any given network, multiple channels can be scheduled without collision by distinguishing the conflicts. Since the number of available channels can be limited, channel assignment need to be done carefully and efficiently. Finally, we propose two channel assignment algorithms (NCA and LCA) that are used in combination with time-slot assignment and scheduling algorithms.

The rest of this paper is organized as follows. In Section 2, some related work is discussed. In Section 3, some background information is presented. Our proposed multi-channel scheduling schemes are presented in Section 4. Our simulation environment and simulation results are presented and interpreted in Section 5. Finally, conclusions and future work are presented in Section 6.

2. Related Work

Multi-channel TDMA algorithms are studied in depth in [9, 3, 10, 11, 12, 13] and [14]. Zhang et al. [9] propose coloring and coding based distributed multi-channel TDMA scheduling in wireless ad-hoc networks. They categorize the conflicts caused by wireless links into two types: explicit and implicit. Explicit conflicts are proposed to be avoided by a time-slot allocation method whereas avoiding of implicit conflicts relies on an algebraic-coding based algorithm that utilizes multiple channels. Jovanovic et al. [3] propose TFMAC, a multi-channel MAC protocol for WSNs that incorporates multiple channels into TDMA. In TFMAC, a node randomly chooses a frequency and broadcasts it to its neighbors before activation period. Then it collects timetables from its neighbors to decide the time-slots to be active. TFMAC employs a control slot to exchange control messages. Our algorithms, however, do not use a contention period. Incel et al. [11] propose local time-slot assignment for raw data convergecast and utilize multiple channels using RBCA channel assignment algorithm proposed in [15]. We give more details about the RBCA with local time-slot assignment scheme in further sections, since we compare our algorithms with this scheme. Multi-channel communication is studied in various wireless networks in [22, 23, 10, 13] and [26]. All these studies differ than our study in terms that they allow packet losses or multi-channel communication algorithms are not TDMA.

Scheduling in industrial WSNs are studied in [16, 17] and [12]. An industrial environment can be harsher due to unpredictable variations in temperature and presence of heavy equipment [18], and therefore can have different link-layer requirements. The MAC layer of WirelessHART [5], for example, combines TDMA and frequency hopping and abandons spatial reuse. It allows only one link to be active on each channel in each time-slot. Our algorithms differ from this approach by utilizing multiple channels and spatial re-use.

Ergen et al. [8] propose node-based and level-based TDMA scheduling algorithms for single-channel WSNs. Our algorithms in this paper extend these single-channel scheduling algorithms and adapt them to be used with multi-channel networks. We give details of these algorithms later in the paper (in section 4). TDMA scheduling algorithms using a single-channel are investigated in various other studies in literature [30, 31, 19, 32, 20, 34, 14].

Various studies consider and use multiple channels that might be available in the wireless communication technology used by sensor nodes. An appropriate channel assignment scheme is needed in order to efficiently assign channels to nodes so that interference is eliminated to enable parallel transmissions. Channel assignment schemes are extensively analyzed in [22, 23, 27, 13, 28, 26]. Incel et al. [11] discuss three channel assignment methods. Joint Frequency Time Slot Scheduling (JFTSS) starts with the link that has the highest number of packets to transmit. If the link loads are equal, then the link that is more-constrained in terms of interference is considered first and the most available slot-channel pair is assigned. The nodes that do not interfere can be assigned the same time-slot and channel. In Tree-Based Multi-channel Protocol (TMCP), network is partitioned into multiple sub-trees where each sub-tree is assigned a different channel. This method is efficient since nodes do not require channel switching. In Receiver-Based Channel Assignment (RBCA), children of a common parent transmit on the same channel. Therefore, each node operates in at most two channels. The algorithms proposed in [11] assume data aggregation is used in WSNs, where our algorithms do not make this assumption.

Network topology can be considered in the design of a scheduling algorithm or MAC protocol. Depending on the algorithm, packet transmission chance of a node can be affected from its location in the topology. Schedule length can be decreased specifically for different type of topologies. Wang et al. [32] propose fairness in end-to end delivery in its spatial TDMA scheduling algorithm so that nodes with different quality and distance to the sink are treated equally. Ergen et al. [8] propose level-based scheduling where movement of the packets across the network is much better balanced for topologies with high density further away from the sink. Their proposed node-base scheduling algorithm gives equal chance to the nodes in the network and performs better in topologies of equal density of packets across the network or higher packet density at low levels. Lu et al. [38] design DMAC to solve data forwarding interruption problem, whereby not all nodes on a multi-hop path to the sink are notified of data delivery in progress, that results in significant sleep delay, and allow continuous packet forwarding by giving the sleep schedule of a node an offset that depends upon its depth in the tree. DMAC, however, is not designed to utilize multiple channels effectively, whereas we consider availability of multiple channels as part of our protocol design to increase concurrency.

We propose centralized multi-channel TDMA scheduling algorithms. We first improve node-based and level-based TDMA scheduling algorithms for single-channel WSNs in [8] by extending them into multi-channel structure. In our proposed channel assignment, children of a parent transmit on the same channel and transmission channel of the children is assigned the same with their parent whenever it does not cause conflicts. Time-slot assignment to nodes is done by coloring conflict graph of the original network. A time-slot can be assigned to more than one node as long as the resulting set of nodes for a time-slot is non-conflicting. Therefore, in our network, the same channel and time-slot can be spatially re-used by more than one node.

3. Background Information

This section introduces network model used in this paper, defines conflict types and details the scheduling problem.

3.1. Network Model

We assume that the network consists of one base station, also referred as sink node, and sensor nodes (also referred as sensors or nodes). Base station constantly collects data from sensor nodes. Sensor nodes generate data packets and transmit these data packets to base station. All the nodes are assumed to be of the same type such that they transmit with the same power using the same hardware, hence nodes have equal transmission range and equal interference range and, adopting the ideal network model, transmission disk is assumed to be circle. We assume node places are fixed. Routing tree is constructed in such a way that each node is connected to sink node either directly or through multiple hops. If a node is not directly connected to sink, it is connected to another sensor node selected as parent. The node selected as parent is a neighboring node with smallest number of hops to the sink node. In the case there are multiple parent choices each with the same hop-count to sink, the one with shortest total path length to sink is chosen. Level of a node is the number of hops from the node to the sink.

The network model proposed in [8] forms the basis of network model in this paper. The network is represented by a graph $G = (V, E)$. Here, V is the set of vertices, in this case sensor nodes; and E is the set of edges, in this case transmission links to be scheduled. $N = |V|$ is the number of nodes in G. The edges $E \subset V \times V$ are undirected. Every sensor node is connected to only one sensor node or base station directly for the transmission of its data packets.

Transmission from a node may interfere with a transmission from another node, causing collisions. Therefore, interfering nodes should not transmit at the same time. Well-known protocol interference model in [35] is used, that identifies the interference at the receiver, based on distance. Interference graph $C = (V, I)$ is assumed to be known. $I \subset V \times V$ is the set of edges such that $(u, v) \in I$ if either nodes u and v are in the interference range of each other or one of them can interfere by a transmission intended for the other although they are far enough to be affected by each other's transmission. If two nodes u, v are connected in the interference graph, v should not be scheduled to receive from another node while u is transmitting.

The conflict-graph corresponding to $G = (V, E)$ and $C = (V, I)$ is called $GC = (V, EC)$. In the conflict-graph, each edge of the original graph G that is a transmission link to be scheduled is represented by a node. Since each sensor node in the original graph has only one transmission link, which is to its parent, that link can be represented by the label of the transmitting node in the original graph. That means a node i in GC corresponds to the transmission link $(i, p_i) \in E$ where p_i is the parent of i ($i \in V$).

In the conflict-graph, EC comprises the edges between node pairs in G that should not transmit at the same time. Since each node has a half-duplex radio interface, it cannot transmit and receive in the same time-slot, and primary and secondary conflicts are considered in determining EC.

3.2. Conflict Types

There are two types of conflicts for the transmissions introduced by the nodes in the network. First type of conflict is called primary conflict that occurs as a node cannot both transmit and receive at the same time-slot as well as cannot receive more than one transmissions destined to it at the same time-slot. This is due to nature of the sensor nodes consisting of half-duplex radios. If $(i, j) \in E$, $(i, j) \in EC$, since a node can not both transmit and receive at the same time-slot. This primary conflict and its representation in GC is illustrated in Figure 1 on the left. Also, if $(i, j) \in E$ and c_j is a child of j in G $i \neq c_j$, $(i, c_j) \in EC$ because a parent can not receive from more than one child at one time-slot. Illustration of this primary conflict and its representation in the conflict-graph GC is given in Figure 1 in the middle.

The other conflict is called secondary conflict that occurs when an intended receiver of a particular transmission is also within the transmission range of another transmission destined to another receiver. If $(i, j) \in I$ and $(i, j) \notin E$, and c_j is a child of j in G, $(i, c_j) \in EC$, because if i is transmitting, child c_j of j cannot transmit at the same time-slot as j would hear from both i and c_j. This situation is illustrated in Figure 1 on the left, together with its representation in GC.

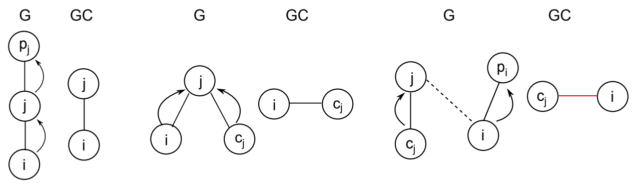


Figure 1: Primary conflicts on the left and in the middle, secondary conflict on the right.

Ergen et al. [8] gives definitions of primary and secondary conflicts which are described above. Since our aim is to decrease schedule length by mitigating interference, our motivation is to distinguish the transmission links that can be resolved when operating on different channels. Therefore, based on conflict types, we consider primary conflicts that are caused by sibling-sibling or parent-child relations as cannot be eliminated by setting to different channels, whereas secondary conflicts can be eliminated by utilizing multiple channels and setting transmission links to different channels.

Moreover, in order to distinguish conflict types in the conflict-graph efficiently, we modified the conflict-graph to be an edge-labeled conflict-graph where edges are associated with two different labels (colors). The edges of primary conflicts are assigned one label and the edges of secondary conflicts are assigned the other label. For instance, in the conflict-graphs of the primary conflicts in Figure 1, edges are associated with black color, whereas edge of the conflict-graph in secondary conflict example is associated with red color. This modification in the conflict-graph reduces computation overhead in the next phases.

3.3. Scheduling Problem

A scheduling frame, in short a schedule, consists of equal-duration time-slots that node or nodes assigned for transmissions. As also stated in [8], we assume duration of a time-slot is enough for a successful transmission of a data packet as well as a guard interval to compensate for synchronization issues. Scheduling frame starts with each node generating a positive number of packets and ends when all these packets reach at the sink node. In the schedule, each edge in G, that is each node in GC are assigned at least one time-slot for transmission. We assumed that interference graph C is given. With this knowledge, scheduling problem is finding a scheduling frame with minimum length during which all nodes can send their packets to sink. Apart from [8], in finding a minimum length scheduling frame, we also take into account utilizing multiple channels.

The scheduling problem mentioned above is proved to be NP-complete in [8] by reducing NP-complete problem of finding the chromatic number of a graph to the scheduling problem under use of a single channel. Moreover, finding an optimum channel assignment to remove secondary conflicts is NP-complete [43]. Therefore, in the solution of this problem we used polynomial time heuristics. Our heuristic is based on reducing the number of transmission links affected from secondary conflicts by assigning them to different transmission channels.

Channel allocation to nodes is based on the approach that a node is assigned channels for transmission and reception states. In this way, a node operates on at most two different channels: transmission channel and reception channel. The channel on which the node will transmit or receive will be calculated centrally. On the transmission state, the node will operate on transmission channel and on the reception state the node will operate on receive channel assigned by the central mechanism.

The central mechanism uses as few numbers of channels as possible to eliminate the interference by spatial reuse of channels such that a channel can be assigned to a number of nodes.

We also study the scheduling problem under limited number of channels available. Having a limited number of channels available, eliminating all secondary conflicts by channel allocation may not be possible. Appropriate channel selection policy is proposed for this case.

The central mechanism produces the schedule proposed by the heuristic algorithms by spatial reuse of a time-slot where none of the conflicting nodes transmit at the same time.

4. Our Multi-channel TDMA Scheduling Schemes

Since both channel assignment and scheduling problems to eliminate secondary conflicts are NP-complete, our proposed solutions depend on heuristics. As part of our solution, two centralized single-channel scheduling heuristics in [8] are modified and improved for multi-channel scheduling.

We perform static channel assignment. Nodes are not frequently and dynamically hopping among multiple channels. A node can change at most between two channels (one channel for reception, one channel for transmission) that are statically and permanently assigned to the node for the lifetime of the network. During data gathering a node either operates in receiving mode or transmitting mode. Channel switching occurs when a node changes its operation mode. If a node is assigned the same channel both for transmitting and receiving operations, then it never switches channels. Static channel assignment is preferred to reduce complexity and power consumption in sensor nodes.

It is possible that multiple channels can be utilized statically without any channel switching. However, this requires a sink node having many radios (as in [22]) where each radio operates on a different channel or requires the network consisting of multiple sinks each using a different channel. Moreover, for networks with tree-shaped traffic, static channel assignment schemes assign unique channels to the sub-trees of the network rooted at the children of the sink. This has its own disadvantages. For example, intra-subtree secondary conflicts cannot be resolved. As a result, our proposed channel assignment schemes use channel switching, but at a minimal level.

Our solution is inspired from the node-based and level-based TDMA scheduling algorithms proposed by Ergen et al. [8]. Their scheduling algorithms are effective in single-channel WSNs, however, do not take into account multi-channel capability of sensor radios. Hence, they cannot utilize multiple channels. Moreover, the algorithms in [8] are not in a form that is directly applicable for multi-channel networks. They need to be modified first to operate in a multi-channel environment.

A key parameter that is important for the effectiveness of our multi-channel scheduling algorithms is the number of channels that can be used by the algorithms (i.e., number of *available* channels). In theory, this number can be unlimited, but in practice it is limited due to several reasons such as the wireless standard restrictions or the interference existing in the environment. Therefore, the proposed algorithms should be effective in assigning channels even with few numbers of available channels.

In this section, we first introduce node-based scheduling proposed in [8] and show our extensions to it together with our channel assignment algorithm. As a result we obtain a multi-channel node-based scheduling algorithm. Then, we introduce level-based scheduling proposed in [8], the extensions we performed to it and our channel assignment scheme to be used with it. As a result we obtain multi-channel level-based scheduling in a similar fashion. We describe our algorithms in detail with some examples.

4.1. Node Based Scheduling

In the multi-channel node-based scheduling, first our channel assignment algorithm NCA (Algorithm 3) is applied. After having assigned the channels, nodes in the network are assigned slots (colored) using the algorithm COLOR (Algorithm 2) such that each node is assigned a time-slot that it can transmit simultaneously with non-conflicting nodes. Finally, nodes are scheduled using algorithm NODE (Algorithm 3) for transmissions according to their slots and channels until all data packets reach to the BS. Multi-channel node-based scheduling is presented together with single-channel node-based scheduling which forms the basis of this multi-channel approach. Extensions to the single-channel base algorithms are shown in bold.

4.1.1. NCA: Our Proposed Node Channel Assignment (phase 1)

Our Node Channel Assignment (NCA) algorithm is a greedy algorithm for multi-channel node-based scheduling. A node operates either in transmission or reception mode whenever it is active in the scheduling phase. Main approach in our channel assignment scheme is that a node is assigned a transmission channel that does not cause any secondary conflict when it is active (when node is transmitting). In this channel assignment scheme, a node operates on at most two channels. A node is preferred to operate on a single channel used for both transmission and reception to avoid channel switching. When this is not possible, two channels are assigned to the node, one for transmission and one for reception. The channels are assigned in such a manner that secondary conflicts are eliminated (if possible).

Our Node Channel Assignment algorithm is given in Algorithm 1. In NCA, a node is assigned a channel that is the same with its parent and respecting interference rules. If not possible, then another available channel is assigned. Among the available channels, a non-conflicting channel is chosen. In the limited channel version of our channel assignment scheme, if a non-conflicting channel is not available, the least conflicting channel is assigned for transmission. Least conflicting channel is determined by the number of conflicts caused if node operates on a channel. As the number of conflicting nodes that a node conflicts increases, it is more likely that a new color (time-slot) is required for the node in the coloring phase.

Algorithm 1 Node channel assignment algorithm - NCA

Input: Graph $G = (V, E)$ with conflict-graph $GC = (V, EC)$, $\sharp$ of channels
Output: Graph $G = (V, E)$ with channels assigned

node $n = sink$
In the depth first traversal of the network:
if $channel_n == null$ **then**
  p_n = parent of n
  assign $channel_{p_n}$ to all children of p_n
  if $\exists j$ assigned to $channel_{p_n}$ s.t. $(j, n) \in EC_c$ is of secondary conflict **then**
    find $channel_{available}$ s.t. $\neg\exists j$ assigned to $channel_{available}$ s.t. $(j, n) \in EC_c$ is of secondary conflict and assign to all children of p_n
  end if
end if

In the case that a non-conflicting channel is unavailable, whichever channel is assigned, node conflicts with some number of other nodes. In this case, the channel with the least number of conflicting nodes is chosen and assigned as transmission channel so as to cause least number of conflicts. Before the algorithm starts, the sink node is assigned a receiving channel and this channel is set as the transmission channel of the sinks children.

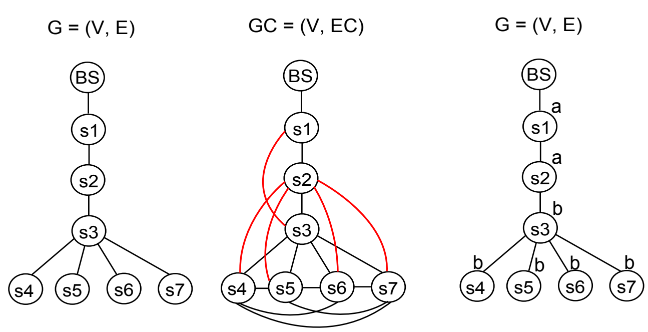


Figure 2: NCA channel assignment of G using GC.

NCA follows a depth first traversal of the nodes in the network starting from the root and is applied to nodes whose channels are not assigned yet. Channel of a node may have been assigned previously, because of being a sibling of a node, since assigned channel of a node is supposed to be the same with channel of its siblings. This is because while a node is transmitting to its parent, node's siblings cannot transmit at the same time and wait for their turn. Therefore, it is unnecessary to assign siblings of the node a channel other than node's channel. This approach has two benefits: first, waste of available channels is avoided; second, a node that operates in the receive state for a while does not necessarily do channel switching for each of its data reception from children.

As an example, assuming $c1_j$ and $c2_j$ are children of node j, even though we assign different transmission channels to $c1_j$ and $c2_j$, they cannot transmit at the same time-slot, because j cannot receive from both of them at the same slot. Besides, assuming $c2_j$ is scheduled to transmit after transmission slot of $c1_j$, if they transmit on different channels, then node j has to switch its channel for $c2_j$. NCA avoids this situation by assigning the same channel to siblings. Moreover, since channel of node j is assigned earlier than its children, NCA prefers to assign $c1_j$ and $c2_j$ the channel of j if it does not cause collisions. In this way unnecessary channel switching is avoided, i.e. j always operates on the same channel whether in receive mode or transmit mode. In the depth first traversal of the network starting from the sink, for every node whose channel is not assigned an available channel is assigned to the siblings and the node itself.

Figure 2 illustrates an example network, its associated conflict-graph with primary and secondary conflicts where edges in red correspond to secondary conflicts, and result of NCA channel assignment on the original network showing transmission channels.

In the limited channel version of the algorithm, as long as there is an available channel, secondary conflicts are resolved. If a non-conflicting channel is not available, then a channel with least number of conflicts is assigned. Unresolved secondary conflicts are resolved further in color assignment phase by assigning different time-slots.

This algorithm assigns a channel to node i in $O(d_{max})$ steps where d_{max} is the maximum degree of a node in GC. So, the running time is $O(d_{max}|V|)$.

4.1.2. COLOR: Extended Slot Assignment Algorithm (phase 2)

The slot assignment algorithm, COLOR (Algorithm 2) is extended from [8]. COLOR algorithm assigns colors, i.e., time-slots, to the nodes, determining their transmission turn in a round initially.

In this algorithm, firstly, nodes are ordered in a non-increasing manner according to number of conflicts existing after the channel assignment phase. Then, a different slot is assigned to each primarily or secondarily conflicting node. For multi-channel networks, this coloring is modified (bold part in the while loop) so that the same color can be assigned to a node which secondarily conflicts with another node whose color is already assigned and who has been assigned a different channel. Thus, previously conflicting nodes because of operating on the same channel and who have secondary conflicts can now do transmissions on the same time-slot if they are assigned different channels. The algorithm assigns a color to a node in $O(V)$ steps, so the running time is

Algorithm 2 Coloring algorithm - COLOR

Input: $V_c = \{2, 3, ..., N\}$, conflict-graph $GC_c = (V_c, EC_c)$

Output: One color assigned to each node $(2, c_2), (3, c_3), ..., (N, c_N)$ in which $c_i \in \{1, 2, ..., M\}$ and M is the number of colors.

```
Order the nodes as (n_1, n_2, ..., n_{N-1}) in non-increasing number of conflict degrees.
for l = 1 to N - 1 do
  i = 1
  while ∃j assigned to color i st. (j, n_l) ∈ EC_c do
    if (j, n_l) ∈ EC_c is of primary type or of secondary type but channel_j == channel_l then
      i = i + 1
    end if
  end while
  assign color i to n_l
end for
```

$O(|V|^2)$. Figure 3 shows coloring of the network with both single-channel and multiple channels assigned, respectively.

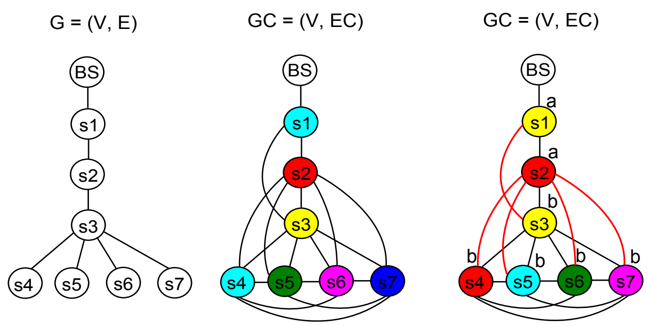


Figure 3: Coloring single-channel and multi-channel network.

4.1.3. NODE: Extended Node Based Scheduling Algorithm (phase 3)

Having assigned channels and time-slots, network can now be scheduled so that all data packets produced by nodes reach to the sink node. Algorithm 3, called NODE, gives details of the node-based scheduling. A *super-slot* in node-based scheduling consists of a number of consecutive time-slots, i.e., nodes with at least one packet at the beginning of a super-slot transmit at least one packet during the super-slot. Length of a super-slot at most equals to the number of colors used in coloring of the original network. In the multi-channel version, this scheduling algorithm is modified (bold part in the center) such that a node can join to a set of nodes for transmission although it has secondary conflicts with the nodes, provided that it is assigned a different channel.

Algorithm 3 Node-based scheduling algorithm - NODE

Input: Graph $G = (V, E)$ with conflict-graph $GC = (V, EC)$, color assignment of the nodes V_c using M colors

Output: Transmission schedule for nodes of G

while at least one packet has not reached BS **do**
    for $s = 1$ **to** M **do**
        set_s = set of nodes corresponding to color s with at least one packet
        $T = set_s$
        if $T \neq \emptyset$ **then**
            set_{os} = set of nodes not corresponding to color s with at least one packet
            for each node $k \in set_{os}$ **do**
                if $(k, j) \notin EC$ **or** $channel_j \neq channel_k$ **in case they have secondary conflict** $\forall\ j \in T$ **then**
                    $T = T \cup \{k\}$
                end if
            end for
            assign current slot to set T
            update the place of the packets
        end if
    end for
end while

In node-based scheduling, Algorithm NODE (Algorithm 3) creates non-conflicting slot sets where each set includes nodes that can transmit in the corresponding slot in a conflict-free manner. This non-conflicting set of nodes in that slot set are scheduled to do transmission at the same time in that slot. The algorithm starts with including all nodes in a set which have at least one packet to transmit for a time-slot (color). Then, other nodes which have at least one packet to transmit and do not conflict with any of the nodes in the set are included one by one, as long as the resulting set is non-conflicting.

In multi-channel networks, in the addition phase of the other nodes that belong to other sets, who are assigned a different slot than the current slot, a node is included to transmit if not only in the case it does not conflict, but also in the case it has secondary conflict with at least one of the nodes in the set but has a different transmission channel. Using multiple channels, a set that corresponds to a color (time-slot) with at least one packet can have greater number of nodes to transmit data compared to using single channel, because interference is eliminated and more transmissions can occur at the same time-slot. Thus, throughput in terms of data packet per time-slot increases. Running time of the algorithm is $O(ld_{max}|V|)$ where d_{max} is the maximum degree of a node in GC and l is the total number of slots in the schedule.

Figure 4 illustrates node-based scheduling of the single-channel and multi-channel networks using NCA algorithm for channel assignment in Figure 3. In

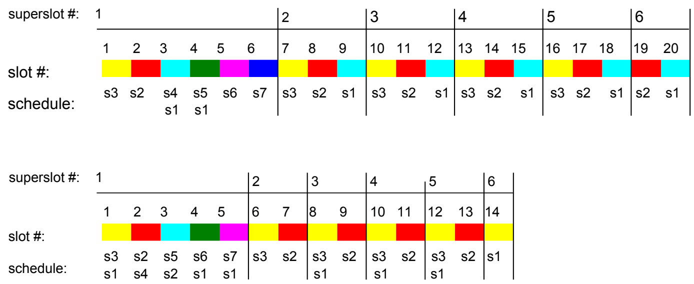


Figure 4: Node-based scheduling of single-channel and multi-channel network.

the schedules, transmitting nodes are shown slot by slot until all packets reach to sink. Multi-channel node-based scheduling using (NCA-NODE) produced a 30% decrease in the schedule length compared to single-channel scheduling algorithm (S-NODE) of [8].

4.2. Level Based Scheduling

Our multi-channel level-based scheduling scheme is based on the level-based scheduling algorithm of [8]. Level of a node is the number of hops to sink. Level-based scheduling balances movement of the packets across the network much better for topologies having higher density further away from the sink. In order to do this, first a linear network, also referred as level-tree, is created using LEVELTREE algorithm (Algorithm 4) that is modified from [8]. Then, our level channel assignment algorithm LCA (Algorithm 5) is used to assign channels to the levels using this linear network representing the original network. After channel assignment, level-tree is colored (time-slots are assigned to levels). Then, the nodes of the original network are assigned channels and colors depending on their levels (a node is assigned the channel and color of its level). Finally, the original network is scheduled with the level-based scheduling LEVEL (Algorithm 6).

4.2.1. LEVELTREE: Extended Linear Network Creation Algorithm

In the linear network, also called as level-tree, each level is represented by a node and a level conflict-graph is generated. In the level-tree, inter-level conflicts are marked such that if at least two nodes in different levels are conflicting, the level nodes in the level-tree are considered to be conflicting as well. Algorithm 4, LEVELTREE, explains how to create such a linear network and its associated interference and conflict-graphs. In the conflict-graph of the level-tree, edges correspond to primary and secondary conflict edges as described in previous sections. $GL = (VL, EL)$ is a linear network with nodes $VL = \{v_1, ..., v_N\}$ where N is the maximum node level in G and EL consists of edges between consecutive level nodes in VL. A node in VL corresponds all nodes belonging to that level in V.

The interference graph of the linear network is $CL = (VL, IL)$ which includes edge (v_j, v_l) if there is an interference edge between a node at level j

and any node at level l in the original network $G = (V, E)$ for j, $l \geq 1$. Conflict graph of the linear network is $GCL = (VL, ECL)$, which includes an edge (v_j, v_l) if the transmissions of a node at level j and a node at level l conflict in the original network.

Initially, the algorithm adds one node for each level. Then it adds edges between node levels. After that, for every conflicting node pairs, the algorithm adds secondary and primary conflict edges. Since consecutive levels have parent and child relationships, they are assigned primary conflict edges in ECL.

Running time of the algorithm is O($|V|^2$).

Algorithm 4 Algorithm to find a linear network corresponding to original network - LEVELTREE

Input: (V, E, I, EC)
Output: (VL, EL, IL, ECL)

add node v_1 to VL
$l = 2$
while $l \leq levelOfTree$ **do**
  add node v_l to VL
  add edge (v_{l-1}, v_l) to EL
  add primary conflict edge (v_{l-1}, v_l) to $IL(ECL)$
  if $\exists(u, v) \in I(EC)$ with u at level l and v at level j satisfying $j < l$ **and** j and l **are not consecutive levels then**
    add secondary conflict edge (v_j, v_l) to $IL(ECL)$
  end if
  $l{+}{+}$
end while

4.2.2. LCA: Our Proposed Level Based Channel Assignment (phase 1)

Channel assignment algorithm NCA is not appropriate to use for level-tree since each node in the level-tree corresponds to the set of nodes belonging to a level. NCA would result in some sequential levels assigned the same channel considering them as parent and child relation which is undesirable since it lowers effectiveness of the approach. Level Channel Assignment algorithm (LCA) is designed to solve the drawback introduced by NCA on the level-tree. Hence, we propose LCA (Algorithm 5) for multi-channel level-based scheduling.

LCA starts in a similar fashion with NCA and assigns a different channel to node in the level-tree where there is a primary or secondary conflict. In the limited version, if a non-conflicting channel is unavailable, then a channel with least number of conflicts is assigned. This algorithms assigns a channel to node i in $O(d_{max})$ steps, so the running time is O($d_{max}|VL|$) where d_{max} is the maximum degree of a node in GCL and $|VL|$is the number of nodes in the linear network.

Algorithm 5 Level channel assignment algorithm - LCA

Input: Graph $G = (V, E)$, $GL = (VL, EL)$ with conflict-graph $GCL = (VL, ECL)$, $\sharp$ of channels

Output: Graph $G = (V, E)$ with channels assigned

node $n = sink$
set_{level_n} = set of nodes in G at $level_n$
In the depth first traversal of the network in GL:
if $channel_n == null$ **then**
  assign $channel_{available}$ to n and set_{level_n}
end if

4.2.3. Slot (Color) Assignment to Levels (phase 2)

The same color assignment algorithm (COLOR) described in the previous section is used determine the color of each level of level-tree.

Figure 5 illustrates the original network, coloring of its associated single-channel level conflict-graph, multi-channel assignment to level graph using LCA, and coloring of its associated level conflict-graph. In the single-channel network, only level 1 and level 4 do not conflict and are assigned the same time-slot. On the other side, assigning multiple channels to level-tree using LCA removes secondary conflicts among levels and results in throughput increase in terms of data packets per time-slot.

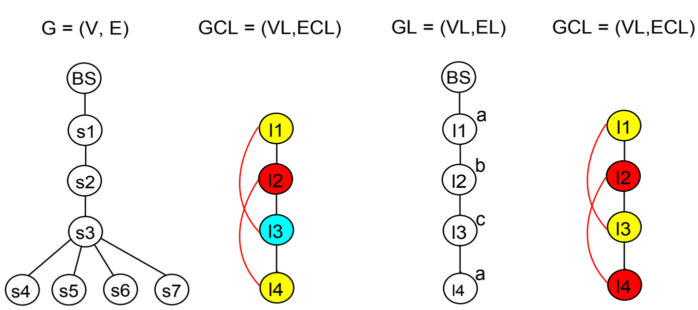


Figure 5: LCA channel assignment and coloring of single-channel and multi-channel network.

4.2.4. LEVEL: Extended Level Based Scheduling Algorithm (phase 3)

After color assignment, level-based scheduling in Algorithm 6, called LEVEL, schedules the network for packet transmissions to sink. A *super-slot* in level-based scheduling consists of consecutive time-slots, i.e., levels with at least one packet at the beginning of a super-slot forwards at least one packet during the super-slot. Length of a super-slot can be at most equal to the number of colors used in coloring level-tree.

First, nodes of the levels corresponding to a slot (color) which have at least one packet to transmit are included in the set. From the set, a non-conflicting set of nodes with at least one packet to transmit is created. Then, other nodes belonging to other levels which have at least one packet to transmit and do not conflict with any of the nodes in the set are included one by one, as long as the resulting set is non-conflicting.

In multi-channel networks, in the phase of adding other nodes belonging to other sets, who are assigned a different slot than the current slot, a node is included to transmit if not only it does not conflict, but also in the case it has secondary conflict with at least one of the nodes in the set, however, has a different transmission channel than the channel of the node it has secondary conflict (bold part in the algorithm).

Running time of the algorithm is $O(ld_{max}|V|)$ where d_{max} is the maximum degree of a node in GC and l is the total number of slots in the schedule.

Algorithm 6 Level-based scheduling algorithm - LEVEL

Input: Graph $G = (V, E)$ with conflict-graph $GC = (V, EC)$, color assignment of the corresponding linear network GCL using M colors
Output: Transmission schedule for nodes of G

while at least one packet has not reached BS **do**
  for $s = 1$ **to** M **do**
    set_s = set of levels corresponding to color s
    $T = \emptyset$
    for $j = 1$ **to** $|set_s|$ **do**
      $T = T \cup$ {a non conflicting set of nodes from level $set_s(j)$ with at least one packet}
    end for
    if $T \neq \emptyset$ **then**
      set_{os} = set of levels not corresponding to color s
      for each node k belonging to a level in set_{os} **do**
        if $(k, j) \notin EC$ **or** $channel_j \neq channel_k$ **in case they have secondary conflict** $\forall\, j \in T$ **then**
          $T = T \cup \{k\}$
        end if
      end for
      assign current slot to set T
      update the place of the packets
    end if
  end for
end while

By eliminating secondary conflicts by assigning different channels to conflicting levels, a greater number of levels can transmit at the same time-slot. Use of NCA, which is specifically designed for channel assignment of nodes, could result in consecutive levels having the same channel since consecutive levels have primary conflict due to sender receiver relation. A better performing approach is obtained with LCA by modifying NCA, assigning different channels to consecutive levels with Level Channel Assignment (LCA) algorithm. Thus, greater number of nodes in a level can be activated in a time-slot either as a transmitter or receiver. For instance, assuming a node in a level is included in the transmission set, and then its sibling cannot transmit at the same. LCA allows that sibling node has the opportunity to be scheduled for reception so that if any of its children in the consecutive level has packet to transmit and does not conflict with any of the nodes has the opportunity to be scheduled for transmission.

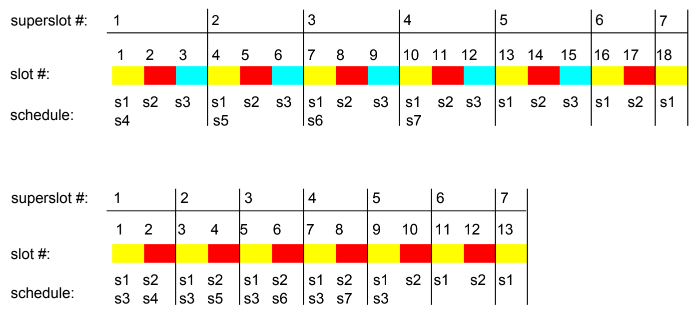


Figure 6: Level-based scheduling of single-channel and multi-channel network.

Figure 6 illustrates Algorithm LEVEL with single-channel and multi-channel networks. This figure also illustrates a network where multi-channel level-based scheduling performs better than multi-channel node-based scheduling.

Moreover, it can be inferred from the schedules shown in Figures 4 and 6 that multi-channel schedules provide better throughput in terms of data packets per time-slot.

5. Performance Evaluation

In this section, we first present our simulation environment, simulation parameters and metrics, and then present the results of our simulation experiments we performed to evaluate our algorithms. For evaluation, we compare our multi-channel scheduling and channel assignment schemes (NCA and LCA) with the single-channel scheduling schemes proposed in [8]. The work of [8] is forming the basis of our work here. Moreover, we compare our algorithms with a multi-channel scheduling scheme from literature, the RBCA with Local Time Slot Assignment proposed in [15, 11].

5.1. Simulation Environment

We developed a custom simulator to evaluate the performance of our algorithms. In the simulated networks, each node produces one packet to be sent to the base station. Delay is defined to be the total number of time-slots required until all packets generated by sensor nodes (one packet per node) arrive at the sink node. That means we define delay to be the data gathering delay in one round. Lower bound of delay is the number of data packets sink receives, since in each time-slot sink can receive only one packet. In the experiments, we measure delay for various values of node density, available number of channels, and the ratio of interference range to communication range.

For the simulations, 1000 nodes are randomly distributed on a circular area (disk) of radius 100 distance units. The sink is located at the center. The node density is defined in the following manner. Two different node densities are used: λ_1 and λ_2. λ_1 is the node density of an inner disk with radius $100/\sqrt{2}$ distance units, having the same center point with the outer disk. λ_2 is the node density of the remaining part of the outer disk, i.e., the part between the radius $100/\sqrt{2}$ and 100 units (a ring). Note that the area of the inner disk and the remaining part of the outer disk (i.e., the ring) are equal to each other.

Effect of density on data gathering delay is investigated with varying values of λ_1/λ_2 for each of the scheduling algorithms. λ_1/λ_2 ratio is a factor that plays an important role in the formation of the network. Low values of this ratio (λ_1/λ_2) forms a network topology with higher density further away from the sink, whereas high values of this ratio results with network topologies that have higher density around the sink. Communication range also plays an important role. We set the communication range to be just enough to have connected network.

Ratio of the interference range to communication range is another parameter that we investigate in our experiments. This ratio is considered to be 2 in experiments of [8], that means interference range is set to be two times the communication range. It is further analyzed from 1 to 4. Another study we compare [11] sets this ratio to 1. In our simulations, we evaluate and compare effects of interference to communication ratio for all algorithms where the ratio is changed from 1 to 5, with an increment of 0.5. When fixed while evaluating the effect of other parameters, this ratio is set to 2, as in [8].

The final parameter used in the evaluation of the algorithms is the available number of channels. We implemented our proposed algorithms to work with both unlimited and limited number of available channels. When unlimited number of channels is used, all secondary conflicts are eliminated at channel assignment time. When limited number of channels is used, however, secondary conflicts in the network cannot be totally eliminated at channel assignment phase. Our algorithms using limited number of channels are evaluated up to the point where increasing the number of available channel does not effectively contribute to shorten data gathering delay. Limited number of available channels is increased up to 7, at which point no further performance improvement could be observed for majority of the algorithms. The effect of number of available channels is observed for different interference to communication ratios as well as for different densities.

We compare our algorithms with a multi-channel scheme from literature, the Receiver Based Channel Assignment (RBCA) with local time-slot assignment proposed in [15, 11], that solves the same problem. In RBCA channel assignment algorithm, first all receivers are assigned a channel. Then, for each receiver, a set of interfering parents is created. And starting from the most interfered parent (the parent with the highest number of interfering links), receivers is assigned the next available channel.After channel assignment, local time-slot assignment algorithm is applied. Each child of the root is said to be a top-subtree. At each time-slot, root receives from one of its children which has the largest number of total remaining packets at its subtree. A node can be scheduled to receive, if its buffer is empty and if there is a child who has packet to transmit respecting interference constraint. This slot assignment is buffer efficient and requires little topology knowledge. The root knows only needs to know the number of nodes in each top-subtree. The authors prove that if all the interfering links are eliminated, the schedule length achieved by this algorithm is the minimum, i.e., $max(2n_k{-}1, N)$ where N is number of nodes and n_k is the number of nodes in top-subtree k. Considering these all, RBCA and local time-slot assignment algorithm proposed in [11] is an appropriate candidate for comparison with our multi-channel scheduling algorithms that can use a limited number of available channels.

With the extensions to the algorithms proposed in [8] for node-based scheduling and level-based scheduling, our multi-channel scheduling results are compared with single-channel scheduling results. Our channel assignment algorithms NCA and LCA are tested with both unlimited number of channels and limited number of channels; and simulated both with node-based scheduling and level-based scheduling. Results are also compared by implementing some other multi-channel algorithms from literature, namely Receiver Based Channel Assignment (RBCA) with local time-slot assignment scheme proposed in [11].

	Algorithm	**Channel Assignment Algorithm**	**Scheduling Algorithm**	**Available # of Channels**
From [8]	S-NODE	-	NODE	1
From [8]	S-LEVEL	-	LEVEL	1
Proposed	NCA-NODE	NCA	NODE	Unlimited
Proposed	LCA-LEV	LCA	LEVEL	Unlimited
Proposed	LNCA-NODE	NCA	NODE	Limited
Proposed	LNCA-LEV	NCA	LEVEL	Limited
Proposed	LLCA-LEV	LCA	LEVEL	Limited
From [11, 15]	LOCAL	RBCA	LOCAL	Limited

Table 1: Algorithms used in the simulations.

In our simulations, in total eight algorithms are compared in terms of delay versus node density, interference-communication range ratio, and available number of channels. In our discussions and figures, the terms S-NODE and S-LEVEL denote the single-channel node-based and level-based scheduling algorithms proposed in [8]. NCA-NODE and LCA-LEV are our multi-channel node-based scheduling algorithm with our NCA channel assignment scheme and multi-channel level-based scheduling algorithm with LCA channel assignment scheme using unlimited number of channels (that means the algorithms can use as many channels as they wish).

The terms LNCA-NODE and LNCA-LEV denote our multi-channel node-based and level-based scheduling algorithms with NCA channel assignment scheme using limited number of channels. That means LNCA can use only a limited number of channels, not as many channels as it wishes. Although NCA is designed for node-based scheduling, its behavior with level-based scheduling is also investigated so as to observe the effects of less intra-level interference to delay. In the implementation of LNCA-LEV, limited version channel assignment of NCA is applied before creating a linear network. In the creation of a linear network, channel assignment is also considered in determining conflicting levels. After coloring the linear network, the original network is scheduled. LLCA-LEV denotes multi-channel level-based scheduling with LCA channel assignment using limited number of channels.

LOCAL denotes the local scheduling algorithm used with RBCA channel assignment proposed by Incel et al. [11, 15]. We implemented this scheme to compare against our algorithms.

Summary of the algorithms used in the simulations is given in Table 1.

5.2. Simulation Results

5.2.1. Delay versus Density

As mentioned earlier, we model the network region as a disk which has an inner disk with the same center. Nodes are deployed in a uniform manner to the inner disk and to the ring between the disk and inner disk, but the density of deployment in the inner disk and in the ring is different. The ratio of these two densities ($\lambda_1/\ \lambda_2$) is the density parameter for the network. If it is 1, both densities are equal and the number of nodes in the inner disk and in the remaining part of the outer disk (i.e., in the ring) is the same. The effect of network density, as defined above, on the delay is presented in Figure 7.

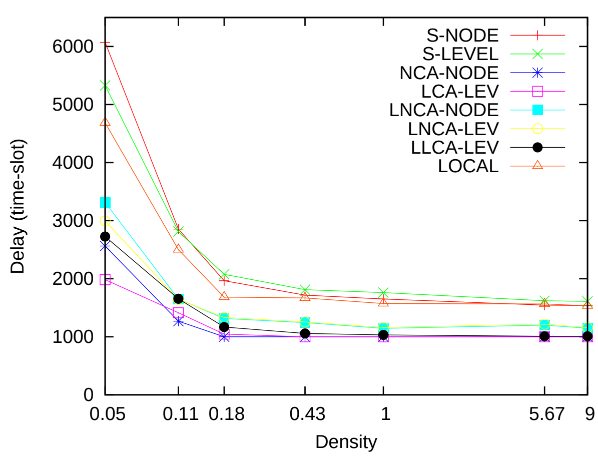


Figure 7: Delay versus density. Interference range = 2 x transmission range. Number of channels = 3.

As can be seen from the figure, the best performing algorithms in terms of data gathering delay are our proposed algorithms LCA-LEV and NCA-NODE. They perform close to lower bound for all different density values considered. For lower densities, LCA-LEV performs better than NCA-NODE. Among the limited-channel algorithms where the number of channels is restricted to 3, LLCA-LEV performs the best. For these experiments, the interference range is set to twice the transmission range. Performance of LOCAL, proposed by [11], is closer to single-channel algorithms, and almost 40% worse than the other limited-channel algorithms for lower densities. Moreover, LCA-LEV and NCA-NODE shows that eliminating all secondary conflicts is not enough to reach optimal schedules and that topology is also an effective factor.

It is also important to note that all curves are decreasing as density is increasing in Figure 7. In topologies with low density, many of the nodes are located further away from the sink and they transmit their packets on multi-hops. As density is increasing, more nodes become closely located around sink, hence they are likely to be able to directly transmit. On top of this, the number of hops a data packet travels until reaching to sink decreases as nodes get closer to sink. This also increases the possibility of having a much balanced network. All considered, increasing density contributes to the performance of all algorithms in the simulations.

5.2.2. Delay versus Interference Range

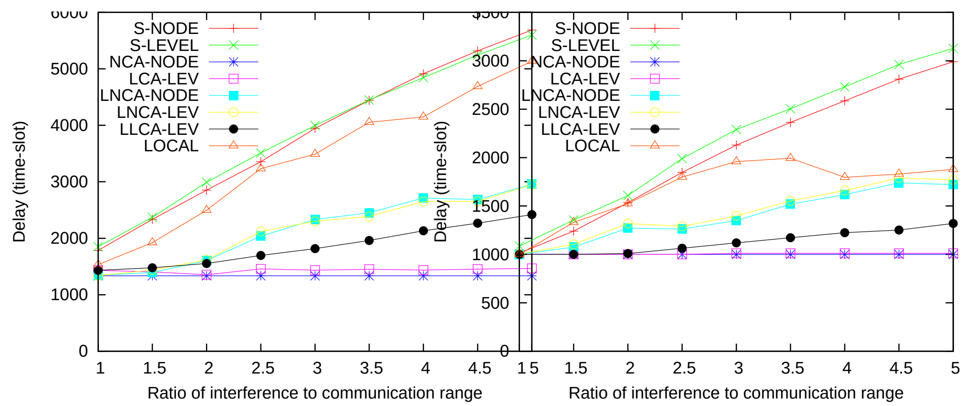
Delay versus interference range is analyzed for two λ_1/λ_2 ratio (density), namely 0.1 and 9. The interference to communication range ratio is shown on the x-axis. The respective results are presented in Figures 8a and 8b.

Figures 8a and 8b indicate that increasing interference significantly affects the delay both for low and high density (λ_1/λ_2 ratio) values. For low interference where interference range equals transmission range, proposed multi-channel algorithms have close results, whereas LNCA-LEV slightly outperforms other limited-channel schemes. Under high interference, proposed multi-channel and limited multi-channel algorithms have better performance compared to others. Besides, LLCA-LEV performs the best among limited channel schemes and produces around 50 % shorther schedule than compared limited multi-channel algorithm where the number of channels is restricted to 3.

In a network with high λ_1/λ_2 ratio, Figure 8b depicts that under low interference, LLCA-LEV with 3 channels performs close to optimum, which is the number of nodes in the network, compared to other limited channel scheduling schemes. NCA-NODE and LCA-LEV performs almost optimum in high-density networks compared to low density.

5.2.3. Delay versus Number of Channels

Delay versus number of channels is analyzed for networks with different low densities (0.1, 0.25, and 0.45) and different interference ranges to communication range (1 to 4). Since S-NODE and S-LEVEL are single-channel algorithms; and NCA-NODE and LCA-LEV are multi-channel algorithms implemented without



(a) Density = 0.1

(b) Density = 9

Figure 8: Delay versus ratio of interference to communication range. Number of channels = 3.

limited number of channels, delay values of these algorithms remain stable, and they are included in the graphics to provide comparison.

Delay versus Number of Channels - with varying density. Effect of available number of channels to delay is analyzed for different low densities when ratio of the interference to communication range is 2.

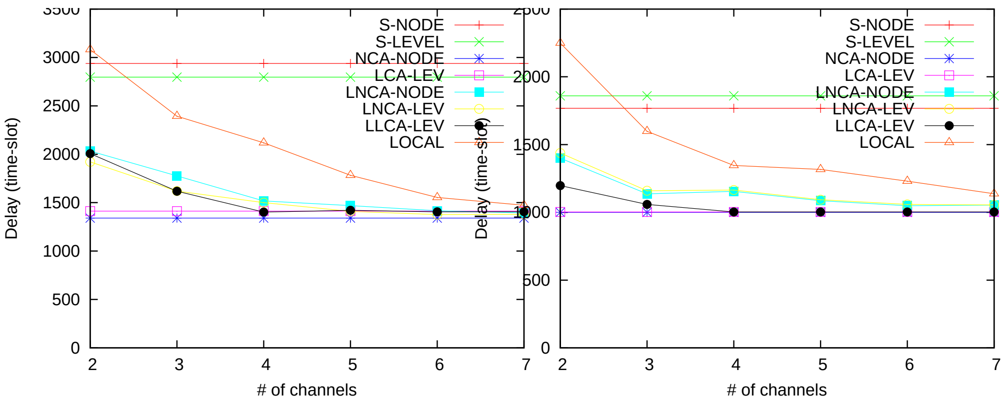
For low-density networks, our proposed limited channel schemes perform similarly as shown in Figure 9a. Besides, proposed schemes have better performance than other methods.

Figure 9b illustrates delay performance on networks with low density values, but which is still greater than the density value of Figure 9a. Increase in density results in almost optimum scheduling of NCA-NODE and LCA-LEV. LLCA-LEV performs better than other multi-channel scheduling algorithms with limited number of channels. Its performance gets close to optimum with less number of channels compared to other multi-channel scheduling algorithms. Moreover, significant delay difference can be observed between proposed and compared schemes when less number of channels are used.

Delay versus Number of Channels - with varying interference range. Effect of available number of channels to delay is analyzed for different ratios of interference to communication range and for a density value of 0.1.

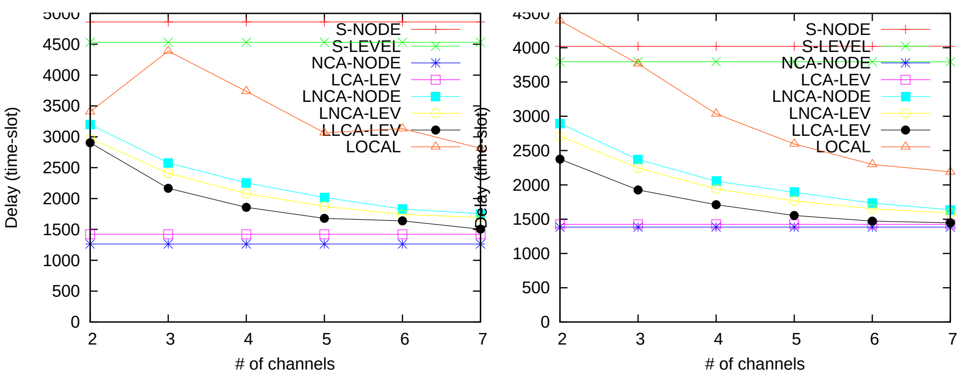
The results shown in Figures 10a, 10b, 11a and 11b indicate that a higher ratio of interference to communication range requires larger number of available channels to eliminate secondary conflicts.

Under high interference, Figures 10a and 10b show that our LLCA-LEV scheme performs better than our other limited-channel schemes as well as the schemes from literature. However, as interference range gets closer to communication range, our proposed limited multi-channel schemes performs similarly



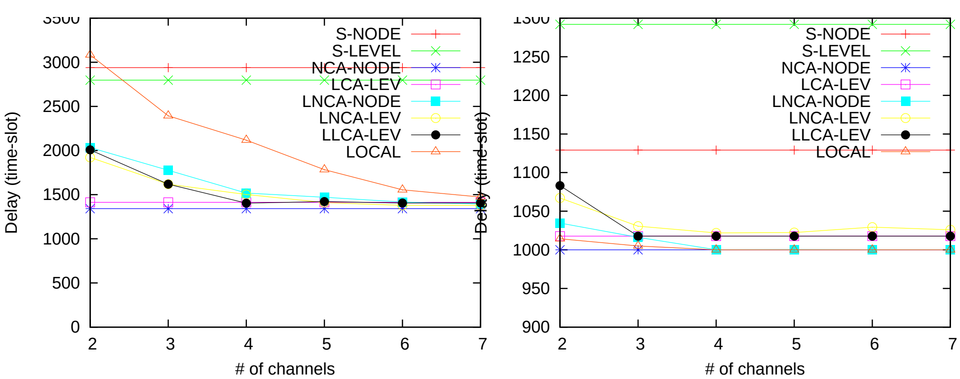
(a) Density = 0.1 (b) Density = 0.45

Figure 9: Delay versus number of channels. Interference range = 2 x transmission range.



(a) int r = 4 * trans r (b) int r = 3 * trans r

Figure 10: Delay versus number of channels. Density = 0.1



(a) int r = 2 * trans r (b) int r = trans r

Figure 11: Delay versus number of channels. Density = 0.1

and better than other methods, as seen in Figure 11a.

When interference range equals to transmission range LOCAL performs close to optimum and outperforms our proposed limited channel algorithms when using less number of available channels. This scenario also shows that NCA-NODE outperforms LCA-LEV.

For networks where interference range is greater than transmission range, proposed limited multi-channel scheduling algorithms have significantly better performance compared to others. Under heavy interference, LLCA-LEV performs best among the limited multi-channel scheduling schemes.

6. Conclusion and Future Work

In this paper, we propose TDMA based multi-channel scheduling algorithms for multi-channel wireless sensor networks with spatial reuse of channels and time-slots. We aim to decrease the required number of time-slots for a round of data gathering. We achieve this by effectively assigning channels and time-slots to sensor nodes. Our proposed algorithms are based on and extended from the single-channel scheduling algorithms proposed in [8]. Node-based and level-based algorithms proposed in [8] color a conflict-graph of the original network to determine the time-slots nodes will use. Then, the original network is scheduled for transmission. In the paper, we first analyze conflict types that may appear in a multi-channel WSN and based on this analysis we identify the conflicts that can be resolved by setting the links to operate in different channels. Then, using this grouping, we modify the existing single-channel algorithms proposed in [8] to operate in a multi-channel network. After that, we propose channel assignment algorithms for node-based and level-based scheduling. Our channel assignment algorithms assign orthogonal channels to links having conflicts that are possible to resolve by assigning different channels. We did extensive simulation experiments and our simulation results show that our proposed scheduling algorithms perform well and achieve low data-gathering latency compared to other alternatives.

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, no. 4, pp. 393 – 422, 2002.

[2] "IEEE 802.15.4-2003 Standard." `http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf`, October 2003. Last access: 7 June 2013.

[3] M. Jovanovic and G. Djordjevic, "TFMAC: Multi-channel MAC Protocol for Wireless Sensor Networks," in 8^{th} *International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services (TELSIKS'07)*, pp. 23–26, 2007.

[4] "ZigBee Specification." `http://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2011/kjb79_ajm232/pmeter/`

ZigBee%20Specification.pdf, January 2008. Last access: 7 June 2013.

[5] "HART Communication Foundation." http://www.hartcomm.org/index.html. Last access: 7 June 2013.

[6] "IEEE 802.11-2007 standard." http://standards.ieee.org/getieee802/download/802.11-2007.pdf, June 2007. Last access: 7 June 2013.

[7] K. Shuaib, M. Boulmalf, F. Sallabi, and A. Lakas, "Co-existence of ZigBee and WLAN - A Performance Study," in *International Conference on Wireless and Optical Communications Networks (IFIP'06)*, pp. 5–, 2006.

[8] S. C. Ergen and P. Varaiya, "TDMA Scheduling Algorithms for Wireless Sensor Networks," *Wireless Networks*, vol. 16, pp. 985–997, May 2010.

[9] X. Zhang, J. Hong, L. Zhang, X. Shan, and V.-K. Li, "CC-TDMA: Coloring- and Coding-Based Multi-Channel TDMA Scheduling for Wireless Ad Hoc Networks," in *IEEE Wireless Communications and Networking Conference (WCNC'07)*, pp. 133–137, 2007.

[10] M. Salajegheh, H. Soroush, and A. Kalis, "HYMAC: Hybrid TDMA/FDMA Medium Access Control Protocol for Wireless Sensor Networks," in *IEEE* 18^{th} *International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)*, pp. 1–5, 2007.

[11] O. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, "Fast Data Collection in Tree-Based Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 86–99, 2012.

[12] H. Zhang, P. Soldati, and M. Johansson, "Time- and Channel-efficient Link Scheduling for Convergecast in WirelessHART Networks," in *IEEE 13th International Conference on Communication Technology (ICCT'11)*, pp. 99–103, 2011.

[13] V. Annamalai, S. Gupta, and L. Schwiebert, "On Tree-based Convergecasting in Wireless Sensor Networks," in *IEEE Wireless Communications and Networking Conference (WCNC'03)*, vol. 3, pp. 1942–1947, 2003.

[14] U. Hunkeler, C. Lombriser, H. L. Truong, and B. Weiss, "A Case for Centrally Controlled Wireless Sensor Networks," *Computer Networks*, vol. 57, no. 6, pp. 1425 – 1442, 2013.

[15] O. Durmaz Incel, *Multi-Channel Wireless Sensor Networks: Protocols, Design and Evaluation*. PhD thesis, University of Twente, Zutphen, March 2009.

[16] X. Zhang, Q. Luo, L. Cheng, Y. Wan, H. Song, and Y. Yang, "CRTRA: Coloring Route-tree Based Resource Allocation Algorithm for Industrial Wireless Sensor Networks," in *IEEE Wireless Communications and Networking Conference (WCNC'12)*, pp. 1870–1875, 2012.

[17] D. Yuan and M. Hollick, “Tree-based Multi-channel Convergecast in Wireless Sensor Networks,” in *IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM'12)*, pp. 1–9, 2012.

[18] K.-S. Low, W. Win, and M.-J. Er, “Wireless Sensor Networks for Industrial Environments,” in *International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, vol. 2, pp. 271–276, 2005.

[19] S. Gobriel, D. Mosse, and R. Cleric, “TDMA-ASAP: Sensor Network TDMA Scheduling with Adaptive Slot-Stealing and Parallelism,” in 29^{th} *IEEE International Conference on Distributed Computing Systems (ICDCS'09)*, pp. 458–465, 2009.

[20] M. O. Daz-Anadn and K. K. Leung, “TDMA Scheduling for Event-triggered Data Aggregation in Irregular Wireless Sensor Networks,” *Computer Communications*, vol. 34, no. 17, pp. 2072–2081, 2011.

[21] G. Hasegawa, T. Hidekuma, M. Sasabe, and H. Nakano, “Power Control Methods for Improving Spatial Reuse in TDMA-based Wireless Mesh Networks,” *ISRN Communications and Networking*, vol. 2011, pp. 1–12, Jan. 2011.

[22] Y. Wu, J. Stankovic, T. He, and S. Lin, “Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks,” in 27^{th} *IEEE Conference on Computer Communications (INFOCOM'08)*, pp. 1193–1201, 2008.

[23] 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 25^{th} *IEEE Conference on Computer Communications (INFOCOM'06)*, p. 7, 2006.

[24] B. E. Bilgin and V. Gungor, “On the Performance of Multi-Channel Wireless Sensor Networks in Smart Grid Environments,” in 20^{th} *International Conference on Computer Communications and Networks (ICCN'11)*, pp. 1–6, 2011.

[25] A. Gonga, O. Landsiedel, P. Soldati, and M. Johansson, “Revisiting Multi-Channel Communication to Mitigate Interference and Link Dynamics in Wireless Sensor Networks,” in 8^{th} *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'12)*, May 2012.

[26] X. Zhang, W. Liang, H. Yu, and X. Feng, “Reliable Transmission Scheduling for Multi-channel Wireless Sensor Networks with Low-cost Channel Estimation,” *IET Communications*, vol. 7, no. 1, pp. 71–81, 2013.

[27] X. Chen, P. Han, Q.-S. He, S.-L. Tu, and Z.-L. Chen, "A Multi-Channel MAC Protocol for Wireless Sensor Networks," in 6^{th} *IEEE International Conference on Computer and Information Technology (CIT'06)*, pp. 224–224, 2006.

[28] N. Abdeddaim, F. Theoleyre, F. Rousseau, and A. Duda, "Multi-Channel Cluster Tree for 802.15.4 Wireless Sensor Networks," in 23^{rd} *IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'12)*, pp. 590–595, 2012.

[29] H. Zhang and S. Li, "A Practical Design of Multi-channel MAC for Cluster-tree WSN," in 6^{th} *International Forum on Strategic Technology (IFOST'11)*, vol. 2, pp. 761–764, 2011.

[30] D. Vergados, A. Sgora, D. Vergados, D. Vouyioukas, and I. Anagnostopoulos, "Fair TDMA Scheduling in Wireless Multihop Networks," *Telecommunication Systems*, vol. 50, no. 3, pp. 181–198, 2012.

[31] P. Djukic and S. Valaee, "Link Scheduling for Minimum Delay in Spatial Re-use TDMA," in 26^{th} *IEEE International Conference on Computer Communications (INFOCOM'07)*, pp. 28–36, 2007.

[32] Z. Wang, F. Yu, L. Tao, and Z. Zhang, "A Fairness Spatial TDMA Scheduling Algorithm for Wireless Sensor Network," in 12^{th} *International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'11)*, pp. 348–353, 2011.

[33] D. Panigrahi and B. Raman, "TDMA Scheduling in Long-Distance WiFi Networks," in 28^{th} *IEEE Conference on Computer Communications (INFOCOM'09)*, pp. 2931–2935, 2009.

[34] W. Wang, Y. Wang, X.-Y. Li, W.-Z. Song, and O. Frieder, "Efficient Interference-aware TDMA Link Scheduling for Static Wireless Networks," in 12^{th} *Annual International Conference on Mobile Computing and Networking (MobiCom'06)*, pp. 262–273, 2006.

[35] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[36] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-hop Wireless Network Performance," in 9^{th} *Annual International Conference on Mobile Computing and Networking (MobiCom'03)*, pp. 66–80, 2003.

[37] W.-Z. Song, F. Yuan, and R. LaHusen, "Time-Optimum Packet Scheduling for Many-to-One Routing in Wireless Sensor Networks," in *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS'06)*, pp. 81–90, 2006.

[38] G. Lu, B. Krishnamachari, and C. Raghavendra, "An Adaptive Energy-efficient and Low-latency MAC for Data Gathering in Wireless Sensor Networks," in 18^{th} *International Parallel and Distributed Processing Symposium (IPDPS'04)*, pp. 224–, 2004.

[39] S. Jeon and C. Ji, "Joint Approximation of Information and Distributed Link-Scheduling Decisions in Wireless Networks," *Computing Research Repository*, 2012.

[40] D. Wu, G.-Y. Wang, and X.-L. Li, "Distributed TDMA Scheduling Protocol Based on Conflict-free for Wireless Sensor Networks," in *International Conference on Intelligent Computing and Integrated Systems (ICISS'10)*, pp. 876–879, 2010.

[41] G. EkbataniFard and R. Monsefi, "A Detailed Review of Multi-Channel Medium Access Control Protocols for Wireless Sensor Networks," *International Journal of Wireless Information Networks*, vol. 19, no. 1, pp. 1–21, 2012.

[42] G. P. Halkes, T. van Dam, and K. G. Langendoen, "Comparing Energy-saving MAC Protocols for Wireless Sensor Networks," *Mobile Networks and Applications*, vol. 10, pp. 783–791, Oct. 2005.

[43] P. Djukic and S. Valaee, "Delay Aware Link Scheduling for Multi-Hop TDMA Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, pp. 870–883, 2009.