On Scheduling and Real-Time Capacity of Hexagonal Wireless Sensor Networks*

K. Shashi Prabh
Department of Computer Science
University of Virginia
shashi@cs.virginia.edu

Tarek F. Abdelzaher
Department of Computer Science
University of Illinois at Urbana-Champaign
zaher@cs.uiuc.edu

Abstract

Since wireless ad-hoc networks use shared communication medium, accesses to the medium must be coordinated to avoid packet collisions. Transmission scheduling algorithms allocate time slots to the nodes of a network such that if the nodes transmit only during the allocated time slots, no collision occurs. For real-time applications, by ensuring deterministic channel access, transmission scheduling algorithms have the added significance of making guarantees on transmission latency possible. In this paper we present a distributed transmission scheduling algorithm for hexagonal wireless ad-hoc networks with a particular focus on Wireless Sensor Networks. Afforded by the techniques of ad-hoc networks topology control, hexagonal meshes enable trivial addressing and routing protocols. Our transmission scheduling algorithm constructs network-wide conflict-free packet transmission schedule for hexagonal networks, where the overhead of schedule construction in terms of message exchanges is zero above and beyond that for topology control and other network control related functions. Furthermore, the schedule is optimal in the sense that the bottleneck node does not idle. We also present an implicit clock synchronization algorithm to facilitate scheduling. We derive the real time capacity of our scheduling algorithm. We present evaluations of our scheduling algorithm in the presence of topological irregularities using simulation.

1 Introduction

Wireless Sensor Networks (WSN) hold a very promising future. These are self-organized ad-hoc networks whose nodes are capable of sensing, gathering, processing and communicating data, especially the data pertaining to the physical medium in which they are embedded. It is envisioned that a typical WSN will consist of a large number of inexpensive nodes, each node covering a small section of the area of deployment. This allows for long-term sensing and monitoring at a fine grained level, both spatially and temporally.

Real-time applications refer to those performance-critical applications that require bounded service latency. There exists a class of WSN applications that is real-time, and requires bounded latency on data delivery. Since the nodes communicate over a shared medium, it is possible that if multiple transmissions overlap in time, some of them will collide. This problem is alleviated by either sensing the medium for possible ongoing transmissions and avoiding collisions, or by carefully timing (scheduling) the transmissions so that no collision occurs. The collision avoidance algorithms that attempt to *schedule* transmissions are referred to as *transmission scheduling algorithms*. While contention based collision avoiding algorithms offer a simpler and more dynamic solution to the medium access problem, schedule based algorithms can provide deterministic service delay bounds. It is crucial for real-time applications that the transmission delays be known and bounded.

Arikan proved that the problem of optimal scheduling in single frequency multi-hop wireless networks is NP-Hard [4]. Coffman *et al*. showed that the problem of scheduling file transfers in a distributed network such that the transfer is completed in minimum amount of time is NP-Complete [16]. Consequently, a number of distributed as well as centralized scheduling heuristics and Time Division Multiple Access (TDMA) protocols have been proposed, but these algorithms have significant set-up overhead and hence a new approach is called for energy constrained wireless sensor sensor networks [3].

Imposing a regular communication topology on networks affords simple but efficient network protocols. Among the well known regular topologies (simple paths, trees, hypercubes etc.), hexagons offer two very desirable properties in a distributed communication system – namely, constant node degree and good bisection width. In sufficiently dense wireless ad-hoc networks, it is possible to adjust the transmission power of radio units to achieve certain connectivity patterns. There exist techniques known as

*This work is supported in part by NSF grants CNS 05-53420 and CNS 06-26342.

topology control that attempt to achieve specified connectivity objective with minimum energy consumption (see [22] for a survey).

The use of hexagonal meshes has already been reported in previous literature. The HARTS system employs processors connected using a hexagonal *torus* network topology [24]. As the consequence of regularity afforded by the hexagonal torus topology, the addressing and resulting routing and broadcasting algorithms are not only simple but efficient as well [10]. Notably, the topology affords $O(1)$ time complexity route computation. An example of simplification of routing algorithms gained due to hexagonal topology in cellular networks can be found in [18]. Afforded by the techniques of ad-hoc networks topology control, we consider wireless ad-hoc networks of hexagonal topology, and show that for such networks, conflict free transmission schedules can be constructed without any message overhead for this purpose. We chose hexagonal topology because it allows optimal spatial reuse in a natural manner (see the section on scheduling). The packet scheduling algorithm presented here can be used on top of a Carrier Sense Multiple Access/ Collision Avoidance (CSMA/CA) MAC protocol to *emulate* a TDMA protocol; or it can be assimilated in TDMA or hybrid CSMA/TDMA MAC protocols.

To enable conflict-free transmission schedule in distributed systems, clock synchronization is essential. Relative synchronization of local clocks suffice for the purpose of transmission scheduling. Using the information contained in data packets, we present a clock synchronization algorithm that has zero message overhead. The protocol relies solely on overhearing neighbors' data transmissions to synchronize clocks and is shown to converge quickly to a common time.

In this paper, we also present an analysis of the real-time capacity of our scheduling algorithm. Traditionally, *network capacity* has referred to the measure of data carrying capability of a network where every communicated bit counts. We define *real-time capacity* of a network to be its information carrying ability for given deadlines, where only those information bits that arrive at their destination within the specified deadline count. For WSN the unit of data is a packet rather than bits. If a packet does not reach its destination by the given deadline, then its contribution to the real-time capacity is 0.

The rest of this paper is organized as follows. We present related work in Section 2, followed by the description of the system model in Section 3. We present our addressing, routing, scheduling, and clock synchronization algorithms in Section 4, and present the real-time capacity of our transmission scheduling algorithm in Section 4.4. We present simulation results in Section 5. We present the real-time implications of the work presented in this paper in Section 6. Finally, we present our conclusions in Section 7.

2 Related Work

Chen *et al*. present addressing and routing algorithms for (wired) multi-processors connected in hexagonal torus topology [10]. The addressing scheme was adapted and simplified for hexagonal cellular networks in [18]. The addressing scheme proposed in [18] and [10] use three axes to determine coordinates (and hence addresses) of nodes in a plane. Consequently, the addressing scheme of [18] has the downside of inability to assign a unique address to the nodes ([10] avoids this problem by imposing an additional ordering constraint). Myoupo [9] and Decayeux *et al*. [12] assign unique coordinates to nodes of a planar hexagonal network using only two axes. Some topological properties of hexagonal networks are presented in all the above mentioned papers. In this paper, we present yet another addressing scheme based on the radial symmetry of the many-to-one (convergecast) communication topology that yields simple algorithm for scheduling. We also outline (see Appendix) a transformation to map the new addresses to coordinates in a frame of two axes aligned at 120 degrees as in [9]. Clock synchronization algorithms in wireless sensor networks are presented in [13]. A survey can be found in [25].

A number of link or node scheduling algorithms for distributed systems have been proposed [6, 11, 20, 19, 14, 21]. These solutions depend upon message exchanges for schedule creation. Funneling MAC is a hybrid CSMA-TDMA MAC protocol designed for convergecast [3]. Funneling MAC is motivated by the scalability problems of the existing TDMA MAC protocols. In Funneling MAC, however, the TDMA scheduling is done in a small neighborhood of the sink only. Network-wide scheduling is needed if deterministic delay guarantee is to be provided for every packet. Caccamo and Zhang have proposed Implicit-EDF for conflict free prioritized transmission in WSN [8]. The authors propose to divide the sensor network in cellular network like (honeycomb) regions, where each cell operates at a different frequency than all of its six neighbors. Within each cell, nodes coordinate to achieve a contention free schedule. The communication is assumed to be uni-hop. If this approach is extended to multi-hop communication using multi-frequency radio, inter cellular coordination will require large number of message exchanges. Communication is expensive in wireless sensor networks. Consequently, optimizing wireless communication in these networks is a valuable optimization. In this paper, we show that by leveraging upon regular hexagonal network topology, network-wide valid and optimal schedule can be created without any message exchange.

In [1], the authors present real time capacity analysis for load balanced and convergecast traffic using time independent fixed priority scheduling algorithms and some simplifying assumptions. The expressions derived use the results for feasibility of tasks in a pipeline, obtained in [2]. A new approach is taken by Schmitt and Roedig in [23]. The authors use and extend the Network Calculus [7] to formulate the tradeoff between the buffer size requirements and power consumption with the worst case delay. A suitable arrival curve is presented. Koubaa *et al*. extend the Sensor Network Calculus formulation of Schmitt and Roedig for a cluster based tree topology and IEEE 802.15.4/ZigBee MAC protocol [17]. The authors incorrectly conclude that the bandwidth requirement grows exponentially with the depth of the tree in WSN. The exponential explosion in bandwidth requirement does not arise since the number of clusters do not grow exponentially with tree depth. The growth is in-fact at-most quadratic in tree depth.

Furthermore, while the formulation using Network Calculus is elegant, it requires further work before it can be applied to determine worst case delays in WSN. In [17], the worst case delay is obtained by using the Concatenation Theorem of Network Calculus. While the theorem has been proved for wired networks, it does not appear to hold in its current form for wireless networks. This happens due to the interference of transmissions on neighboring links. A Concatenation Theorem for links with mutual exclusion constraints needs to be furnished.

3 System model

We consider multi-hop transmission of data packets in wireless ad-hoc networks over single frequency shared radio channel. We consider convergecast traffic. Convergecast refers to the communication topology where nodes transmit packets to a common sink node, also called base-station or aggregation point. For densely deployed wireless sensor networks, we assume a two-tired cluster-based network topology. At the first tier, the cluster heads collect sensed data in their neighborhood. At the second tier, cluster heads send and route data packets to their destination. This cluster-based two-tier topology is similar to the original version [5] but with the exception that the cluster heads also assume the role of gateway nodes. Similar to the LEACH [15] architecture, we assume that nodes belonging to one cluster use CDMA to transmit data to the cluster head, and use appropriate radio power level to do so. Cluster heads maintain code distribution and transmission schedules of the active sensor nodes in their cluster. Thus, transmissions to different cluster heads can occur simultaneously. For inter-cluster communication however, we do not assume CDMA transmissions since this approach is not scalable for multi-hop transmissions. We shall focus on constructing schedules for conflict-free inter-cluster communication in the rest of this paper, and shall not dwell on intra-cluster transmissions anymore.

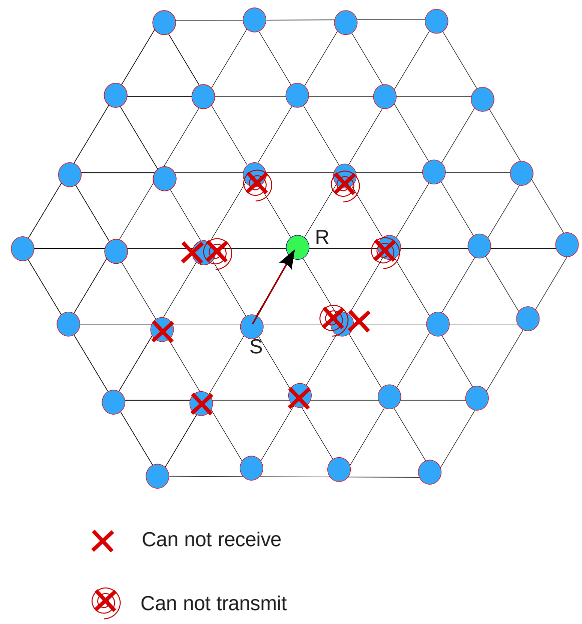

Figure 1. Interference in hexagonal networks

We consider nodes (cluster heads) of the network to be connected in regular hexagonal pattern. Only those nodes that are connected by an edge can hear some node's transmission. Appropriate node placement at deployment time and topology control algorithms at run-time can ensure the abstraction of a hexagonal network topology. While this logical topology might remain a slight over-simplification of the underlying actual connectivity, we show in the evaluation section that our algorithm is surprisingly robust to deviations of the actual topology from the hexagonal mesh.

Figure 1 illustrates the interference model used in this paper. Let us consider a transmission from node S to R shown in the figure. In the neighborhood of these two nodes, the one-hop neighbors of S can not receive any transmission, and that of R can not transmit any packet successfully while the transmission $S \rightarrow R$ is taking place. Such nodes are indicated with an $\times$ and an $\times$ with a spiral in the figure.

Hextants and Sides

Of the three principal diagonals, we select a pair inclined at 120 degrees to be the axes, labelled as X and Y (Figure 2.) Let XY denote the line bisecting the X and Y axes. These three lines drawn through the origin divide the plane into six regions. We refer to these six regions as *hextants*. Each of the six hextants are marked with roman numerals in the figure.

Consider a set of concentric hexagons. A hexagon ring at distance h edges from the origin contains $6h$ nodes. We

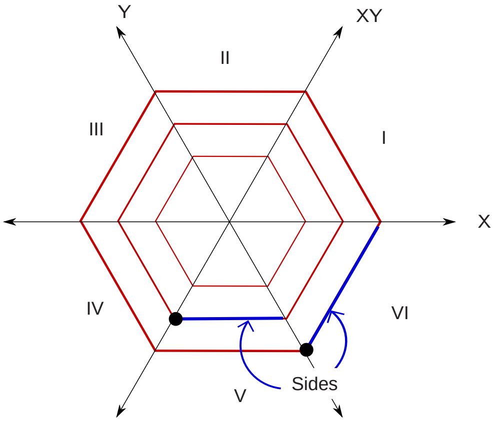

Figure 2. Hextants

define the set of h consecutive nodes contained within any hextant, inclusive of one node on one of the diagonals, as the nodes on one *side* at distance h. In this paper, we adopt the convention of including the first diagonal in the anti clockwise sense to the definition of a side.

4 TDMA Scheduling for Convergecast

In this section, we present addressing, routing, clock synchronization and distributed transmission scheduling algorithms for wireless ad-hoc networks of hexagonal network topology. The scheduling algorithm creates bounded latency TDMA schedules with spatial reuse. We consider convergecast traffic. The aggregation point is thus the bottleneck for this communication topology.

4.1 Addressing and Routing

For convergecast, the arrangement of nodes can be seen as that of concentric hexagons centered at the aggregation point, where the neighboring hexagons are separated by one hop. Due to this symmetry, we choose the aggregation point to be the origin. In such an arrangement, all nodes on any given concentric hexagon are equidistant from the origin. We assign addresses of the form $[h, i]$ to the nodes, where h is the shortest hop-count of the node from the origin and i denotes the index of a node located on the hop-h hexagon. The index starts at the x-axis and increases in the counter-clockwise direction. Hence the first hop nodes are addressed as $[1, 0], [1, 1] \dots [1, 5]$. For brevity, we use $[h, i]$ to refer to a node as well as its address. Observe that nodes of the form $[h, .]$ are all located on the same hexagonal ring at distance h form the origin(. denotes wildcard). Since the number of nodes on h^{th} hop hexagon is $6 \times h$, the node addresses range from $[h, 0]$ to $[h, 6h - 1]$ (see Figure 4 for an example).

For routing, we use the following algorithm. All the nodes falling on the X, Y or XY axes route packets along the straight line joining them to the origin (Figure 3). All other nodes route packets as follows: in hextants I and IV, packets are routed parallel to the XY-axis towards the origin, in hextants II and V, packets are routed parallel to the Y-axis towards the origin, in hextants III and VI, packets are routed parallel to the X-axis towards the origin (see Figure 4 for an example). Once the packet reaches one of the diagonals, it is routed along the diagonal to the base-station. In other words, all non-diagonal nodes on a given side route towards the diagonal at 60 degrees. This algorithm keeps the traffic flow in all regions of the network nearly balanced, and routes the packets along a shortest path.

VAR: $q = \lceil i/h \rceil$
Route: $[h, i] \Rightarrow [h - 1, i - q]$

Algorithm 1: Routing

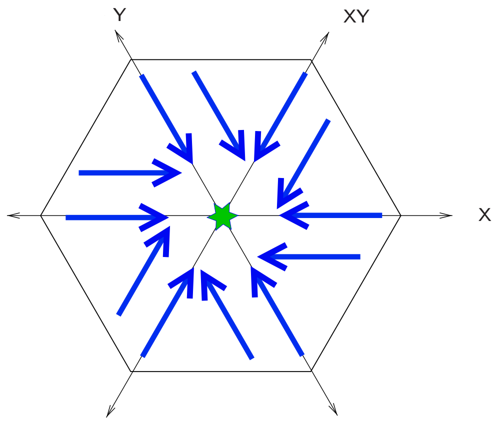

Figure 3. Routing

4.2 Scheduling

In the following, we derive closed form expressions for construction of cyclic TDMA schedule with spatial re-use such that:

- during each cycle, every node can send one locally originated packet to the sink
- by the end of each cycle, all such packets are received at the sink

Let H be the radius (the maximum shortest distance hop-count of any node to the aggregation point) of the hexagonal network. Since the number of nodes on the h^{th} hop hexagon is $6h$, the total number of nodes (excluding the sink node) is given by $\sum_{h=1}^{H} 6h = 3H^2 + 3H$. Therefore, the total number of packets that must be received by the sink every cycle is $3H^2 + 3H$. Since the sink node can receive only one packet at a time, a minimum of $3H^2 + 3H$ time slots are needed to transmit all of these packets to the sink, where each time slot is assumed to be large enough to transmit one packet over one hop (packets are assumed to be of the same size).

The scheduling algorithm employs spatial reuse – during every time slot, up-to H transmissions are scheduled. Additionally, the transmissions are scheduled in such a way that packets flow continuously to the sink node, and by the end of $3H^2 + 3H$ time slots, the sink receives all packets.

4.2.1 Intuition into Scheduling

Of the six hextants, no two odd hextants have any edge in common. Similarly, all three even hextants do not have any edge in common. Therefore, except for the last hop transmissions to the sink node, any transmission in hextant I can not interfere with any other transmission in hextants III and IV. Furthermore, transmission(s) on a given side do not interfere with transmission(s) on the sides that are 3 hops or more apart – even if in the same hextant. We accomplish the goal of conflict free scheduling with spatial reuse into two steps. First, we divide the nodes into six disjoint sets where, except for the neighboring nodes on a side, no two nodes have an edge in common. And in the second step, we allocate time slots to the nodes such that the neighboring nodes of a given side transmit in separate time slots.

Formally speaking, we partition[1] the nodes of the network into six subsets such that all nodes of the form $[h, .]$ in any of the subsets do not interfere with any other node $[h', .]$ of the same subset where $h \neq h'$. For illustration, let us consider a three hop network (Figure 4). Nodes $[1, 0]$, $[2, 4], [2, 5]$ and $[3, 12], [3, 13], [3, 14]$ belong to the three sides of odd hextants (which implies that the three diagonal nodes on the sides are mutually separated by 120 degrees.) In this arrangement, none of the second hop nodes $[2, 4], [2, 5]$ interfere with any of the first or the third hop nodes. To generalize, starting at any of the first hop node $[1, i]$, we choose all nodes on the side at 120 degrees of the next concentric hexagon, where the angle is measured between the diagonals. This partitioning procedure ensures that any two set of nodes located at two different hops but in the same hextant are at-least 3 hops apart, and thus do not interfere. For example, in a 4-hop network, the set of nodes of the same partition as $[1, 0]$ and falling into the same hextant are $[4, 0], [4, 1], [4, 2], [4, 3]$. The nearest node to $[1, 0]$ is $[4, 0]$ which is 3 hops away.

However, being neighbors, not all nodes on any given side may transmit simultaneously. For example, in Figure 4 not all of $[2, 4], [2, 5]$ or $[3, 12], [3, 13], [3, 14]$ can transmit simultaneously without conflict. We structure TDMA schedule cycles as a series of sub-cycles where each sub-cycle consists of six time slots. We schedule the neighboring nodes in successive sub-cycles, hence in different time slots. For example, the set of nodes corresponding to a valid schedule for this partition is $\{\{[1, 0], [2, 4], [3, 12]\}, \{[1, 0], [2, 5], [3, 13]\}$ $\{[1, 0], [2, 4], [3, 14]\} \ldots \{[1, 0], [2, 4]\} \ldots \{[1, 0]\}\}$. This example is a 3-hop network. Hence it is sufficient to allocate only one time slot to all nodes at the third hop since each originate one packet and route none. It suffices to allocate two slots to $[2, 5]$ since it originates one and routes one. Similarly, it suffices to allocate three slots to $[2, 4]$ and fifteen time slots to the nodes at first hop respectively. Schedules for the rest five partitions can be constructed symmetrically. All the six subsets can then be interleaved to form a set of sub-cycles. Since there are $3H^2 + 3H$ time slots in one cycle, it contains $(H^2 + H)/2$ sub-cycles.

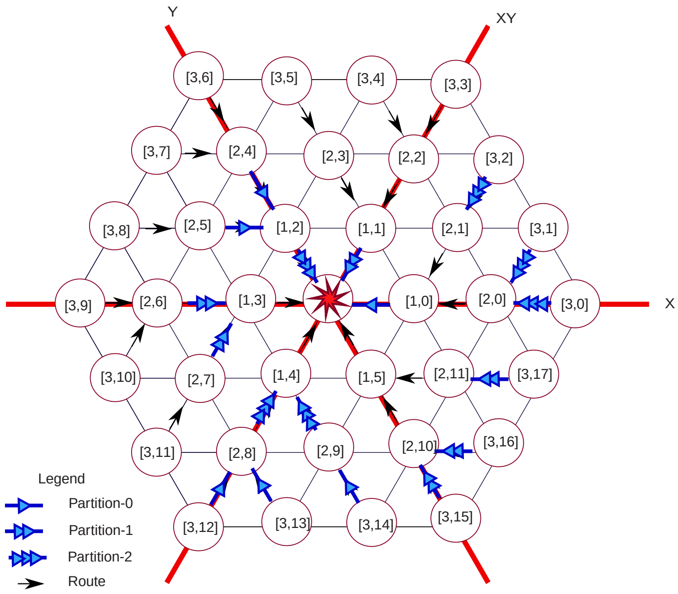

Figure 4. Example: Addressing, routing and partitions of a 3-hop network

4.2.2 Closed form expressions for scheduling

We now derive a set of expressions that determine time slots for transmission as a function of the tuple $[h, i]$. First, we obtain an expression to determine the partition of the nodes and then we obtain expressions for the time slots when a node may transmit.

Let R be defined as

$$R([h, i]) \triangleq (h - 1) \bmod 3. \tag{1}$$

[1]Partition of a set is defined as the disjoint set of its subsets such that the union of the subsets is the set.

The variable R indicates spatial reusability. Nodes of different h but the same R are at-least 3 hops away. Let Q be defined as

$$Q([h,i]) \triangleq \left\lfloor \frac{i}{h} \right\rfloor. \quad (2)$$

Observe that $Q+1$ gives the hextant of a node. Then the partition $P([h,i])$ of $[h,i]$ is given by

$$P = (Q - 2R) \bmod 6,^{2} \quad (3)$$

where the arguments are omitted for the sake of brevity.

The complimentary expression $P = (2R - Q) \bmod 6$ gives another partition where the partitions are in the clockwise order.

Lemma 1. *The expression $P = (Q - 2R)$ mod 6 partitions h consecutive nodes of one side and no other node of an h^{th}-hop hexagon in the same subset.*

Proof. We shall show that $P = (Q - 2R) \bmod 6$ assigns the consecutive nodes $\{[h,kh], \dots [h,(k+1)h-1]\}$ to one partition, where $0 \le k < 6$ is an integer. Let us denote this set by $\mathcal{N} \triangleq \{[h,kh+j] | 0 \le j < h\}$. From (1), $R(\mathcal{N})$ is the same for all nodes of the form $[h,.]$, and hence is the same for all nodes in $\mathcal{N}$. From (2), $Q(\mathcal{N}) = \left\lfloor \frac{kh+j}{h} \right\rfloor = k$. Hence Q is also the same for all nodes in $\mathcal{N}$. Furthermore, $Q(\mathcal{N}) \neq Q(\mathcal{N}')$ if $k \neq k'$. Since k takes 6 unique values, this lemma follows. □

Lemma 2. *If a partition p includes nodes of some side in hextant q, then, p contains nodes from the hextants $(q \pm 2)$ mod 6. In other words, the diagonal nodes from neighboring hops in p are 120 degrees apart.*

Proof. We first show that if some diagonal node $N = [h,kh]$, where $0 \le k < 6$ is some integer, is in partition p, then another diagonal node $N' = [h+1, (k+2)(h+1) \bmod 6(h+1)]$ is also in the same partition.

$$\begin{aligned} R(N) &= (h-1) \bmod 3 \\ R(N') &= h \bmod 3 \\ Q(N) &= k \\ Q(N') &= (k+2) \bmod 6 \end{aligned} \quad (4)$$

Therefore, $P(N') - P(N) = 2 \bmod 6 - 2 \bmod 3 = 0$, or N and N' belong to the same partition. Similarly, $N'' = [h-1, (k-2)(h-1) \bmod 6(h-1)]$ also belongs to the same partition as N. From (4), if N is in hextant q, then the other two are in hextants $(q \pm 2) \bmod 6$. But our choice of N was arbitrary. Hence using Lemma 1, this lemma follows. □

Theorem 1. *The expression $P = (Q - 2R)$ mod 6 partitions the network such that nodes on any two different sides in the same partition do not interfere.*

[2] $0 \le P \le 5$. According to modulus algebra conventions, if the residue is negative, P is made positive by adding 6.

Proof. From Lemma 2, nodes that belong to the same partition are either in all odd or all even hextants. Since the diagonal nodes on each neighboring-hop sides are 120 degrees apart, it also implies that if two sides are the same hextant and in the same partition, they must be $3n$ hops apart, where $n > 0$ is an integer. Hence the proof. □

The nodes falling on the principal diagonals have address of the form $[h,kh]$, where integer $0 \le k \le 5$ and that of the rest of the nodes is of the form $[h, kh+j]$ where integer $1 \le j < h$. It is easily seen that packet accumulation on the nodes on the six principal diagonals, $[h,kh]$, is given by $(H-h+1)(H-h+2)/2$ and that on the other nodes $[h, i \neq kh]$ by $H-h+1$. The nodes farther from the sink thus need smaller number of slots than those that are nearer to the sink. Recall that, in any given partition, there are $h-1$ consecutive nodes located on any one side at hop-distance h. We schedule all nodes on any given side alternately until the diagonal and non-diagonal nodes are scheduled for $H-h+1$ time slots, and then schedule the diagonal nodes for an additional $\{(H-h+1)(H-h+2)/2\} - \{(H-h+1)\} = (H-h+1)(H-h)/2$ time slots.

Let

$$K = i - \left\lfloor \frac{i}{h} \right\rfloor h. \quad (5)$$

Then the series of nodes having index $\{kh, \dots (k+1)h-1\}$ get $K = 0, 1, \dots h-1$.

Proof. Let

$$i = kh + j. \quad (6)$$

Then we want to show that $j = K$.

$$\left\lfloor \frac{i}{h} \right\rfloor = k + \left\lfloor \frac{j}{h} \right\rfloor = k, \quad (7)$$

since $j = 0, \dots h-1$. Therefore, on plugging (6) and (7) into (5), we get $K = kh + j - kh = j$. □

Since P uniquely identifies one partition, we start scheduling nodes of partition P starting at time slot P. Assuming that the time slots are sequenced as $0, 1, 2, \dots$, nodes in partition $0 \le P \le 5$ are scheduled in time slots of the form $P + 6n$, where n is a positive integer. Since the neighboring nodes of a side have consecutive K values, assigning them the time slots of the form $P + 6K$ allocates slots to neighboring nodes in consecutive sub-cycles.

Since there are $6h$ nodes on an h^{th}-hop hexagon, the adjacent slots allocated to a given node are separated by $6h$. Nodes $[h,i]$ are scheduled during the time slots:

$$t = P + 6K + 6nh, \text{ where } n = 0, 1, \dots, (H-h). \quad (8)$$

The diagonal nodes $[h,kh]$ are scheduled during the additional time slots:

$$t = P + 6(H-h+1)h + 6m, \quad (9)$$

where $m = 0, 1, \ldots, (H-h)(H-h+1)/2 - 1$. Nodes can optimize energy consumption by going to "sleep" state after transmitting all packets.

During every time slot, one packet is transmitted to the sink and one packet to a node at one hop from the sink (except for the last six time slots). Due to the symmetry of traffic, at the end of every sub-cycle, all the first hop nodes contain exactly the same number of packets. Hence follows the continuity of packet flow to the sink node.

Theorem 2. *The transmission schedule obtained by the algorithm presented above is optimal in the sense that it achieves the minimum schedule length.*

Proof. The algorithm schedules every node for u time slots if the number of packets routed by the node is $u-1$. One slot is allocated for the packet originated at the node. From (9), during any given cycle, the last node to be scheduled has $P = 5, h = 1$, and $m = (H-h)(H-h+1)/2 - 1$. Thus it is scheduled in the time slot $t = 3H(H+1) - 1$. Thus, from (9), the length of one cycle is $3H(H+1)$, which is equal to the total number of packets received at the sink. The sink node can receive only one packet at a time. Therefore, $3H(H+1)$ slots are necessary for it to receive all the packets. Since, the algorithm schedules all the packets in the minimum amount of slots, the optimality follows. □

4.3 Clock Synchronization

Since the order of transmissions is pre-determined by the scheduling algorithm presented in the previous section, over-hearing of transmissions can be used to deduce neighboring nodes' own schedules and also to synchronize their clock. The algorithm for implicit clock synchronization is presented in Algorithm 2. Observe that the proposed algorithm does not ensure monotonicity of time as adjustments can set the clock into the past. While this is, in general, a deprecated practice, in the special case of data collection sensor networks, where the primary network task is data transmission to a base-station, no serious adverse consequences ensue.

VAR: t {local time}
loop
  Do upon listening/receiving from node $[h', i']$
  $t \leftarrow (t + t')/2$
end loop

Algorithm 2: Clock synchronization

4.4 Real-Time Capacity

During one cycle, all the $6h$ nodes at every h^{th} hop hexagon send one locally originated packet to the base-station. Thus each such packet contributes sh byte-hops to the real-time information transmission, where s is the packet size. Thus, the total amount of information transmission by all nodes in one cycle is

$$\sum_{h=1}^{H} 6.s.h^2 = sH(H+1)(2H+1). \tag{10}$$

The algorithm presented in this paper transmits $3H^2 + 3H$ packets in the same number of time slots. Each time slot is s/W seconds, where W is the bandwidth. Thus, the real-time capacity using the transmission scheduling algorithm presented here is:

$$\begin{aligned} RTC &= \frac{sH(H+1)(2H+1)}{s/W(3H^2+3H)} \\ &= \frac{W(2H+1)}{3} \text{ byte-hops/sec.} \end{aligned} \tag{11}$$

Therefore, at the cost of linear increase in latency (due to the schedule cycle size), the real time capacity grows sub-linearly with the size of the network.

5 Evaluation

We implemented a simulator to evaluate the robustness of our transmission scheduling algorithm in the presence of topological irregularities, namely long links, and to evaluate the rate of convergence of the implicit clock synchronization algorithm. Long links refer to those that arise when the interference range of a node is larger causing it to interfere with neighbors that it would not normally reach in a hexagonal topology. Observe that another form of irregularity is when the transmission range is too short, precluding connectivity along some of the hexagonal edges. This latter type, however, can be solved by deploying the nodes closer together until only long links remain. The above argument justifies considering only long links in this evaluation.

Figure 5 shows the cumulative distribution of separation between receiver node(s) and the other sender nodes for three networks of size $H = 5$ (90 nodes), $H = 10$ (330 nodes) and $H = 20$ (1260 nodes). That there are no nodes of 2 or less hop separation can be seen as an experimental validation of our scheduling algorithm. The point of inflexion of the curves fall around $H/2$, meaning that the largest fraction of transmissions are separated by $H/2$ hops. Notably, as the network size increases, the curves get flatter, indicating that even larger fraction of transmissions happen at farther separations. Thus, our formalism should be fairly resilient in the presence of random long links.

To verify our assumption, we simulated networks of irregular topology. We introduced long links where the excess length (> 1) of the links causing interference, x, was

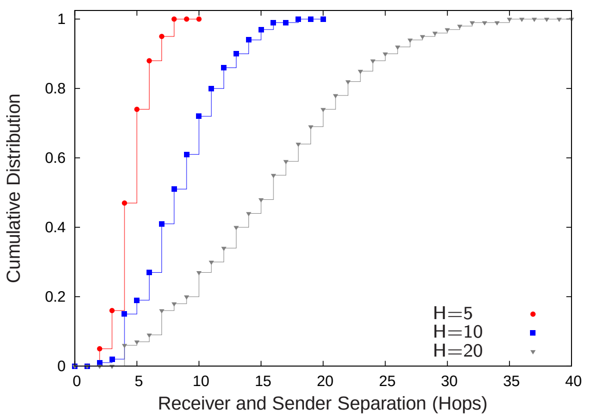

Figure 5. Cumulative distribution of separation between receiver and simultaneous senders (to different other receivers)

Poisson distributed with mean λ (i.e., the probability density function $\mathbf{p}(x) = 1/\lambda \exp^{-x/\lambda}$). In Figure 6, the percentage of collisions is reported as a function of the mean of the Poisson distribution, λ. Consistent with Figure 5, as the size of the network increases, the effect of irregularity diminishes. Collisions show up after $\lambda = 0.2$, indicating that the scheduling algorithm is robust in the presence of occasional long links. The performance degradation is graceful with the increase in the size of irregularity.

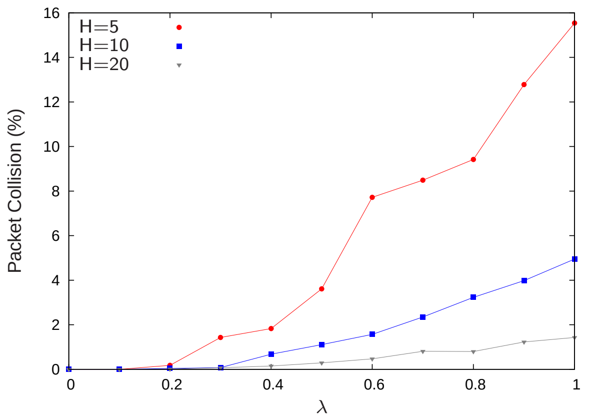

Figure 6. Percentage packet collision in the presence of radio irregularity.

We simulated the our clock synchronization algorithm to evaluate its convergence rate. We started schedule cycles with nodes having uniformly randomly distributed clock times. We obtained the standard deviation of local times of all nodes as a function of simulation time where time granularity was the slot size. First we kept the initial range of random errors, E, in $[0, 100]$ and varied the network size. Shown in Figure 6, we observed a sharp exponential decline in the coefficient of variation initially, followed by a slow convergence to finer synchronization. Our clock synchronization algorithm achieves standard deviation < 1 within one schedule cycle. Next, to evaluate the effect of initial range of clock asynchrony on the convergence, we varied the error range exponentially, keeping the network size fixed to $H = 10$. In Figure 8, we observe that even though the initial range of errors are varied exponentially, the rate of convergence is only linear. This is a cumulative consequence of time averaging.

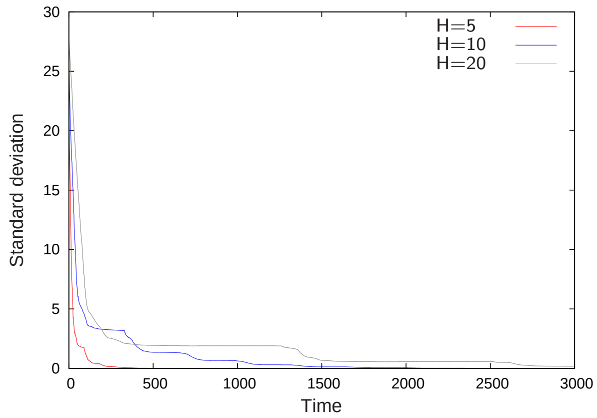

Figure 7. Convergence of clocks. $E = 100$ for all plots

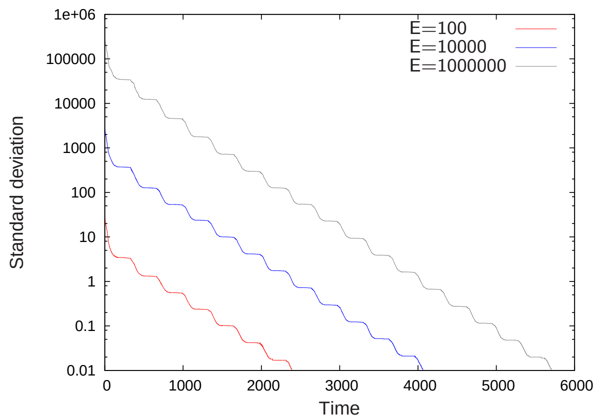

Figure 8. Convergence of clocks parameterized by initial error size. H=10 for all plots.

6 Discussion and Scheduling Implications

The scheme presented and evaluated in this paper has significant implications on the ability of sensor networks to provide real-time guarantees. We have shown that given proper (i.e., hexagonal) node placement and topology control algorithms (that create a hexagonal mesh), it is possible to develop trivial addressing, routing, and clock synchronization algorithms that guarantee each node a fixed portion of network bandwidth and guarantee a bounded latency to the sink node. In particular, each node will be able to send for one time unit every $3H^2 + 3H$ time units. Its communication will reach the sink node in no more than $3H^2 + 3H$ time units. Since our algorithm transmits all packets in minimum time, in the case where all packets have the same deadline, our scheduling algorithm is optimal.

Previously proposed schedulability algorithms for real-time broadcast LANs (where each node can send for x out of y seconds on the LAN) can now be applied to schedule real-time traffic on the multi-hop sensor network with the added observation that a transmitted packet will incur a latency of no more than $3H^2 + 3H$ time units before it is delivered. Our simple addressing, routing, and clock synchronization algorithms provide a practical solution to ensure predictability of the network, making a wealth of prior schedulability analysis techniques applicable. Moreover, the sink node is 100% utilized, which is optimal. The protocol is self-synchronizing and easy to implement.

7 Conclusions

In this paper we presented a novel way to design wireless ad-hoc and sensor networks, namely constructing a hexagonal network topology. We presented addressing and constant time routing algorithms for multi-hop hexagonal networks. We presented closed form expressions that creates network-wide conflict free TDMA schedule with zero message overhead. The resulting MAC protocol gives equal bandwidth to every node, and is optimal in the sense that the base-station does not idle. It affords real-time guarantees on packet delivery. Using simulation, we showed that our algorithms are robust.

Appendix: Oblique Coordinates and Transformations

In this section, we present a transformation from addresses of the form $[h, i]$ to the coordinates of the nodes in oblique Cartesian system. As we shall see, the use of Cartesian coordinates makes calculation of distance between any pair of nodes simple.

We choose one of the principal diagonals as the X axis and the other inclined at 120 degrees to be the Y axis.

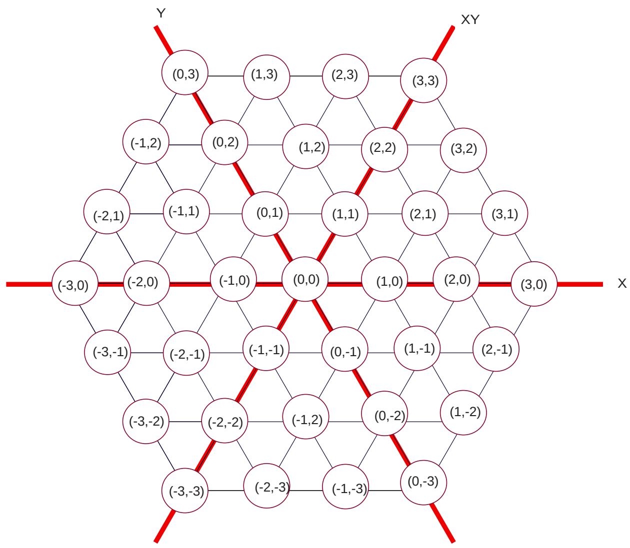

Figure 9. Oblique coordinate system for hexagons

An example of assignment of coordinates in this system is shown in Figure 9. To transform $[h, i]$ to (x, y), we use the equations for hextants, Q (2) and K (5). Recall that

$$Q = \left\lfloor \frac{i}{h} \right\rfloor, \quad K = i - \left\lfloor \frac{i}{h} \right\rfloor h.$$

The transformation rules are as follows:

Q	Transformation
0	$[h, i] \Rightarrow (h, i)$
1	$[h, i] \Rightarrow (h - K, h)$
2	$[h, i] \Rightarrow (-K, h - K)$
3	$[h, i] \Rightarrow (-h, -K)$
4	$[h, i] \Rightarrow (K - h, -h)$
5	$[h, i] \Rightarrow (K, K - h)$

Table 1. Transformation rules

Inverse transformation can be obtained similarly.

Theorem 3. *The distance between nodes* (x_1, y_1) *and* (x_2, y_2) *is given by* $\text{MAX}\{|x_1 - x_2|, |y_1 - y_2|, |x_1 - x_2 - y_1 + y_2|\}$.

Proof of this theorem is similar to that of Theorem 2 of [12].

References

[1] T. Abdelzaher, K. S. Prabh, and R. Kiran. On real-time capacity limits of multihop wireless sensor networks. In *Proceedings of the 25th IEEE Real-Time Systems Symposium*

(RTSS'04). IEEE Computer Society Press, Los Alamitos, CA, 2004.

[2] T. Abdelzaher, G. Thaker, and P. Lardieri. A feasible region for meeting aperiodic end-to-end deadlines in resource pipelines. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*, pages 436–445. IEEE Computer Society Press, Los Alamitos, CA, 2004.

[3] G.-S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell, and F. Cuomo. Funneling-MAC: a localized, sink-oriented MAC for boosting fidelity in sensor networks. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 293–306, New York, NY, USA, 2006. ACM Press.

[4] E. Arikan. Some complexity results about packet radio networks. *Information Theory, IEEE Transactions on*, 30(4):681–685, Jul 1984.

[5] D. J. Baker. Distributed control of broadcast radio networks with changing topologies. In *INFOCOM*, pages 49–55, 1983.

[6] D. J. Baker, J. Wieselthier, and A. Ephremides. A distributed algorithm for scheduling the activation of links in a self-organizing, mobile, packet-radio network. In *IEEE ICC '82: Proceedings of the IEEE International Conference on Communications*, pages 2F.6.1–2F.6.5. IEEE Computer Society Press, Los Alamitos, CA, 1982.

[7] J.-Y. L. Boudec and P. Thiran. *Network calculus: a theory of deterministic queuing systems for the internet*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[8] M. Caccamo and L. Y. Zhang. The capacity of implicit EDF in wireless sensor networks. In *Real-Time Systems, 2003. Proceedings. 15th Euromicro Conference on*, pages 267 – 275. IEEE Computer Society Press, Los Alamitos, CA, July 2003.

[9] J. Carle and J.-F. Myoupo. Topological properties and optimal routing algorithms for three dimensional hexagonal networks. *High Performance Computing in the Asia-Pacific Region, 2000. Proceedings. The Fourth International Conference/Exhibition on*, 01:116, 2000.

[10] M.-S. Chen, K. G. Shin, and D. D. Kandlur. Addressing, routing, and broadcasting in hexagonal mesh multiprocessors. *IEEE Trans. Comput.*, 39(1):10–18, 1990.

[11] I. Cidon and M. Sidi. Distributed assignment algorithms for multihop packet radio networks. *Computers, IEEE Transactions on*, 38(10):1353 –1361, Oct. 1989.

[12] C. Decayeux and D. Seme. 3D hexagonal network: Modeling, topological properties, addressing scheme, and optimal routing algorithm. *IEEE Trans. Parallel Distrib. Syst.*, 16(9):875–884, 2005.

[13] J. Elson and D. Estrin. Time synchronization for wireless sensor networks. In *IPDPS '01: Proceedings of the 15th International Parallel & Distributed Processing Symposium*, page 186, Washington, DC, USA, 2001. IEEE Computer Society.

[14] S. Gandham, M. Dawande, and R. Prakash. Link scheduling in sensor networks: distributed edge coloring revisited. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, pages 2492–2501, 2005.

[15] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, page 8020, Washington, DC, USA, 2000. IEEE Computer Society.

[16] E. G. C. Jr., M. R. Garey, D. S. Johnson, and A. S. LaPaugh. Scheduling file transfers in a distributed network. In *PODC '83: Proceedings of the 2nd annual ACM symposium on Principles of Distributed Computing*, pages 254–266. ACM, 1983.

[17] A. Koubaa, M. Alves, and E. Tovar. Modeling and worst-case dimensioning of cluster-tree wireless sensor networks. In *Proceedings of the 27th IEEE Real-Time Systems Symposium (RTSS'06)*, pages 412–421, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[18] F. G. Nocetti, I. Stojmenovic, and J. Zhang. Addressing and routing in hexagonal networks with applications for tracking mobile users and connection rerouting in cellular networks. *IEEE Trans. Parallel Distrib. Syst.*, 13(9):963–971, 2002.

[19] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient collision-free medium access control for wireless sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 181–192, New York, NY, USA, 2003. ACM Press.

[20] S. Ramanathan. A unified framework and algorithm for channel assignment in wireless networks. *Wirel. Netw.*, 5(2):81–94, March 1999.

[21] I. Rhee, A. Warrier, M. Aia, and J. Min. Z-MAC: a hybrid mac for wireless sensor networks. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 90–101, New York, NY, USA, 2005. ACM Press.

[22] P. Santi. Topology control in wireless ad-hoc and sensor networks. *ACM Comput. Surv.*, 37(2):164–194, 2005.

[23] J. B. Schmitt and U. Roedig. Sensor network calculus - a framework for worst case analysis. In *DCOSS*, pages 141–154, 2005.

[24] K. G. Shin. HARTS: A distributed real-time architecture. *Computer*, 24(5):25–35, 1991.

[25] B. Sundararaman, U. Buy, and A. D. Kshemkalyani. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3(3):281–323, 2005.