A Decentralized Scheduling Algorithm for Time Synchronized Channel Hopping
(Invited Paper)

Andrew Tinka[1], Thomas Watteyne[2], and Kris Pister[2]

[1] Electrical Engineering,
University of California, Berkeley, USA
tinka@berkeley.edu
[2] Berkeley Sensor & Actuator Center
University of California, Berkeley, USA
{watteyne,pister}@eecs.berkeley.edu

Abstract. Time Synchronized Channel Hopping (TSCH) is an existing medium access control scheme which enables robust communication through channel hopping and high data rates through synchronization. It is based on a time-slotted architecture, and its correct functioning depends on a schedule which is typically computed by a central node. This paper presents, to our knowledge, the first scheduling algorithm for TSCH networks which both is distributed and which copes with a mobile nodes.
Two scheduling algorithms are presented. Aloha-based scheduling allocates one frequency channel for broadcasting advertisements for new neighbors. Reservation-based scheduling augments Aloha-based scheduling with a dedicated slot for targeted advertisements based on gossip information. A mobile ad-hoc network with frequent connectivity changes is simulated, and the performance of the two proposed algorithms is assessed against the optimal case. Reservation-based scheduling performs significantly better than Aloha-based scheduling, suggesting that the improved network reactivity is worth the increased algorithmic complexity and resource consumption.

Key words: time-synchronized channel hopping, mobile ad-hoc networks, decentralized scheduling, simulation

1 Introduction

The Floating Sensor Network (FSN) project [1] at UC Berkeley builds autonomous floating sensor packages for deployments in rivers and estuaries (see Fig. 1). The floating sensors (or "drifters", in the terminology of the hydrodynamic community) are untethered; once deployed in the river, they are carried by the current. Some versions of the drifters can affect trajectory modifications using propellers. The Berkeley FSN devices carry two communication systems: a GSM module for transmissions to a central server, and an low-power low-range IEEE802.15.4-2006 [2] radio for communication between nodes.

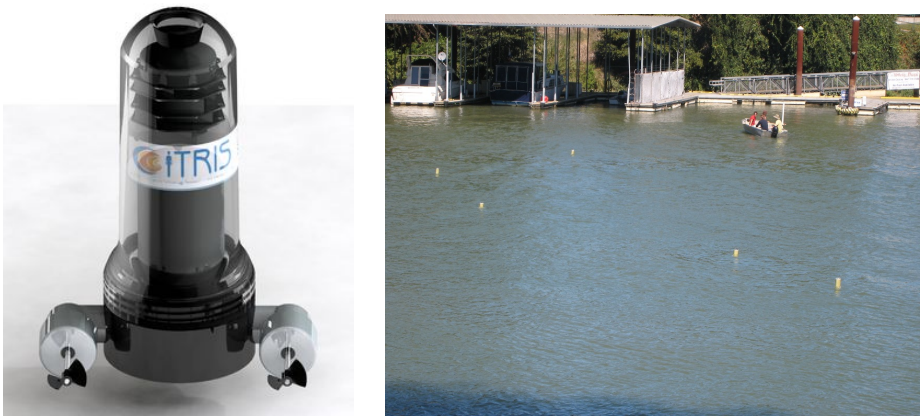

Fig. 1. Prototype of a motorized drifter (left). Five drifters in a river (right).

The GSM communication channel is both expensive (both monetary and in terms of energy) and unreliable (due to variable GSM coverage). One strategy for delivering data from individual nodes to a remote server is to have one or several nodes with good GSM connection act as *ad hoc* sink nodes. Nodes connected by IEEE802.15.4-2006 links that do not have their own GSM connections available can send their data to one of the sinks, which retransmits it via GSM to the server. Since the nodes with GSM connectivity are not known a priori, the design objective for the IEEE802.15.4-2006 network must be to maximize point-to-point connectivity.

We define the *physical connectivity graph* to be the ensemble of wireless links good enough to be used for communication at a given instant in time. We define the *logical connectivity graph* to be the set of links scheduled to be used at the same instant.

The mobility of the nodes means that the physical connectivity between nodes changes significantly over time. Global connectivity is not guaranteed. Therefore, centralized schemes for determining a communication schedule are poor fits for the problem. Our goal is to develop an algorithm which schedules intermittent bi-directional links between neighboring nodes as these links become available. We assess candidate schemes by evaluating how close the logical connectivity gets to the physical connectivity; that is, how many of the possible links are actually scheduled by the protocol.

Time Synchronized Channel Hopping (TSCH) is a Medium Access Control (MAC) scheme which enables robust communication through channel hopping and high data rates through synchronization. It is based on a time-slotted architecture, where a schedule indicates to the nodes on which slot and which frequency channel to transmit/receive data to/from which neighbor. TSCH is being standardized by the the IEEE802.15.4e Working Group [3], and expected to be included in the next revision of the IEEE802.15.4 standard.

TSCH only defines the mechanism, and makes no assumption on how the schedule is built. Typically, nodes report their communication needs (expressed in terms on throughput, reliability and latency) to a central scheduler which computes a schedule and injects this into the network. This technique has proven perfectly adequate for static networks such as industrial control Wireless Sensor Networks (WSNs). A distributed solution seems more appropriate for mobile networks. In those type of networks, each topological change would have to be reported to the central scheduler, which would have to recompute a schedule and inform the nodes about the change.

This paper presents distributed scheduling algorithms to be used on top of a TSCH MAC protocol, and which deals with mobile nodes. It is, to our knowledge, the first of its kind.

The remainder of this paper is organized as follows. Section 2 provides a comprehensive overview of MAC protocol approaches and standardization activities, and highlights the need for a distributed scheduling algorithm for TSCH. Section 3 then details the two scheduling algorithms proposed in this paper, called Aloha-Based Scheduling and Reservation-Based Scheduling. We evaluate the goodness of the connectivity achieved by both algorithms when compared to the physical connectivity in Section 4. The metric of interest are relative connectivity (a metric for goodness in the static case) and link duration (dynamic case). Finally, Section 5 concludes this paper and presents future work.

2 Time Synchronized Channel Hopping

There are two main approaches for regulating access to a shared wireless medium: contention-based and reservation-based approaches. Any derived Medium Access Control (MAC) protocol is based on one of those two approaches, or a combination thereof.

Contention-based protocols are fairly simple, mainly because neither global synchronization nor topology knowledge is required. In a contention-based approach, nodes compete for the use of the wireless medium and only the winner of this competition is allowed to access to the channel and transmit. Aloha and Carrier Sense Multiple Access (CSMA) are canonical representative schemes of contention-based approaches. They do not rely on a central entity and are robust to node mobility, which makes it intuitively a good candidate for dynamic mobile networks.

Preamble-sampling is a low-power version of contention-based medium access, widely popular in WSNs. All nodes in the network periodically sample the channel for a short amount of time (at most a few milliseconds) to check whether a transmission is ongoing. Nodes do not need to be synchronized, but all use the same check interval. To ensure all neighbors are listening, a sender pre-pends a preamble which is at least as long as the check interval. Upon hearing the preamble, nodes keep listening for the data that follows. The optimal check interval, which minimizes the total energy expenditure, is a function of the average network degree and its load. Durations of 100 ms are typical. Numerous works

have proposed ways to optimize the sampling [4], reducing the preamble length by packetization [5] or by synchronizing the nodes [6].

Despite its success, contention-based protocols suffer from degraded performance in terms of throughput when the traffic load increases. In addition, the distributed nature prevents them from achieving the same efficiency as ideal reservation-based protocols. Finally, frequency agility is hard to achieve by such protocols as nodes are not synchronized.

Reservation-based protocols require the knowledge of the network topology to establish a schedule that allows each node to access the channel and communicate with other nodes. The schedule may have various goals such as ensuring fairness among nodes, or reducing collisions by avoiding that two interfering nodes or more access to the channel and transmit at the same time. TDMA (Time Division Multiple Access) is a representative example for such a reservation-based approach.

In TDMA, time is divided into slots which are grouped into superframes which repeat over time. A schedule is used to indicate to each node when it has to transmit or receive, to/from which neighbor. Provided the schedule is correctly built, transmissions do not suffer from collisions, which guarantees finite and predictable scheduling delays and also increases the overall throughput in highly loaded networks.

The reliability of a wireless link is mainly challenged by external interference and multi-path fading. [7] and [8] show how channel hopping combats both of these, respectively. If a transmission fails, the sender retransmits the packet on a different frequency channel. Because this frequency change causes the wireless environment to be different, the retransmission has a higher probability of being successful than if it were retransmitted on the same channel.

Channel hopping was first applied to Wireless Sensor Networks (WSNs) in a proprietary protocol called Time Synchronized Mesh Protocol (TSMP) [9]. In TSMP, nodes in the network are synchronized on a slotted time base. An individual slot is long enough for a sender to send a data frame, and for a receiver to acknowledge correct reception (a slot of 10 ms is common). L consecutive slots form a superframe, which indefinitely repeats over time. A schedule of length L slots indicates, for each slot, whether the node is supposed to transmit or receive, to/from which neighbor and on which frequency channel. TSMP runs on IEEE802.15.4-2006 [2] compliant radios, which offers 16 frequency channels in the 2.4GHz ISM band. A central scheduler is used to compute a schedule, which is then injected and used in the network.

TSMP, which combines time synchronization and frequency agility, has been shown to achieve reliabilities of over 99.999% [10]. Its core idea has been standardized for industrial applications in WirelessHART [11–13] and ISA100.11a [14]. In 2009, it has been introduced in the draft standard IEEE802.15.4e under the name Time Synchronized Channel Hopping (TSCH). This draft standard will replace the current IEEE802.15.4-2006 standard in its next revision.

All of the above standards rely on a central controller to compute a schedule for the network to use. The goal of this paper is to propose a distributed alternative, targeted at mobile nodes.

3 Distributed Scheduling Algorithms

3.1 Goal and Metrics

The goal of the proposed schedule is to achieve full connectivity, which is achieved when each node of the neighbor has established a bidirectional link to each of its physical neighbors. A bidirectional link is established between nodes A and B when, in the superframe, there is at least one slot scheduled from A to B, and one from B to A. The unreliability of the wireless link and the movement of the nodes are challenges the scheduling algorithm needs to cope with.

If a link is present in the physical graph, it is *feasible*; if a link is present in the physical but not in the logical graph, it is said to be *unscheduled*; a link which still appears in the logical while is disappeared from the physical graph is called *stale*. We use the ratio between the scheduled and feasible links as a metric for the static goodness of the scheduling algorithm.

Node mobility causes links to come and go. A link therefore has a finite lifetime, or *link duration*. To take advantage of a link, the scheduling algorithm needs to establish a logical link as soon as the physical link appears, and unscheduled it as soon as it disappears from the physical graph. We quantify the dynamic goodness of the scheduling algorithm by comparing the link duration between the physical and logical graphs.

Results presented in Section 4 are normalized against the optimal case, i.e. the physical connectivity graph. The variables to be used in this paper are listed in Table 1.

c	a channel
s	a slot number
L	number of slots in a superframe
$\mathcal{S} = \{\mathcal{S}0, \mathcal{S}_2, \ldots, \mathcal{S}_{L-1}\}$	state for each slot
$\mathcal{C} = \{\mathcal{C}_0, \mathcal{C}_2, \ldots, \mathcal{C}_{L-1}\}$	data frequency channel for each slot
$\mathcal{N} = \{\mathcal{N}_0, \mathcal{N}_2, \ldots, \mathcal{N}_{L-1}\}$	neighbor for each slot (can be NULL)
$\mathbb{P} = \{(r, c)_1, \ldots\}$	list of potential neighbors (contains identifier and channel)
$\{\mathbb{C}\}$	list of neighbors self has a connection to

Table 1. Variables used in this paper.

To be able to communicate, two nodes need to schedule a slot to one another. They hence need to communicate to agree which slot in the superframe to use, and which channel. We present two variants of the proposed scheduling mechanism. Aloha-based scheduling (Section 3.2) is a simple, canonical algorithm, in

which neighbor nodes opportunistically discover each other and establish links. Reservation-based scheduling (Section 3.3) builds upon that. By adding an explicit reservation channel, nodes discover each other faster, which is desirable in the presence of mobile nodes.

3.2 Aloha-Based Scheduling

Each of the L slots in the superframe is attached a state $\mathcal{S}$, a channel $\mathcal{C}$, and a neighbor $\mathcal{N}$. There are five states: "*Aloha*", "*Transmit Connection Request*", "*Receive Connection Request*", "*Transmit Data*", "*Receive Data*". A slot is assigned a channel $\mathcal{C}$ and a neighbor $\mathcal{N}$ only in the latter four states.

The "*Aloha*" state is the default. When establishing a unidirectional link from A to B, the scheduling algorithm causes a slot in A's schedule to transition from "*Aloha*" to "*Transmit Connection Request*", to "*Transmit Data*". Similarly, the same slot in B's schedule transitions from "*Aloha*" to "*Receive Connection Request*", to "*Receive Data*". When both A and B's slots are in the "*Transmit Data*" and "*Receive Data*" state, resp., data packets can be transmitted from A to B, once per superframe if exactly one slot is scheduled in the superframe. If A has no data to send, it sends an empty keep-alive message. While communicating, A monitors whether its data packets are acknowledged; B monitors whether it receives data at all. If for 5 consecutive superframes no data is successfully transmitted, the slot returns to the "*Aloha*" state; connection is then lost.

Three types of packets move through the network:

- **Advertisement** packets contain a list of "*Receive Connection Request*" slots of the sender node. This can be unseed by neighbors to know where it can be reached to establish a link. Each entry is a tuple (s, c) of slot and associated channel. Advertisements are broadcast and always exchanged on channel 0;
- **Connection Request** packets are sent in response to Advertisements; they are unicast on one of the slots announced in the Advertisement (at the announced frequency channel, always different from channel 0);
- **Data** packets flow over the slot when a link is established. Their content is determined by the application, but their successful transmission is monitored by the scheduling algorithm to detect stale links. An empty data packet is used as a keep-alive. Data packets are always sent on a frequency channel different from channel 0.

Note that there are L slots in a superframe, each of which can be used for an independent link. That is, a independent state machine is running for each slot. IEEE802.15.4-2006 compliant radios can transmit on 16 independent frequency channels. We dedicate channel 0 exclusively to Advertisements, and channels 1-15 exclusively to Connection Requests and Data packets.

Algorithm 1 presents Aloha-based scheduling in pseudo-code. It is executed by every node in the network. Upon startup (lines 1-4), all the slots are set to the "*Aloha*" state. The main loop (lines 5-49) iterates at each slot; different actions are taken according to the state of the slot. When in an "*Aloha*" slot, a node

listens for Advertisements 90% of the time (on channel 0, lines 15-26), 10% of the time it transmits an Advertisement (lines 8-15).

Sending Advertisement packets. The idea of sending an Advertisement is for a node to announce different *rendez-vous* slot/channel tuples so that interested nodes can establish a link to it. When sending an Advertisement, a node converts all of its "*Aloha*" slots to the "*Receive Connection Request*" state, and assigns each of those a random channel other than channel 0 (lines 10-13). It puts that list in an Advertisement which it sends on channel 0. It then waits to be contacted on one of the "*Receive Connection Request*" slots it just announced.

Receiving Connection Request packets. When reaching a slot in the "*Receive Connection Request*" state (lines 27-36), a node listens to the channel it has previously randomly picked and announced in its Advertisement. If it does not receive anything (lines 33-34), it converts that slot back to "*Aloha*" state. If it does receive a Connection Request (lines 29-32), it converts that slot to "*Receive Data*" state and records the identifier of the requester.

Receiving Advertisement packets. When receiving an Advertisement (lines 15-26), a node learns about the presence of a neighbor and is given the opportunity to contact it. If it has no slot scheduled to that neighbor, it picks one of the slots announced in the Advertisement where itself is in the "*Aloha*" state, i.e. it picks a *rendez-vous* slot and channel. If case there are multiple slots which satisfy these requirements, it picks one of them randomly. It changes the state of that slot in its scheulde to "*Transmit Connection Request*" (line 21), records the channel announced in the Advertisement (line 22), and the sender of that packet (line 23).

Transmitting Connection Request packets. When reaching a slot in the "*Transmit Connection Request*" state (lines 36-42) a node sends a Connection Request to the neighbor recorded in $\mathcal{N}$, at the channel recorded in $\mathcal{C}$ (line 37). If it receives an acknowledgment, it puts that slot in the "*Transmit Data*" state, and the logical link is established. If the Connection Request is not established (due to e.g. a collision or nodes moving apart), the slot is reset to the "*Aloha*" state.

3.3 Reservation-Based Scheduling

The Reservation-Based Scheduling protocol behaves like the Aloha-based protocol, with the following additions:

- Slot 0 is a permanent rendez-vous slot, i.e. only Advertisements can be exchanged. Unlike other slots, Advertisements can be exchanged on any of the 16 available channels, in slot 0. Each node picks a channel on which it listens for Advertisements. Using slot 0 as a reservation slot gives nodes more opportunities to establish links to one another.
- In their Advertisements, nodes also include the list of the neighbors they are connected to, and the channel those neighbors are listening on in slot 0. This means that nodes learn about their two-hop neighbors.

– Each node maintains a list $\mathbb{P}$ of potential neighbors and the channel they are listening on in slot 0. This information is obtained by listening to Advertisements. Each node also maintains a list $\mathbb{C}$ of neighbors it is currently connected to. The scheduling algorithm tries to get as many nodes from $\mathbb{P}$ to $\mathbb{C}$.
– A node only announces the *even* slots in its Advertisement. When the state of even slot i becomes "Transmit data" (resp. "Receive data"), the state of odd slot $i+1$ is implicitly changed to "Receive data" (resp. "Transmit data"). This means that links are scheduled in pairs, one in each direction, establishing only bidirectional links.

Algorithm 2 presents Reservation-based scheduling in pseudo-code. The part of p. 10 details the execution in slot 0, while the part on p. 11 focuses on the other slots.

4 Simulation Setup and Results

We use a Python-based simulator[1] to model the mobility and RF propagation characteristics for a fleet of 25 mobile nodes. The superframe size was chosen to be 17 slots. The size must be co-prime with 16 in order to gain the benefits of the channel offset scheme; a relatively small superframe size was chosen to ensure that the scheduling constraints would be significant.

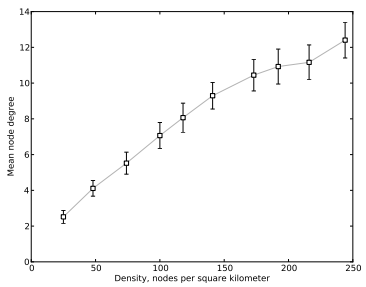

Fig. 2. Mean node degree vs density of nodes in simulated environment

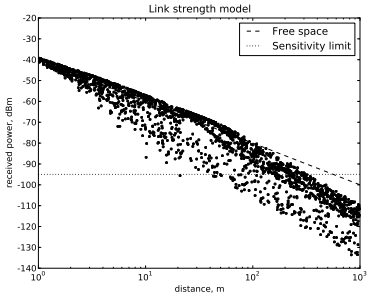

Fig. 3. Received power from randomly chosen locations in simulated environment

4.1 Propagation Model

The design objective for the RF propagation model is to create a deterministic model which captures the variance of the distance-to-received-power relationship observed in empirical studies of static spatial configurations, while also providing plausible spatial correlation of link strength. Approximately 30% of the simulated environment is covered with obstacles. The radiated power from a

[1] As an on-line addition to this paper, the source code of the simulator is made freely available at `http://lagrange.ce.berkeley.edu/`.

Algorithm 1 Aloha-Based Scheduling *(executed by each node).*

for each slot $i \in L$ **do**
  $\mathcal{S}_i$ ="Aloha"
  $\mathcal{N}_i$ =NULL **and** $\mathcal{C}_i$ =NULL
end for
loop
  Go to the next slot i
  if $\mathcal{S}_i$ =="Aloha" **then**
    if uniform(0,1)< 0.1 **then**
      Enumerate all other slots with state $\mathcal{S}$ =="Aloha", $\{s_j\}$
      for each of these slots **do**
        $\mathcal{S}$="Receive Connection Request"
        $\mathcal{C}$ =uniform(1,15)
      end for
      Send Advertisement listing these slots and their channels $\{(s_j, c_j)\}$, on channel 0
    else
      Listen for an Advertisement on channel 0
      if Advertisement $\{(s_j, c_j)\}$ received **then**
        Find own set of slots $\{s_k\}$ which are of state $\mathcal{S}_i$ =="Aloha"
        if $\{s_k\} \cap \{s_j\} \neq \emptyset$ **then**
          Choose common slot $s_n \in \{s_k\} \cap \{s_j\}$ randomly
          $\mathcal{S}_n$="Transmit Connection Request"
          $\mathcal{C}_n$ set to the receiving channel for that slot, read from Advertisement
          $\mathcal{N}_n$ set to the node that sent the Advertisement
        end if
      end if
    end if
  else if $\mathcal{S}_i$ =="Receive Connection Request" **then**
    Listen for a Connection Request to self on channel $\mathcal{C}_i$
    if valid Connection Request received **then**
      Send Acknowledgment
      $\mathcal{S}_i$ ="Receive Data"
      $\mathcal{N}_i$ set to the ID of the requesting node
    else
      $\mathcal{S}_i$ ="Aloha"
    end if
  else if $\mathcal{S}_i$ =="Transmit Connection Request" **then**
    Send Connection Request on channel t_i to node c_i
    if Acknowledgment received **then**
      $\mathcal{S}_i$ ="Transmit Data"
    else
      $\mathcal{S}_i$ ="Aloha"
    end if
  else if $\mathcal{S}_i$ =="Receive Data" **or** $\mathcal{S}_i$ =="Transmit Data" **then**
    if no successful communication for n consecutive superframes **then**
      $\mathcal{S}_i$ ="Aloha"
      $\mathcal{N}_i$ =NULL
    end if
  end if
end loop

Algorithm 2 Reservation-Based Scheduling *(for slot 0).*

```
1: for each slot i do
2:    S_i ="Aloha"
3:    N_i =NULL
4: end for
5: P = ∅
6: C = ∅
7:
8: Go to the next slot i
9: if this is slot i == 0 then
10:   if P ≠ ∅ and uniform(0,1)< 0.1 then
11:      Choose (id, c) (identifier, channel) randomly from neighbors of interest in P
12:      Transmit Advertisement to node id on channel c
13:      if Acknowledgment received then
14:         set state of all advertised slots to S ="Receive Connection Request"
15:      end if
16:   else
17:      Listen for an Advertisement on channel C_0
18:      if Advertisement received then
19:         Send Acknowledgment
20:         If neighbor of interest, choose common slot n (similar to Algorithm 1)
21:         S_n ="Transmit Connection Request"
22:         N_n set to the ID of the node that sent the Advertisement
23:         C_n set to the receiving channel for that slot in the Advertisement
24:      end if
25:   end if
26: end if
27:
28: Continues on p. 11
```

transmitting antenna is attenuated by an inverse square law as it moves through "obstacle-free" space, but is attenuated by an inverse fourth power law as it moves through "obstacle" space. This "higher power attenuation" scheme is inspired by empirical models of the effect of foliage on line-of-sight transmission [15]. The foliage model and density of obstacles is intended to represent an outdoor estuarial environment similar to that encountered by the Floating Sensor Network project. The multipath effect of the signal reflecting off the ground is modeled. The reflection is assumed to result in a 180° phase change and no attenuation.

The size of the simulated enviroment is modified as needed to yield desired node densities. The minimum and maximum densities are 25 and 250 nodes per square kilometer. Figure 2 shows the mean node degree (number of neighbors in the physical connectivity graph) for the different simulated densities. The bars represent the 95% certainty interval for the estimate of the mean.

Algorithm 2 *[cont.]* Reservation-Based Scheduling *(for slots other than 0).*

1: *Continues from p. 10*
2:
3: **if** this is slot $i! = 0$ **then**
4: **if** $\mathcal{S}_i$ ==“Aloha” **then**
5: **if** uniform(0,1)< 0.1 **then**
6: Enumerate all other *even* slots with state $\mathcal{S}$ ==“Aloha”, $\{s_j\}$
7: **for** each of these slots **do**
8: $\mathcal{S}$=“Receive Connection Request”
9: $\mathcal{C}$ =uniform(1,15)
10: **end for**
11: Send Advertisement listing these slots channels $\{(s_j, c_j)\}$, on channel 0; also include all (id, c) tuples present in $\mathbb{C}$
12: **else**
13: Listen for an Advertisement on channel 0
14: **if** Advertisement $\{(s_j, c_j)\}$ received **then**
15: Add new possible neighbors to $\mathbb{P}$ using the information in the Advertisement
16: Find own set of slots $\{s_k\}$ with state $\mathcal{S}$ ==“Aloha”
17: **if** $\{s_k\} \cap \{s_j\} \neq \emptyset$ **then**
18: Choose common slot $s_n \in \{s_k\} \cap \{s_j\}$ randomly
19: $\mathcal{S}_n$ =“Transmit Connection Request”
20: $\mathcal{N}_n$ set to the ID of the node that sent the Advertisement
21: $\mathcal{C}_n$ set to the receiving channel for that slot in the Advertisement
22: **end if**
23: **end if**
24: **end if**
25: **else if** $\mathcal{S}_i$ ==“Receive Connection Request” **then**
26: Listen for a Connection Request for self on channel $\mathcal{C}_0$
27: **if** valid Connection Request received **then**
28: Send Acknowledgment
29: $\mathcal{S}_i$ =“Receive Data”; $\mathcal{S}_{i+1}$ =“Transmit Data”
30: $\mathcal{N}_i$ and $\mathcal{N}_{i+1}$ set to the ID of the requesting node
31: **else**
32: $\mathcal{S}_i$ =“Aloha”
33: **end if**
34: **else if** $\mathcal{S}_i$ ==“Transmit Connection Request” **then**
35: Send Connection Request on channel $\mathcal{C}_i$ to node $\mathcal{N}_i$
36: **if** Acknowledgment received **then**
37: $\mathcal{S}_i$ =“Transmit Data”; $\mathcal{S}_{i+1}$ =“Receive Data”
38: $\mathcal{N}_{i+1}$ set to the ID of the requesting node
39: $\mathcal{N}_i$ = to $\mathbb{C}$ and remove it from $\mathbb{P}$
40: **else**
41: set slot i to “Aloha”
42: **end if**
43: **else if** $\mathcal{S}_i$ ==“Receive Data” **or** $\mathcal{S}_i$ ==“Transmit Data” **then**
44: **if** no successful communication for n consecutive superframes **then**
45: $\mathcal{S}_i$ =“Aloha”
46: move $\mathcal{N}_i$ from $\mathbb{C}$ to $\mathbb{P}$
47: $\mathcal{N}_i$ =NULL
48: **end if**
49: **end if**
50: **end if**

4.2 Co-channel interference model

The interfering effects of two nodes transmitting on the same channel at the same time (usually called a "collision") is one of the main constraints on the decentralized schedule.

The IEEE802.15.4-2006 standard specifies required jamming resistance for interference coming from an adjacent channel (1 channel away) or an alternate channel (2 channels away), but does not specify a required resistance to interference on the same channel. The Texas Instruments CC2420 2.4 GHz IEEE802.15.4-2006 compliant transceiver [16] has a specified co-channel rejection of -3 dB; in other words, if node A receives a transmission from node B with p dBm power, and a simultaneous transmission from node C with $(p-3)$ dBm power, the transmission for B will be received correctly and the transmission from C rejected. We use this model for our simulation. Adjacent and alternate channel interference are not modeled in this simulation.

4.3 Node Mobility Model

Each node is modeled as a mobile device moving at a constant speed in the environment described above. The speed of each node is drawn from a uniform distribution over [0.8,1.2] m/s. Each node transmits at 1 mW using an isotropic antenna. The height of the antenna from the ground (used for the multipath calculations) is drawn from a uniform distribution over [0.7,1.3] m for each node. Node motion is controlled by a random waypoint procedure: nodes select a cardinal direction randomly, then a distance to move in that direction. When they reach their destination, they repeat the selection process. The nodes are confined to a square area with dimensions determined by the desired node density.

Fig. 3 shows the received power for randomly located transmitter and receiver nodes in the simulated environment.

4.4 Static Metric: Relative Connectivity

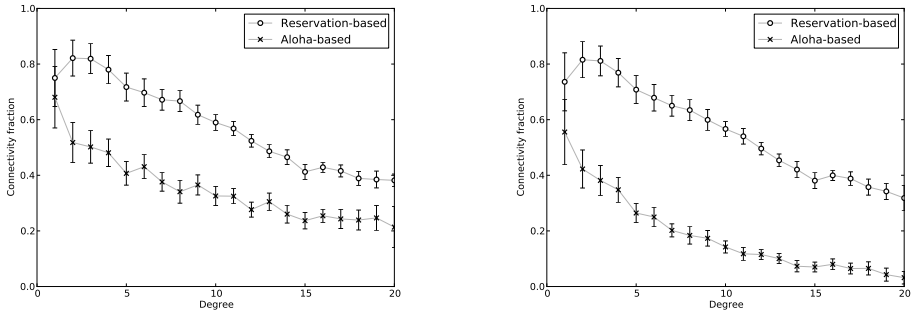

Fig. 4. Unidirectional (left) and bidirectional (right) mean connectivity ratio vs degree

The static connectivity test proceeds as follows:

1. Simulate 25 mobile nodes for 60 seconds;
2. pick a node and a superframe at random;
3. from the physical connectivity graph, count the number of unique edges incident to that node over the superframe (that is, the number of one-hop neighbors connected for at least 1 slot during the superframe); this is the degree of the node;
4. from the logical connectivity graph, find the number of outbound edges (for the unidirectional test), or find the number of neighbors with both an outbound and inbound edge (the bidirectional test);
5. the ratio of the logical connection count to the node degree is the *connectivity ratio* for the node.

Fig. 4 shows the mean connectivity ratio vs. the node degree for 1250 simulations, for both unidirectional and bidirectional connections. The bars represent the 95% confidence interval in the estimate of the mean.

The reservation-based algorithm outperforms the Aloha-based algorithm at almost all node degrees (the confidence intervals overlap for degree 1). The reservation-based algorithm has more resources allocated to neighbor discovery, and a successful advertisement/connection request exchange results in a bidirectional connection. For both algorithms, increased node degree results in decreased relative connectivity ratios. More local nodes means more collisions between Aloha advertisements, which reduces the effectiveness of neighbor discovery, and more cases of multiple nodes responding to an advertisement, resulting in collisions and lost connectivity. The superframes also fill up when more neighbors are present; since the superframe size is 17 slots, a node cannot have bidirectional links with more than 8 neighbors. The difference between the Aloha-based and reservation-based algorithm performance at high node degrees, however, demonstrates that both collisions and saturation must be significant.

4.5 Dynamic Metric: Link Durations

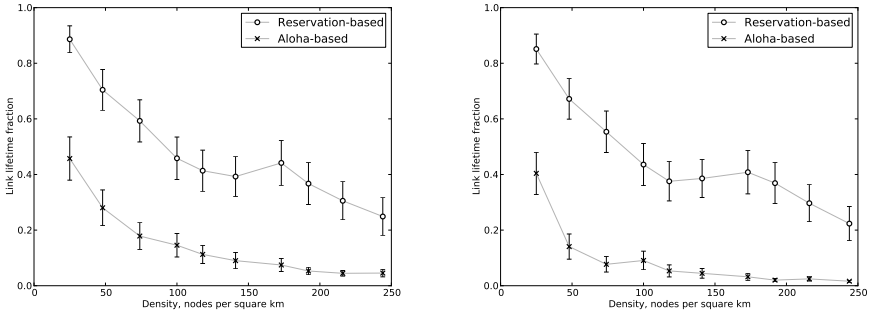

Fig. 5. Unidirectional (left) and bidirectional (right) mean link lifetime ratio vs density

The dynamic link duration test proceeds as follows:

1. Simulate 25 nodes for 60 seconds;

2. pick a node and a superframe at random;
3. pick one of the edges on the physical connectivity graph incident to that node at random; this is the link we will test;
4. count the number of consecutive superframes (forward and backward in time) in which this link is in the physical connectivity graph; this is the *physical link duration*;
5. starting from the beginning of the link's lifetime, find the first superframe in which the link exists in the logical connectivity graph, either as a unidirectional link (the original node to the destination) or as a bidirectional link;
6. count the number of consecutive superframes until the link no longer exists in the logical connectivity graph; this is the *logical link duration*;
7. the ratio of the logical link duration to the physical link duration is the *link lifetime ratio*.

Fig. 5 shows the mean link lifetime ratio vs. the density of the nodes in the simulated environment for 1250 simulations. The bars represent the 95% confidence interval for the estimate of the mean. The degree of the node is not well defined over many superframes, as the physical and logical connectivity change. While the static connectivity test could use the node degree as the independent variable, for the dynamic link duration test we use the node density as a surrogate. See Fig. 2 for the relationship between the mean node degree and node density.

The dynamic performance also shows that the reservation-based algorithm outperforms the Aloha-based algorithm. Again, the Aloha-based algorithm is at a disadvantage, because its advertisement/connection request transactions build unidirectional links, not bidirectional links. At low densities, the ratio between the algorithms' performances is roughly 2, which suggests that the unidirectional/bidirectional allocation difference dominates in this regime. But at higher densities, the difference between the two algorithms widens, which means other effects must be significant as well.

The saturation effects at work in the connectivity tests are also significant in the dynamic case. Links can be broken by co-channel interference, if another pair of nodes begins transmitting at the same channel/slot as an existing link. Nodes that have many active links also have less vacant slots available to form new links. Saturation effects alone cannot explain the decreased performance at high density, however, since the Aloha-based algorithm's performance decreases significantly more than the reservation-based algorithm's performance.

The reservation-based algorithm benefits when advertisements are exchanged frequently, because information about connected neighbors is carried by the advertisement packets. The reactivity of the reservation-based algorithm therefore increases at higher densities, as nodes learn about possible new neighbors more quickly. Because the advertisements in the reservation-based algorithm carry more information than the advertisements in the Aloha-based protocol, the reservation-based algorithm gains relative performance at higher node densities.

5 Conclusions and Future Work

In this paper, we present what is, to our knowledge, the first scheduling algorithm for Time Synchronized Channel Hopping networks which both is distributed and which copes with mobile networks. The two variant algorithms are based on an advertisement and *rendez-vous* scheme: nodes continuously advertise their presence which allows neighbor nodes to discover and contact one another. A inactivity threshold mechanism is used to tear down previously established links.

The algorithms are tuned for a network of 25 drifter nodes randomly moving inside a lake or river. Simulation results show, under realistic propagation and mobility models, the efficiency of the algorithms. Figs. 4 and 5 demonstrate that the reservation-based algorithm outperforms the Aloha-based algorithm in practically all density conditions. This suggests that in an environment with highly dynamic connectivity, including networks of mobile nodes, devoting additional resources to neighbor discovery and coordination pays off.

The goal of the scheduling algorithms presented in this article is to establish two-way connections between neighbor nodes, subject to the constraints of the superframe structure and the physical connectivity. We did not make assumptions about what kind of data is sent over the links; the latency, throughput, and reliability requirements are not specified. These scheduling algorithms could be adapted to meet either pre-determined or dynamic provisioning requirements. For example, a pair of nodes that need to exchange a large amount of data might wish to schedule more than one transmission slot per superframe.

Many wireless sensor network applications are highly energy-constrained. Our scheduling algorithms, as described here, require the radios to constantly either receive or transmit. This would consume too much power for many applications. An obvious modification is to reduce the duty cycle of the Aloha coordination activities; the algorithms could be implemented exactly as written, while only performing Aloha listen/transmit actions on a subset of the idle slots. The obvious tradeoff is between the energy consumed for Aloha coordination versus the reactivity of the network to changes in the physical connectivity graph. Further work could focus on characterizing the rate of change of the connectivity graph, and determining a method for balancing power consumption and reactivity.

References

1. A. Tinka, I. Strub, Q. Wu, and A. M. Bayen, "Quadratic Programming based data assimilation with passive drifting sensors for shallow water flows," *International Journal of Control*, 2010, to appear.
2. *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Std., Rev. 2006, 8 September 2006.

3. *IEEE P802.15.4e/D0.01 Draft Standard for Information technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) Amendment 1: Add MAC enhancements for industrial applications and CWPAN*, IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) Std. IEEE Std 802.15.4e, Rev. D0.01/r3, 13 September 2009.
4. J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in *Second ACM Conference on Embedded Networked Sensor Systems (SenSys)*. Baltimore, MD, USA: ACM Press, November 3-5 2004, pp. 95–107.
5. M. Buettner, E. Yee, Gary V.and Anderson, and R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks," in *4th international conference on Embedded networked sensor systems (SenSys)*. Boulder, Colorado, USA: ACM, 31 October - 3 November 2006.
6. W. Ye, F. Silva, and J. Heidemann, "Ultra-Low Duty Cycle MAC with Scheduled Channel Polling," in *4th ACM Conference on Embedded Networked Sensor Systems (SenSys)*. Boulder, Colorado, USA: ACM, November 1-3 2006, pp. 321–334.
7. T. Watteyne, A. Mehta, and K. Pister, "Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense," in *6th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, Tenerife, Canary Islands, Spain, 26-30 October 2009.
8. T. Watteyne, S. Lanzisera, A. Mehta, and K. Pister, "Mitigating Multipath Fading Through Channel Hopping in Wireless Sensor Networks," in *IEEE International Conference on Communications (ICC)*. Cape Town, South Africa: IEEE, 23-27 May 2010.
9. K. Pister and L. Doherty, "TSMP: Time Synchronized Mesh Protocol," in *Parallel and Distributed Computing and Systems (PDCS)*, Orlando, Florida, USA, 16-18 November 2008.
10. L. Doherty, W. Lindsay, and J. Simon, "Channel-Specific Wireless Sensor Network Path Data," in *16th International Conference on Computer Communications and Networks (ICCCN)*. Turtle Bay Resort, Honolulu, Hawaii, USA: IEEE, August 13-16 2007, pp. 89–94.
11. *HART Field Communication Protocol Specifications, Revision 7.1, DDL Specifications*, HART Communication Foundation Std., 2008.
12. D. Gustafsson, "WirelessHART - implementation and evaluation on wireless sensors," Master's thesis, Kungliga Tekniska högskolan, April 2009.
13. J. Song, S. Han, A. K. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt, "WirelessHART: Applying wireless technology in real-time industrial process control," in *IEEE Real-Time and Embedded Technology and Applications Symposium*, 2008, pp. 377–386.
14. ISA, *ISA-100.11a-2009: Wireless Systems for Industrial Automation: Process Control and Related Applications*, International Society of Automation Std., 11 September 2009.
15. C. Oestges, B. Montenegro Vollacieros, and D. Vanhoenacker-Janvier, "Radio Channel Characterization for Moderate Antenna Heights in Forest Areas," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 4031–4035, October 2009.
16. CC2420, *2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver (Rev. B)*, Texas Instruments, Inc., 20 March 2007, data Sheet SWRS041B [available online].