Joint Synchronization, Routing and Energy Saving in CSMA/CA Multi-Hop Hybrid Networks EPFL Technical Report: IC/2004/35

Dan Jurca and Jean-Pierre Hubaux
Laboratory of Computer Communications and Applications (LCA)
Swiss Federal Institute of Technology (EPFL)
Lausanne, Switzerland
{dan.jurca, jean-pierre.hubaux}@epfl.ch

Abstract

Multi-hop hybrid networks can help providing both high bandwidth and broad coverage for wireless data networks. We focus on CSMA/CA-based networks and take IEEE 802.11 as a concrete example. We show that the three fundamental operations of synchronization, routing and energy saving can be implemented in an integrated way.

Our integrated solution is based on the periodic computation of a connectivity tree among the nodes reporting to the same Access Point, starting from the Access Point itself. We use the nodes that are tree vertices as relays for both data and control packets. We propose a distributed neighbor discovery protocol and a simple centralized algorithm for computing the connectivity tree. Our analysis and simulation results show that the proposed solution has low protocol overhead in terms of message passing and execution time, and performs well even if nodes are mobile.

Key Words: connectivity tree, relaying nodes, synchronization, routing, energy saving

1 Introduction

Wireless access to the Internet is currently provided by two families of networks: cellular networks (e.g., GSM/GPRS), operating on licensed frequencies, and offering good geographic coverage but limited bitrate; and Wi-Fi networks (typically based on IEEE 802.11), operating on unlicensed frequencies and offering high bitrate but very limited coverage.

The research community is devoting a lot of effort to devise networks that would provide the best of both worlds, namely high bitrate with broad coverage; an important goal of this research is to avoid having to deploy too many additional fixed antennas. Many ongoing projects are focusing on smart, directional antennas, by which mobile nodes will be able to assess the availability of the spectrum in their neighborhood, and identify in real time the best way to transmit information; in most cases, this information will reach an access point, possibly in a multi-hop way over several other mobile nodes. This vision raises a number of technical challenges, and will not be accomplished in the near future.

In this paper, we take a more pragmatic, short term approach. We show that the concept of multi-hopping can already be implemented with existing, CSMA/CA-based networks; in order to be very concrete, we focus our proposal on IEEE 802.11. More specifically, we show that the three fundamental mechanisms of synchronization, routing, and energy saving can be implemented in an integrated way in a Multi-hop Hybrid Network (MHN), by leveraging on the superior characteristics of the Access Point (AP). We will call a *control area* (CA)[1] the geographic area which is under the responsibility of a given AP.

Inside a given control area, the wireless nodes can reach the responsible AP through multiple hops with the help of other nodes (Figure 1). The advantages of multi-hop communication include a reduced energy consumption of the mobile nodes, a lower interference and an increased coverage [26, 16].

Synchronization of the nodes at the MAC layer is usually needed by the underlying physical data transmissions or by power saving mechanisms [2]. In a multi-hop environment, nodes must be aware of existing routes in order to transmit their packets towards the destination. The limited energy of wireless nodes imposes the implementation of efficient energy saving strategies. Both the synchronization mechanism (in order to keep the clock drift below a certain threshold), and the routing solution in the presence of mobile nodes (in order to maintain the freshness of the routes), need periodic updating.

Our integrated solution is based on the periodic computation of the connectivity tree[2] inside the control area. We call *subset* S the set of nodes that are tree vertices at a given time. These nodes are *relaying nodes* until a new tree is computed; they forward all data packets and control messages (e.g., synchronization signals, referred to as *beacons*).

We make use of a distributed neighbor discovery protocol (performed at all nodes in the control area) and of two centralized algorithms (performed at the AP), to periodically compute the connectivity tree of the control area. The solution runs on the common wireless channel governed by the CSMA/CA principle. The periodicity in the construction of the subset S helps our solution cope with nodes' mobility and with topology changes.

Nodes synchronize with the help of periodic beacons initiated by the AP and relayed by the subset S. In the intervals between beacons, nodes can go to sleep in order to save energy. Data packets are relayed hop-by-hop to/from the AP, using the forwarding services of subset S.

We assess the performance of our solution by means of an analysis and of extensive simulations.

[1] We refrain from using the term "cell" because (i) the purpose of the described solution is not necessarily to provide full coverage and (ii) the power range of the AP is in general not large enough to reach a given node of the area in a single hop.

[2] The connectivity tree is the tree that connects all nodes inside the CA with the AP (the root of the tree).

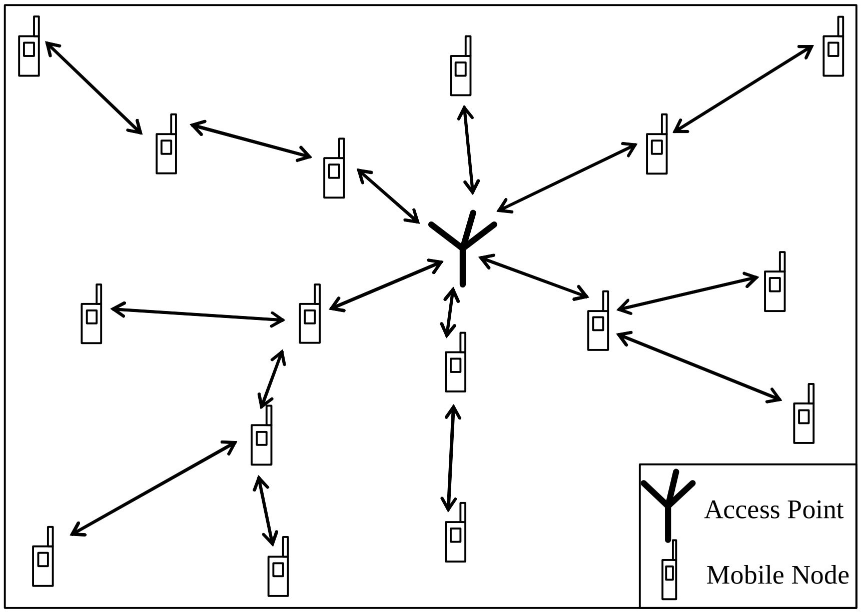

Figure 1: Control area in a Multi-hop Hybrid Network

The rest of the paper is organized as follows. Section 2 introduces our system model. We present our solution in Section 3 and provide an analysis of our protocol and algorithms in Section 4. In Section 5 we present and discuss the simulation results. We compare our solution with the related work in Section 6. Section 7 concludes the paper.

2 System Model

This section presents our assumptions and requirements on the MHN architecture.

Our goal is to show how the three fundamental mechanisms of synchronization, routing and energy saving can be jointly implemented, by leveraging on the superior resources of the APs.

In order to minimize the hardware cost, we assume that all traffic inside a MHN control area (including all control packets such as beacons) goes over the same radio channel regulated by the CSMA/CA principle. Nodes contend for the channel using contention windows and back-off mechanisms. The way our control messages are sent over the channel is explained in details in Section 3.

The MHN is divided into control areas, corresponding to the APs on top of the ad-hoc network. The APs are interconnected with wired or directed wireless, high bandwidth links. Inside the MHN control area, the power range of the AP and all wireless nodes, is smaller than the radius of the CA, inducing a multi-hop environment. Furthermore, we assume that the energy and the computational power are much larger at the AP than at the wireless nodes (APs are wire-powered). The algorithms used to compute the connectivity tree of the CA and the subset S are performed at the AP.

We assume that all traffic in the control area is directed to/from the AP. We assume a small number of hops between the AP and the most remote wireless node in the CA (a maximum of four seems to be a reasonable number). All nodes in the CA, including the AP, transmit with the same fixed power and all links in the CA are symmetric and bidirectional. All nodes cooperate inside the CA and there are no malicious or misbehaving nodes. We assume that all wireless nodes run on individual batteries with limited power. In order to join a CA, a node must run a membership request protocol. Each member node can be identified with the help of a two byte address (allocated by the AP), unique inside the CA.

Inside the MHN control area, our model treats collisions in the same way as the IEEE 802.11 standard. If a node receives two packets at the same time from two nodes situated within its power range, we assume a collision (and the loss of both packets). Collisions may also occur, depending on the interference range and on the value of the signal to noise and interference ratio ($SNIR$). The interference due to nodes belonging to adjacent control areas is partially addressed in Section 5.

We assume that each node contains local clocks with parameters similar to the ones defined in the IEEE 802.11 standard [2].

3 Proposed Solution

This section presents the details of our solution for synchronization, routing and energy saving in CSMA/CA MHNs. Our solution uses broadcasting inside the MHN control area to achieve its goals locally. Simulations show that our solution performs very well even if nodes are mobile.

3.1 Rationale of the solution

Our solution runs locally and independently in each MHN control area. It realizes the synchronization of all wireless nodes in the same CA, while providing routes to/from the corresponding AP. It also enables the nodes to remain synchronized with the AP.

The AP periodically constructs the connectivity tree of its MHN control area. In each CA, all nodes that are members of the tree at a given time form the subset S. The AP inserts the identities of these nodes into the beacon body. Due to the nodes' small size identities (two bytes), the variable size of subset S has a marginal impact on the size variations of the beacons. When receiving a beacon, each wireless node checks if its identity is present in the beacon body. If so, the node is a relaying node and rebroadcasts the beacon. If not, the node just retrieves the time information in the time-stamp and adjusts its internal clock.

Our solution only uses beacons to disseminate local topology information (subset S), inside the control area. While the periodicity at which our protocols must be run depends on the pace of topology changes, we require the periodicity of the beacon transmission (T_{beacon}) to be smaller or equal to the periodicity of our solution. Otherwise, the overhead induced by the unused iterations of our solution is not justified.

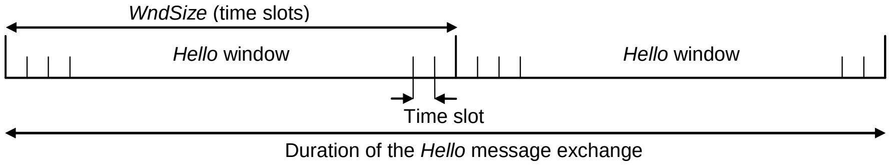

Figure 2: *Hello* windows used for back-off in the NDP

Because of the nodes limited energy, we propose a solution that implies low protocol overhead in terms of message passing and execution time. Furthermore, we limit the computational overhead at the nodes by proposing a centralized algorithm performed periodically by the APs.

The solution is adapted to the network's specifics. It comprises distributed parts performed at each wireless node (like the Neighbor Discovery Protocol), and centralized parts performed at the AP (building the connectivity tree). These parts are explained in detail later in this section. Each node periodically gathers information about its present neighbors. This topology information is then forwarded to the AP, using an efficient scheduling mechanism in order to avoid collisions. The AP aggregates the messages and, based on the neighboring information, constructs the connectivity tree in the CA and assigns the subset S. The nodes in subset S are then used to forward control messages and data packets until the next iteration of the protocols.

3.2 Neighbor Discovery Protocol (NDP)

The NDP is performed periodically at each member node (including the AP). The time between two consecutive iterations of the protocol (T_{NDP}) depends on the mobility assumptions (typical values being between half a second and a few seconds [21]).

All nodes in the control area start performing the NDP protocol at the same time[3]. This approach offers the possibility to reserve a special time window for performing NDP in each CA (while all other traffic is buffered and delayed). It leads to a good performance of our solution while imposing a hard bound on its execution time (Section 4).

The purpose of the protocol is to provide the AP with fresh topology information. Based on the received information, the AP performs the algorithm presented in Section 3.3 in order to compute the connectivity tree.

When running NDP, each node i (member of the CA), constructs a data structure containing the received information about its neighboring nodes (identity, member of subset S or not, number of hops to the AP). From now on we refer to this data structure as the *Neighbor Table* of node i or *NT(i)*, and to the information received about a neighboring node j as the *status* of node j in $NT(i)$.

Each Neighbor Table is sent to the AP using the relaying nodes in subset S assigned in the previous iteration of our solution. The AP uses the received Neighbor Tables to

[3] This is easily achievable if the network CA is already initialized and all member nodes have been synchronized in a previous iteration of our solution. We explain how our solution works during the CA initialization phase in Section 3.4.

construct the connectivity tree and to assign a new subset S that reflects the recent changes in the topology of the CA (Section 3.3).

During one iteration of NDP, each node constructs its Neighbor Table with the information received from messages broadcasted by neighboring nodes. These are short messages (referred to as *Hello* messages) containing the present status of the sender. Nodes receiving a Hello message update their Neighbor Table with the status extracted from the message. Since all Hello messages use the same CSMA/CA channel and NDP is performed at the same time at all nodes, it is important to efficiently avoid collisions.

During each iteration of the protocol, each node sends two Hello messages, during two separate *Hello* windows (Figure 2). Hello windows are contention windows divided in time slots. They are in synchrony across all nodes in one CA. Each window contains a number of time slots referred to as *WndSize*. The choice of WndSize influences the probabilities of message collision and the total execution time of the protocol (Section 4.1). Each node randomly chooses a time slot between 0 and $WndSize - 1$ for each window, according to a uniform distribution. When that time slot comes, the node senses the medium and, if idle, it broadcasts its Hello message. If the medium is busy at the beginning of the time slot, the node defers for one time slot and senses the channel again.

This technique leads to a very low probability of *significant* Hello message collisions. A significant collision is defined as a double collision: at a particular receiving node, both Hello messages from one neighbor are lost. In the case of a significant collision, the Neighbor Table of the receiving node will not contain one neighbor. However, by sending two Hello messages at each protocol iteration, the probability of a significant collision is much smaller than the probability of a simple message collision. Also, this probability can be controlled by network designers, by assigning the value of WndSize as a function of the maximum number of neighbors a node can have. A detailed computation of these probabilities can be found in Section 4.1.

Once the two Hello windows elapse, the nodes have to forward the fresh Neighbor Tables to the AP. The protocol uses the subset S assigned in the previous iteration (Section 3.3) to gather all Neighbor Tables and to relay them to the AP.

Each relaying node consequently polls all its neighbors (nodes present in its Neighbor Table) and gathers their Neighbor Tables. Nodes only send the differences between the fresh Neighbor Table and the previous version of it. When there are no differences, the nodes simply acknowledge the received poll. If a node is not polled, after a timeout, it attempts a direct Neighbor Table transmission to a neighboring relaying node. This relaying node is chosen from the node's fresh Neighbor Table. The relaying nodes aggregate all received Neighbor Tables (including their own) into one or more messages and forward them to the AP.

We use a **scheduling mechanism** for these transmissions in order to avoid collisions. The scheduling mechanism is based on the consistency of the information regarding subset S, received from the beacons. Each relaying node extracts subset S from the beacon body. The subset S contains the identities of the relaying nodes as they are introduced by the AP. During one iteration of NDP, the order of the nodes identities in subset S is maintained, since all relayed beacons inherit the same original beacon generated by the AP. Our scheduling mechanism imposes different delays (before message relaying towards the AP), at each relaying node, based on its position in subset S. Since subset S is extracted from the beacons body and is the same at all relaying nodes, each relaying node knows its position and the position of all other relaying nodes in the current subset S. This information is consistent with all relaying nodes that receive the beacons. Hence, each relaying node can delay its transmission with a different time, based on its known position in subset S and on the length of the message that needs to be relayed. The mechanism efficiently avoids the collisions of the messages that are relayed closer towards the AP. It requires no additional message exchange and can be performed individually at each relaying node in a distributed fashion.

As seen before, we use the subset S chosen in a previous iteration of our solution to forward the messages in the present iteration. However, the topology of the control area can change, due to the mobility of the nodes. Therefore, the periodicity of the protocol, (T_{NDP}), is strictly related to the assumptions of mobility of the nodes [11, 24, 21].

3.3 Connectivity Tree Computation Algorithm (CTC)

The purpose of the CTC algorithm is to compute the connectivity tree of the CA, based on the received Neighbor Tables. The AP performs the algorithm with the same periodicity as the NPD protocol, immediately after it receives all Neighbor Tables.

The result of the algorithm is the subset S that ensures full CA connectivity. The proposed algorithm meets two requirements:

1. All nodes in the CA must have at least one neighbor that is a member of subset S;

2. All nodes in the CA must be able to access a minimum route (in hops) to/from the AP through the nodes in subset S.

We want to inform all nodes in the CA, as soon as possible, about the freshly discovered topology. The AP disseminates the topology information (subset S) by means of beacons. We require the periodicity in beacon transmission (T_{beacon}) to be smaller than the periodicity of the NDP protocol (T_{NDP}) and we want the next scheduled beacon (after performing the NDP protocol) to already contain the newly discovered subset S. Hence, the AP has only little available time to perform the CTC algorithm. Assuming that the NDP protocol starts immediately after a beacon broadcast, the AP must compute the new subset S in the time between the end of NDP until the scheduled time of the next beacon.

An optimal solution for computing the connectivity tree is presented in [9] as the Geometric Connected Dominating Set Problem. The problem is NP-complete and the optimal solution cannot be computed under the time constraints presented above. Hence we present a simpler and suboptimal solution that computes the connectivity tree in feasible time inspired by [17]. We evaluate the average cardinality of subset S in Section 5.

We provide a formal description of the problem. Let *MaxNodes* be the maximum number of nodes allowed in the CA and *NrNodes* be the actual number of nodes in the considered CA. All nodes that are members in one CA (including the AP) form the set $\mathcal{N}$. We also define *MaxHop* as the maximum number of hops in the CA (from the AP to the periphery). Let *NT(i)* be the Neighbor Table of node i received at the AP with $1 \leq i \leq MaxNodes$. We say that node i, belonging to the CA, is *covered* if it is contained in the Neighbor Table of at least one of the relaying nodes. Otherwise, node i is *not covered*. We define $distance(i)$ as the distance between node i and the AP (in number of hops), using the existing subset S. We have $1 \leq distance(i) \leq MaxHop$ if node i is *covered*, $distance(i) = \infty$ if node i is *not covered*; also, $distance(AP) = 0$. Finally, we define $potential(i)$ as the potential increase in the number of *covered* nodes in the CA if node i becomes a relaying node (e.g. the number of nodes *not covered* present in $NT(i)$). Initially subset S=∅.

Assuming that each node has at least one neighbor in its transmission range, the algorithm has to construct the subset S such that node i is *covered* and $distance(i)$ is minimum $\forall i, 1 \leq i \leq NrNodes$.

Algorithm 1 Connectivity Tree Computation

```
Input: $NT(i)\ \forall\ i \in \mathcal{N}$
Output: fresh subset $S$
  Initialization: subset $S := \emptyset$
  for $d = 0$ to $MaxHop - 1$ do
    repeat
      for all nodes $i$ such that $distance(i) = d$ do
        compute $potential(i)$
      end for
      pick a node $k$ s.t. $potential(k) = \max\{potential(i)\}$
      if $potential(k) \neq 0$ then
        $S := S \bigcup \{k\}$
        for all $t$ such that node $t \in NT(k)$ do
          node $t :=$ "covered"
          $distance(t) := d + 1$
        end for
      end if
    until $potential(k) = 0$
  end for
```

`Algorithm 1` is based on the analysis in [21] and is adapted to the considered network topology. It takes as input data all Neighbor Tables gathered at the AP after performing NDP. The algorithm builds the connectivity tree inside the CA by looking at each node's neighbors. It first places the AP at the root of the tree. The tree grows by connecting the nodes that are closest to the AP (one hop away). Then, the connectivity spreads towards the periphery. Theoretically, the algorithm ensures a minimum route for each node in the CA to/from the AP. Section 5.4 presents our simulation results.

The algorithm is "greedy". It designates as tree vertices the mobile nodes that cover the largest number of neighbors *not covered* yet. In this way, it ensures a fast coverage

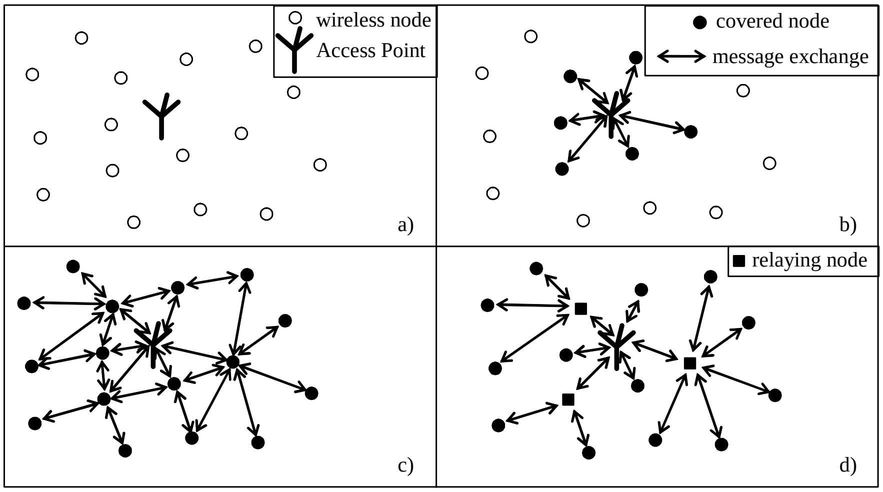

Figure 3: MHN control area initialization: a) initial state; AP is the only member in the CA; b) the AP performs the NDP protocol and starts broadcasting beacons; nodes that hear them join the CA; c) the AP and the nodes in the CA perform the NDP protocol; nodes that are two-hops-away from the AP join the CA; d) the AP knows the topology of the entire CA and constructs the connectivity tree

of the CA and reduces the number of relaying nodes in the subset S.

The algorithm stops only when the connectivity tree of the control area is complete, so all nodes already *covered* have only *covered* neighbors. In this way, the algorithm ensures full connectivity inside the CA. Therefore, when the algorithm stops, all nodes in the CA are *covered*. It terminates since it contains no infinite loops or blocking/waiting statements.

`Algorithm 1` constructs the connectivity tree according to the topology constraints. However, we must be careful in our choice of the subset S. Since relaying nodes must forward traffic (therefore spend more energy than a normal node), the nodes' battery levels and fairness issues should be taken into consideration. Hence, an improvement of the algorithm would consist in taking these parameters into account. The AP knows the current power levels of the nodes, if they are part of the node's status contained in the Hello messages; therefore, the AP would strive to allocate the relaying effort between nodes in a fair manner. A concrete solution is part of our future work.

3.4 Network Initialization and Node Admission to the Control Area

Each MHN control area is initialized by its corresponding AP. In the beginning, the AP is the only member of the CA. It starts to perform the NDP protocol and to broadcast periodic beacons. Once neighboring nodes hear the messages, they perform the membership protocol and become members of the CA.

The new member nodes synchronize with the CA through the beacons, and perform together the NDP protocol. The nodes that are two hops away from the AP hear the

Hello messages and perform the membership protocol. As the AP is aware of all nodes that are two hops away, it can construct the connectivity tree and the subset S needed to relay beacons and to gather the Neighbor Tables.

The previous procedure is employed iteratively to discover nodes that are at a growing distance from the AP. As more Neighbor Tables reach the AP, the resulting connectivity tree extends until it reaches the periphery.

A wireless node can be member of only one MHN control area. If more CAs are available, it must select and join one of them. The selection process takes into account the hop distance to each of the APs and the $SNIR$ of the received messages.

A node becomes a member of a CA through message exchange with neighboring member nodes. The member node that receives a membership request, sends the membership information to the AP. In the same time it inserts the new node in its Neighbor Table. The AP allocates the unique identity of the new member node and sends it back. Throughout the membership process, until the AP has enough information to update the connectivity tree, the member node that received the membership request becomes a temporary relaying node. A full description of the membership protocol is outside the scope of this paper and is part of our future work.

Figure 3 presents the initialization process for a CA containing nodes up to two hops away from the AP.

3.5 Synchronization, Routing and Energy Saving

The NDP protocol and the CTC algorithm represent the building blocks for our integrated solution to synchronization, routing and energy saving in MHNs.

Once the nodes have run the NDP protocol and the AP has performed the CTC algorithm, the AP knows the subset S containing the nodes that must be used as relays to ensure the connectivity of all nodes in the MHN control area.

The **synchronization** of the nodes in the CA starts at the AP. The AP generates a periodic beacon that contains a time-stamp with the present time of its internal clock. The beacon also contains the current subset S with the identities of all member nodes.

The AP broadcasts the beacon. Every node that hears it, updates its local clock with the time information contained in the beacon's time-stamp. The beacon also tells to the node whether it is a member of subset S. If it is, it becomes a relaying node and rebroadcasts the beacon. In order to avoid collisions at receiving nodes, we use the same scheduling mechanism as presented in Section 3.2 to forward the beacons. The robustness of this solution will be discussed in Section 4.2.

Routing of data packets to/from the AP is performed with the help of relaying nodes in subset S. Since the subset S ensures full connectivity of the control area, there exists a route for each node in the CA. The AP can compute a path to each node in the CA using the connectivity tree. Moreover, each node learns the identities of the relaying nodes in subset S from the beacon body. A node can intersect this subset with its own Neighbor Table to find a neighbor closer to the AP that is also a relaying node in the actual connectivity tree. Then, it sends the data packet to this relaying node, which forwards it on the uplink to the AP.

An important strength of our routing solution is that it consists only of packet forwarding. The transmitting nodes only need to know the identity of the next hop. This information is available in the nodes Neighbor Tables and consists of the address of the neighboring relaying node. On the down-link (packets that are routed from the AP to a specific node in the CA), the AP introduces in the header of the packet the identities of all relaying nodes needed for forwarding.

Other major strengths of our routing solution are:

- Routing can be performed with no additional protocol overhead. No extra exchange of messages is needed in order to compute a route to and from the AP.
- The routes are periodically updated by the AP based on fresh topology information; there is no stale route problem.
- Since the solution is based on the tree structure of the subset S, there are no loops in the routing path.

Routing between two nodes members of different MHN control areas is performed in three stages: from the sender to the corresponding AP; between the two APs using the infrastructure link between them; from the second AP to the receiver.

Finally, implementing an **energy saving** mechanism is straightforward, as the connectivity tree is available at all nodes inside the MHN control area.

All nodes that are not members of subset S can enter in sleep mode, in the interval between two beacons; the relaying nodes must stay awake at all times to ensure connectivity. All nodes must wake up periodically (with T_{beacon}) to receive the beacons and synchronize. They also need to perform the NPD protocol (with T_{NDP} periodicity). As we will see in Section 4.3, a node that is not a member of subset S and has no data to send or receive, must send three short NDP messages and update one Neighbor Table in the time interval when it is awake. This time represents about 3-7% of the total time during which the node is a member of the CA. This means that for at least 93% of the time (if we disregard the transition from the sleep state to awake state and vice-versa), the node can stay in sleep mode and conserve energy.

In a CA that implements this energy saving mechanism, data packets must be buffered at the AP and advertised in the beacon body. Techniques to buffer and advertise data packets in MHNs that implement energy saving are outside the scope of this paper, however, the IEEE 802.11 standard [2] can be a useful source of inspiration. Figure 4 is an example of routing between two nodes and the AP, in a MHN control area that implements energy saving.

3.6 Completion of Neighbor Tables Based on the Symmetry Assumptions

It is easily observed that the CTC algorithm performs well only if the AP gathers complete and accurate Neighbor Tables from the nodes. If Neighbor Tables are lost on the up-link or if they are incomplete, the AP fails to construct a correct image of the topology, therefore the resulting connectivity tree can be suboptimal.

Under the assumptions on symmetric, bidirectional links and based on the superior characteristics of the AP, we can further improve the performance of our solution. We

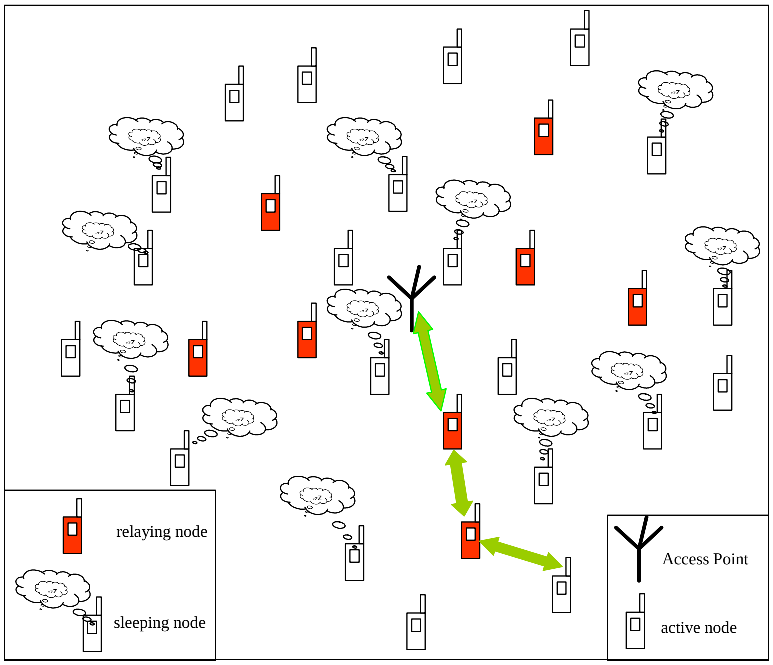

Figure 4: Routing and Energy Saving in a MHN control area

make these assumptions because all nodes inside the CA (including the AP) use the same fixed transmission power. However, even in this context, we are aware of the possibility of existing asymmetric links. We shortly discuss this case at the end of this section.

We let the AP complete the received Neighbor Tables and (in some cases) even recreate the missing ones from the received information. `Algorithm 2` is performed at the AP after the NDP protocol and before running the CTC algorithm.

The algorithm parses all received Neighbor Tables. Based on the symmetry assumption, it completes them with undiscovered neighboring nodes. Two neighboring nodes are not discovered only if they both lose each other's Hello messages during the same iteration of the NDP protocol. We refer to this event as a *symmetric* loss. The probability of such a loss is much smaller than the probability of a significant collision; it is computed in Section 4.1. Also, some lost Neighbor Tables can be reconstructed from the information gathered from neighboring nodes. Assume that the Neighbor Table of node i, *NT(i)*, received by the AP, contains node j but the AP did not receive *NT(j)*. The AP creates the table $NT(j)$ and adds to it node i, as a neighbor of node j. Furthermore, the AP parses through all other received Neighbor Tables and completes $NT(j)$ with all other nodes that have node j as a neighbor.

Even for the more general case (assuming the existence of asymmetric links), `Algorithm 2` could help the AP construct a connectivity tree based only on symmetric links. A small change in the algorithm, that would lead to the removal of all asymmetric links from the Neighbor Tables (instead of forcing their symmetry), would make the AP keep only

Algorithm 2 Neighbor Table Completion

```
Input: received Neighbor Tables
Output: reconstructed symmetric Neighbor Tables
  for all $i \in \mathcal{N}$ do
    if $\exists\ NT(i)$ at $AP$ then
      for all $j$ such that node $j \in NT(i)$ do
        if $\exists\ NT(j)$ then
          if node $i$ not in $NT(j)$ then
            insert node $i$ in $NT(j)$
          end if
        else
          create $NT(j)$
          insert node $i$ in $NT(j)$
        end if
      end for
    end if
  end for
```

the discovered symmetric links. Consequently, the AP would apply `Algorithm 1` on the resulting symmetric Neighbor Tables and construct the connectivity tree.

4 Analysis

This section presents the analysis on the key parts of the proposed solution. We concentrate on the distributed part of the NDP protocol and compute the probabilities of a Hello message collision, of a significant collision and of a symmetric loss. We also present a synchronization analysis of our solution and address complexity and overhead issues.

4.1 NDP Analysis

We derive formulas for the collision probabilities mentioned above.

The duration of transmission, (T_{Hello}), of a Hello message, expressed as a number of time slots, is computed as a function of the message total size, (L), in bytes, (including the MAC and physical headers), the transmission rate, (R), in $Mbps$, and the size of a time slot, (T_{slot}), in μs: $T_{Hello} = \lceil \frac{L \cdot 8}{R \cdot T_{slot}} \rceil$ time slots[4].

The member nodes in the MHN control area synchronously perform the NDP protocol, so that all Hello messages of one protocol iteration are transmitted in the same pair of Hello windows. Each node randomly and independently picks one time slot for each window, and when that time slot comes, it senses the channel and, if idle, transmits its Hello messages. If the channel is busy, the node defers until the next time slot. We

[4] This is a general formula for computing the duration of a Hello transmission. However, in our analysis and simulations the transmission of one Hello message fits into one time slot.

assume that each node in the CA can have at most K neighbors and that no other traffic is transmitted while performing the protocol.

We first derive general formulas for our analysis, and then we attribute numerical values for our parameters to obtain also quantitative results.

We compute the probability that node i loses one Hello message from a neighboring node j, due to message collisions. We denote this probability as $P_{collision}$. The Hello messages of node j collides at node i, if its transmission overlaps with another neighboring Hello message transmission. Since all nodes start transmitting only at the beginning of a time slot, we can identify the time slots that can cause a collision at node i. We call these time slots as *occupied*.

$$P_{collision} = \frac{\textit{number of occupied time slots}}{\textit{WndSize}} \tag{1}$$

The number of occupied time slots depends on K and on T_{Hello}.

The value $1 - P_{collision}$ represents the accuracy with which the nodes can construct their Neighbor Tables if, during one iteration of the NDP protocol, each node transmits only one Hello message. It has a great influence on the AP ability to reconstruct a correct CA topology.

We define $P_{significant}$ as the probability that both Hello messages from node j cause collisions at node i during the same iteration of the protocol. Since the two Hello messages are sent in separate disjoint Hello windows, $P_{significant}$ can be computed as the product of two independent Hello collisions:

$$P_{significant} = P_{collision_1} \cdot P_{collision_2} = [P_{collision}]^2 \tag{2}$$

The value $1 - P_{significant}$ represents the accuracy with which the nodes can construct their Neighbor Tables after all nodes in the CA performed the NDP protocol as described in Section 3.2.

Let $P_{symmetric}$ denote the probability of a symmetric loss event between the Hello messages of nodes i and j during the same iteration of the NDP protocol. This event implies that both Hello messages from node j collide at node i and at the same time, the Hello messages from node i collide at node j. We define $P_i(\textit{j-neighbors})$ as the probability that, during one Hello window, a message from node i collides with another message of any of the neighbors of node j; we have $P_i(\textit{j-neighbors}) \leq \frac{(K-1) \cdot T_{Hello}}{WndSize}$. Similarly, we define $P_j(\textit{i-neighbors})$. We also define P_{ij} as the probability that nodes i and j transmit at the same time during one Hello window: $P_{ij} = [\frac{T_{Hello}}{WndSize}]^2 \cdot WndSize$. The event of a symmetric loss in both Hello windows happens therefore with the probability:

$$P_{symmetric} = [P_i(\textit{j-neighbors}) \cdot P_j(\textit{i-neighbors}) + P_{ij}]^2 \tag{3}$$

The value $1 - P_{symmetric}$ represents the accuracy of the received Neighbor Tables after the AP performs `Algorithm 2`.

We exemplify these probabilities using a very unfavorable scenario for nodes i and j presented in Figure 5. Each of the two nodes has the maximum number of neighbors allowed (K). Moreover, nodes i and j are located in such a way that none of the

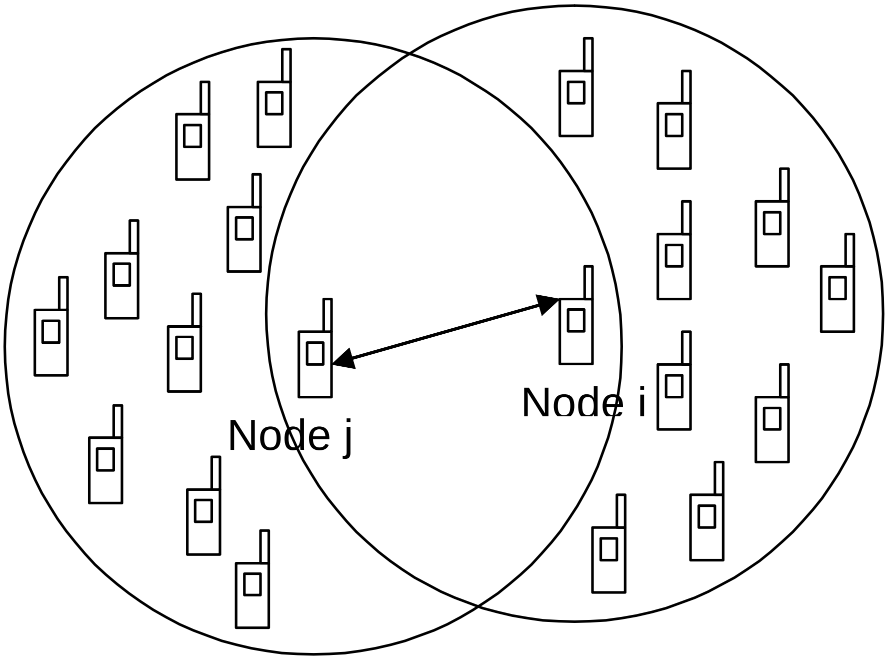

Figure 5: Worst Case Scenario for NDP Analysis

Table 1: Parameters for the NDP analysis

NDP periodicity T_{NDP}	$1\ s$
Hello window $WndSize$	50 time slots
time slot size T_{slot}	$40\ \mu s$
Tx-Rx Turnaround time	$5\ \mu s$
Transmission rate R	$11\ Mbps$
Hello message size L	40 bytes
Maximum neighbors K	10

two nodes senses any of the other node's neighbors. This scenario contains $2K$ nodes, maximizing the number of occupied time slots and the probabilities presented above. We use the values in Table 1 to assign numerical values to the probabilities. The values for the Tx-Rx Turnaround time, the transmission rate R and the headers contained in L are taken from [3]. In our choice of K, we are aware of the results obtained in [5].

With the values in Table 1, we compute T_{Hello} ($T_{Hello} < 1$ time slot) and we compute the probability that node i loses one Hello message from node j as the probability that node j chooses the same time slot for transmission as node i or any of its other neighbors:

$$P_{collision_i} \leq \frac{10 \ \textit{occupied time slots}}{50 \ \textit{total time slots}} = 0.2 \tag{4}$$

Equality happens when node i and all its neighbors except node j choose different time slots for their message transmission. Hence, using the NDP protocol with only one Hello message per iteration would lead to an error of up to 20% in the Neighbor Table construction. This would translate in turn, into a large error in the CA topology reconstruction at the AP.

We now compute the probability of the loss of both Hello messages of node j in the two disjoint Hello windows:

$$P_{significant} = [P_{collision}]^2 \leq 0.04 \tag{5}$$

Thus, the presented NDP protocol increases the accuracy of the Neighbor Table construction to more than 96%, increasing also the chances of a correct CA topology reconstruction at the AP.

In order to compute $P_{symmetric}$, we first notice that each node transmits its Hello message in the chosen time slot and that the transmission ends before the beginning of the next time slot. Also, a transmitting node has enough time to switch its antenna from Tx state to Rx state before the beginning of the next time slot. The nodes choose their transmitting time slots randomly and independently from all other nodes, and all nodes in the CA are previously synchronized as seen in Section 3.4. Since no other traffic is allowed in the CA during the Hello windows, each node will sense the channel idle and will transmit at the beginning of the chosen time slot. Hence, we can compute $P_{symmetric}$ for our NDP protocol:

$$P_{symmetric} = \{[P_i(j\text{-}neighbors)]^2 + P_{ij}\}^2 \tag{6}$$

$$P_{symmetric} \leq \{[\frac{9}{50}]^2 + \frac{1}{50}\}^2 \approx 0.0027 \tag{7}$$

Thus, the AP can obtain an over 99% accuracy in the received Neighbor Tables, if it performs `Algorithm 2`.

The numerical results obtained above motivate the choice we made to send two Hello messages during one iteration of the NDP protocol. They also encourage the use of `Algorithm 2` at the AP. The values depend on the size of the Hello windows and on the maximum number of neighbors in the CA. The probabilities can be decreased by increasing the size of the Hello window for a constant number of neighbors at the cost of a longer protocol execution time. Network designers must take into consideration the trade-off between the probability of losing neighbors and the execution time of the protocol.

The probabilities hold if the relaying of the Neighbor Tables towards the AP is free of collisions. This also motivates the use of the presented scheduling mechanism at each relaying node instead of direct relaying with normal back-off. The probabilities also increase if we consider traffic from adjacent control areas.

In Section 5 we will compare the obtained theoretical results with our simulations.

4.2 Synchronization Analysis

We analyze the performance of our synchronization solution in the most adverse conditions. We compute the maximum number of consequent beacons that a node can lose

Table 2: Parameters for the synchronization analysis

beacon period T_{beacon}	100 ms
propagation delay p	1 μs
maximum clock drift C	$\pm 0.01\%$
maximum clock difference D	0.02%
FHSS hop time H	224 μs
maximum nr. of hops in the CA $MaxHop$	4

and still be synchronous with the AP and the other nodes in the CA.

We perform an analysis similar to [10]. The values of the parameters are taken from the FHSS specifications in [2] (except our assumption on $MaxHop$). They are presented in Table 2. We assume that the relaying nodes forward the beacons generated by the AP towards the periphery of the CA. We assume that all delays caused by the transfer of the beacon frame from the MAC layer to the physical layer when transmitting, and vice-versa when receiving, are known. The relaying nodes compensate these delays by adjusting the time stamp in the forwarded beacon. The only delays that cannot be compensated are related to the propagation delays through the wireless medium[5].

We say that two nodes are no longer synchronized (or reach "asynchronism") if the time difference (drift) between their local clocks is larger than the FHSS hop time parameter. We use this parameter since it represents the strictest requirement in clock synchronization. When the clock drift exceeds this limit, if the nodes use FHSS, they are no longer able to switch frequencies at the same time. Therefore, they are no longer able to exchange messages.

Assuming the maximum clock difference D between the AP and the node considered, the drift Δ between their clocks after one beacon period is:

$$\Delta = D \cdot T_{beacon} = 2 \cdot 10^{-4} \cdot 10^{-1}\ s = 20\ \mu s \tag{8}$$

We add the propagation delay to this value. It results a maximum clock drift between AP and any node in the CA for one beacon period:

$$\Delta = D \cdot T_{beacon} + MaxHop \cdot p = 24\ \mu s \tag{9}$$

Given the constraint on synchronization, the number n of consequent beacons a node can lose before reaching asynchronism is:

$$n = \frac{H}{\Delta} = \frac{224\ \mu s}{24\ \mu s} = 9.33 \tag{10}$$

Hence, $n = 9$ represents an upper bound on the number of beacons a node can lose and still be considered synchronized. This bound holds for a node at the CA periphery, thus at the maximum hop number away from the AP. The computation shows how many beacons this node can lose and still be in the synchronization limits with the

[5] We assume a transmission range of at most 300 m which motivates our choice of p in Table 2.

AP. However, for our MHN model, the nodes only need synchronization with their next hop neighbors and not with the AP. By performing the same computations for two neighboring nodes we obtain $n = 10.66$.

If a node is not covered by the connectivity tree constructed by the AP during one iteration of our solution and $T_{NDP} \leq 10 \cdot T_{beacon}$, the node still does not lose synchronization. The AP has the chance to construct an updated connectivity tree in the next iteration.

In conclusion, with the proposed synchronization solution, a node remains synchronous with its neighbors even if it loses 10 consecutive beacons. If $T_{NDP} \leq 10 \cdot T_{beacon}$ (with the value for T_{beacon} from Table 2), the node remains synchronized even if it loses all beacons during one iteration of our solution.

4.3 Complexity and Overhead Evaluation

We now estimate the complexity of the two algorithms presented above and the overhead of our protocol in terms of number of required messages and execution time. While an optimal algorithm for computing the connectivity tree is NP-complete [9], `Algorithm 1` is simple and can be performed in polynomial time at the AP. Our choice of the algorithm is motivated by the limited execution time available at the AP as seen in Section 3.3.

We define $N = NrNodes$ and $M = MaxHop$. We also assume that each node in the CA has at most K neighbors. `Algorithm 1` needs $O(K)$ boolean operations to compute $potential(i)$. $potential(i)$ is computed for all nodes with the same $distance(i)$, therefore, $O(N)$ times. It takes $O(N)$ boolean operations to choose a node k with maximum $potential(k)$, and another $O(K)$ operations to set the parameters of the nodes in $NT(k)$. Therefore, we require $O(N \cdot K)$ operations during one iteration of the algorithm. We repeat this until $potential(k) = 0$, thus $O(N)$ times, for every hop in the CA. The total complexity of the algorithm is $O(M \cdot N^2 \cdot K))$. For M fixed, (M is of $O(\sqrt{N})$) and for K of order $O(N)$, the complexity is $O(N^3\sqrt{N})$.

`Algorithm 2` requires $O(K)$ boolean operations to search node i in $NT(j)$. It performs this search in all $NT(j)$ with $j \in NT(i)$, thus $O(K)$ times. The iteration is performed for all i; $1 \leq i \leq N$. The algorithm requires $O(N(K^2))$ operations. For K of $O(N)$, the total complexity is of $O(N^3)$ operations.

During one iteration of the NDP protocol, all nodes in the CA broadcast two Hello messages (40 bytes each) and update one Neighbor Table ($O(10K)$ memory locations needed). One additional message containing the Neighbor Table ($O(10K)$ bytes) is sent towards the AP. Relaying nodes in subset S send additional short poll messages ($O(K)$ messages) and need additional memory ($O(10K^2)$ memory locations).

As all nodes in the CA perform the NDP synchronously (as seen in Section 3.2), the NDP execution time is bounded. It can be theoretically computed, knowing only the size of all messages that need to be sent and the cardinality of subset S. NDP consumes between $2-5\%$ of the total time available for communication (dependent on the cardinality of subset S). Beacon relaying (using the scheduling mechanism) takes an additional $1-2\%$ of the total time (Section 5).

In conclusion, the proposed algorithms work in polynomial time and can be performed by the AP under the identified time constraints. The NDP protocol and the relaying of the beacon have a low protocol overhead in terms of exchanged messages and execution time.

5 Simulation Results

We simulate the proposed protocol and algorithms using the ns-2 simulator with the CMU wireless and mobility extension [6]. We evaluate our solution, emphasizing the following aspects:

- *NDP performance* and comparison with the theoretical values obtained in Section 4.1 for $1 - P_{collision}$, $1 - P_{significant}$ and $1 - P_{symmetric}$;
- *Evaluation of the scheduling mechanism* employed in both beacon and Neighbor Table relaying to/from the AP; We compare our solution using the scheduling mechanism with the same solution that implements relaying with random back off;
- *Control area connectivity* obtained by using the subset S generated by our solution in both fixed and mobile scenarios;
- *Execution time* of our solution; We compute the time needed for the NDP protocol and for beacon relaying, as the percentage of the total time available for message transmission inside the CA;
- *Optimality of the obtained routes* for all nodes inside the MHN control area; We compute the nodes distribution (as distance from AP counted in number of hops) given by our solution and compare it with the real nodes distribution in the considered topology;
- *Synchronization performance* of our solution inside the MHN control area; We compute the percentage of nodes inside the CA that lose synchronization as defined in Section 4.2.

In our simulations we use the same parameters as in Section 4 and we set the transmission range and interference range to 250 m and 550 m respectively. All transmissions use the Two Ray Ground propagation model. We program our solution for the worst case, where during each iteration, all nodes send their complete Neighbor Tables, and the AP computes the new subset S starting from zero. Since, for the same transmission power, the simulator forces symmetric links between nodes, we test our solution with `Algorithm 2` incorporated.

5.1 NDP Simulation

We simulate the unfavorable scenario analyzed in Section 4.1. We test the NDP protocol performance at nodes i and j on a fixed topology (1000×1000 m^2) of 40 nodes. Since we are only interested in the distributed part of NDP, we do not include the AP and the relaying nodes in this scenario. Each of the two nodes has the maximum number of neighbors, $K = 10$. Nodes i and j are in transmission range but they do not have any other common neighbors. We place the remaining 20 nodes on the topology so that they are within interference range of nodes i and j. We use them in order to assess the

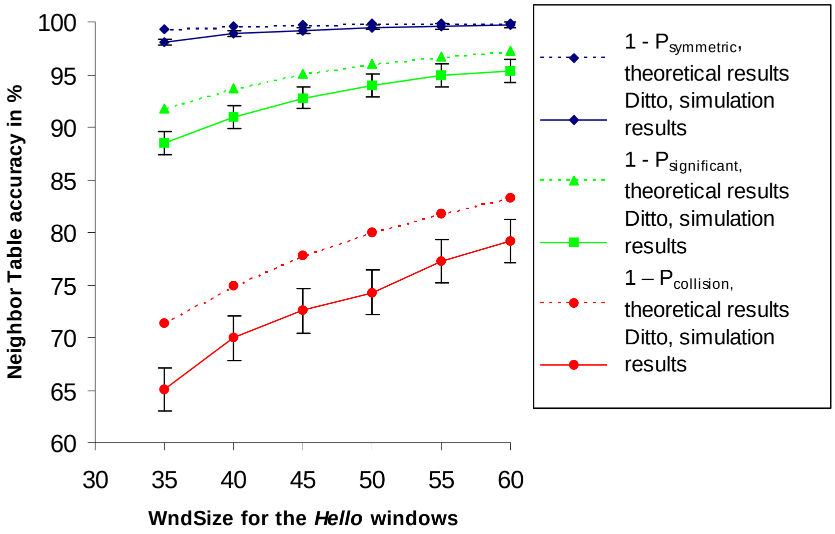

Figure 6: Performance of the Neighbor Discovery Protocol

impact of interference (coming from other nodes inside the CA, or from adjacent MHN control areas) on our protocol performance.

The results present the probabilities of constructing complete Neighbor Tables defined as $1 - P_{collision}$, $1 - P_{significant}$ and $1 - P_{symmetric}$. We average our results and compute the 95% confidence intervals over 5000 iterations of the NDP protocol (10 simulations of 500 s, each with a different random seed).

Figure 6 provides the Neighbor Table accuracy as a function of WndSize. As expected, that the performance of the protocol improves with the size of the windows. The simulation results are comparable to the theoretical ones obtained in Section 4.1. The impact of interference can also be observed since our theoretical model does not take it into account. Although the interference lowers the values of $1 - P_{collsion}$, its impact on $1-P_{significant}$ and $1-P_{symmetric}$ is greatly reduced. The results obtained for $1-P_{collision}$, $1-P_{significant}$, $1-P_{symmetric}$, and the robustness of $1-P_{symmetric}$ against interference, motivate our choice of the NDP protocol and the use of `Algorithm 2`.

5.2 Evaluation of the Scheduling Mechanism

As presented in Section 3, we employ a scheduling mechanism in both beacon and Neighbor Table relaying. The scheduling mechanism distributively imposes different delays in the transmissions, based on the position of the relaying nodes in the subset S advertisement in the beacon body. The purpose of the mechanism is to reduce the number of collisions when messages are relayed to/from the AP. The drawback of the mechanism is that it leads to a longer execution time of our solution compared e.g. to random back-off relaying.

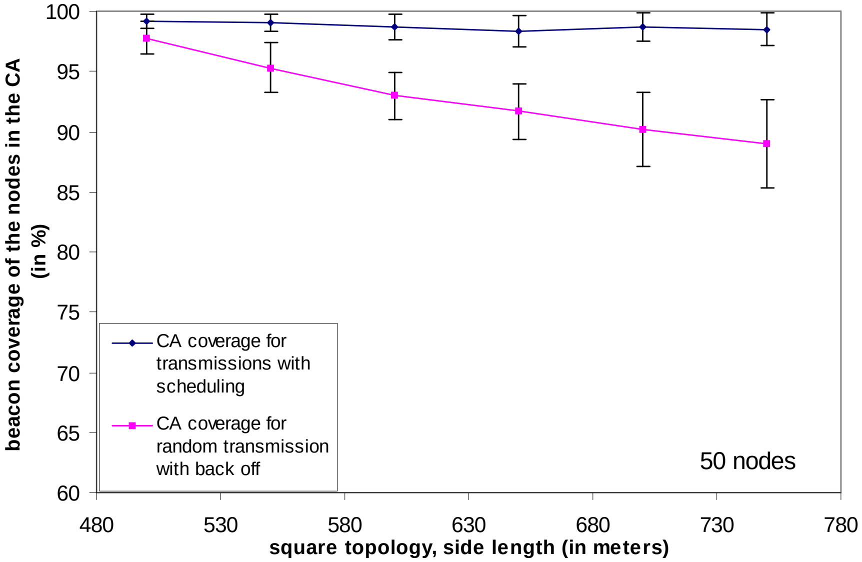

Figure 7: Beacon coverage of the nodes in the CA using scheduled and random back-off relaying, function of CA size

We compare the two alternatives (scheduled versus random back-off transmissions), through simulations of fixed networks and vary the topology size. An increase in topology size is translated in the increase of the average distance from the nodes in the CA and the AP (in hops). We compute the percentage of CA coverage obtained by using the subsets S derived from the two solutions.

We simulate the two proposals on square topologies of length varying in size between 500 and 750 m, using 50 nodes (scattered on the topologies according to a random uniform distribution). The AP is placed at the center of all topologies[6]. We average the simulation results over 5000 iterations of each of the two alternatives (10 simulations of 500 s).

Figure 7 presents the results for the scheduling and random back-off transmissions. As expected, while our solution scales well with the increase in topology size, the relaying with random back-off does not. The poor results of the random back-off relaying are explained by the fact that messages containing Neighbor Tables collide while they are relayed towards the AP. Hence the AP does not have the complete image of the topology and it cannot compute an accurate subset S.

The obtained results motivate us to choose the scheduling mechanism in the beacon and Neighbor Table relaying.

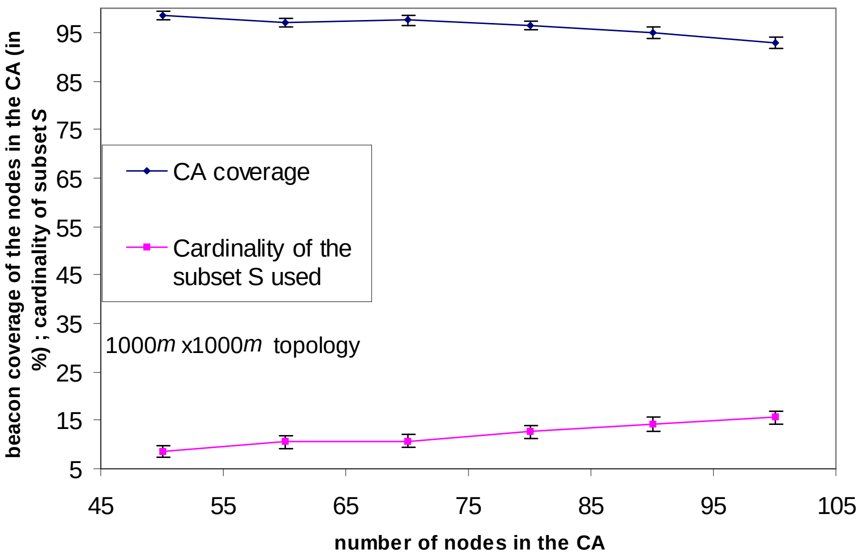

Figure 8: CA connectivity using subset S, function of number of nodes

5.3 Control Area Connectivity in Fixed and Mobile Networks

We simulate our solution in fixed and mobile topologies, and we test the efficiency with which the obtained subset S connects all nodes in the CA. In all our experiments, during each iteration of our solution, we count the percentage of nodes in the CA that receive beacons. If a beacon is received at node i, we deduce that there is a viable path between node i and the AP using the existing subset S, hence node i and the AP are connected.

We first consider a fixed network and test our solution on a 1000×1000 m^2 topology containing between 50 and 100 nodes. The AP is placed at the center of the topology. In all scenarios, the nodes are placed according to a tree topology, with the AP as the root of the tree[7]. We average our results and we compute the confidence intervals over 5000 iterations of our solution (10 simulations of 500 s each).

Figure 8 presents the percentage of nodes covered in the CA and the average cardinality of subset S computed by our solution. We observe that our solution scales well with the increase in node density. From simulation results, we observe that the cardinality of subset S depends on the total number of nodes in the CA. The cardinality of subset S is of order $O(\sqrt{N})$.

We then test our solution in mobile networks. We use a square topology of 800×800 m^2 with 50 nodes. We use the Random Waypoint Mobility model to induce nodes average speeds[8] between 5 and 25 m/s. The AP is immobile and is placed at the center

[6] The shape of the simulated CA is a square; however, the shape of the CA does not significantly affect the results of our simulations.

[7] We choose this topology in order to compensate for some implementation limitations. However, our protocol performance should not be influenced by the nodes distribution on the topology.

[8] Note that according to [25], the real average speed of the nodes is less than the declared one.

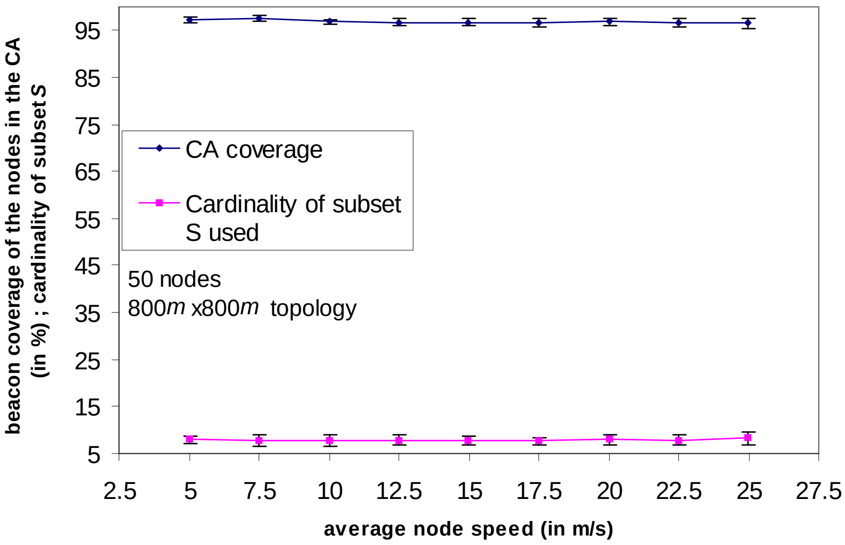

Figure 9: CA connectivity using subset S, function of mobility

of the topology.

Our simulation results are presented in Figure 9. We observe that our solution performance remains high and the size of subset S does not increase even at high node speeds. This is partly due to the fact that, in the given mobility model, the nodes average hop distance from the AP decreases in time.

The execution time of our solution employing the scheduling mechanism for beacons and Neighbor Table relaying is presented in Figure 10. It is computed for the fixed network scenarios discussed above. It varies between 3 and 7% of the total time allocated for message transmission inside the CA. The total execution time (for both NDP protocol and beacon relaying) is highly dependent on the cardinality of the subset S mainly because of the scheduling mechanism. The differences between the simulation and theoretic results are due to the extra guard times that we program for coping with the variable number of neighbors.

5.4 Route Optimality and Synchronization Performance

We now simulate our solution in order to assess the optimality of the routes obtained for all nodes in the MHN control area. We consider a fixed square topology of 1000×1000 m^2 containing 100 nodes. The AP is placed at the center of the topology.

For each of the nodes, we compute the minimum distance (in hops) to the AP, and we compute the distribution of the nodes according to this distance. Then we run our solution and again, we compute the nodes distribution according to the new distances to the AP (using the discovered subset S).

The two distributions are presented in Figure 11. We observe that only around 5%

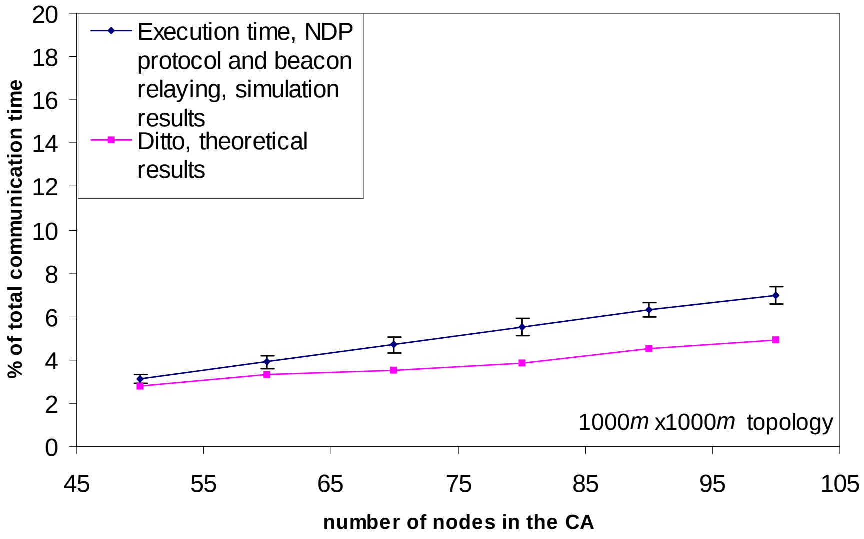

Figure 10: Execution time of NDP and beacon relaying with scheduling mechanism

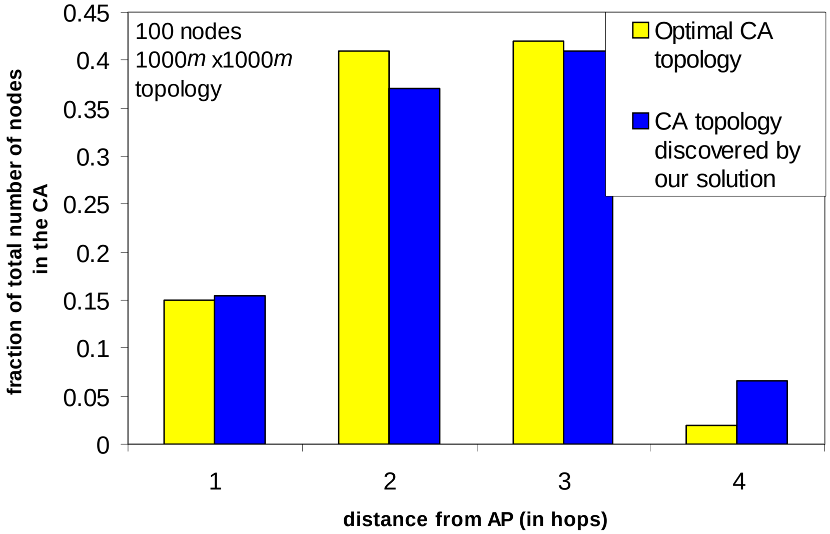

Figure 11: Comparison between the optimal and discovered node distribution, function of number of hops to the AP

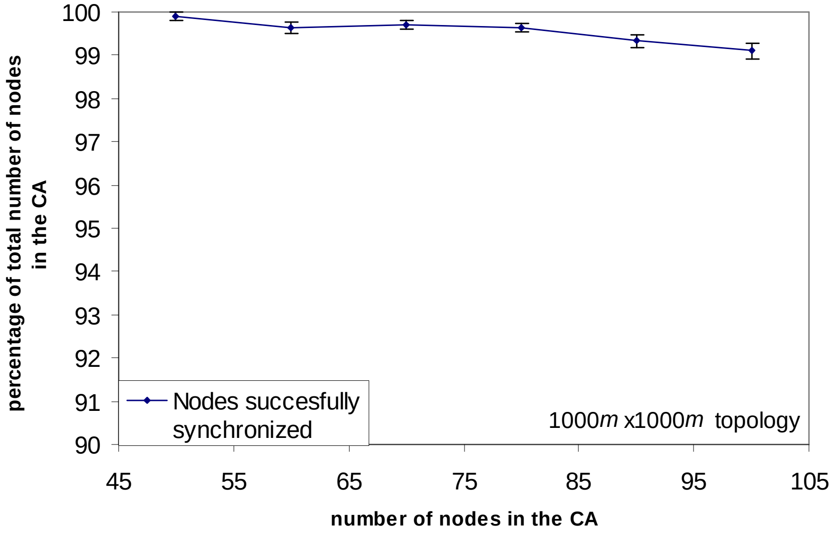

Figure 12: Percentage of successfully synchronized nodes

of the nodes in the CA do not benefit from the minimum route to the AP, using the discovered subset S. This result is in accordance to the purpose of our solution stated in Section 3.3.

We also test our synchronization solution on the fixed network topologies discussed before and make the number of nodes vary between 50 and 100. We compute the percentage of nodes that lose synchronization as defined in Section 4.2 and present the results in Figure 12. We observe that our solution behaves very close to the theoretical value of 100% node synchronization inside the CA.

6 Related Work

MHNs start to receive the attention of the scientific community. Different ideas, like multi-hop cellular networks [14], wireless TAPs [12] or MIT's Roofnet [22], are tested or implemented. They benefit from the stability and high bandwidth links of the fixed access points, while keeping the infrastructure costs low.

Routing in MHNs is discussed by Kumar et al. in [13] and by Ananthapadmanabha et al. in [1]. The authors present a multi-hop cellular architecture with one control channel over which all nodes are one hop away from the AP. Messages are exchanged on-demand between nodes and AP in order to establish routes in the cell. Consequently, multiple data channels with different power levels are used in order to route the packets. The AP receives the Neighbor Tables of all nodes in the cell. Based on them, the AP can compute on-demand routes inside the cell. Unlike our solution, the two papers do not address the synchronization and energy saving aspects of MHNs.

Our solution does not make the assumption of a high power control channel. All our control messages are exchanged on the same channel, governed by the CSMA/CA principle. Our solution starts from a fully distributed scenario and converges towards a routing solution for all nodes in the MHN control area. It needs no additional message exchange between AP and nodes in order to obtain on-demand routes. Compared to [13] and [1], where the Neighbor Discovery Protocols use TDMA or FDMA and requires *a priori* scheduling of the time slots or frequencies, our protocol is distributed, and runs on the same shared medium.

Pepe and Vojcic [18] present a routing protocol for MHNs with CDMA, that combines routing decisions with physical layer characteristics. Other approaches in ad-hoc networks with overlay, using CDMA, can be found in [7, 19, 26], where the authors find the optimal routing strategy and the channel allocation under given power constrains. In [15] Liu et al. compute the minimum number of base stations overlayed on top of an ad-hoc network in order to achieve a significant increase in network capacity.

The IEEE 802.11 standard represents one of the most concrete examples of synchronization and energy saving mechanisms for wireless communication [2]. In infrastructure mode, nodes synchronize on periodic beacons generated by the AP. In the interval between beacons, idle nodes can enter sleep mode. Our synchronization and energy saving mechanisms are also based on broadcasted beacons, and we extend their functionality to cover the multi-hop environment of the considered topology. An energy saving mechanism that does not require the synchronization of the nodes can be found in [27].

Energy efficient broadcast trees for wireless networks are studied by Wieselthier et al. in [20] and by Das et al. in [8]. Our connectivity tree algorithm takes into account the specificity of the considered MHN topology (fixed and equal transmission power for all nodes) and satisfies our routing constraint inside the CA (minimum number of hops between any node in the CA and the AP). The analysis in [21] proves that similar broadcasting techniques in ad-hoc networks perform very well in the case of increased node density and congested networks. Through simulations, we show that our solution is also robust in the case of mobile networks, leveraging on the extensive use of the APs.

Steps towards multi-hop hybrid wireless networks are taken by Bao and Garcia Luna Aceves in [4]. They present a topology management mechanism for ad-hoc networks that elects cluster heads, used for packet forwarding. Based on the topology information, further routing and energy saving mechanisms can be implemented. In [23] as well, neighboring nodes form virtual clusters. Inside the cluster, nodes synchronize with each other and decide on sleep schedules. Our topology management mechanism is based on the superior characteristics of the AP. The AP computes the topology of the CA and elects the relaying nodes. Our centralized solution is less computationally expensive for the wireless nodes.

7 Conclusion

In this paper we have addressed the aspects of synchronization, routing and energy saving in MHNs. Leveraging on the superior characteristics of the APs, that divide the underlying ad-hoc network into control areas, we have proposed an integrated and local solution to the three aspects. We periodically compute the connectivity tree inside each MHN control area; the relaying nodes perform packet forwarding (for both control and data packets). The designated relaying nodes stay awake to ensure CA connectivity while the rest of the nodes can enter sleep mode. Our solution is performed periodically and locally in each CA, to adapt to the changes in topology and to the mobility of the nodes. To the best of our knowledge, no integrated solution has been published so far.

Our solution has low complexity and works with low protocol overhead. This is compliant with the scarce resources of the mobile nodes. We present the trade-off between our solution's performance and the execution time needed. It is a designer's choice to emphasize more or less either one of the aspects.

Unlike other related works in the field, we do not make the assumption of an existing high-power control channel through which all nodes can communicate directly to the AP. Simulations show that our solution performs well without these assumptions, even when nodes are mobile.

Although focused on CSMA/CA networks, this paper can be used as a framework to design MHNs based on other operating principles, such as CDMA or UWB.

In terms of future work, we will extend our solution to provide peer-to-peer routing paths inside the CA (without involving the AP). We will extend our algorithms to the more complex case in which nodes (including the AP) adapt their power dynamically, notably in order to maintain connectivity. We will also address the fairness problem of the energy saving model. Also, the proposed protocol and algorithms will be extended to perform well in the presence of asymmetric links. Finally, we will provide concrete examples of membership protocols for MHNs and a full analysis of our solution taking into consideration the traffic generated by adjacent control areas.

8 Acknowledgements

The authors would like to thank Edward W. Knightly for his useful feedback, and Naouel Ben Salem, Mario Cagalj and Imad Aad for helpful discussions and comments.

References

[1] R. Ananthapadmanabha, B. Manoj, and C. Murthy. "Multi-hop Cellular Networks: The Architecture and Routing Protocols". *PIMRC*, IEEE 2001.

[2] ANSI/IEEE. "802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications". 1999.

[3] ANSI/IEEE. "802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Higher-Speed Physical Layer Extension in the 2.4GHz Band". 1999.

[4] L. Bao and J. J. Garcia Luna Aceves. "Topology Management in Ad Hoc Networks". *MobiHoc*, ACM 2003.

[5] D. Blough, M. Leoncini, G. Resta, and P. Santi. "The K-Neigh Protocol for Symmetric Topology Control in Ad Hoc Networks". *MobiHoc*, ACM 2003.

[6] CMU Monarch. "The CMU Monarch Project's Wireless and Mobility Extensions of NS", http://www.monarch.cs.cmu.edu/, 1998.

[7] R. L. Cruz and A. V. Santhanam. "Optimal Routing, Link Scheduling and Power Control in Multi-Hop Wireless Networks". *INFOCOM*, IEEE 2003.

[8] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray. "Minimum Power Broadcast Trees for Wireless Networks: Integer Programming Formulations". *INFOCOM*, IEEE 2003.

[9] M. R. Garey and D. S. Johnson. *"Computers and Intractability, A Guide to the Theory of NP-Completeness"*. W. H. Freeman and Company, 23^{rd} edition, 2002.

[10] L. Huang and T.-H. Lai. "On the Scalability of IEEE 802.11 Ad Hoc Networks". *MobiHoc*, ACM 2002.

[11] A. Jardosh, E. Belding-Royer, K. Almeroth, and S. Suri. "Towards Realistic Mobility Models for Mobile Ad hoc Networks". *MobiCom*, ACM 2003.

[12] R. Karrer, A. Sabharwal, and E. Knightly. "Enabling Large-scale Wireless Broadband: The Case for TAPs". *HotNets*, 2003.

[13] K. J. Kumar, B. S. Manoj, and C. Murthy. "MuPAC: Multi-Power Architecture for Cellular Networks". *PIMRC*, IEEE 2002.

[14] Y. D. Lin and Y.-C. Hsu. "Multihop Cellular: A New Architecture for Wireless Communications". *INFOCOM*, IEEE 2000.

[15] B. Liu, Z. Liu, and D. Towsley. "On the Capacity of Hybrid Wireless Networks". *INFOCOM*, IEEE 2003.

[16] O. C. Mantel, N. Scully, and A. Mawira. "Radio Aspects of Hybrid Wireless Ad Hoc Networks". *VTC*, IEEE 2001.

[17] W. Peng and X. Lu. "AHBP: An efficient broadcast protocol for mobile ad hoc networks". *Journal of Science and Technology - Beijing, China*, 2001.

[18] K. M. Pepe and B. Vojcic. "Cellular Multihop Networks and the Impact of Routing on the SNIR and Total Power Consumption". *Workshop on Multiaccess, Mobility and Teletrafic 2002.*

[19] C. A. St. Jean, A. N. Zadeh, and B. Jabbari. "Combined Routing, Channel Scheduling and Power Control in Packet Radio Ad Hoc Networks with Cellular Overlay". *VTC*, IEEE 2002.

[20] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. “Energy-Efficient Broadcast and Multicast Trees in Wireless Networks”. *Mobile Networks and Applications no. 7*, 2002.

[21] B. Williams and T. Camp. “Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks”. *MobiHoc*, ACM 2002.

[22] www.pdos.lcs.mit.edu/roofnet.

[23] W. Ye, J. Heidemann, and D. Estrin. “An Energy-Efficient MAC Protocol for Wireless Sensor Networks”. *INFOCOM*, IEEE 2002.

[24] J. Yoon, M. Liu, and B. Noble. “Sound Mobility Models”. *MobiCom*, ACM 2003.

[25] Y. Yoon, M. Liu, and B. Noble. “Random Waypoint Considered Harmful”. *INFOCOM*, IEEE 2003.

[26] A. N. Zadeh and B. Jabbari. “Self Organizing Packet Radio Ad Hoc Networks with Overlay (SOPRANO)”. *IEEE Communications Magazine*, IEEE 2002.

[27] R. Zheng, J. C. Hou, and L. Sha. “Asynchronous Wakeup for Ad Hoc Networks”. *MobiHoc*, ACM 2003.