Joint Synchronization, Routing and Energy Saving in CSMA/CA Multi-Hop Hybrid Networks

Dan Jurca and Jean-Pierre Hubaux

Laboratory of Computer Communications and Applications (LCA)
EPFL, 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 broadcast 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 broadcast 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.

Keywords: Broadcast 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 [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.

Our integrated solution is based on the periodic computation of the *broadcast tree*[2] (*BT*) inside the control area. We call *subset S* the set of nodes that are tree vertices at

---

[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 control area in a single hop.

[2]The broadcast tree is the tree that connects all nodes inside the CA with the AP (the root of the tree).
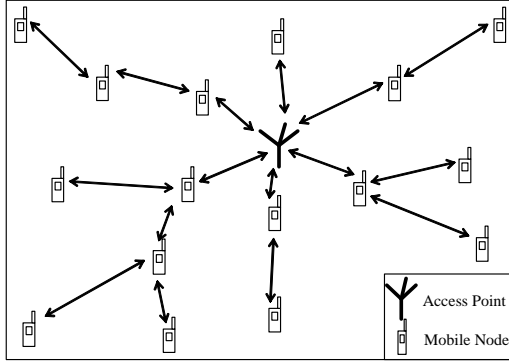
Figure 1. Control area in a MHN

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 a centralized algorithm (performed at the AP), to periodically compute the *BT* of the control area. The solution runs on the common wireless channel governed by the CSMA/CA principle.

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. A complete presentation and a more detailed analysis of our solution can be found in [13].

The rest of the paper is organized as follows. Section 2 introduces our system model. We present our solution in Section 3 and provide its analysis in Section 4. In Section 5 we 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 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. We assume that all traffic in the CA 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. 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 an 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. 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.

All nodes in a CA perform a distributed Neighbor Discovery Protocol to periodically update their local view of the CA. This information is forwarded to the AP using the relaying nodes. Based on the received information, the AP computes the new topology of the CA and constructs a new *BT* inside the CA.

The nodes that are vertices in the *BT* form the subset S and become the new relaying nodes in the CA. Their identity is advertised by the AP with the help of broadcasted beacons. All nodes receiving a beacon synchronize their internal clocks according to the time-stamp inserted by the AP in the beacon body. Additionally, nodes that discover that they belong to the actual *BT*, become relaying nodes and rebroadcast the beacon. They help the beacons propagate towards all nodes in the CA and forward data and other control packets between any node in the CA and the AP.

3.1. Neighbor Discovery Protocol (NDP)

The NDP is performed periodically at each member node in the CA (including the AP). Its purpose is to provide the AP with fresh topology information based on which, the AP constructs a new *BT* and assigns a new set of relaying nodes (Section 3.2).

All nodes in the CA start performing the NDP protocol at the same time. This approach offers the possibility to reserve a special time window for performing NDP in
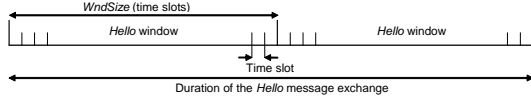
Figure 2. *Hello* **windows used for back-off in the NDP**

each CA (while all other traffic is buffered and delayed)[3]. It leads to a good performance of our solution while imposing a hard bound on its execution time (Section 4).

When running NDP, each node i (member of the CA), broadcasts information about itself in short $Hello$ messages. This information concerns the identity of the node, its known distance from the AP in hops, and its membership in subset S. We organize this information as a data structure that we call the $status$ of node i. Each Hello message contains the $status$ of the transmitting node.

Furthermore, each node i constructs a data structure containing the $status$ of its neighboring nodes. From now on we will refer to this data structure as the *Neighbor Table* of node i or *NT(i)*.

During one iteration of NDP, node i constructs its *NT(i)* with the $status$ extracted from all heard Hello messages broadcasted by its neighbors.

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*. 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 *NT* of the receiving node will not contain one neighbor.

Once the two Hello windows elapse, the nodes have to forward the fresh *NT*s to the AP. The protocol uses the relaying nodes in the most recently advertised subset S to gather all *NT*s and to relay them to the AP.

---

[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 towards the end of this section.

Each relaying node consequently polls all its neighbors (nodes present in its *NT*) and gathers their Neighbor Tables. Nodes only send the differences between the fresh *NT* and the previous version of it. If a node is not polled, after a timeout, it attempts a direct *NT* transmission to a neighboring relaying node. This relaying node is chosen from the node's fresh Neighbor Table. The relaying nodes aggregate all received *NT*s (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. Since the identities of all relaying nodes are included as an ordered structure in the beacon's body, we exploit the position of each relaying node in this structure in order to distributively schedule non-overlapping transmissions.

As seen before, we use the subset S chosen in a previous iteration of our solution to relay protocol messages in the present iteration. However, the topology of the CA can change, due to the mobility of the nodes, and the previous *BT* might fail to cover all nodes at the time of the relaying. Therefore, the periodicity of the protocol, (T_{NDP}), is strictly related to the assumptions of mobility of the nodes [12, 26, 23].

Last, we present a short description of our protocol during the CA's initialization phase.

Only an AP can initialize a new CA. It starts broadcasting beacons that advertise the parameters of the new CA and it starts performing the NDP protocol in order to discover its neighboring nodes. Nodes that hear the Hello messages, perform the membership protocol and join the CA. At their turn, they start performing the NDP protocol and forwarding their *NT*s to the AP. In this way, the AP iteratively discovers nodes that are more and more distant. Figure 3 presents the initialization process for a CA containing nodes up to two hops away from the AP.

3.2. Broadcast Tree Computation Algorithm (BTC)

The purpose of the BTC algorithm is to compute the broadcast tree of the CA, based on the received Neighbor Tables.

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

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

• 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.

In the same time, we want to minimize the cardinality of subset S in order to reduce the energy needed to relay messages, and to avoid message collisions. The problem of computing the minimum broadcast tree can be reduced to
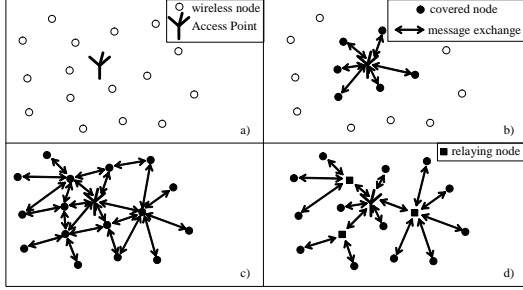
Figure 3. MHN CA initialization: a) initial state; AP is the only member in the CA; b) the AP performs the NDP protocol and starts broadcasting beacons; nodes that hear them join the CA; c) the AP and the nodes in the CA perform the NDP protocol; nodes that are two-hops-away from the AP join the CA; d) the AP knows the topology of the entire CA and constructs the connectivity tree.

Algorithm 1 Connectivity Tree Computation

Input: $NT(i) \, \forall \, i \in \mathcal{N}$
Output: fresh subset S
1: **Initialization:** subset $S := \emptyset$
2: **for** $d = 0$ to $MaxHop - 1$ **do**
3:    **repeat**
4:       **for all** nodes i such that $distance(i) = d$ **do**
5:          compute $potential(i)$
6:       **end for**
7:       pick a node k s.t. $potential(k) = \max\{potential(i)\}$
8:       **if** $potential(k) \neq 0$ **then**
9:          $S := S \bigcup \{k\}$
10:         **for all** t such that node $t \in NT(k)$ **do**
11:            node $t :=$ "$covered$"
12:            $distance(t) := d + 1$
13:         **end for**
14:       **end if**
15:    **until** $potential(k) = 0$
16: **end for**

---

the Geometric Connected Dominating Set Problem [10]. It is NP-complete and an optimal polynomial time solution is hard to find. Hence we present a simpler and suboptimal algorithm that computes the broadcast tree in feasible time inspired by [20].

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). We say that node i, belonging to the CA, is *covered* if it is contained in the *NT* 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 $S = \emptyset$.

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 takes as input data all *NT*s gathered at the AP after performing NDP. The algorithm builds the *BT* 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. The algorithm ensures a minimum route for each node in the CA to/from the AP.

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

The algorithm stops only when the *BT* 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.

Algorithm 1 constructs the *BT* according to the topology constraints. In a mobile scenario, the subset S computed by our algorithm will change at each iteration, hence distributing the relaying load on more nodes. In a static scenario, our algorithm will always pick the same subset S, which may exhaust the nodes battery power fast, due to frequent relaying. This can be avoided by introducing other constraints like node battery levels or fairness issues, which can be easily integrated in our algorithm while computing the broadcast tree.

We further improve our solution under the assumption of symmetric bidirectional links. We let the AP complete the received *NT*s of some nodes, or even create the missing ones from the information received from neighboring nodes. This further decreases the probability of not discovering some nodes in the CA before performing the BTC algorithm. We refer to this event as a $symmetric$ loss, which only occurs when two neighboring nodes lose both each others Hello messages during the same iteration of the NDP.

In the more general case, when we acknowledge the presence of asymmetric links, a small change in our improvement could help the AP construct the *BT* based only on symmetric links.

3.3. Synchronization, Routing and Energy Saving

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

4

Once the nodes have run the NDP protocol and the AP has performed the BTC algorithm, the AP knows the subset S containing the relaying nodes that ensure the connectivity of all nodes in the CA.

The **synchronization** of the nodes in the CA starts at the AP. The AP generates a periodic beacon (with periodicity T_{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. Also, relaying nodes rebroadcast the beacon using the same scheduling mechanism mentioned in Section 3.1 in order to avoid collisions.

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 *BT*. Each node in the CA has in its neighborhood at least one relaying node that can forward its data packets 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 *NT*s and consists of the address of the neighboring relaying node. On the down-link route (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 *BT* is available at all nodes inside the CA.

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 NDP protocol (with T_{NDP} periodicity). This takes around 3-7% of the total lifetime of a node inside a CA (Section 4.3) which leaves at least 93% of the time for sleep mode, in the case of a node that does not send or receive data packets.

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.

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 CA 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 assume that each node in the CA can have at most K neighbors and that no other traffic is transmitted while performing the protocol. Table 1 contains the numerical values used in our analysis. 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 [6].

Figure 4 presents the worst case scenario that we take into consideration. 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 two nodes senses any of the other node's neighbors. This scenario contains $2K$ nodes, maximizing the number of possible collisions.

Let $P_{collision}$ be the probability that node i loses one Hello message from a neighboring node j due to collision. This happens if j transmits its Hello message in the same time slot as node i or any other neighbor of node i. We refer

---

[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.
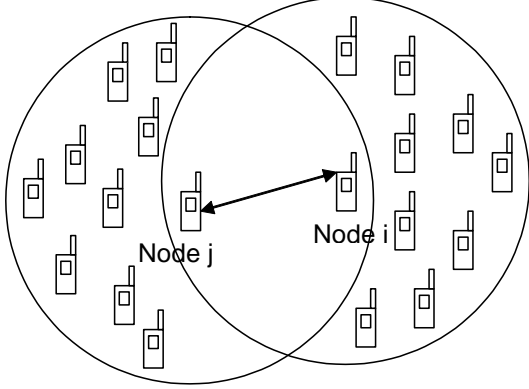
Figure 4. Worst Case Scenario for NDP

to these time slots as *occupied*. The number of occupied time slots depends on K and on T_{Hello}. For the values in Table 1:

$$P_{collision} = \frac{number\ of\ occupied\ time\ slots}{WndSize} \leq \frac{10}{50} = 0.2$$

Equality happens when node i and all its neighbors except j choose different time slots for their transmissions.

The value $1 - P_{collision}$ represents the accuracy with which the nodes can construct their *NT*s if, during one iteration of the NDP protocol, each node transmits only one Hello message.

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:

$$P_{significant} = P_{collision_1} \cdot P_{collision_2} = [P_{collision}]^2 \leq 0.04$$

The value $1 - P_{significant}$ represents the accuracy with which the nodes can construct their *NT*s after all nodes in the CA performed the NDP protocol as described in Section 3.1.

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.

We define $P_i(j\text{-}neigh)$ 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 $P_i(j\text{-}neigh) \leq \frac{(K-1) \cdot T_{Hello}}{WndSize}$. Similarly, we define $P_j(i\text{-}neigh)$. 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:

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

$$\begin{aligned} P_{symmetric} &= [P_i(j\text{-}neigh) \cdot P_j(i\text{-}neigh) + P_{ij}]^2 \\ &\leq \{[\frac{9}{50}]^2 + \frac{1}{50}\}^2 \approx 0.0027 \end{aligned}$$

In the computations we use the facts that nodes choose their transmitting time slots randomly and independently from all other nodes, that the time slots are big enough for the transmission of the whole Hello message before the beginning of the next slot, and that nodes are previously synchronized.

The value $1 - P_{symmetric}$ represents the accuracy of the received *NT*s after the AP performs the symmetric Neighbor Table completion as presented in Section 3.2.

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 symmetric Neighbor Table completion 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 hold if the relaying of the *NT*s 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.

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

We perform an analysis similar to [11]. 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].

---

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

Table 2. Parameters for the synchronization analysis

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

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:

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

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

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

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. The computation shows how many beacons this node can lose and still be in the synchronization limits with the 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 $n = 10.66$.

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.

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 $O(NK)$ 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 $O(MN^2K)$. For M fixed, (M is of $O(\sqrt{N})$) and for K of order $O(N)$, the complexity is $O(N^3\sqrt{N})$. The algorithm for symmetric Neighbor Table completion performed by the AP has a complexity of $O(N^3)$ operations.

All nodes in the CA need memory of order $O(10K)$ locations, must broadcast two Hello messages and send an additional message containing the *NT* towards the AP. Relaying nodes send additional short poll messages and need additional memory.

As all nodes in the CA perform the NDP synchronously (as seen in Section 3.1), 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 and 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.

5. Simulation Results

We simulate the proposed protocol and algorithms using the ns-2 simulator with the CMU wireless and mobility extension [8]. We evaluate our solution, namely, the performance of the NDP, the obtained connectivity inside the control area, the optimality of discovered routes, the nodes synchronization and the execution time. A comparison between our solution implementing the scheduling mechanism mentioned in Section 3.1, and our solution with random transmissions can be found in [13]. The comparison motivates our choice in scheduling the relaying of beacons and of messages to the AP.

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.

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 ($1000 \times 1000\ m^2$) 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
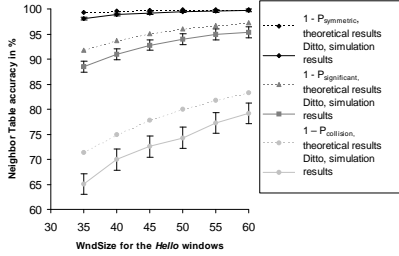
Figure 5. Performance of NDP

Figure 6. Number of nodes influence on CA connectivity

Figure 7. Mobility influence on CA connectivity

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 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 5 provides the Neighbor Table accuracy as a function of WndSize. As expected, 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. 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 the symmetric Neighbor Table completion.

5.2. 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[6]. We average our
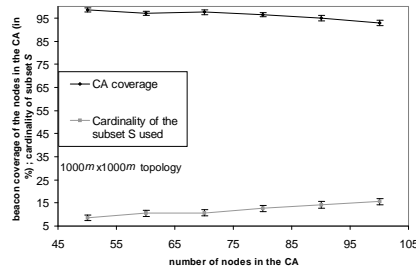
results and we compute the confidence intervals over 5000 iterations of our solution (10 simulations of 500 s each).

Figure 6 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[7] between 5 and 25 m/s. The AP is immobile and is placed at the center of the topology.

Our simulation results are presented in Figure 7. 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 8. 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 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.

5.3. 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

---

[6]We choose this topology in order to compensate for some implemen-

tation limitations. However, our protocol performance should not be influenced by the nodes distribution on the topology.

[7]Note that according to [27], the real average speed of the nodes is less than the declared one.
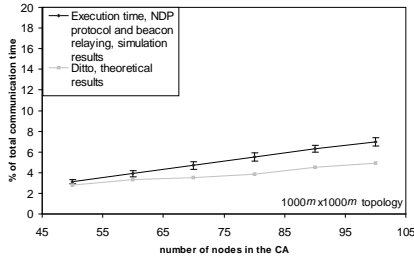
Figure 8. Execution time of NDP and beacon relaying

Figure 9. Optimal and discovered node distribution, function of hops to AP

Figure 10. Percentage of successfully synchronized nodes

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 9. We observe that only around 5% of the nodes in the CA do not benefit from the minimum route to the AP, using the discovered subset S.

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 10. 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. Our solution can be used (or adapted) in the framework of proposals such as: multi-hop cellular networks [17], wireless TAPs [14], iCAR [24] or MIT's Roofnet [25]. These proposals benefit from the stability and high bandwidth links of the fixed access points, while keeping the infrastructure costs low. Commercial products such as Nokia RoofTop(TM) [19] manage to connect fixed ad-hoc communities to the infrastructure with the help of dedicated line-of-sight routers. With this respect, the novelty of our work lies in our idea of integrating the 802.11 based communication in a multihop environment benefiting from the presence of a coordinating access point.

Routing in MHNs is discussed by Kumar et al. in [15] 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. 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.
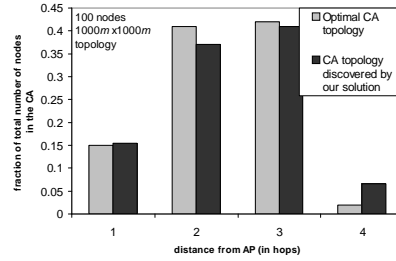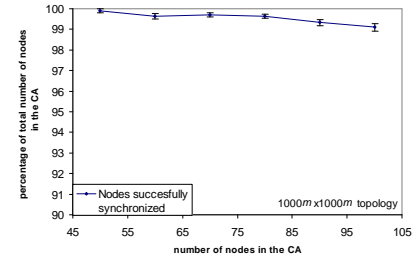
Pepe and Vojcic [21] present a routing protocol for MHNs with CDMA, that combines routing decisions with physical layer characteristics. Other approaches in ad-hoc networks with overlay, using CDMA, can be found in [9, 28], where the authors find the optimal routing strategy and the channel allocation under given power constrains. In [18] 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. An analysis of power efficient communication in multi-hop wireless networks with impact on routing, scheduling and power control, can be found in [5].

Energy efficient broadcast trees for wireless networks are studied by Wieselthier et al. in [22] and by Cagalj et al. in [7]. 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 [23] 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. 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, hence, our 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 broadcast 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.

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.

Acknowledgments

References

[1] R. Ananthapadmanabha, B. Manoj, and C. Murthy. "Multihop 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] R. Bhatia and M. Kodialam. "On Power Efficient Communication over Multi-hop Wireless Networks: Joint Routing, Scheduling and Power Control". *INFOCOM*, IEEE 2004.

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

[7] M. Cagalj, J.-P. Hubaux, and C. Enz. "Minimum-Energy Broadcast in All-Wireless Networks: NP-Completeness and Distribution Issues". *MobiCom*, ACM 2002.

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

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

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

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

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

[13] D. Jurca and J.-P. Hubaux. "Joint Synchronization, Routing and Energy Saving in Multi-hop Hybrid Networks". Technical Report IC/2004/35, EPFL, 2004.

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

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

[16] S. Lee, S. Banarjee, and B. Bhattacharjee. "The Case of a Multi-hop Wireless Local Area Network". *INFOCOM*, IEEE 2004.

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

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

[19] Nokia Corporation. "Nokia RoofTop Wireless Routing". *White Paper*, 2001.

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

[21] 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*.

[22] 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.

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

[24] H. Wu, C. Qiao, S. De, and O. Tonguz. "Integrated Cellular and Ad Hoc Relaying Systems: iCAR". *IEEE Journal on Selected Areas in Communications*, 19(10), Oct. 2001.

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

[26] J. Yoon, M. Liu, and B. Noble. "Sound Mobility Models". *MobiCom*, ACM 2003.

[27] Y. Yoon, M. Liu, and B. Noble. "Random Waypoint Considered Harmful". *INFOCOM*, IEEE 2003.

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