GETRF delivrable 6:

Combining and improving the mechanisms designed in the GETRF project

Cédric Adjih, Ichrak Amdouni, Philippe Jacquet, Antonia Masucci, Paul Mühlethaler
Hipercom2 Project-Team
Inria Paris-Rocquencourt
78150 Le Chesnay cedex, France

30 September 2014

Contents

1 Introduction

In the previous deliverables of the project we studied the four following mechanisms:

- energy saving mechanisms using synchronous techniques,
- energy saving mechanisms using asynchronous techniques,
- network coding,
- mobile routing.

In this deliverable, we study how these techniques can be combined. We also present how to improve asynchronous techniques for energy saving and how to adapt mobile routing to other assumptions.

2 Combining energy saving mechanisms based on synchronous and asynchronous approaches

The idea is to split the time frame between a first time interval during which the node operates with a synchronous low-duty-cycle scheme and a second time interval during which the node operates with an asynchronous low-duty-cycle mechanism. The time frame is split as shown in Figure 1. An internal clock of the nodes must maintain the frame synchronisation. Inside the interval dedicated to the synchronous low-duty-cycle scheme all the mechanisms follow the algorithm described in [4] and inside the interval dedicated to the low-duty-cycle synchronous scheme all the mechanisms follow the algorithm described in [3].

An application of such an architecture can be appropriate if a small number of nodes periodically send packets to the sink, while a large number nodes is monitoring an infrequent event over a large area, see Figure 2. The former nodes use a synchronous low-duty-cycle mechanism; whereas the latter nodes employ a low-duty-cycle asynchronous mechanism.

The dual mode of operation is beneficial since the periodic traffic is better supported by a synchronous approach, while reporting an infrequent event to the sink is more efficiently handled with an asynchronous low-duty-cycle mechanism.

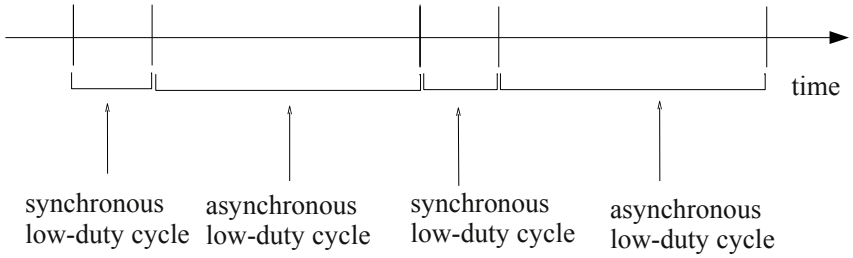

Figure 1: Combined synchronous and asynchronous low-duty-cycle mechansims

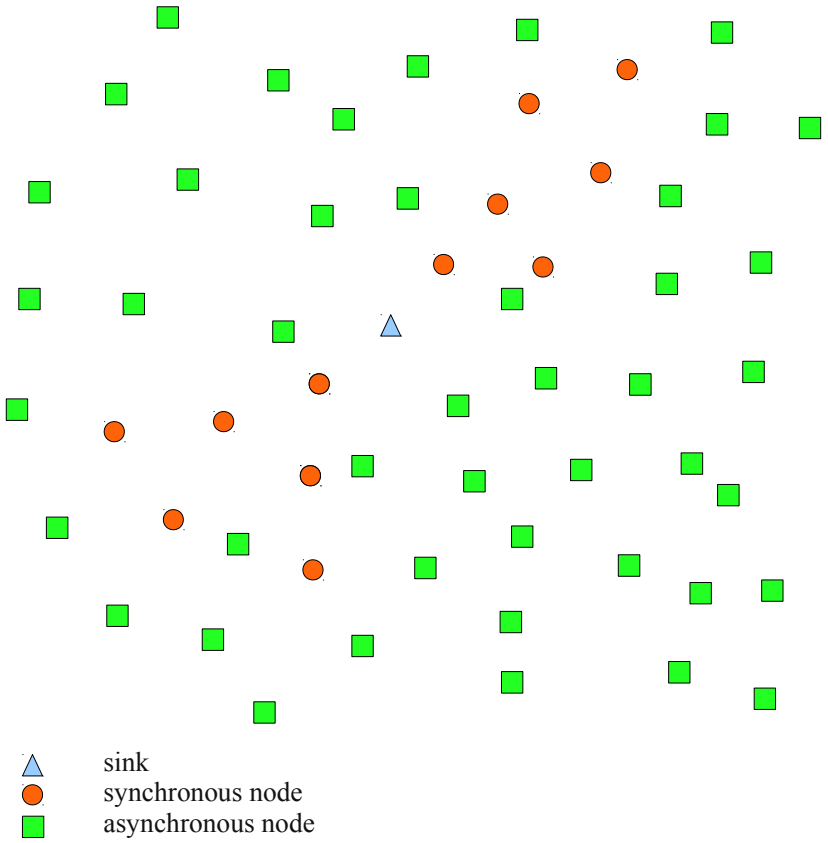

Figure 2: Nodes in synchronous and asynchronous low-duty-cycle modes

3 Combining synchronous energy saving mechanisms and mobile routing

This combination is complex since the synchronous energy saving mechanisms need a stable topology and, in contrast the mobile routing schemes

imply a very dynamic topology. Combining these two techniques would be unlikely to prove fruitful.

4 Combining an energy saving mechanism based on a synchronous approach and network coding

The energy saving mechanism based on a synchronous approach is designed to transmit packets from nodes to a sink. There are usually many routes to link two distant nodes in the network and thus network coding is possible. In the following we study network coding when the nodes are on a grid.

4.1 Presentation of the problem

We consider a unicast communication, one source and one destination, in a torus grid where each node has 4 neighbors. We assume that initially the source broadcasts the packets to its neighbors, then, once they have received the information, they transmit the packets as in a wired network according to the ORCHID protocol [5].

In this communication setting we apply network coding. Network coding, introduced by Ahlswede, Cai, Li and Yeung in [1], is different from the classic routing strategy since intermediate nodes combine information from different traffic flows before forwarding it. The use of network coding enables us to attain the maximum capacity, given by the min-cut of the associated graph.

In particular, we analyze the case when all the nodes in the network have the same transmission rate (packets per second). We compute the min-cut in the network considered and we find that is strictly linked to the number of neighbors. In order to prove this, we use tools from additive combinatorics.

4.2 Background

We consider a unicast communication in a torus grid G of width n_X and height n_Y.

The connectivity in this network is described as a hypergraph [8].

A *hypergraph* is a graph where the edges are replaced by hyper-arcs which are generalizations of arcs that may have more than one end node.

Let us consider a source s and a destination $t \in \mathcal{V}$.

Definition 1. *An* s-t cut *is a partition of the set of nodes* $\mathcal{V}$ *into two sets* S, T *such that* $s \in S$ *and* $t \in T$.

Let $Q(s,t)$ be the set of such *s-t cuts*: $(S,T) \in Q(s,t)$.

We denote Δ_S the set of nodes of S that are neighbors of at least one node of T:

$$\Delta_S \triangleq \{v \in S | \mathcal{N}(v) \cap T \neq \emptyset\}. \tag{1}$$

The *capacity of the cut* $(S,T) \in Q(s,t)$, denoted by $C(S)$, is defined as the maximum rate between the nodes in S and the nodes in T:

$$C(S) \triangleq \sum_{v \in \Delta_S} C_v. \tag{2}$$

In other words, the idea is to cut the network into two parts, and check the total quantity of information transmitted from nodes in the part which contains the source, to nodes in the other part. The *min-cut* between s and t, that we denote by $C_{\min}(s,t)$, is the cut of $Q(s,t)$ with the minimum capacity:

$$C_{\min}(s,t) \triangleq \min_{(S,T) \in Q(s,t)} C(S). \tag{3}$$

We define the neighborhood of the node $(0,0)$ as

$$R \triangleq \left\{(x \bmod n_X, y \bmod n_Y) : (x,y) \in \mathbb{Z}^2 \text{ and } x^2 + y^2 \leq 1\right\}. \tag{4}$$

The number of elements in R less 1 represents the number of neighbors of a node. From the definition of R in (4) the cardinality of $|R|$ is 5 and each node has 4 neighbors.

One way to express the neighborhood of an area is given by the Minkowski addition [12].

Definition 2. *Given two subsets A and B of a group, the Minkowski sum $A \oplus B$ is defined as the set of all sums generated by all pairs of points in A and B, respectively:*

$$A \oplus B \triangleq \{a + b \ : \ a \in A, b \in B\}. \tag{5}$$

In the torus grid, the closed set of neighbors of one node t, denoted by $\mathcal{N}[t]$, can then be redefined in terms of the Minkowski sum as follows

$$\mathcal{N}[t] = \{t\} \oplus R. \tag{6}$$

This is indeed a translation of R, the set of neighbors of the node at $(0,0)$ to the node t.

4.3 Min-cut between source and destination

We now introduce the essential notion of "large neighborhood" related to the fact that we have no problem of neighborhood for the nodes that are on the border of the grid.

Definition 3. *If for all subsets $A \subset G$ at least one of these conditions is verified:*

$$A \oplus R = G \tag{7}$$

$$\text{or} \quad |A \oplus R| \geq |A| + |R| - 1 \tag{8}$$

we say that R verifies the large neighborhood condition.

Lemma 1. *[13] When R verifies the large neighborhood condition, let (S, T) be a partition of G and $\Delta_S = \mathcal{N}[T] \setminus T$, where $\mathcal{N}[T] \triangleq \mathcal{N}(T) \cup T$ is the "closed neighborhood" of nodes of T. If there exists $\sigma \in S$ such that $\sigma \notin \Delta_S$ then*

$$|\Delta_S| \geq |R| - 1.$$

Theorem 1. *We consider a network G which is represented by a torus grid with a neighborhood defined by the set R, and with rate selection such that $C_v = C_s$ with $v \in \mathcal{V} \setminus \{s\}$. If R verifies the large neighborhood condition then the min-cut between s and t is given by $C_{\min}(s, t) = C_v(|R| - 1)$.*

Proof: Consider the source s and the destination t, see Figure 3. Let us consider any cut $(S, T) \in Q(s, t)$.

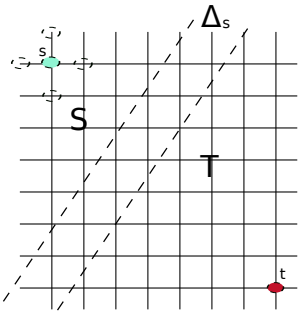

Figure 3:

The capacity of this cut is

$$C(S) \triangleq \sum_{v \in \Delta_S} C_v \quad \text{with} \quad \Delta_S \triangleq \{v \in S : \mathcal{N}(v) \cap T \neq \emptyset\}. \tag{9}$$

First of all, we assume that we are near the source: $s \in \Delta_S$. Moreover, we consider Δ_{TS} the set of neighbors of s which are not in Δ_S and whose cardinality is denoted by ℓ and $\ell \geq 1$ because $s \in \Delta_S$. We can have two cases: either all the neighbors of the source are included in $\Delta_S \cup T$ or some of the neighbors of the source are not included in $\Delta_S \cup T$.
First, we analyze the case $\mathcal{N}(s) \subset \Delta_S \cup T$, see Figure 4.

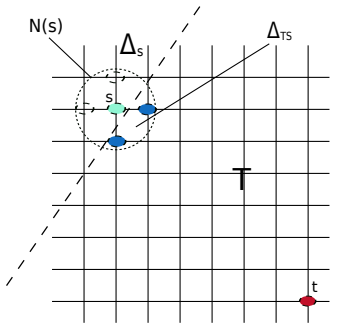

Figure 4: $\mathcal{N}(s) \subset \Delta_S \cup T$

We observe that

$$\mathcal{N}(s) = (\mathcal{N}(s) \backslash \Delta_S) \cup (\mathcal{N}(s) \cap \Delta_S) = \Delta_{TS} \cup (\Delta_S \cap \mathcal{N}(s)) \quad \text{and} \quad \Delta_{TS} \cap (\Delta_S \cap \mathcal{N}(s)) = \emptyset$$

and since $|\mathcal{N}(s)| = |R|$ we have

$$|R| = |\Delta_{TS}| + |\Delta_S \cap \mathcal{N}(s)|.$$

Therefore,

$$|\Delta_S| \geq |\Delta_S \cap \mathcal{N}(s)| = |R| - |\Delta_{TS}| = |R| - \ell. \tag{10}$$

Therefore

$$\begin{aligned} C(S) &\geq C_s + C_v(|\Delta_S| - 1) \\ &\overset{(a)}{=} C_v|\Delta_S| \\ &\geq C_v(|R| - \ell) \\ &\geq C_v(|R| - 1) \end{aligned} \tag{11}$$

since $\ell \geq 1$ and (a) comes from the hypothesis $C_s = C_v$.

When $\mathcal{N}(s) \not\subset \Delta_S \cup T$, see Figure 5: the fact that $\mathcal{N}(s) \not\subset \Delta_S \cup T$ corresponds exactly to the conditions of Lemma 1. Applying Lemma 1, we have: $|\Delta_S| \geq |R| - 1$. Therefore, we have:

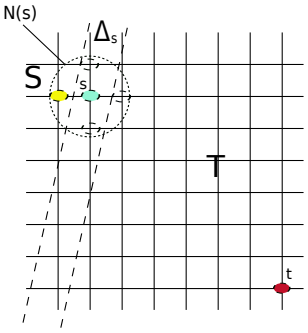

Figure 5: $\mathcal{N}(s) \not\subset \Delta_S \cup T$

$$\begin{aligned} C(S) &\geq C_s + C_v(|\Delta_S| - 1) \\ &= C_v|\Delta_S| \\ &\geq C_v(|R| - 1). \end{aligned} \tag{12}$$

This means that in either case we have

$$C(S) \geq C_v(|R| - 1). \tag{13}$$

Now, we consider the case far from the source : $s \notin \Delta_S$. The capacity of the cut is

$$C(S) = \sum_{u \in \Delta_S} C_u = C_v|\Delta_S|. \tag{14}$$

Using Lemma 1, we have

$$|\Delta_S| \geq |R| - 1, \tag{15}$$

therefore,

$$C(S) \geq C_v|\Delta_S| \geq C_v(|R| - 1). \tag{16}$$

If we consider the cut $(\mathcal{V} \setminus \{t\}, t)$ since

$$\Delta_S = \Delta_{\mathcal{V}} \setminus \{t\} = \{u \in S | \mathcal{N}(u) \cap \{t\} \neq \emptyset\} = R \setminus \{t\}$$

the capacity is

$$C(S) = \sum_{u \in \Delta_S} C_u = C_v|\Delta_S| = C_v(|R| - 1). \tag{17}$$

■

5 Combining an asynchronous low-duty-cycle mechanism and mobile routing

The principle of the mobile routing defined in [10] is that the nodes carry the packet when their heading vector is close to their bearing vector, see Figure 6. In what follows we summarize the mobile routing algorithm. Let us assume that node A is carrying a packet for node D. The velocity of node A is denoted by $\mathbf{v}(A)$.

- If node A is within range of node D, it transmits the packet to D directly; otherwise:
- if the relative bearing angle is smaller than θ_c, node A continues to carry the packet; otherwise:
- node A transmits the packet to a random mobile neighbor inside the cone of angle θ_e, with the bearing vector as the axis, and then takes no further interest in the packet.

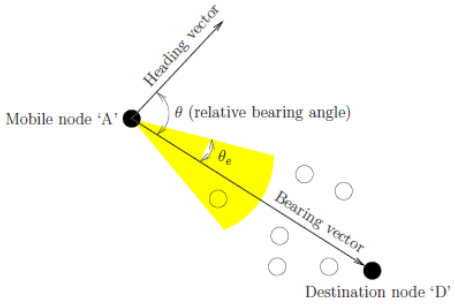

Figure 6: Mobile routing

We now suppose that the mobile nodes switch to sleep mode and wake up asynchronously. We assume that a node awakes after an exponentially distributed time of rate λ_{off} and remains on for one time unit. We also assume that if the relative bearing angle of a node carrying a packet is no longer smaller than θ_c this node wakes up and transmits the packet to a mobile random neighbor inside the cone of angle θ_e, with the bearing vector as the axis. This node switches back to the sleep mode as soon it has successfully transmitted its packet. The mobile routing defined in [10] will correctly operate if there is always an awake node in the cone of angle θ_e with a high probability. With the assumption of [10] it is reasonable to believe that the

awake nodes in the mobile network follow a Poisson point proccess of spatial rate $\frac{\lambda\lambda_{off}}{1+\lambda_{off}}$ if λ is the spatial density of the nodes. If we have n nodes in the network, the mean number of awake nodes is $\frac{n\lambda_{off}}{1+\lambda_{off}} = \alpha n$.

5.1 Mobile routing with a receiver-oriented low-duty-cycle MAC scheme

We can adopt a receiver-oriented low-duty-cycle MAC scheme: RI-MAC see Figure 7. The node holding the packet will wait for a beacon from another suitable node i.e. this node has a heading vector inside the cone of angle θ_e with the bearing vector as the axis. This means that the nodes which are awake have to send their positions and speed vectors in their beacon.

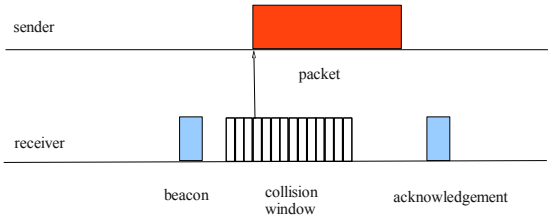

Figure 7: The Receiver-Initiated MAC protocol: RIMAC

5.2 Mobile routing with a sender-initiated low-duty-cycle MAC scheme

We can also adopt a sender-initiated low-duty-cycle MAC scheme such as X-MAC, see Figure 8. When the current node must relay its packet, it sends a preamble until a suitable node awakes and sends back an acknowledgement. In our mobile routing the potential relays can determine whether or not they satisfy the relaying condition (i.e. they have a heading vector within the cone of angle θ_e with the bearing vector as the axis).

If we use the results of [10], the achievable throughput of the mobile routing using the low-duty-cycle MAC is:

$$O\Big(\frac{\alpha n}{\log(\alpha n)\log(\log(\alpha n))}\Big)$$

and the delivery delay is:

$$0\Big(\frac{1}{v}\Big).$$

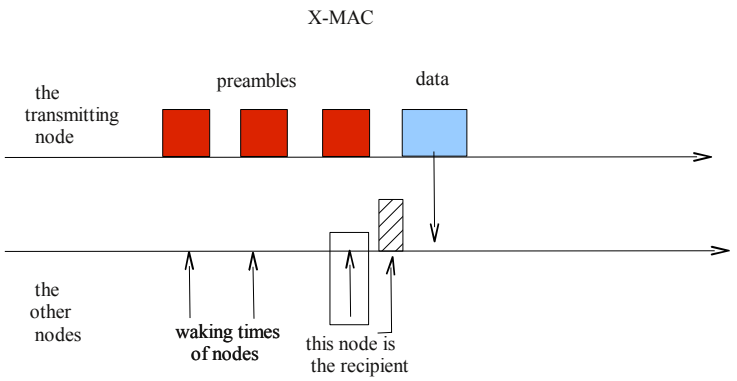

Figure 8: The Sender-Initiated MAC protocol: XMAC

6 Combining an energy saving mechanism based on a asynchronous approach and network coding

This combination is complex. Energy saving mechanisms based on a asynchronous approach are appropriate for a small throughput, but network coding needs a significant throughput. This combination is possible in principle but would generate very long delays for the packet delivery.

7 Combining mobile routing and network coding

Since mobile routing can be considered as a standard routing scheme it can be used for network coding. Moreover the mobility of the nodes and the mobile routing scheme create diverse options for the routes from the source to the destination. Thus network coding can be beneficial in this case.

8 Other approaches for mobile routing

8.1 Random walks with angle dependencies

The condition regarding random walks with angle independence can be relaxed and the result concerning the expected number of relay changes will still be valid. In other words, the angle independent random walk model can

be seen as a worst case compared to realistic mobility models. If the mobile relays move like cars in an urban area, then we can expect that their mobility model will significantly depart from the random walk. Indeed cars move toward physical destinations and in their journey along the roads toward their destination, their heading after each turn is positively correlated with the heading before the turn. This implies that the probability that a relay change is needed after a turn is smaller than it would be under a random walk model, where headings before and after a turn are not correlated. Furthermore on a street, the headings are positively correlated (consider Manhattan one-way streets) and in this case a relay change due to a hnad over will have more chance of arriving at a relay with good heading (one half instead of θ_c/π). Again this would lead to fewer relay changes due to hand over.

8.2 Restricting the possible angles

In the model of [10] the angles of the trajectory with a given direction can be chosen in the interval $[-\pi, \pi]$. Instead of selecting the turning angle uniformly in the interval $[0, 2\pi[$, the motion vector angle can be non uniformly distributed in the interval $[0, 2\pi[$, or even be constrained to a discrete set of values as for example the Manhattan street configuration *e.g.* $\{0, \pi/2, \pi, 3\pi/2\}$. The scheme is expected to work, although θ_c should be larger than the largest gap in the angle spectrum.

8.3 Non flat, non simply connected network map

The protocol can be adapted when the region is a connected but non-convex space (i.e. the region may have holes like lakes and parks). But in this case each node needs to have knowledge about the cartography of the region. In this scenario, the bearing angle will be replaced by the angle between the current motion vector of the mobile relay and the tangent of the geodesic path to the destination. This clearly requires a deeper knowledge of the cartography of the area. Each relay node computes its geodesic path to the destination and derive its relative bearing angle and emission cone accordingly. With this modification the routing scheme described in [10] remains valid.

8.4 Turn rates dependent on density n

The result still holds if we assume that the turn rate τ depends on n and $\tau = \tau_n = O(\log n)$. In this case, the mobility model would correspond even better to the realistic mobility of an urban area. Indeed the trajectories of cars should be fractal or self-similar, showing more frequent turns when cars

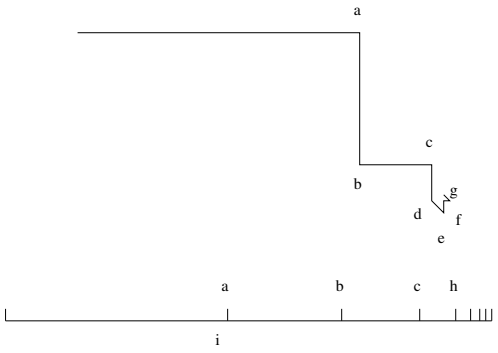

Figure 9: Illustration of self-similar trajectories in urban areas.

are close to their physical destination (different from packet destination) or when leaving their parking lot. In this case, the overall turn rate tends to be in $O(\log n)$ with a coefficient depending on the Hurst parameter of the trajectory. This would lead to the same estimate of $O(\log n)$ relay changes per packet.

Figure 9 illustrates a self-similar trajectory in an urban area. It shows a two-dimensional trajectory (upper half) and its distance traveled (lower half). The successive turns are indicated by $T_1, \ldots, T_7$. The trajectory after any turn T_i looks like a reduced copy of the original trajectory. The CRB scheme may need some adaptation to cope with some unusual street configurations, *e.g.*, to replace the Cartesian distance by the Manhattan distance in the street map.

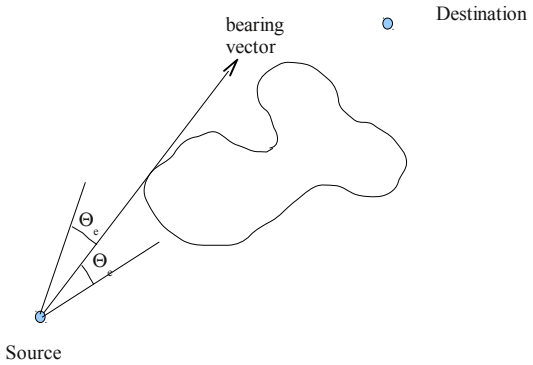

Figure 10: Mobile routing with holes

8.5 Fixed nodes as relays

In this approach,we propose to use fixed nodes as relays. For instance, these nodes can be on a grid. With such nodes we can assume a lower density of mobile nodes. If the cone of angle θ_e in the direction of the destination does not contain any nodes, then the packet must be sent to the fixed neighbor which is the closest to the destination. This fixed node must emit this packet immediately to its destination node or to any mobile relay in the neighborhood. Note that this will also help increase the connectivity of the network. Indeed, we expect that the global capacity rises to

$$O\Big(\frac{\alpha n}{\log(\alpha n)}\Big)$$

omitting in the denominator the factor $\log(\log(\alpha n))$ which was before needed to insure that each relay change was successful. This factor can now be removed, since the fixed relay nodes will store and forward the packet toward the destination until a suitable mobile node can receive and carry the packet. The average number of fixed relays between two mobile relay changes is expected to be finite, therefore the average number of overall relays remains $O(\log n)$.

8.6 Open questions

Are the results of [10] still valid if we have a general law for the duration of the straight movement?

9 Optimizing an asynchronous low-duty-cycle mechanism

9.1 Opportunistic Routing based on the Sender-Initiated X-MAC Protocol

In this section, we firstly present the main details about the LPL-based `X-MAC` protocol and its variant implemented in the UPMA package [11]. Then, we describe our new opportunistic cross-layer scheme based on this MAC protocol principle.

9.1.1 X-MAC: a Short Preamble MAC Protocol

Typically, asynchronous low duty-cycle MAC protocols are sender-initiated. This means that the sender starts off the communication process. To do

that, these MAC protocols rely on the Low Power Listening technique (LPL-based) to maintain the receiver awake as they have independent schedules. Thus, the sensor nodes periodically and asynchronously (depending on the sleeping period) check the channel to detect if they are concerned by a data transfer. When a sender wants to request a receiver for data transmission, it disseminates a preamble (which can be long or short and is closely dependent on sleeping period length) as a wake-up signal. When the receiver activates its transceiver and detects the preamble, it keeps its transceiver on and waits for the sender's data. Many protocols fall within this category, especially `B-MAC` [14] and `X-MAC` [7].

When `B-MAC` is used, a node that has data to send transmits a long preamble whose duration exceeds the sleeping period length. Usually, the preamble does not contain any information about the destination. Then, all the nodes that detect the preamble have to maintain their activity at least until the end of the preamble dissemination. The data is then transferred and only the receiver stays awake. Such an approach is efficient in terms of delivery with light traffic conditions in terms of delivery. However, the one-hop delays are substantial and estimated to be as long as the sleeping period. This fact also induces additional energy consumption in a multi-hop communication scenario. To overcome this situation, the short preamble-based `X-MAC` protocol was proposed.

In this paper, we make use of the principle of the `X-MAC` protocol. As we have stated, `X-MAC` swaps the long preamble for a strobed one that consists of a sequence of short preambles that precede data transmission, as depicted in Figure 11. The strobes contain the address of the destination which allows unconcerned neighbors to toggle their transceivers to off. Moreover, it enables the intended receiver to answer by an early acknowledgement so that the sender suspends the preamble dissemination and initiates data transmission. Operating in this way is a major factor in reducing delays and saving energy.

The UPMA (Unified Power Management Architecture for wireless sensor networks) package implements a variation of `X-MAC` in TinyOS. The specific feature here is that the strobed preamble is replaced by a chain of data frame transmissions, as illustrated in Figure 12. This strategy simplifies implementation and helps a sender to ensure that the data has been successfully received from the ACK sent by the receiver. We are interested in this variation and we consider it as the cornerstone of our new opportunistic cross-layer scheme proposal. As a final note, we assume that the sender indicates its coordinates in each strobe it disseminates.

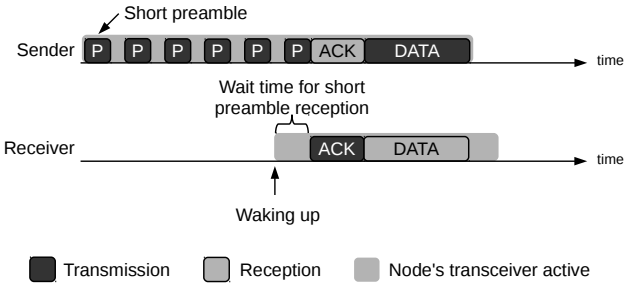

Figure 11: `X-MAC` communication scheme

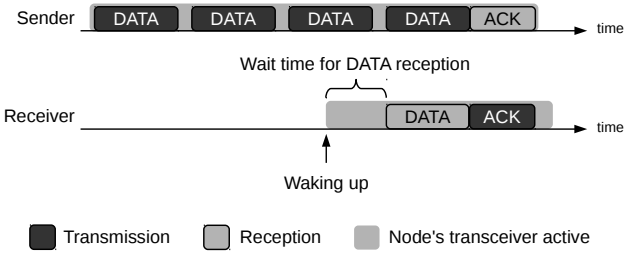

Figure 12: `X-MAC` communication scheme implemented in the UPMA package

9.1.2 Geographic opportunistic routing based on X-MAC

We design our opportunistic cross-layer scheme by using the `X-MAC` variant implemented in the UPMA package. So, when a node $\mathcal{N}$ has data to forward to the sink $\mathcal{S}$, it transmits strobed preambles where each strobe consists of a chain of data frames. A sender knows neither its neighbors nor their low duty-cycles. At the end of each strobe transmission, an election process starts and concerns all the neighbors that have detected the strobe. The next hop is designated if it satisfies a progression threshold toward the sink. We detail this point later. We propose to use a geographic opportunistic protocol as advanced in [6]. We assume that sensor nodes know their location and the location of the sink. In the GETRF project, the sensors are manually provided with these details when deployed. Thus, they are able to determine both their own position and that of the sink $\mathcal{S}$ [15]. To forward information, a sender, as mentioned, has to disseminate a chain of data frames as a short preamble. It can send several series of strobes interspersed by a gap. During this gap, the neighbors that have received data launch an election process to

identify the candidate that most reduces the distance toward the sink. The election process that we define takes the principle of the signalling bursts with logarithmic coding of the rank proposed in [9]. We present an example in Figure 13 in which four nodes compete for the next hop designation. It is important to highlight that this process is fully distributed and does not need an extra information exchange.

When a short preamble is broadcast, each neighbor $\mathcal{N}_i$ that receives the data checks if it is geographically further from the sink $\mathcal{S}$ than the sender $\mathcal{N}$ (as we assume that the coordinates of the sender are indicated in the data string). If such a condition is verified, it turns off its transceiver. Otherwise, it maintains its activity and declares itself as a candidate to be the next hop. At the end of the short preamble transmission, the candidates convert their distance to the sink into a binary code and compute its complement to obtain a K-bit code sequence. This sequence expresses the remaining distance to the sink and the closest candidate obtains the strongest sequence. If we consider the example of Figure 13, the binary sequence length K is set to 14 bits.

Afterwards, a competition takes place where potential candidates analyze their sequence, periodically exploit a bit starting from the most significant bit toward the lowest one and execute an action according to the bit value. If this value is **1**, the candidate sends a short frame and remains active. If it is **0**, it senses the medium. When activity is detected, the sensor concludes that there is at least one candidate which has a stronger sequence i.e. the candidate is closer to the sink and the sensor leaves the election competition by switching off its transceiver. By referring to Figure 13, nodes "C" and "D" are knocked out of the process as they conclude that there are other contenders that have stronger binary sequences. Otherwise, if no activity is detected, the sensor deduces that no other candidate has a stronger current bit and its keeps its transceiver on for the next part of the competition. After examining the K bits, it is possible to have more than one candidate remaining if the "winners" are the same distance from the sink $\mathcal{S}$. In order to designate one "winner", R additional random bits are concatenated to K bits and the same elimination process is applied. From the example in Figure 13, node "A" and node "B" are the remaining candidates after executing actions related to the K bits (they are the same distance from the sink). Nevertheless, node "A" has a stronger random sequence than node "B" (R is set to 3) and is declared the winner of the competition. As shown, this process is fully distributed and there is no need to exchange information between nodes.

The sender is able to decrypt the signaling burst activity and to discover the distance between the winner and the sink. According to the progression provided toward the destination, the sender decides whether it will send a new chain of data frames or not (so that the winner forwards data). As an

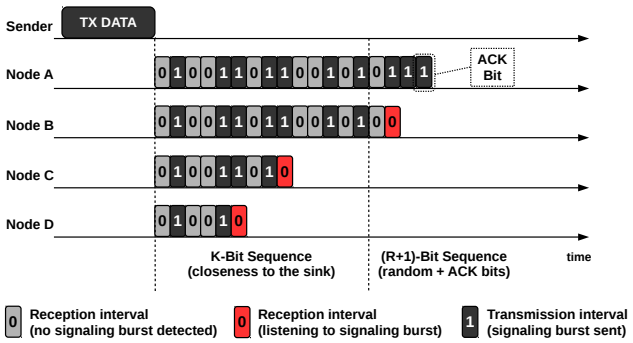

Figure 13: Burst mode: illustration of an election process

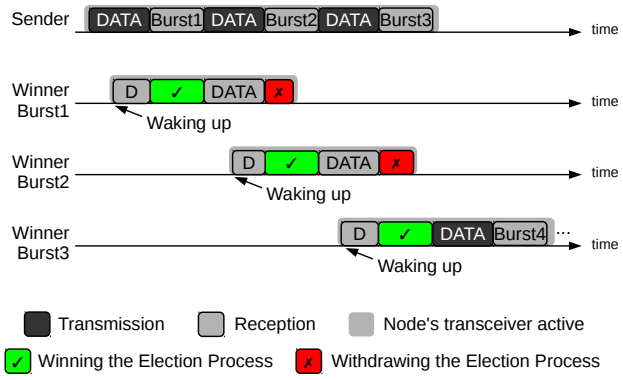

Figure 14: Next hop selection with the `X-MAC`-based scheme

example, the decision can be made whether the progression toward the sink is greater or not than some percentage of the communication range. If we apply this condition to the scenario illustrated in Figure 14, we can highlight that the winners of the two first election processes are not fulfilling the cited requirement. So, they do not disseminate data and maintain their transceiver active for the next election phase as they detect a retransmission of a new chain of data frames from the sender. Nevertheless, if the winner satisfies the geographical progression requirement (as does the third winner in the example), the sender stops its activity. The winner, in this case, interprets this "silence" as an acknowledgement of its status of next hop and can start disseminating data within its neighborhood.

It is worth noting some important details that concern data transmission. First, if the progression toward the sink is not satisfactory, the sender

disseminates a new chain of data frames. When the winner of the election process detects the sender's signal, it remains awake until the next election phase as finding a closer candidate is not straightforward. Otherwise, if the sender stops its activity, this means that the winner has provided a good progression and can forward data. Second, a sender can send short preambles as long as the sleeping period length if the progression is not satisfactory. When no winner provides the expected progression, the node elected after the last strobe is designated as the next hop. It is necessarily the closest neighbor to the sink as all intermediate winners are maintained active for the following election process. Finally, the election process is executed during the gap between two successive strobes. The duration of this gap must be shorter than the time during which a node remains active to check the activity of the medium. Hence, all the neighbors can be requested if the expected progression is not achieved.

9.2 Simulation results

9.2.1 Simulation Settings

To assess the effectiveness of our proposal, we implemented our discrete-event-based simulator with C++. The implementation considered opportunistic routing schemes based on `X-MAC`, `B-MAC` and `RI-MAC` [2] to compare their performance. It also took into consideration slotted CSMA/CA and a physical layer with respect to the standard specification of IEEE 802.15.4 [**?**]. The physical signal propagation was the two-ray-ground model.

We define a deployment area $\mathcal{A}$ as a square unit area[1]. Sensor nodes are deployed over $\mathcal{A}$ and their coordinates are determined according to a homogeneous Poisson Point Process with a density λ fixed at 4000. They are configured to operate in a low duty-cycle mode; a node activates its transceiver for one time unit[2] to check the network activity and switches to a sleeping state, if appropriate, whose duration is fixed to 100 time units. We assume that the all nodes have the same communication range throughout the network, denoted by $\mathcal{R}_{com}$ and that they use the `CC2420` chipset which performs the slotted CSMA of IEEE 802.15.4 [**?**]. We consider that an event occurs at $(0.1, 0.1)$ and that the sink $\mathcal{S}$ is located at $(0.9, 0.9)$ (i.e., diagonal). Each sensor that detects the monitored event generates data to deliver to the sink $\mathcal{S}$.

We compare the performance of our opportunistic routing scheme based

[1] A scaling factor can be applied to match the figures of a real deployment.

[2] Corresponding to 6.1 ms, the time needed to send 19 bytes with a transmisson rate of 250 $Kbps$

on X-MAC with the schemes built either on B-MAC or RI-MAC. We study the following:

- The probability of packet delivery, denoted by P_{path},
- The average number of hops per path, denoted by $\mathcal{N}_{hop}$,
- The average packet delay per hop, denoted by T_{hop},
- The average end-to-end packet delay, denoted by T_{tot}.

We compute each of the above values by evolving $\mathcal{R}_{com}$ from 0.01 to 0.089 over an average of 50 simulations. The error bars correspond to a confidence level of 95%. In X-MAC- and B-MAC-based schemes, the election process duration is fixed to 0.02 time units when setting K and R respectively to 14 and 3 bits. The transmission of the long preamble lasts a little over 100 time units for the B-MAC-based scheme and the chains of data frames (the length of one data frame is fixed at 0.7 time units) are sent for a period that slightly exceeds 10 time units (1/10 of a long preamble). With the X-MAC-based technique, the winner of an election process is designated as the next hop if it ensures a geographical progression toward the sink that is greater than some percentage $\mathcal{P}$ of the communication range. We vary $\mathcal{P}$ from 20% to 80% with increments of 20%. Finally, the length of an acknowledgement is 0.3 time units and the length of a beacon is set to 0.1 for RI-MAC.

In Figures 15, 16, 17 and 18, we depict all the results for our X-MAC-based scheme and with different values for $\mathcal{P}$. For example, by "XMAC-based-40", we mean that the result is obtained for the cross-layer scheme using X-MAC and applying the progression condition of 40% to designate a signaling burst winner as the next hop.

9.2.2 Performance Evaluation

As mentioned in the previous subsection, we focus on some performance metrics. As expected, the opportunistic routing cross-layer scheme based on B-MAC ensures a robust delivery, especially with high communication ranges and it considerably reduces the number of hops. Nevertheless, although the delays (hop and end-to-end) do not represent a hard constraint in rare event monitoring WSNs, they may well be substantial due to the use of long preambles. The scheme using RI-MAC satisfies delay constraints but its robustness can be affected by collisions caused by beacons. By building our scheme on X-MAC, our proposal aims to offer the same robustness as B-MAC while reducing the delays to a minimum. In the following, we compare the three schemes

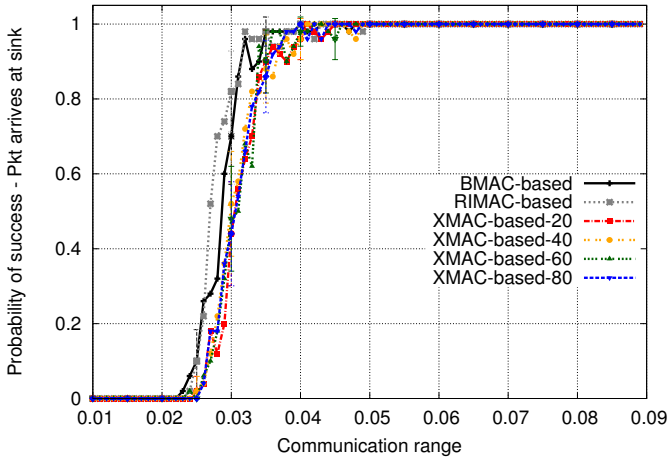

Figure 15: Probability of packet delivery P_{path}

according to the parameters cited above.

Probability of packet delivery P_{path}

Figure 15 depicts the evolution of the average packet delivery probability with an increasing communication range. For schemes based on `X-MAC` and `B-MAC`, we find the same performances. Even if they use different approaches to request the sender's neighbors, they share the LPL scheme. Such a technique avoids collisions (which is not the case for the `RI-MAC`-based scheme which only remains efficient due to retransmissions) with a well-set contention backoff and ensures the same performance whichever scheme is used. It might be expected that a route would be longer when `X-MAC` is used, but, as LPL is applied at each hop, this length does not impact on the packet delivery rate.

Average number of hops per path $\mathcal{N}_{hop}$

The average number of hops per path, for each scheme, is represented in Figure 16. The opportunistic routing cross-layer scheme based on `B-MAC` enables us to obtain shorter paths as, at each hop, it identifies the closest neighbor to the sink. The paths are slightly longer when `X-MAC` is used. This can be explained by the condition that must be verified after each election process; a winner is selected as the next hop if it provides a geographical progression toward the sink which represents at least a percentage $\mathcal{P}$ of the communication range. This favours the selection of one of the closest neighbors even if it is not necessarily the "best" one. That said, looking for the closest neighbor does not result in a great difference between the two schemes' performance. As our proposal guarantees a same order of magnitude regarding the number

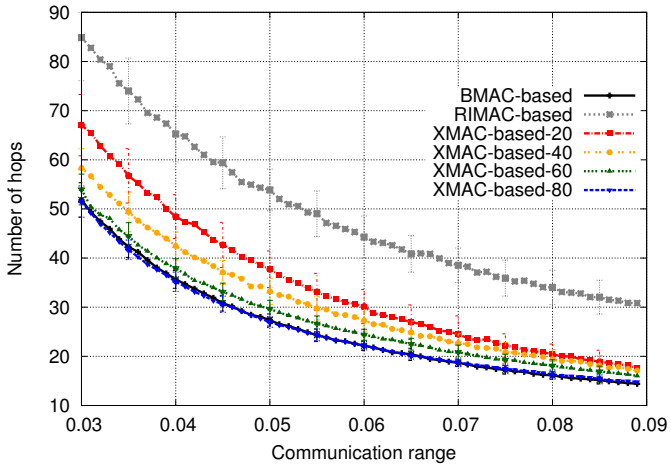

Figure 16: Average number of hops per path $\mathcal{N}_{hop}$

of hops per path (especially with high communication ranges), it reflects a non-negligible assurance of reliability and robustness against failures. We also remark that the higher $\mathcal{P}$ is, the closer the average number of hops for both schemes becomes. This is not suprising as by increasing $\mathcal{P}$, the scheme tends to pick the closest candidate to the sink. The `RI-MAC`-based scheme, being highly opportunistic gives the longest paths as it does not use a selection process (the first candidate that sends a beacon and which is closer to the sink than the current node is selected).

Average packet delay per hop T_{hop}

We focus here on the delay per hop obtained for all the schemes and based on the first packet that reaches the destination. Figure 17 shows the performance of each solution. The `B-MAC`-based scheme registers a delay per hop on the basis of the sleeping period (around 100 time units). Finding the "best" neighbor for a sender requires using a long preamble (slightly longer than the sleeping period) to ensure that this neighbor is awake to receive data. However, with the `X-MAC`-based proposal, a sender would potentially have a subset of neighbors that satisfy the progression toward the sink condition. Disseminating chains of data frames interspersed by election processes would make it possible to designate the next hop earlier than `B-MAC`. When no neighbor satisfies the progression condition, the scheme designates the closest candidate to the sink. The per hop delay with our new proposal is significantly shorter and the smaller the percentage $\mathcal{P}$ is, the greater the difference in performance becomes. Moreover, increasing the communication range results in a larger subset of eligible neighbors for a sender and sub-

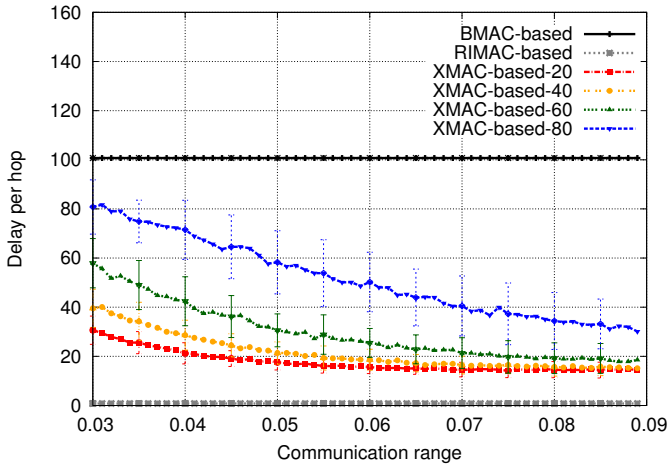

Figure 17: Average packet delay per hop T_{hop}

sequently shorter delays to designate the next hop. More flexible selection constraints allow the `X-MAC`-based scheme to be more opportunistic and it provides results comarable with those of `RI-MAC`.

Average end-to-end packet delay T_{tot}

Now, we focus on the end-to-end delay for the three opportunistic routing cross-layer schemes based on the first packet that arrives at the sink. The results are presented in Figure 18. As expected, the end-to-end delays are significantly shorter with our new scheme compared to the `B-MAC`-based one and get closer to the delays given by the `RI-MAC`-based scheme with a small percentage $\mathcal{P}$. This result is confirmed by the analyses presented in Sections 9.2.2 and 9.2.2. Even if the `X-MAC`-based technique builds longer routes than the `B-MAC`-based one, the number of additional hops is small. On the other hand, the considerable gain in per hop delays obtained by the new proposal is substantial, as can be seen in Figure 17.

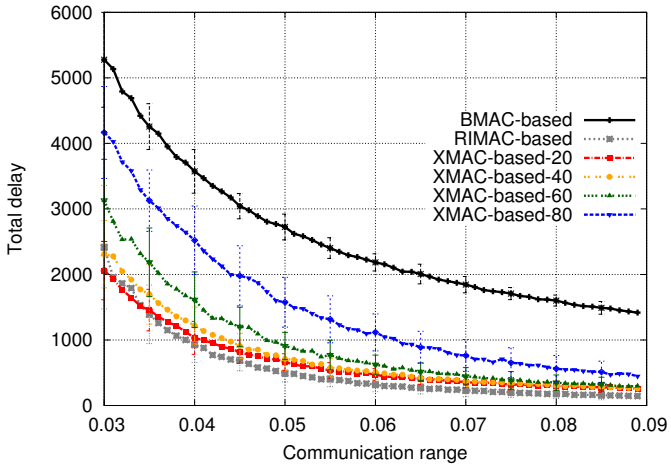

Figure 18: Average end-to-end packet delay T_{tot}

10 Conclusion

The combination of the techniques that have been designed in the GETRF project are summarized in the table below.

	Synchronous	Asynchronous	Network coding	Mobile routing
Synchronous		I	I	D
Asynchronous	I		D	I
Network coding	I	D		I
Mobile routing	D	I	I	

where I stands for interesting and easy, and D stands for difficult and usually not very useful. We observe that the four techniques developed for Getrf are quite complementary.

We have also shown that the asynchronous techniques can be optimized. The mobile routing can be adapted to other assumptions than the "classical" ones where the nodes move randomly.

References

[1] Rudolf Ahlswede, Ning Cai, Shuo-Yen Robert Li, and Raymond W. Yeung. Network information flow. *IEEE TRANSACTIONS ON INFORMATION THEORY*, 46(4):1204–1216, 2000.

[2] N. Aitsaadi, B. Blaszczyszyn, and P. Mühlethaler. Performance of opportunistic routing in low duty-cycle wireless sensor networks. In *Wireless Days (WD), 2012 IFIP*, 2012.

[3] Nadjib Aitsaadi, Paul Muhlethaler, and Mohamed-Haykel Zayani. Getrf delivrable 4: Architecture of low duty-cycle mechanisms, February 2014.

[4] Ichrak Amdouni, Cedric Adjih, and Pascale Minet. Getrf delivrable 2: Joint routing and tdma-based scheduling to minimize delays in grid wireless sensor networks, February 2014.

[5] Ichrak Amdouni, Cedric Adjih, and Pascale Minet. Joint routing and stdma-based scheduling to minimize delays in grid wireless sensor networks. *CoRR*, abs/1402.7017, 2014.

[6] F. Baccelli, B. Blaszczyszyn, and P. Mühlethaler. Time-space opportunistic routing in wireless ad hoc networks: Algorithms and performance optimization by stochastic geometry. In *Computer Journal, vol. 53, no. 5*, 2010.

[7] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems, ser. SenSys 06. New York, NY, USA: ACM, 2006.*

[8] Supratim Deb, Michelle Effros, Tracey Ho, David R. Karger, Ralf Koetter, Desmond S. Lun, Muriel Medard, and Niranjan Ratnakar. Network coding for wireless applications: A brief tutorial. In *International Workshop on Wireless Ad-hoc Networks (IWWAN)*, 2005.

[9] P. Jacquet, P. Minet, P. Mühlethaler, and N. Rivierre. Priority and collision detection with active signaling - the channel access mechanism of HiPERLAN. In *Wirel. Pers. Communication, vol. 4, no. 1, pp. 11–25*, Jan. 1997.

[10] P. Jacquet and Alonso Silva. Deliverable 5: Improving throughput in multihop networks using node mobility, February 2014.

[11] K. Klues, G. Hackmann, O. Chipara, and C. Lu. A component-based architecture for power-efficient media access control in wireless sensor networks. In *Proceedings of the 5th international conference on Embedded networked sensor systems, ser. SenSys 07. New York, NY, USA: ACM, 2007.*

[12] In kwon Lee, Myung soo Kim, Gershon Elber, and Tu Wien. The Minkowski sum of 2d curved objects, 1998.

[13] Antonia M. Masucci and Cedric Adjih. Efficiency of broadcast with network coding in wireless networks. *INRIA RR-8490*, Feb. 2014.

[14] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems, ser. SenSys 04. New York, NY, USA: ACM, 2004.*

[15] V. Zeimpekis, G. M. Giaglis, and G. Lekakos. A taxonomy of indoor and outdoor positioning techniques for mobile location services. In *SIGecom Exchanges, vol. 3, no. 4, pp. 19–27*, 2003.