GETRF delivrable 4:

Architecture of low duty-cycle mechanisms

Nadjib Aitsaadi, Paul Mühlethaler, Mohamed-Haykel Zayani
Hipercom Project-Team
Inria Paris-Rocquencourt

February 2014

Contents

1 Introduction **4**

2 Principles of the architecture **5**

2.1 Routing protocols 5

2.2 MAC rendezvous protocols 6

2.2.1 Sender-oriented rendezvous algorithm 6

2.2.2 Receiver-oriented rendezvous algorithms 9

2.3 Building a low duty-cycle protocol in multihop wireless networks 9

3 Description of our contributions **10**

3.1 Receiver-oriented proposal 10

3.1.1 Description 10

3.1.2 Analytical model 12

3.1.3 Simulation results 15

3.2 Sender-oriented proposal 24

3.2.1 Description 24

3.2.2 Analytical model 27

3.2.3 Simulation results 28

3.3 Comparison and discussion 32

4 Conclusion **34**

1 Introduction

A Wireless Sensor Network (WSN) is composed of sensor nodes deployed within an area to monitor predefined phenomena (e.g. temperature, humidity, movement, etc.). Once something is detected, a packet is generated and sent to a specific node, named the base station or sink, which then informs the control room.

The main challenge is to convey a packet from any sensor node to the sink while optimizing the energy consumed and ensuring the required quality of service in terms of packet delivery and forwarding delay. In the literature, several routing protocols have been proposed [20] [3] which can be classified in three main groups. The first group includes reactive protocols such as AODV [17]. A path is established on demand by flooding a route-request packet and backtracking a route-reply packet. The second group includes proactive protocols such as OLSR [2] and the Dijkstra shortest path protocol [8]. A routing table is maintained and updated periodically by exchanging control packets. Finally, the third group contains opportunistic protocols [10] in which each hop depends on local neighbours' conditions. Generally this is done with a self-election process of the relay.

Opportunistic routing has recently emerged and demonstrated both simplicity and scalability. It is worth noting that a node does not use a routing table. This kind of routing exploits the broadcast nature of wireless communication: when a node transmits a packet, all its active neighbours can receive it. However, only one neighbour must forward the packet. There are various means to build such a mechanism. It may involve a self-election process or the transmitter can select the next hop according to information about the nodes in its neighbourhood, see [5].

To maximise the lifetime of the network, whenever possible nodes' transceivers are periodically and asynchronously switched off; the WSN operates with a low duty-cycle. However, we have to ensure that the network remains connected. This property can be obtained by calibrating the communication range or density of deployed sensors. We also have to guarantee that the packet delivery delay is acceptable, which leads to a constraint on the length of time that the sensors are turned off.

In this deliverable of task 4 (mécanismes d'économie d'énergie), we study the performance of opportunistic routing and how it can be used in WSNs operating in a low duty-cycle mode. *In contrast to the previous studies, "opportunism" is used to take advantage of awake nodes and not to benefit from all the receptions in the neighbourhood of a node such as in [4] [12].*

The main contributions of this task are as follows. First, we describe the principles of low duty WSN architectures. Then, we summarize the main

WSN asynchronous MAC low duty-cycle schemes found in the literature; they are mainly two schemes, the first technique is sender-oriented and the second technique is receiver oriented. We explain how MAC low duty-cycle schemes and routing algorithms can be combined in cross-layer approach. We propose two low duty WSN architectures: one is based on `B-MAC` (Sender-oriented), the second on `RI-MAC` (Receiver-oriented). The routing is an opportunistic routing algorithm. We will depict several variants for each cross-layer scheme. Depending on the transmission range, we evaluate the average i) packet delivery probability, ii) delay at each hop, iii) number of hops to reach the sink and iv) end-to-end delivery delay. We compare the simulation results, for each scheme, with those obtained by a simple analytical model using a Poisson point process. We study the energy consumed when no event is detected and the additional energy to convey a packet to the sink. We show that the gain in energy obtained with a low duty-cycle is very significant.

2 Principles of the architecture

There are two main algorithms in the sensor network architecture. The first is a synchronisation algorithm which allows the nodes which most often sleep to be awake at the same time and exchange packets. The second algorithm is a routing algorithm which route the packets towards a (the) sink(s) of the network.

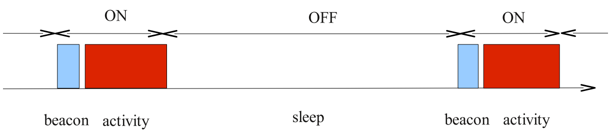

Figure 1: Duty-cycle of sensor's transceiver

2.1 Routing protocols

There are two main sets of mechanisms to route packets in a WSN.

The first set of mechanisms consists in sending topological packets to build the network topology. The topological packets can be sent proactively or reactively. When we use proactive protocols, different kinds of routing techniques may be used such as link-state or distance vector. These mechanisms are useful if the entire topology must be known. If we only need to know the route to the sink we can also use a reactive tree-based protocol.

An example of a reactive protocol is given in [17] whereas an example of a proactive protocol is in [2].

In the second set of protocols we assume that the nodes know their own location and the location of the destination node. We will use geographic protocols to route the packets to their destination and no topological packets will be used to build routing tables. For instance, we can use a greedy algorithm [13] to route data to the destination; the packets are forwarded to a node which is at a smaller distance from the sink. This scheme is recursively applied until the packet is within transmission range of the final destination. Then the packet is simply sent to the destination. To perform such a scheme, the relay must know the position of its neighbors.

We can also use geographic opportunistic routing. The main idea is that the relay does not directly select the next hop. This next hop will be determined by the potential relays themselves. This can be done for instance by using different backoffs. When a potential relay receive the packet it calculates the progression towards the final destination and computes its backoff as a function of this progression: the greater this progression is, the smaller the backoff will be. Doing so will favor the relay which gives the greatest progression towards the final destination. This selection can also be performed using signalling bursts in the acknowledgement packets to elect a winner. This winner will be the relay which gives the greatest progression towards the final destination.

2.2 MAC rendezvous protocols

How the radio medium is shared is one of the main problems of WSNs. In fact, the performance of the network in terms of bandwidth, delays, etc. strongly depends on the efficiency of the MAC algorithm. It is obvious that the percentage of collisions directly impact the network performance. This task is more complex when sensor nodes asynchronously schedule their on-off transceiver activity. To be able to communicate nodes must set up "rendezvous".

Hereafter, we summarize the main asynchronous MAC rendezvous protocols which can be classified in two groups: sender-oriented rendezvous algorithms and receiver-oriented rendezvous algorithms.

2.2.1 Sender-oriented rendezvous algorithm

The main idea of sender-oriented rendezvous algorithms is that the source transmits frames while unaware of the state of the receiver (sleep or awake).

In the following we describe the most relevant protocols within this group.

In `B-MAC` [18] each node periodically wakes up to check if there is any activity currently on the wireless channel. If so, the node remains active to receive a possible incoming packet. When a transmitter has a packet to send, it transmits a preamble whose duration exceeds that of the receivers sleep interval. Doing so allows each node to wake up or sleep based on its own schedule while the transmitter can transmit its packet to its receiver, see Figure 2. The `B-MAC` protocol is very energy efficient under light traffic conditions because a node spends only a very short period of time checking channel activity at each scheduled wakeup time. However, a node executing `B-MAC` may wake up and remain awake due to channel activity, only to, in the end, receive one or more data frames actually destined for other nodes. Another interesting aspect of `B-MAC` is that the nodes do not transmit at all except when a packet must be sent. This property can be very useful when the sensor network is a surveillance network.

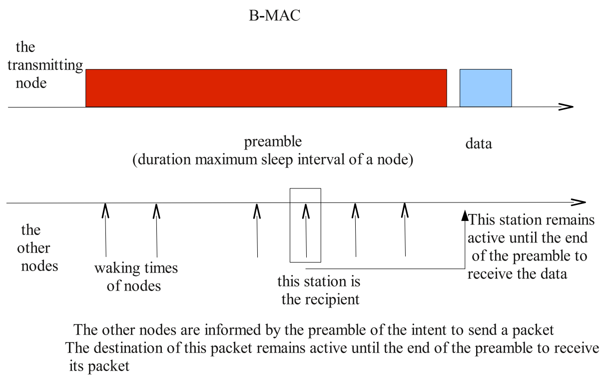

Figure 2: The BMAC protocol

`X-MAC` [7] solves this overhearing problem in `B-MAC` by using a strobed preamble that consists of a sequence of short preambles prior to data transmission, as illustrated in Figure 3. The target address is embedded in each short preamble, which not only allows unconcerned nodes to go to sleep immediately but also allows the intended receiver to send an early ACK to the sender so that the sender stops preamble transmission and starts transmitting the data frame immediately. In this way, `X-MAC` saves energy by avoiding overhearing while reducing latency almost by half on average. After receiving a data frame, a receiver in `X-MAC` stays awake for a duration equal to the maximum backoff window size to allow queued packets to be transmitted immediately.

The `UPMA` (Unified Power Management Architecture for wireless wensor

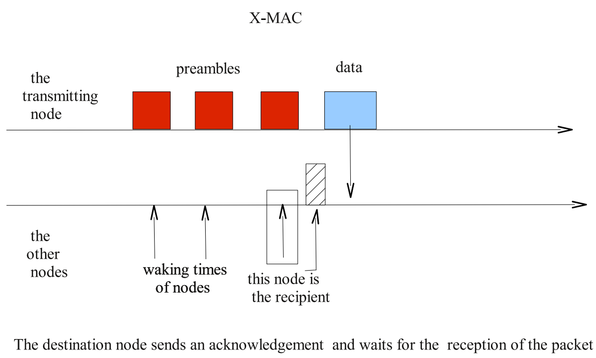

Figure 3: The X-MAC protocol

networks) package [15] implements a variation of X-MAC in TinyOS, in which the data frame itself is used as the short preamble, as illustrated in Figure 4. This strategy simplifies implementation and helps a sender to determine whether the DATA is successfully delivered from the ACK of the receiver.

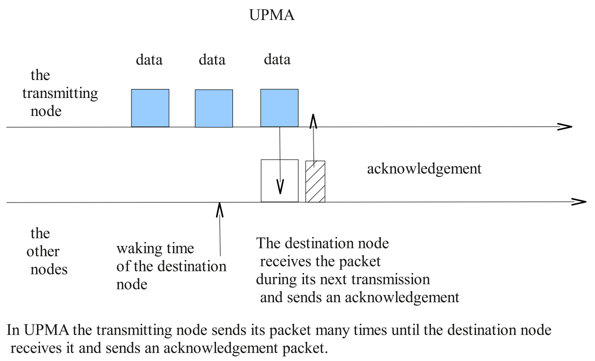

Figure 4: The UPMA protocol

WiseMAC [9] uses a scheme similar to B-MAC however additional information is used to synchronize a node with its direct neighbors. This information can be implemented using acknowledgement packets. When the transmitter sends its packet after the long preamble which has woken up all its neighbors, the recipient node sends an acknowledgement packet. This packet also contains the remaining time until its next sampling time. With this information, and taking possible clock drifts into account, the sender of its next data frame to this receiver estimates when the receiver will next wake up, and starts transmitting its preamble just before then.

Another interesting aspect of B-MAC and X-MAC is that the nodes do not transmit at all except when a packet has to be sent. This property can be very useful when the sensor network is a surveillance network.

2.2.2 Receiver-oriented rendezvous algorithms

In contrast to the previous subsection, receiver-oriented rendezvous algorithms use a technique where the transmitter waits for the receiver to wake up before sending it the packet. This is the Receiver-Initiated MAC [19] protocol. In this approach, when a node wakes up and switches on the transceiver, it sends a beacon to let its neighbours know that its transceiver is on. In order to avoid collisions, the awake node indicates in its beacon the collision window that must be used for the nodes which want to communicate with it. Thus, the network operates as shown in Figure 5; the sender uses CSMA and will not transmit if it senses a prior transmission.

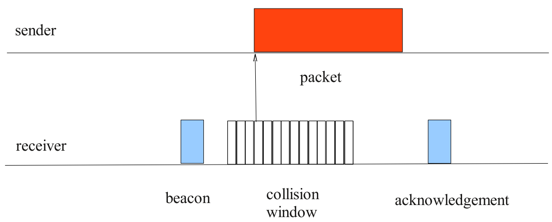

Figure 5: The `RI-MAC` protocol

2.3 Building a low duty-cycle protocol in multihop wireless networks

A low duty-cycle protocol can be obtained by combining a routing algorithm with a MAC rendezvous algorithm.

If we use a reactive or proactive routing algorithm, it is necessary when the routing algorithm is running that the MAC rendezvous is turned off. When the network has acquired the topology, the MAC rendezvous is turned off. We assume that there is no mobility and that the links are stable. When the MAC rendezvous is turned on, the routing tables can be used to determine the next hop of the packet and to forward it, see Figure 6. The MAC rendezvous is simply used to perform the binding between the relay and its next hop. This operation is also convenient for geographic routing.

If we use a geographic routing scheme, the MAC rendezvous can be turned off as is done with reactive or proactive routing algorithms. In this case the nodes can learn the topology by receiving for instance “hello” messages. A node will learn the position of its neighbor and use this information to forward a packet towards the destination. For instance a node can use a “greedy” algorithm to select the next hop among its neighbors.

The MAC rendezvous scheme can also always be turned on if we use a geographic routing scheme combined with an opportunistic forwarding scheme. In this case, the next hop will be selected on the fly during the rendezvous. When a node wakes up, the relay node can select this node to forward the packet if it offers a suitable progression towards the final destination. In this case the time interval during which the MAC rendezvous is turned off is no longer necessary. However, this operating mode requires that location information be available in the network.

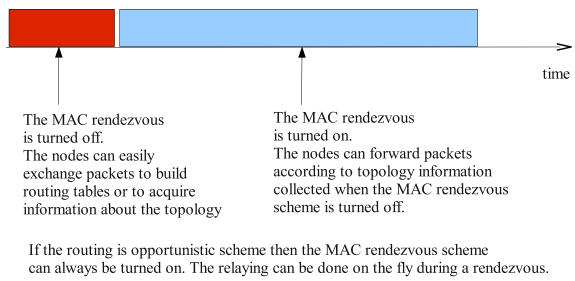

Figure 6: Rendezvous and routing schemes

3 Description of our contributions

In task 4 we propose two main contributions. The first one is based a receiver-oriented approach (`RI-MAC`). The second one uses the sender-oriented approach (`X-MAC`). However, we use a similar routing approach with an opportunist “flavor”. It is worth noting that our our two contributions are totally asynchronous.

3.1 Receiver-oriented proposal

3.1.1 Description

We build our proposal on the `RI-MAC` protocol that we presented above in subsection 2.2.2. `RI-MAC` must be complemented at the routing level to ensure the connectivity of the network. We combine the `RI-MAC` protocol which operates with nodes in a low duty-cycle with an opportunistic routing scheme.

We assume that the sensors are equipped with a GPS or apply geo-localization methods proposed in the literature [11, 16, 21]. Then, each node communicates its position to its neighbors and the sink node floods its coordinates over the whole network. This location can also be written in the sensor nodes when the network is set up.

In that sense our approach is a cross-layer scheme. When a sensor wants to transmit a packet, it waits for the next awake sensor node and checks whether or not it reduces the remaining distance to the sink. If so, the packet is sent to this node just at the beginning of its active period. We consider three variants of the geographic opportunistic routing protocol, which conveys a packet as follows. First, each sensor node determines its geographic position.

To forward a packet from a node $\mathcal{N}_i$ to the sink $\mathcal{S}$, the next hop can be selected according to the variant of opportunistic routing as explained below.

1. **Basic-opportunistic**: The next hop $\mathcal{N}_i$ is selected as the next active neighbor that decreases the remaining distance to the sink $\mathcal{S}$. If the remaining distance between the sink and the next awake node is not smaller than the distance between the sink and the current sensor node, the packet waits in $\mathcal{N}_i$ for possibly an unlimited period. This variant is greedy, the distance between the sink and the next sensor relay never increases.

2. **Opportunistic with delay**: This is similar to the basic-opportunistic variant except that in each hop a packet can wait for, at most, a predefined and fixed duration. If a packet is not transmitted during this period, it will be discarded. In other words, a packet only waits for a limited amount of time in each node.

3. **Opportunistic with backtracking**: This is similar to the opportunistic with delay variant except that it is not greedy. If a next hop minimising the remaining distance is not found during the maximum waiting time, the packet is not discarded and the protocol allows the packet to initially move further away from the sink until a path to the destination is found. Moreover, a node $\mathcal{N}_i$ will be tagged as a forbidden hop in the future for the packet concerned (i.e. a packet cannot visit the node $\mathcal{N}_i$ again). We know that, in random networks, packets using greedy opportunistic routing can be blocked by holes. The opportunistic with backtracking protocol is a response to this problem.

4. The **Dijkstra** approach does not use a greedy routing but instead a shortest path routing. A node relaying a packet will wait for the beacon of the next node on the shortest path routing to send its packet.

When a sensor node has a packet to forward it does not turn off its transceiver until it has successfully transmitted the packet to the next hop towards the destination. In the next section, we study the performance of this

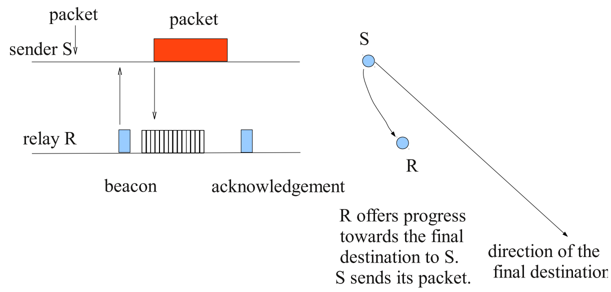

Figure 7: The `RI-MAC` protocol combined with opportunistic routing

protocol and an example of the execution of the above variants is illustrated in Figure 8. We observe that opportunistic with backtracking is the variant which is the closest to Dijkstra's protocol. This is because the backtracking allows the protocol to explore more routes than the basic-opportunistic or the opportunistic with delay protocols.

These protocols require only a very limited computation power and memory. The basic opportunistic protocol only has to compute the remaining distance to the sink when a beacon is received. There is no obligation for the node to keep this information. The opportunistic with delay only requires an additional timer to limit the stay of the packet in the node. The opportunistic with backtracking is little more complex. In fact each transmitting node needs to add its ID to the packet if it sends it back. In doing so, the packet cannot be sent back to the previous nodes. In the following section, we will see that this incurs a very limited overhead.

3.1.2 Analytical model

We consider a target deployment area denoted by $\mathcal{A}$. We assume that $\mathcal{A}$ is a square unit area[1]. Sensor nodes are deployed in $\mathcal{A}$ and their positions are the points of a homogeneous Poisson point process with density λ. We assume the same communication range throughout the network, denoted by $\mathcal{R}_{com}$. We consider a WSN with an asynchronously low duty-cycle. A sensor node's transceiver is active for one time unit and it sleeps for an exponentially distributed time with density λ_{off}, as illustrated in Figure 1.

We assume an infrequent event is being monitored. Only one packet can be transmitted within any sensor's neighborhood. Hence, we can ignore collisions between packets. In order to generate a long path, we assume that a packet is sent from a sensor $\mathcal{O}$ located at $(0.1, 0.1)$ to the sink node $\mathcal{S}$ deployed at $(0.9, 0.9)$ (i.e. diagonal), see Figure 8.

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

In what follows, we study the following parameters:

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

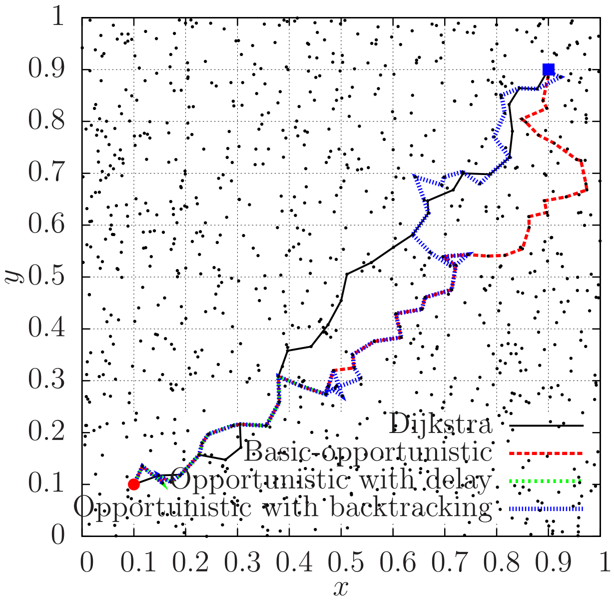

Figure 8: Opportunistic routing variants

A direct analysis of the above network model does not seem to be possible. Therefore, we use a Poisson rain model [6] and we analyse the opportunistic with delay variant. To simplify, we assume that the maximum delay at each hop is equal to $1/\lambda_{off}$. The main difference with respect to the network model described above is that the nodes $\{N_i\}$ are not fixed in time. Instead, we may think of these nodes as being "born" at some time T_i and being active one unit of time and "disappearing" immediately after. The joint space-time distribution of node locations and transmission instances $\Psi = \{(X_i, T_i)\}$ is modelled by a homogeneous Poisson point process in $2+1$ dimensions with intensity $\lambda_s = \lambda\lambda_{off}/(\lambda_{off}+1)$. We assume that the packet in a node is immediately sent to a "newly-born" node which has a positive projection on the direction towards the sink, see Figure 9[2]. We denote $\Delta_n = |X_{n+1} - X_n|$ where X_n and X_{n+1} are two successive node locations on the path of the packet sent towards the sink $\mathcal{S}$. We denote $S_n = T_{n+1} - T_n$ and θ_n is the angle between $\overrightarrow{X_n \mathcal{S}}$ and $\overrightarrow{X_n X_{n+1}}$ as illustrated in Fig. 9.

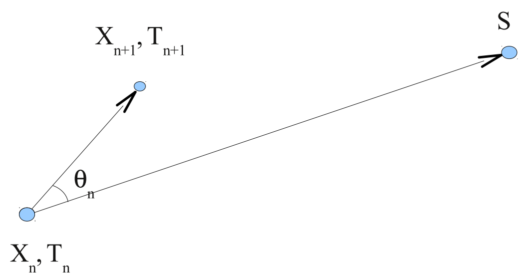

Figure 9: Packet progression towards the sink

The mean value of Δ_n, $\cos(\theta_n)$ and S_n can be easily computed: $E(\Delta_n) = \int_0^{\mathcal{R}_{com}} (1 - \frac{r^2}{\mathcal{R}_{com}^2}) dr = \frac{2\mathcal{R}_{com}}{3}$, $E(\cos(\theta_n)) = \frac{1}{\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos(\theta) d\theta = \frac{2}{\pi}$, $E(S_n) = \frac{\lambda_{off}+1}{\frac{\pi}{2}\mathcal{R}_{com}^2 \lambda\lambda_{off}}$.

Thus the packet propagation speed is equal to:

$$\mathcal{V} = \frac{E(\Delta_n)E(\cos(\theta_n)}{E(S_n)} = \frac{2\lambda\lambda_{off}\mathcal{R}_{com}^3}{3(1+\lambda_{off})}.$$

If we denote by $\mathcal{D}$ the Euclidian distance between the source and the sink, we can compute the mean number of hops as:

$$\mathcal{N}_{hop} = \frac{\mathcal{D}}{E(\Delta)\, E(\cos\theta)} = \frac{3\pi\mathcal{D}}{4\mathcal{R}_{com}}. \tag{1}$$

The end-to-end packet delay can be easily expressed[3] by:

$$T_{tot} = \frac{\mathcal{D}}{\mathcal{V}} + \mathcal{N}_{hop} \times (T_{pk} + T_{bc}) \tag{2}$$

where T_{pk} and T_{bc} are packet and beacon transmission delays respectively. The average packet delay per hop is equal to:

$$T_{hop} = \frac{T_{tot}}{\mathcal{N}_{hop}}. \tag{3}$$

[2]With the opportunistic with delay variant, the selected node decreases the remaining distance to the sink which is slightly different here.

[3]We add the transmission delay which is not considered in the analytical model.

It is also possible to roughly evaluate P_{path}. A packet is not blocked on its path towards the sink if, at each hop, it finds an active node within its communication range (i.e. $\mathcal{R}_{com}$) with a positive projection on the direction towards the sink, and within the maximum waiting delay (i.e. $1/\lambda_{off}$). This occurs with a probability of $1 - \exp(-\frac{\pi\lambda\mathcal{R}_{com}^2}{2(\lambda_{off}+1)})$. Thus we have:

$$P_{path} = \left(1 - \exp(-\frac{\pi\lambda\mathcal{R}_{com}^2}{2(\lambda_{off}+1)})\right)^{\frac{3\pi\mathcal{D}}{4\mathcal{R}_{com}}} \tag{4}$$

We now compute the energy consumed by a sensor node with and without a low duty-cycle. We recall that a transceiver has four states: i) off, ii) idle, iii) transmission and iv) reception and it consumes $\mathcal{E}_{off}$, $\mathcal{E}_{idle}$, $\mathcal{E}_{tr}$ and $\mathcal{E}_{rv}$ respectively. Assuming the `CC2420` chipset, the energy consumed in the different states is: $\mathcal{E}_{off} = 0.06\ mW$, $\mathcal{E}_{idle} = 1.27\ mW$, $\mathcal{E}_{tr} = 52.2\ mW$ and $\mathcal{E}_{rv} = 59.1\ mW$. When we do not turn off transceivers (no duty-cycle), sensor nodes do not need to send beacons. Since we assume an infrequent event, we can neglect the energy consumed during a packet transmission. The average energy consumed by a sensor is: $\mathcal{E}_{tot}^{on} = \mathcal{E}_{idle}$. If we assume that sensors use a low duty-cycle, they need to notify their neighbors when they wake up. Hence, the average energy consumed by a sensor during a whole duty-cyle can be expressed as:

$$\mathcal{E}_{rimac}^{on/off} = \left(\tfrac{1}{\lambda_{off}}\mathcal{E}_{off} + T_{bc}\mathcal{E}_{tr} + T_{pk}\mathcal{E}_{idle}\right) \tag{5}$$

The mean additional energy to convey a packet from a source node to the sink is:

$$\mathcal{E}_{rimac}^{p} = \mathcal{N}_{hop} \cdot (T_{pk} + T_{ack}) \cdot (\mathcal{E}_{tr} + \mathcal{E}_{rv}), \tag{6}$$

where T_{ack} is the transmission time of an acknowledgement.

3.1.3 Simulation results

In this section, we first consider simple simulations where there is no contention nor collision on the medium even though we use the timing values of IEEE 802.15.4. In the second part of this section we will consider the effect of the collisions.

In these first simulations, we assume that only one node detects the event. The transmission on the radio medium is perfect, there is no contention nor collision on the medium even though we use the timing values of IEEE 802.15.4 given above. We fix the duty-cycle by using $\lambda_{off} = 1/100$.

We obtain the simulation results by considering a packet propagation from the source node $\mathcal{O}$ to the sink $\mathcal{S}$. For the simulation results, we use $\lambda = 4000$

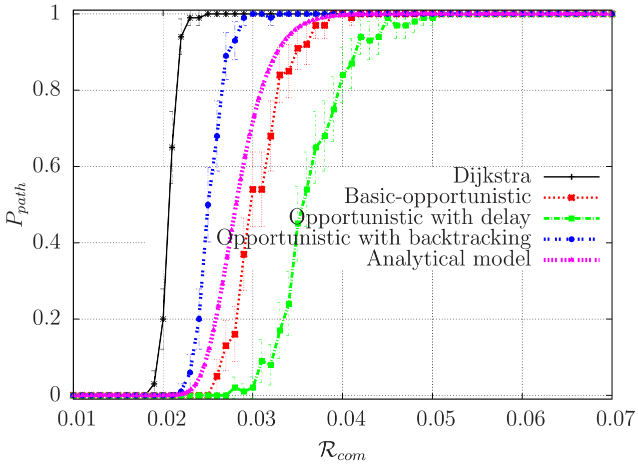

Figure 10: Probability of packet delivery to the sink - P_{path}

and $\lambda_{off} = 0.01$. We set the packet's maximum waiting delay at each hop to 100 time units, which is equal to the mean sleeping period (i.e. $1/\lambda_{off}$). We set the packet, the acknowledgement and the beacon transmission delays T_{pk}, T_{ack} and T_{bc} to, respectively, 0.7, 0.3 and 0.1 time units. We set the simulation duration to 10000 time units. We study each parameter over an average of 100 simulations. Moreover, the results are always presented with error bars corresponding to a confidence level of 95%.

To obtain real figures, we can multiply the distances by 1000, thus the network area is 1 km^2 and the average distance between a node and its closest neighbor is approximately $15m$. We can also assume that the sensor nodes use the `CC2420` chipset and the same beacon frame as in IEEE 802.15.4 [1]. The transmit bit rate is equal to 250 *bps* and the size of the beacon is equal to 19 bytes. In our model, a beacon transmission delay T_{bc} is equal to 0.1 time unit. Thus 1 time unit is equal to 6.1 ms. It is worth noting that we cannot compare the opportunistic routing with proactive (e.g. OLSR) or reactive (e.g. AODV) routing protocols due to the asynchronous random on/off activity of transceivers within the network. In fact, routes frequently change and the route request does not exist when the route reply packet is sent.

In Figure 10, we compare the probability of packet delivery, P_{path}, from the source node $\mathcal{O}$ to the sink $\mathcal{S}$ according to the sensors' transmission range, $\mathcal{R}_{com}$. We evaluate P_{path} by simulation for the three opportunistic routing variants and Dijkstra's protocol. We observe that Dijkstra's packet delivery probability converges quickly ($\mathcal{R}_{com} = 0.025$) to 1. If Dijkstra's protocol cannot convey a packet to the sink, a path does not exist in the physical network because Dijkstra's protocol computes all the possible routes. We remark that the opportunistic with backtracking protocol finds a path to the sink with a smaller communication range ($\mathcal{R}_{com} = 0.030$) than the two other variants. This is because with the opportunistic with backtracking protocol, the packet may be moved further back and is not blocked indefinitely in a node. Dijkstra's protocol only requires $\mathcal{R}_{com} > 0.025$ thus the difference in the communication range required for the opportunistic with backtracking protocol and Dijkstra is very small. Moreover, based on extensive simulations, we notice that the opportunistic with backtracking protocol moves the packet back by 5.89 ± 0.86 and 0.18 ± 0.11 hops when $\mathcal{R}_{com}$ is equal to 0.030 and 0.040 respectively. It is straightforward to see that opportunistic with backtracking protocol incurs a penny overhead. Besides, Figure 10 shows that the opportunistic with delay is the protocol that requires the largest communication range ($\mathcal{R}_{com} = 0.051$) to ensure $P_{path} = 1$. Finally, we observe that all opportunistic variants can ensure packet delivery ($P_{path} = 1$) when the transmission range is large enough. Indeed, a delay does not affect a packet's progression to the sink.

Figure 11 illustrates the average packet delay per hop, T_{hop}, generated with Dijkstra's protocol and all the opportunistic variants. We notice that the opportunistic routing variants significantly outperform the shortest path algorithm by roughly 4 times. We remark that even if we increase $\mathcal{R}_{com}$, the average packet delay per hop for Dijkstra's protocol is constant and equal to $\frac{1}{\lambda_{off}} = 100$, as could be expected. We observe that the opportunistic with delay offers the shortest delay per hop, then comes the basic-opportunistic. The opportunistic with backtracking protocol offers the longest delays. However when $\mathcal{R}_{com}$ increases, all the opportunistic variants offer the same performances.

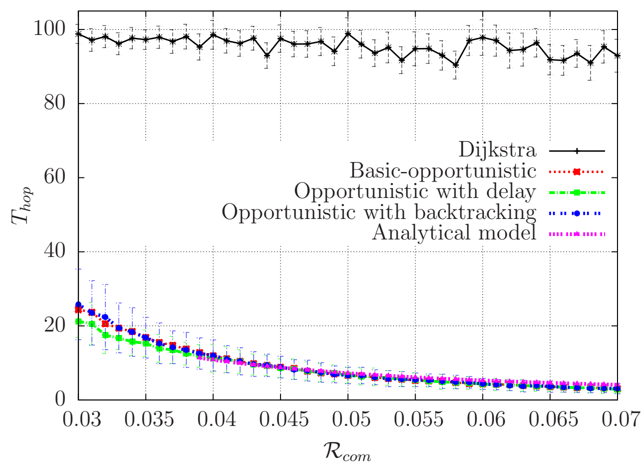

Figure 11: Average packet delay per hop - T_{hop}

In Figure 12, we compare the average length of the paths, $\mathcal{N}_{hop}$, obtained by the simulations. We notice that the opportunistic routing protocols never more than double the length of the path between the source node $\mathcal{O}$ and the sink $\mathcal{S}$ compared to Dijkstra's protocol. Nevertheless, as already pointed out and shown in Figure 13, the end-to-end delay is notably decreased (i.e. the gain is roughly between 2 and 4 times). Moreover, we observe that with small values of $\mathcal{R}_{com}$, opportunistic with backtracking has the longest path which can be explained by the fact that the packet may move further back from the sink. In addition, we observe that when $\mathcal{R}_{com}$ increases, all the opportunistic routing variants have the same path length.

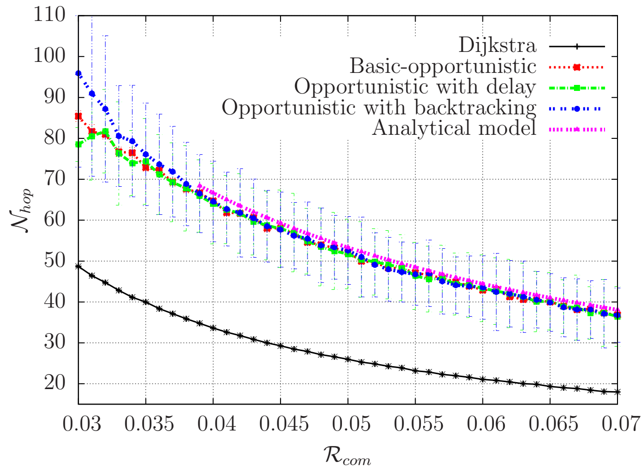

Figure 12: Average number of hops per path - $\mathcal{N}_{hop}$

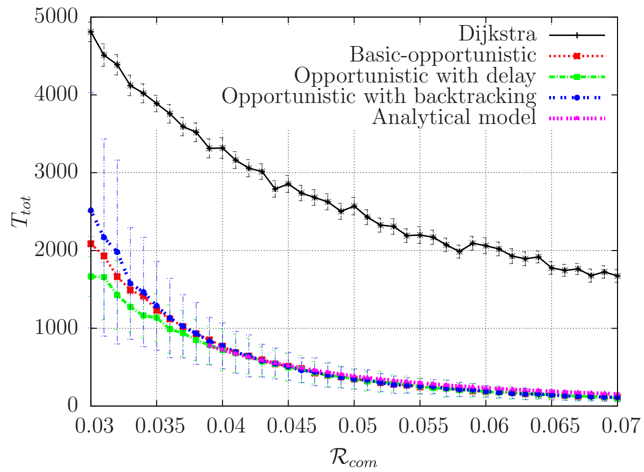

Figure 13: Average end-to-end packet delay - T_{tot}

In Figure 13, we evaluate the packet's end-to-end delay, T_{tot}, from the source node $\mathcal{O}$ to the sink $\mathcal{S}$. We remark that, with all the protocols, the end-to-end delay decreases when $\mathcal{R}_{com}$ increases. Moreover, we notice that all opportunistic routing variants notably outperform Dijkstra's protocol and the gain is approximately between 2 and 4 times. In fact, Dijkstra's algorithm builds a shortest path in terms of hops but not in terms of delay. Since sensors' transceivers use duty-cycles, a packet waits in each node for a long period (as shown in Figure 11), until the next node computed by Dijkstra's protocol wakes up. Moreover, we note that when $\mathcal{R}_{com}$ is small, opportunistic with delays is the fastest protocol. However, its packet delivery probability remains smaller than the two other variants. Opportunistic with backtracking routing is the protocol which exhibits the largest delivery delay which occurs because the packet can be moved further back from the sink. Hence, the path generated is longer than with the other opportunistic variants (as shown in Figure 12). However, we observe that all the opportunistic variants have the same packet end-to-end delay when $\mathcal{R}_{com}$ is large.

Concerning the analytical model, we have the following results. In Figure 10 we compare the probability of packet delivery to the sink (i.e. P_{path}) of the three opportunistic variants with the analytical model. Although the analytical model is closer to the opportunistic with delay protocol, we do not find a perfect matching between the analytical model and this variant. This may be explained by the algorithmic differences between the two approaches. However, the model gives a good order of magnitude for the transmission range for which the opportunistic with delay protocol has a path to the sink. In Figures 11, 12 and 13, we also compare the average packet delay per hop, the mean number of hops and the end-to-end delayobtained by simulation with the values given by the analytical model. It is clear that the matching between the two approaches is very good.

In the following simulations, we take into account the contentions and the collisions on the radio medium. The nodes use the IEEE 802.15.4 MAC to access the channel. We will specify whether the event is detected by only one node or by all the nodes in the neighborhood of the event detected.

We maintain $\lambda_{off} = 1/100$ but we use the assumptions of a real IEEE 802.15.4 network. We also assume that only one node detects the event in the network.

In Figure 14, we study the probability of a packet reaching the sink. We observe that in contrast to the ideal case where we do not take into account the MAC we do not reach 100% probability of success even with

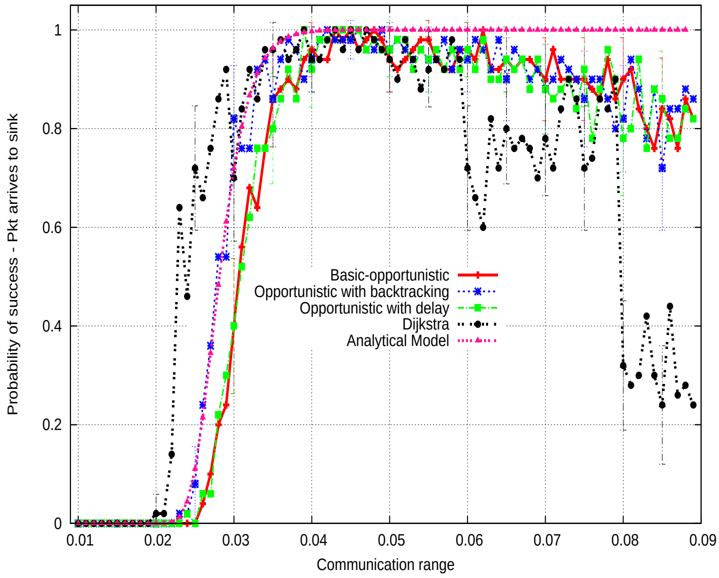

Figure 14: Probability of reaching the sink (`RI-MAC` 1pkt).

a large communication range, this is because of the collisions incurred by the beacons. For our opportunistic schemes based on `RI-MAC` we reach a probability of success between 80% to 90% but for Dijkstra we note a very low probability of success. This can be explained by the collisions; they are more likely on the long links produced by Dijkstra's scheme. There are no significant differences between the opportunistic routing variants.

In Figure 15 we study the delay per hop. This delay is much larger for Dijkstra's scheme, which is expected. There are no significant differences between the opportunistic routing variants.

In Figure 16, we study the mean number of hops to reach the sink. This mean number of hops is, as expected, much larger for opportunistic routing. There are no significant differences between the opportunistic routing variants except for small values of the communication range.

In Figure 17, we study the mean total delay to reach the sink. This mean delay is, as expected, much smaller for opportunistic routing even if the mean number of hops to reach the sink is much larger. The improvement in the per-hop delay explains this gain. There are no significant differences between the opportunistic routing variants except for small values of the communication range.

In the following scenarios, we still have $\lambda_{off} = 1/100$ but we use the assumptions of a real IEEE 802.15.4 network. We also assume that all the

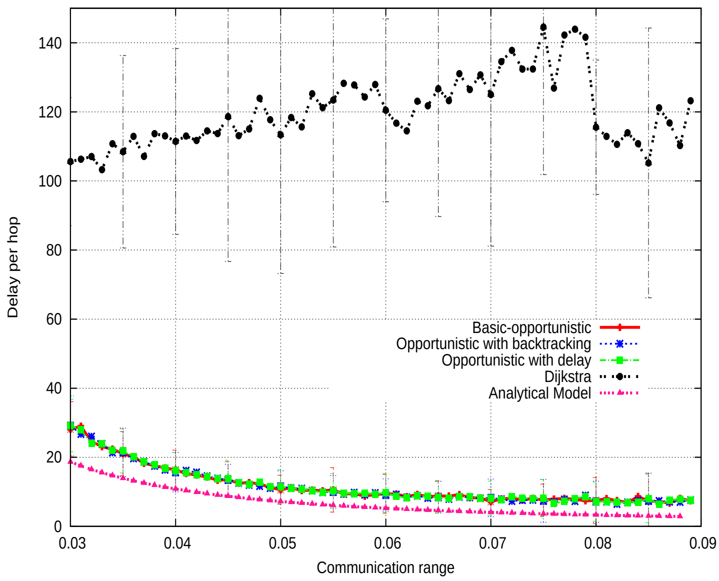

Figure 15: Mean packet delay per hop (`RI-MAC` 1pkt).

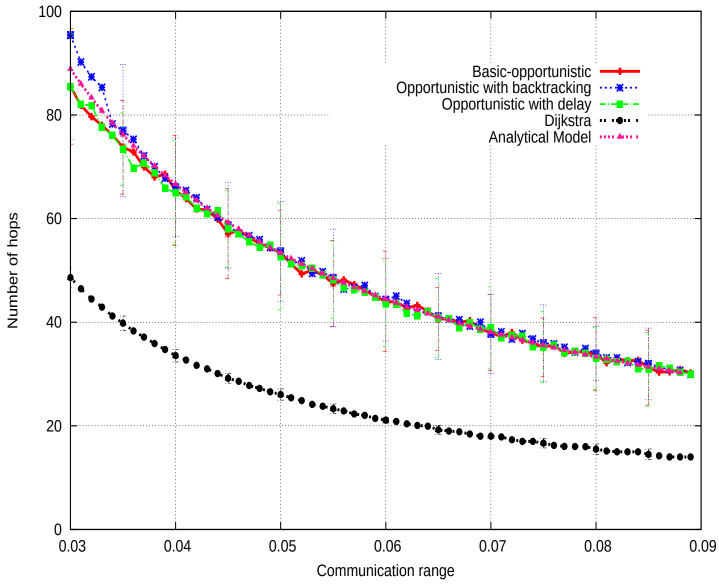

Figure 16: Mean number of hops (`RI-MAC` 1pkt).

nodes in the neighborhood detect the event in the network.

In Figure 18 we study the probability of a packet reaching the sink. We observe that the opportunistic routing offers much better performances than Dijkstra's shortest path routing. When the communication range is larger

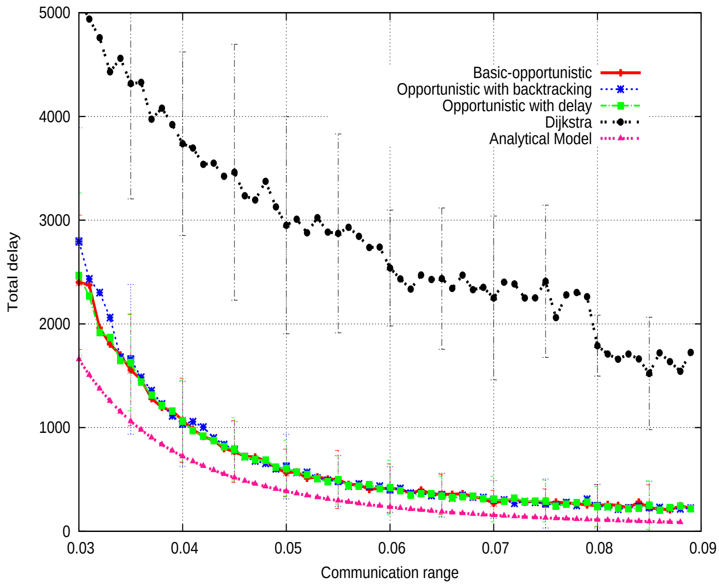

Figure 17: Mean total delay (`RI-MAC` 1pkt).

than 0.05 the oppportunistic routing offers 100% of success rate. This is probably because Dijkstra's protocol uses longer links and these links are less reliable that the shortest links used by the opportunistic routing scheme.

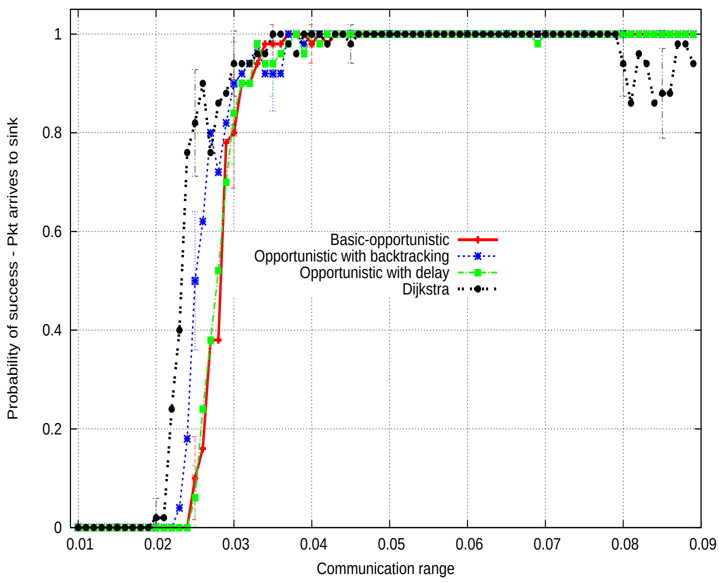

Figure 18: Probability of reaching the sink (`RI-MAC` multi packet).

In Figure 19, we study the mean delay per hop. As expected, this delay is much larger for Dijkstra's scheme. There are no significant differences between the opportunistic routing variants.

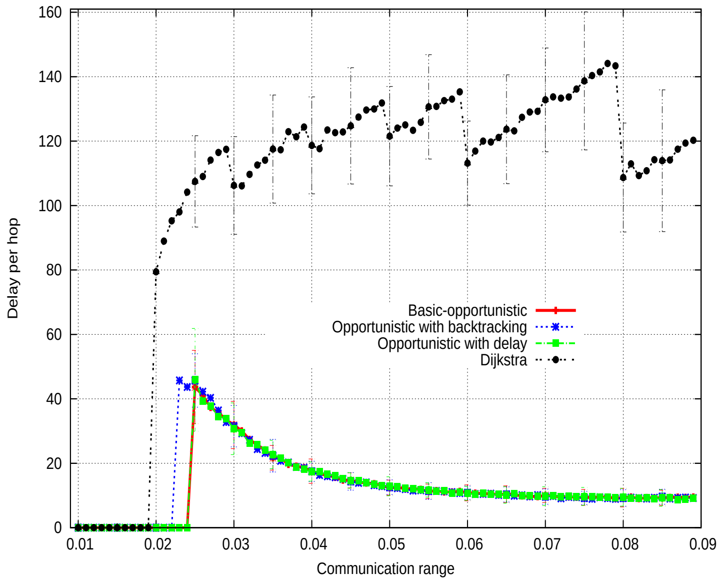

Figure 19: Mean packet delay per hop (`RI-MAC` multi packet).

In Figure 20, we study the mean number of hops to reach the sink. This mean number of hops is, as expected, much larger for opportunistic routing. There are no significant differences between the opportunistic routing variants except for small values of the communication range.

In Figure 21, we study the mean total delay to reach the sink. This mean delay is, as expected, much smaller for opportunistic routing even if the mean number of hops to reach the sink is much larger. The improvement in the per-hop delay explains this gain. There are no significant differences between the opportunistic routing variants except for small values of the communication range.

Energy consumption with `RI-MAC`

We obtain an energy consumption with the `RI-MAC`-based scheme $\mathcal{E}^{on/off}_{rimac} = 75.4$ mJ. The consumption without a low duty-cycle scheme is 782.6 mJ. This cleary shows the benifit of our low duty-cycle approach with regards to energy consumption.

If we assume that $T_{ack} = 0.3$ time units and set $\mathcal{R}_{com} = 0.05$ we obtain : $\mathcal{E}^{p}_{rimac} = 36$ J.

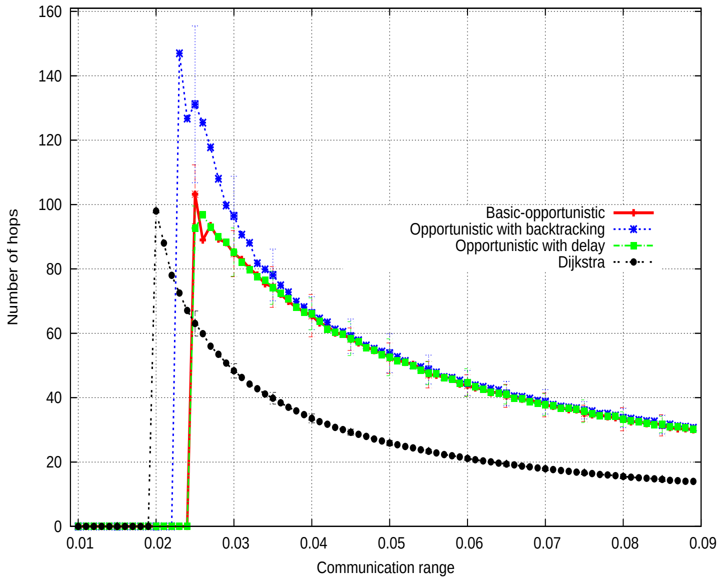

Figure 20: Mean number of hops (`RI-MAC` multi packet).

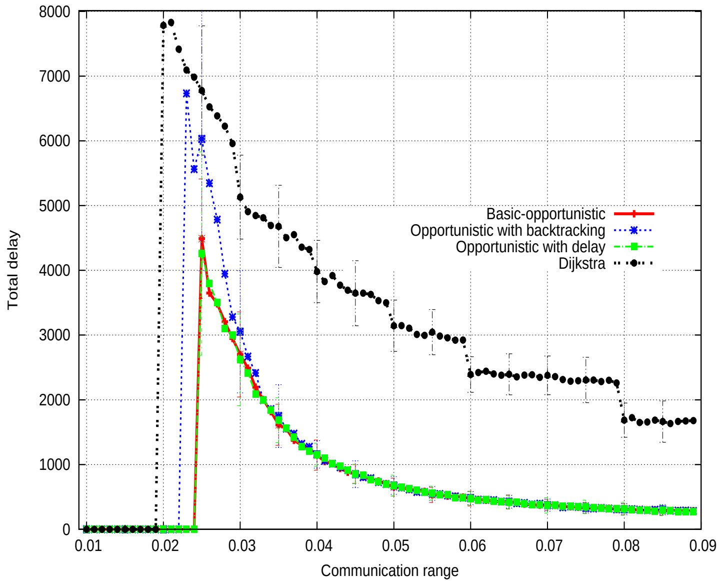

Figure 21: Mean total delay (`RI-MAC` multi packet).

3.2 Sender-oriented proposal

3.2.1 Description

We build our proposal on the `B-MAC` protocol that we presented above in subsection 2.2.2. `B-MAC` must be complemented at the routing level to ensure the connectivity of the network. We have to combine the B-MAC protocol with a routing protocol. We also assume that the sensors are equipped with a GPS or apply geo-localization methods.

The first way to combine the B-MAC protocol with a routing protocol is to assume that there is a warning period during which all the nodes are awake and exchange hello messages which contain with their position. In this case all the nodes know their neighborhood and can determine the next relay to reach the sink using geographic routing. After this warming period the network nodes return to the standard B-MAC protocol.

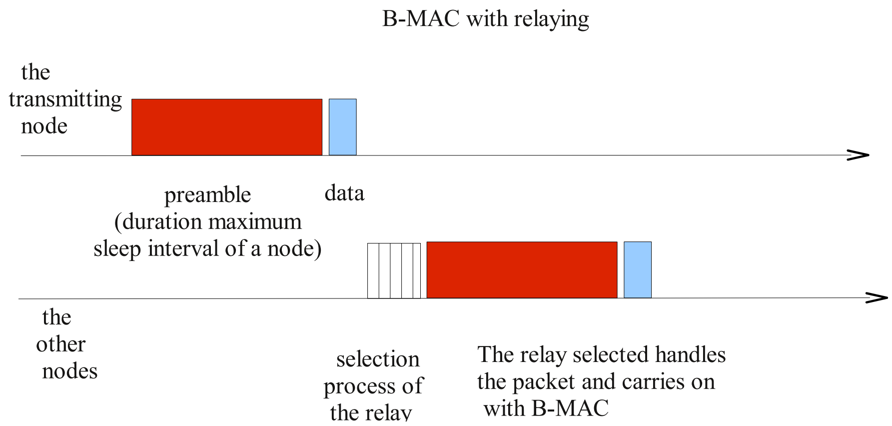

Figure 22: Opportunistic selection process

The second way to combine the B-MAC protocol is to use an opportunistic geographic routing. A selection process is used to select the next relay (see Figure 22). The most efficient way to do this is to use an active signalling scheme implemented in the acknowledgement of the packet sent by the transmitter after the preamble of B-MAC. Our proposed acknowledgement active signalling burst is quite similar to the HiPERLAN active signalling scheme, except that we propose to use it as an acknowledgement. We propose that each node receiving a packet has to transmit a short acknowledgement. In this case, several acknowledgements could be sent at the same time by all the nodes capturing the packet. In classical communication, this operation leads to several collisions between the acknowledgements, but not in our case. In fact, each acknowledgement, denoted as a burst in the following, is not carrying any data at all but consists of a sequence of intervals of the same length in which a given receiver can either transmit or listen. During the transmitting period each node sends a signal on the medium and during a listening period each node listens on the medium. These signalling bursts can be represented by binary sequences denoted in the following as burst codes. 0 denotes a listening interval and 1 denotes a transmitting interval. We propose to use the following relay selection rule: if a receiver detects a signal (energy) during any of its listening intervals, it quits the selection process. This means that the receiver stops transmitting during the entire remaining part of the burst. We consider that the detection of a transmission during a listening interval implies that a better relay has also captured the data packet. When a node receives a data packet it computes its own burst code as a function of the criterion that we wish to optimize to select our "best" relay. The only condition that we need to satisfy is that the better the relay is, the higher the burst code is. For instance, if we consider that the best relays are the farthest nodes, we should code the distance separating the source node and the relay nodes in base 2; the binary complement of the distance to the sink in base 2 will be the burst code. Thus, we can easily check that the selection mechanism will always select the relay nodes having the highest burst code (i.e. the greatest distance), since there will always be an interval in which another relay with a smaller burst code (i.e. a smaller distance) listens when the relay with the highest burst code is transmitting. Finally, in order to discriminate between nodes using the same burst code (e.g. at the same distance from the transmitter) we add r randomly selected bits.

In Figure 23, we show an example of the selection process of three potential relays, a, b and c. We can notice that node a, with a highest burst code wins the selection process. In this case, only node a will relay the packet.

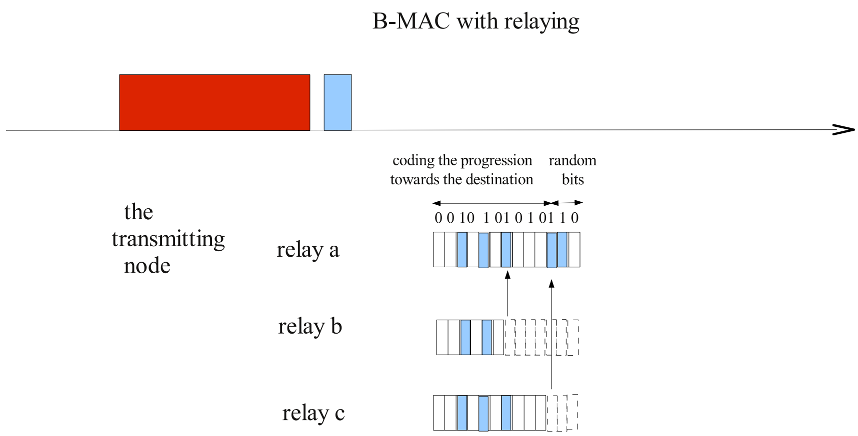

Figure 23: Opportunistic selection process

3.2.2 Analytical model

We denote by T_{pk} the duration of the packet in time units, T_s the duration (in time units) of the selection process of the packet offering the best progress towards the destination. The preamble has a duration of $1/\lambda_{off}$. In the following model, we do not consider the contention and collision periods.

If we call γ the distance between the current node and the destination node, the mean value of Δ_n is (see [14]):

$$\Delta_n \cos(\theta_n) \simeq \mathcal{R}_{com} - \frac{\Gamma(5/3)}{(\lambda a_0)^{2/3}}$$

with $a_0 = \sqrt{\frac{2R_{com}}{\gamma(\gamma-\mathcal{R}_{com})}}\frac{4(\gamma-\mathcal{R}_{com})}{3}$. If we assume that $\gamma \gg \mathcal{R}_{com}$ we obtain: $a_0 \simeq \frac{4}{3}\sqrt{2R_{com}}$.

The dependence of $\Delta_n \cos(\theta_n)$ with the distance to the sink and the dependence between two successive hops (see [14]) are not taken into account, thus we have:

$$\mathcal{N}_{hop} \simeq \frac{\mathcal{D}}{\mathcal{R}_{com} - \frac{\Gamma(5/3)}{(\frac{4\lambda}{3})^{2/3}(2R_{com})^{1/3}}}. \tag{7}$$

The delay for one hop encompasses the duration of the preamble λ_{off}, the duration of a packet T_{pk} and the duration of the selection process T_s. We obtain:

$$T_{hop} = \frac{1}{\lambda_{off}} + T_{pk} + T_s \tag{8}$$

Thus we have:

$$T_{tot} \simeq \mathcal{N}_{hop} \times \left(\frac{1}{\lambda_{off}} + T_{pk} + T_s\right) \simeq \frac{\mathcal{D}}{\mathcal{R}_{com} - \frac{\Gamma(5/3)}{(\frac{4\lambda}{3})^{2/3}(2R_{com})^{1/3}}} \times \left(\frac{1}{\lambda_{off}} + T_{pk} + T_s\right) \tag{9}$$

At each hop the probability of having a relay is approximately $(1 - \exp(-\pi\lambda\mathcal{R}_{com}^2)$ (we assume that we are far from the sink and we do not consider the dependence between two successive hops), thus we obtain the probability of reaching the sink:

$$P_{path} \simeq \left(1 - \exp(-\pi\lambda\mathcal{R}_{com}^2)\right)^{\frac{\mathcal{D}}{\mathcal{R}_{com} - \frac{\Gamma(5/3)}{(\frac{4\lambda}{3})^{2/3}(2R_{com})^{1/3}}}} \tag{10}$$

We do not take into account the energy consumption when an event is detected, because there is a very low probability of the event occurring. Concerning the power consumption when no event is detected, the nodes wake up with a periodicity of $\frac{1}{\lambda_{off}}$ for a listening period T_l. The power consumed by a node in the stationary mode during a whole cycle is :

$$\mathcal{E}_{bmac}^{on/off} = (\tfrac{1}{\lambda_{off}} + 1 - T_l) \cdot \mathcal{E}_{off} + T_l \cdot \mathcal{E}_{idle}, \tag{11}$$

since, during the listening period, the receiver does not receive any signal. The additional energy when a packet must be transmitted to the sink is:

$$\mathcal{E}_{bmac}^{p} = \mathcal{N}_{hop} \cdot \Big(\frac{\mathcal{E}_{tr}}{\lambda_{off}} + \pi\lambda\mathcal{R}_{com}^{2} \cdot (T_l\mathcal{E}_{rv} + \frac{\mathcal{E}_{idle}}{2\lambda_{off}})\Big). \tag{12}$$

The three terms correspond to the transmission of the preamble (by the transmitter), the reception of the information of the preamble (during T_l) and the idle duration until the end of the preamble (for the potential forwarder). It is very important in `B-MAC` that the potential relays return to the idle state to wait for the end of the preamble, otherwise energy consumption would be even greater than it is. However, we should bear in mind that the protocols we have designed are mostly devoted to the surveillance of very infrequent events.

3.2.3 Simulation results

In this subsection, we maintain $\lambda_{off} = 1/100$ and we use the assumptions of a real IEEE 802.15.4 network and `B-MAC`. We also assume that only one node in the network detects the event.

In Figure 24, we study the probability of the packet reaching the sink. We observe that Dijkstra's shortest path routing offers significantly better results than opportunistic routing schemes with `B-MAC`. This is because Dijkstra's shortest path routing takes into account all the possible routes to the sink whereas the opportunistic routing schemes only consider routes locally built with a greedy scheme. The back-tracking scheme offers slightly better results than the other opportunistic schemes.

In Figure 25, we study the delay per hop. There are no significant differences between the opportunistic routing variants and the matching with the analytical model is very good.

In Figure 26, we study the mean number of hops to reach the sink. There are no significant differences between the opportunistic routing variants except for small values of the communication range. The matching with the analytical model is very good.

In Figure 27, we study the mean total delay to reach the sink. There are no significant differences between the opportunistic routing variants except for small values of the communication range. The matching with the analytical model is very good.

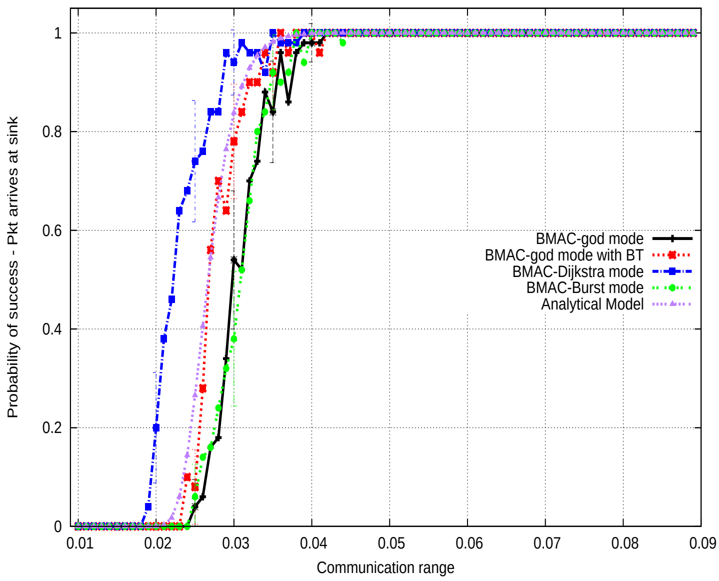

Figure 24: Probability of reaching the sink (`B-MAC` 1pkt).

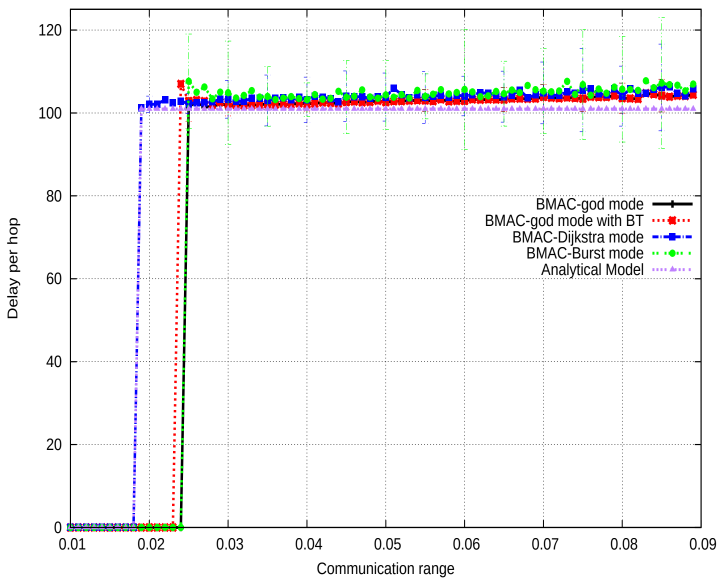

Figure 25: Mean packet delay per hop (`B-MAC` 1pkt).

We assume now that all the nodes in the neighborhood detect the event in the area covered by the network. We maintain $\lambda_{off} = 1/100$ and we still use the assumptions of a real IEEE 802.15.4 network.

In Figure 28, we study the probability of at least one packet reaching

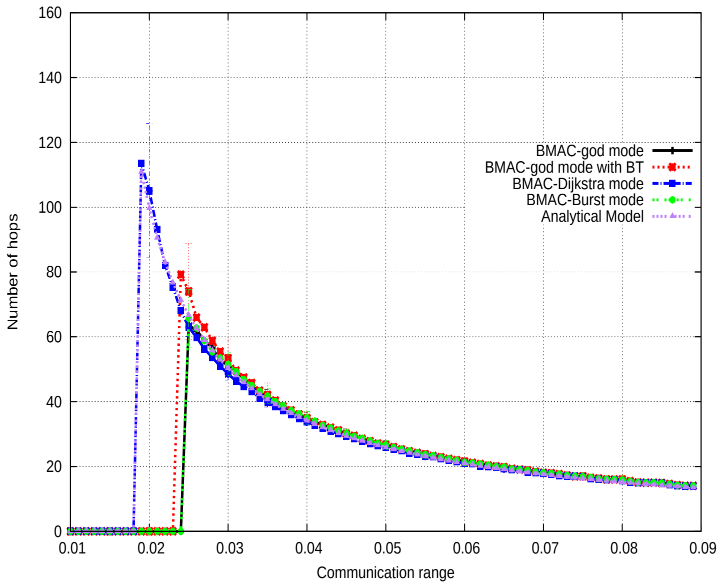

Figure 26: Mean number of hops (`B-MAC` 1pkt).

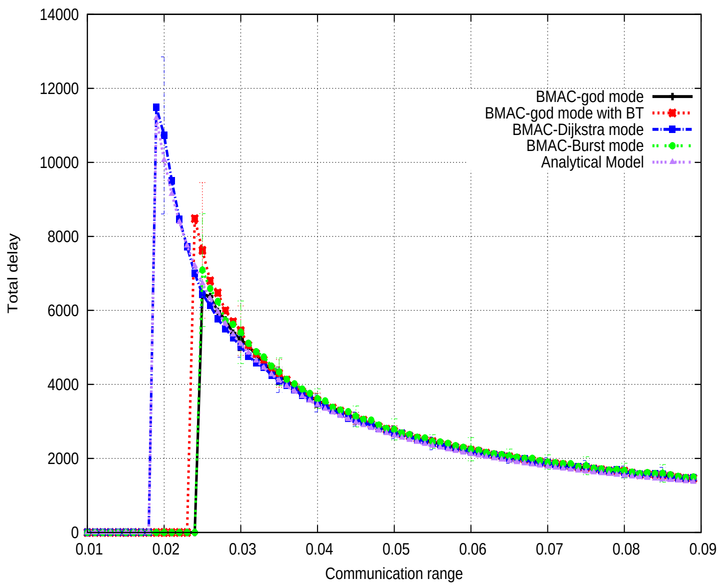

Figure 27: Mean total delay (`B-MAC` 1pkt).

the sink. We observe no significant improvement compared with the case where only one emergency packet is generated. This is because, when a single packet is generated, the relaying scheme is already a robust algorithm. As already stated for the single packet case, the back-tracking scheme offers slightly better results than the other opportunistic schemes.

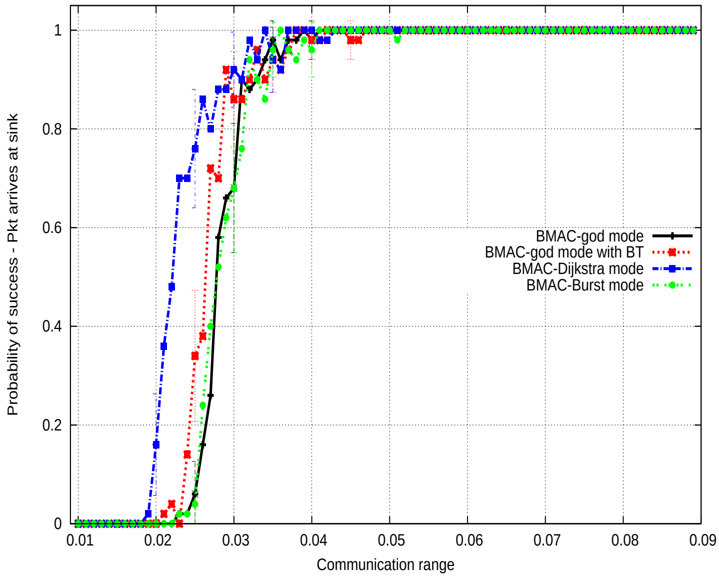

Figure 28: Probability of reaching the sink (`B-MAC` multi packet).

In Figure 29, we study the mean delay per hop. We observe that the delay increase with the transmission radius . This is because the contentions increase with the transmission radius. In contrast to the multi-packet detection there is no contention in the single packet detection scenario. There are no significant differences between the opportunistic routing variants.

In Figure 30, we study the mean number of hops to reach the sink. There are no significant differences between the opportunistic routing variants except for small values of the communication range.

In Figure 31, we study the mean total delay to reach the sink. There are no significant differences between the opportunistic routing variants except for small values of the communication range.

Energy consumption with `B-MAC`

For a whole duty-cycle, the `B-MAC`-based scheme consumes $\mathcal{E}_{bmac}^{on/off} = 41.05$ mJ. When $\mathcal{R}_{com} = 0.05$, the additional energy to convey a packet to the sink is : $\mathcal{E}_{bmac}^{p} = 869,9$ J.

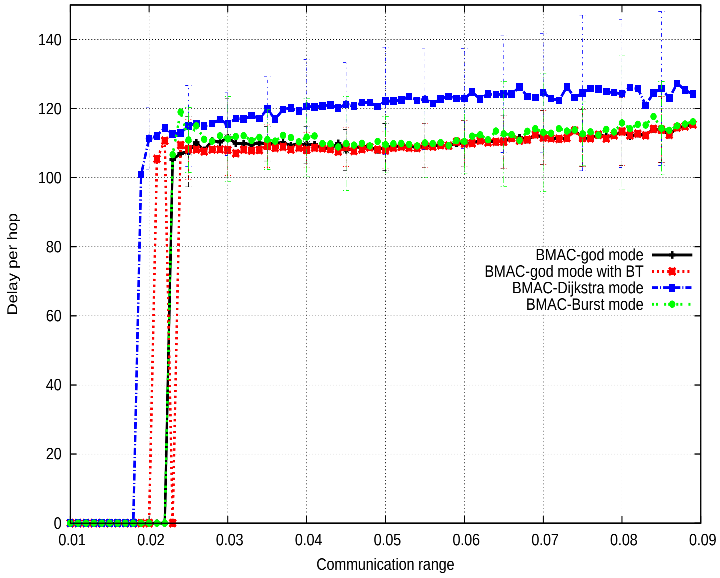

Figure 29: Mean delay per hop (`B-MAC` multi packet).

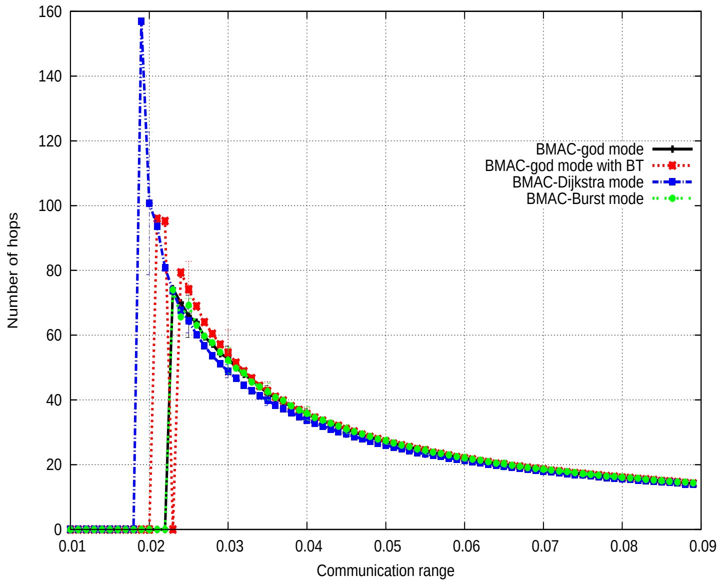

Figure 30: Mean number of hops (`B-MAC` multi packet).

3.3 Comparison and discussion

We present a comparison of the schemes using `B-MAC` and `RI-MAC` in Figure 32. This comparison is built with $\mathcal{R}_{com} = 0.05$. We observe that all the variants of

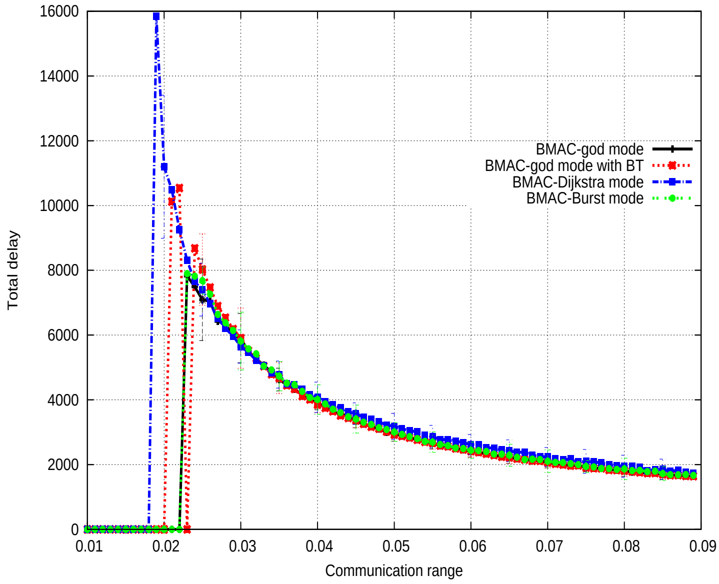

Figure 31: Mean total delay (B-MAC multi packet).

B-MAC and RI-MAC offer a perfect probability of detection except the RI-MAC with one node detection scenario. In this case, the RI-MAC-based approach only reaches 80% of detection probability. The B-MAC-based scheme does not need to use many packets to reach 100% of detection probability because B-MAC with one packet is already a very robust protocol.

RI-MAC offers a much better average total delay i.e. 3s than B-MAC which exhibits an average total delay of 16.5s. However, the total delay given by B-MAC remains compatible with many applications which are not too "real time"-oriented.

When there are no event detected, B-MAC and RI-MAC consume much less than a network operating without low duty-cycle mechanism, namely 41.05 mJ for B-MAC, 75.4 mJ for RI-MAC and 782.6 mJ when there is no duty-cycle involved in the scheme. This is 19 times more than the consumption of the opportunistic approach using B-MAC and 10 times more than the consumption of the opportunistic approach using RI-MAC. In the steady state the opportunistic based on B-MAC approach is the most energy efficient as it consumes around 45% less than the RI-MAC-based approach. However when there is a detection, the cost of sending a packet to the sink is $869,9$ J for the B-MAC-based scheme and only 36 J for the RI-MAC-based scheme which is 24 times less even if the number of hops to reach the sink is smaller with the B-MAC approach: 27 rather than 53 with RI-MAC. Given these values and if the expected rate of detection is known, it is straightforward to decide which

	Success rate	Total delay	Energy per cycle (stationary)	Energy for one packet
RIMAC 1 pkt	~ 80 %	~ 3 s	~ 75 mJ	~ 36 J
RIMAC multi pkt	100 %	~ 3 s	~ 75 mJ	
B-MAC 1 pkt	100 %	~ 16.5 s	~ 41 mJ	~ 870 J
B-MAC multi pkt	100 %	~ 16.5 s	~ 41 mJ	

Figure 32: Comparison `B-MAC`/`RI-MAC`.

is the better solution in terms of energy consumption.

The schemes based on `B-MAC` and `RI-MAC` are suitable to convey emergency packets in dense and large WSNs.

4 Conclusion

In this document, we have reviewed two main asynchronous low duty-cycle MAC schemes, one is receiver-oriented and the other is sender-oriented. We have explained how a low duty-cycle architecture for multihop WSNs can be built using these MAC schemes. We have proposed two low duty-cycle architectures: one uses a receiver approach: `RI-MAC`, the other using a sender approach: `B-MAC`. The routing scheme, in both of these architectures, is an opportunistic greedy scheme.

For each scheme, we have proposed a simple analytical model to analyse the performance of our proposals. These models can be used to determine if the schemes proposed satisfy the constraints of the application, for instance the delay. We have performed extensive simulations to study these two low duty-cycle architectures. We have shown that if correctly tuned both of them can satisfy the constraints of a dense surveillance network.

References

[1] IEEE standard, part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans). *IEEE Computer Society*, 2003.

[2] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized link state routing protocol (OLSR). *RFC 3626, Network Working Group*, 2003.

[3] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Elseiver Ad Hoc Networks*, 3:325–349, 2005.

[4] F. Baccelli, B. Blaszczyszyn, and P. Muhlethaler. Time-space opportunistic routing in wireless ad hoc networks, algorithms and performance. *The Computer Journal - Oxford Journals*, 53:592–609, 2009.

[5] Franois Baccelli, Bartlomiej Blaszczyszyn, and Paul MhlethalerVasileios Zeimpekis. Time-space opportunistic routing with an optimized relay self selection technique in wireless ad hoc networks. *Computer Journal*, June 2009.

[6] B. Błaszczyszyn and P. Mühlethaler. Stochastic analysis of non-slotted Aloha in wireless ad-hoc networks. In *Proc. of IEEE INFOCOM*, San Diego, CA, 2010.

[7] Michael Buettner, Gary Yee, Eric Anderson, Richard Han, and Michael Buettner Gary Yee. X-mac: A short preamble mac protocol for duty-cycled wireless sensor networks. In *in SenSys*, pages 307–320, 2006.

[8] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[9] Amre El-Hoiydi and Jean-Dominique Decotignie. Wisemac: An ultra low power mac protocol for multi-hop wireless sensor networks. In *In Proceedings of the First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), Lecture Notes in Computer Science, LNCS 3121*, pages 18–31, July 2004.

[10] Liu Haitao, ZhangBaoxian, H. Mouftah, Shen Xiaojun, and Ma Jian. Opportunistic routing for wireless ad hoc and sensor networks: Present and future directions. *IEEE Communications Magazine*, 47:103–109, 2009.

[11] Liu Hui, H. Darabi, P. Banerjee, and Liu Jing. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37:1067–1080, 2007.

[12] P. Jacquet, B. Mans, P. Muhlethaler, and G. Rodolakis. Opportunistic routing in wireless ad hoc networks: upper bounds for the packet propagation speed. *IEEE Journal on Selected Areas in Communications*, 27:1192–1202, 2009.

[13] Brad Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pages 243–254, New York, NY, USA, 2000. ACM.

[14] Holger Paul Keeler and Peter G Taylor. A stochastic analysis of a greedy routing scheme in sensor networks. *SIAM Journal on Applied Mathematics*, 70(7):2214–2238, 2010.

[15] Kevin Klues, Gregory Hackmann, Octav Chipara, and Chenyang Lu. A component-based architecture for power-efficient media access control in wireless sensor networks. In *In Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, pages 59–72, 2007.

[16] Hyuk Lim and Jennifer C. Hou. Distributed localization for anisotropic sensor networks. *ACM Trans. Sen. Netw.*, 5:1–26, 2009.

[17] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing. *RFC 3561, Network Working Group*, 2003.

[18] Joseph Polastre, Jason Hill, and David Culler. Stochastic analysis of non-slotted Aloha in wireless ad-hoc networks. In *Proceedings of the Second International Conference On Embedded Networked Sensor Systems (SenSys 2004)*, pages 95–107, Baltimore. USA, 2004.

[19] Yanjun Sun, Omer Gurewitz, and David B. Johnson. RI-MAC: A receiver initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. *SenSys: ACM Conference on Embedded Networked Sensor Systems*, pages 1–14, 2008.

[20] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. A survey on routing protocols for wireless sensor networks. *Elseiver Computer Networks*, 52:2292–2330, 2008.

[21] Vasileios Zeimpekis, George M. Giaglis, and George Lekakos. A taxonomy of indoor and outdoor positioning techniques for mobile location services. *ACM - SIGecom Exch.*, 3:19–27, 2003.