Routing protocols for wireless sensor networks-based network

Technical Report

José Carlos Castillo, Teresa Olivares and Luis Orozco-Barbosa

Albacete Research Institute of Informatics
University of Castilla-La Mancha 02071 Albacete, SPAIN

{josecarlos, teresa, lorozco}@dsi.uclm.es

1 State of the art in WSN protocols

In WSN applications, it is usual to locate nodes close or inside the observed phenomenon. This concept of network needs new network protocols which adjust to the new raised requirements. Traditional ad-hoc techniques can not cover those requirements because they cannot adjust to the design principles of this kind of networks [1]:

- Take into account power constraints.
- Data aggregation is useful only if do not hinder collaboration between nodes.
- An ideal sensor network has attribute-based routing and knowledge

In this section we will take a tour through the routing techniques for WSN which appear in the bibliography. In general, we can classify these techniques in three categories depending on the network structure [2]: flat-based routing where every node play the same roll in the network, hierarchical-based routing where the nodes play different roles and location-based routing where data are routed according to nodes position.

Furthermore, we said a protocol is adaptive if it gives the ability of control certain system parameters and to adapt themselves to the network conditions and the available power. Some examples of this kind of protocols are multipath routing, query-based routing, QoS-based routing and coherent routing.

The combination of both proposals gives as a result three classifications depending on how the source finds the destination: proactive protocols, reactive protocols and hybrid protocols. In proactive protocols routes where calculated at first, whereas reactive protocols routes where calculated dynamically when needed. Hybrid protocols use a combination of both techniques. So, proactive protocols are better for environment with fixed nodes and reactive protocols for environment with mobile nodes.

Another kind of routing protocols are cooperative routing protocols where nodes send data to a central node that join the data to reduce the cost in terms of energy consumption.

Table 1. Network protocols overview

	SPIN	Directed Diffusion	Rumor Routing	LEACH	TEEN & APTEEN	GAF	GEAR	SAR	RMR
Classification	Flat	Flat	Flat	Hierarchical	Hierarchical	Location	Location	QoS	Flat
Mobility	Possible	Limited	Limited	Fixed BS	Fixed BS	Limited	Limited	No	Limited
Power Usage	Limited	Limited	N/A	Maximum	Maximum	Limited	Limited	N/A	Maximum
Negotiation Based	Yes	Yes	No	No	No	No	No	Yes	No
Data Aggregation	Yes	Yes	Yes	Yes	Yes	No	No	Yes	No
Localization	No	Yes	No	Yes	Yes	No	No	No	No
QoS	No	No	No	No	No	No	No	Yes	No
Scalability	Limited	Limited	Good	Good	Good	Good	Limited	Limited	Good
Multipath	Yes	Yes	No	No	No	No	No	No	No
Query Based	Yes	Yes	Yes	No	No	No	No	Yes	No

Recently, a new classification for WSN routing protocols has arisen. It classifies the routing techniques according to packet destination (a single node, a set of nodes or every node in network) [3]. So, the following types can be found:

- Gossiping and agent-based unicast forwarding.
- Energy-efficient unicast.
- Broadcast and multicast.

- Geographic routing.
- Mobile nodes.

Gossiping and agent-based unicast forwarding: These schemas are an attempt of working without routing tables in order to minimize the overflow needed to build the tables, as much as a result of the initial stages in which the tables were not built yet. The simplest choice is flooding (forwarding each message received), but it is not very efficient.

Energy-efficient unicast: these techniques analyze the network nodes distribution to set the cost of transmitting over the link between two nodes and select an algorithm to calculate the minimum cost.

There are many aspects to consider about the energy awareness:

- Minimize energy per packet
- Maximize network's lifetime
- Set routes according to the remaining energy
- Minimize the amount of transmission power

Broadcast and multicast: earlier protocols, gossiping and unicast, try to find efficient ways to send data between nodes, possibly over several hops. For this, many nodes must collect or distribute the information to every node in the network (broadcast). In fact, broadcast is a common operation in WSN applications. In a similar way, sometimes is necessary to distribute data to a subset of previously known nodes. This process is called multicast.

Geographic routing: This kind of routing appeared due to two main motivations.

- Many applications need the node location as a reference address to allow destinations of the type: "every node in a given region" or "the closer node to a point". If these requirements are needed, an appropriated routing scheme must be provided.

- When the source and destination position is known and also the nodes among them, this information can be used to improve the routing process. For that, the destination node location must be specified geographically or relatively (with a location service).

The first idea, sending data randomly to every node in a given region is called geocasting. The second is called position-based routing.

Mobile nodes: We find three aspects with motion ability in this kind of Sensor Networks: Mobile sensor nodes, mobile base station and mobile sensed phenomenon.

2 Routing protocols for real WSNs

The earlier protocols where only tested in simulations. It is worth pointing out the proposals of routing protocols already implemented for WSN.

In first place we will analyze AODV protocol, created by Charles E. Perkins for ad-hoc mobile networks [4]. Ad-Hoc On-demand Distance Vector is a reactive protocol. This kind of protocols is known for calculating the route only when needed. This way, they try to reduce the overflow generated by the route updating messages in proactive protocols. The inconvenient of these protocols is the initial delay when trying to send messages through a new route.

Other protocol developed for WSNs is the so called GEM protocol. Its goal is to find a tree structure with minimum packet loss and low power consumption.

GEM is based in the optimization of two metrics: number of received packets, which depends on the network topology and environmental features; and energy consumption, which depends on the system performance. This approach needs the optimization of both metrics and takes into account two suppositions: Firstly, it only considers two independent parameters, retransmissions counter and transmission power to tune up the optimization; and secondly, the earlier parameters are the only ones allowed to change in time. They will be managed by the application.

3.1 Routing protocols for real WSNs

As we could see in previous pages, there are many protocols proposed for WSN but most of them were not implemented or, in the best cases, they are in a developing stage. For these reasons it has been carried out a tour over the routing techniques already implemented in TinyOS. Many proposals have been found along the different TinyOS contributions, but most of them are similar. For this reason, the center of our studies was based in one of the contributions that gave us the most number of choices: Xbow contribution.

Xbow contribution offers four kinds of routing: Basic Routing (with normal or improved variants), Reliable Routing, Low Power Routing and XMesh Routing. The main aspects of these routing techniques are captured in the following table:

Table 2. Features of TinyOS routing algorithms

Routing Technique	Use Watchdog	Update Interval	Metric	Consider Old Cost	Implementation Date
Basic	No	20 sec	Goodness	Yes	10/07/03
Basic Improved	Yes	20 sec	# lost packets	No	11/29/04
Reliable	No	20 sec	# lost packets	Optional	04/14/04
Low Power	Yes	> 6 min	# lost packets	Optional	04/02/04
XMesh	Yes	> 6 min	# lost packets	Optional	01/14/05

Initial suppositions expected a clear evolution in terms of network lifetime among the proposed algorithms, which would help us to generate an optimized algorithm. As the distinct algorithms were analyzed, differences in the algorithms resulted not so significant. In fact, the only differences observed in the implementations were the use of a watchdog component (which controls the correct operation of the node), the cost calculation (using the old cost for the calculation or not) and the increase of the packet send interval to update the neighbor tables.

In any case, the nodes energy level is not considered. It may cause that a node selects as father another node with almost depleted batteries causing a degradation of network performance.

After TinyOS implemented algorithms review, it was decided to test its features on a real sensor network. Every test was done over the IntellBuilding network due to the absence of simulators for our micaZ node platform.

The IntellBuilding network is located at the “Albacete Research Institute of Informatics” (i3a)[8]. The network consists of ten micaZ nodes provided with MTS400 sensorboards, a MIB600 network interface to program the nodes and a MIB510 serial interface as base station. The IntellBuilding application measures humidity and temperature every 8 seconds and it sends the data to the base station [5] (see figure 1).

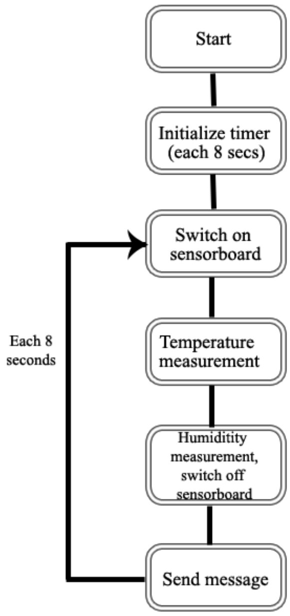

Figure 1. IntellBuildApp operation

Five tests were made to check up the features of Xbow routing algorithms. An application pointed by Crossbow as the specific for our sensorboards, XMTS400 application, was also tested. The results among the implementations were not very significant with only a few hours of difference among them. Even so, the Basic Improved Routing gave us the best lifetime with 97 hours. Network's lifetime in the five tests was calculated as the mean of the lifetime of three nodes acting as gateway. Figure 2 shows network's lifetime for the five tests.

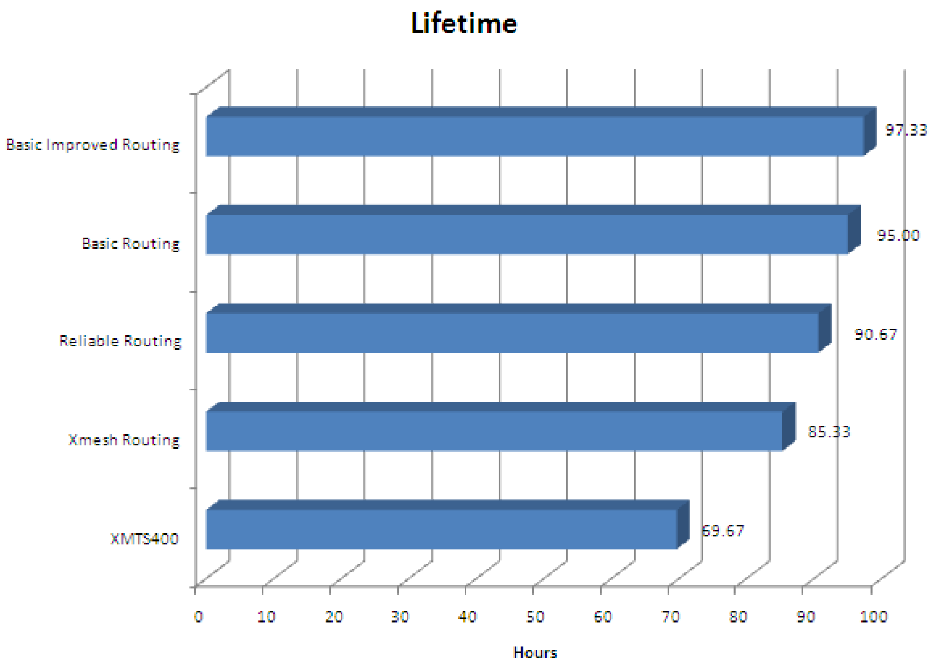

Figure 2. Network's lifetime

The next stage in the study was proposing improvements over the routing strategy to increase network's lifetime. In addition, a BMAC implementation for micaZ [6], which makes possible to sleep the nodes in order to make the routing protocol more power-efficient, was available for our researches.

3 RMR implementation for CC2420 radio chip

As can be appreciated in the bibliography there are many routing metrics, but in no case is considered the control of the sensed information. As a result, a new routing protocol that brings together the benefits of the existing protocols with this philosophy was implemented.

The improved design is based in some of IDEALS/RMR ideas [7]. This approach tries to extend the lifetime of the network through a combination of energy management and information control. Basically, each node decides its own network involvement as a result of a balance between available energy resources and the information content in each packet. The information content is ascertained through a system of rules which describe prospective events in the sensed environment. These rules specify when reporting should occur, and the importance of each packet. While energy management and information content have been individually considered elsewhere, this technique utilizes a combination of both to incur greater benefits. Results obtained from a simulation depicting an industrial Wireless Sensor Network (WSN) monitoring a water pumping station have shown that a considerable increase in lifetime and connectivity can be obtained. In addition, when coupled with energy harvesting, this technique permits sustained operation.

4.1 IDEALS/RMR

The concept of IDEALS/RMR is that a node with a high energy reserve acts for the good of the network by participating in routing all packets that come to it, and by generating its own packets from all locally detected events. However, a node with a near-depleted energy reserve acts selfishly, by only generating or forwarding important packets. Through this, IDEALS extends the network lifetime for important data, through the possible loss of more trivial data. The computational costs introduced are low, as only simple mathematical operations are required.

RMR

The purpose of RMR is to determine if an event worth reporting has occurred, and how important such an event is. A range of methods exist for deciding when a node should report that an event has occurred. The simplest method is to report periodically, every *t* minutes (meaning that packets are transmitted even when the sensed parameter has not significantly changed). The second option is a querying approach, where the base station instigates data transfer by requesting data from a subset of the nodes. The third method is for the sensor node to decide locally when events should be reported. This is the method that RMR uses. Before deploying a network, the designer creates a set of rules describing differing events that can be detected in the sensed environment. These rules include *threshold rules* (the sensed value crosses a preset value), *differential rules* (the change in the sensed value is larger or smaller than a preset value), *feature rules* (a pattern or feature is noticed in the sensed value), *periodic rules* (periodically, every *t* time), and *routine rules* (a packet of that importance or higher has not been sent for a period of time). For examples of these rules, see table 1. The rules are also assigned a message priority (MP), relating to the importance of the event. A high message priority (MP1) relates to an important event (e.g. a large temperature change). Conversely, a low message priority (MP5) relates to a low importance event (e.g. no temperature change). Intermediate priorities MP2-MP4 are allocated to events whose information content lies between the two extremes. On receiving sensor data, if any rules are fulfilled, RMR generates a packet with the associated MP.

IDEALS

IDEALS continuously assigns the node a power priority (PP) based on the state of the energy reserve and harvesting environment. Nodes with high energy reserves are allocated a high power priority (PP5), while near depleted energy resources are allocated a low power priority (PP1). Intermediate priorities PP2–PP4 relate to the power levels which lie between these extremes. When a packet is to be sent or forwarded, IDEALS compares the MP and PP. A packet will be sent if the PP ≥ MP. Therefore, as the residual energy drops, packets will be selectively discarded in order of their information content. For example, if the battery is full (PP5), packets with any information content (MP1– MP5) will be transmitted. However, if the battery is low (PP1), only packets with high information content (MP1) will be transmitted. A fraction of the energy is allocated to PP0 to maintain an energy store for power management, during which no sensing or communications takes place.

4.2 Proposed optimizations

After the analysis of the tests outcomes, it was noticed a large amount of generated data. In applications like ours, entrusted with monitoring environmental conditions in a building, these data are, most of all, irrelevant. For this reason the application works as detailed previously and reduces the amount of transmitted messages.

Our approach tries to carry out a processing of the information at the node, before the message is sent [9]. The algorithm lets the node check the importance of the information. If it is worthwhile, it is sent to the base station. To establish the sending conditions, a set of rules to check the data significance has been proposed. These rules can be classified in two of the categories pointed out in the RMR definition, *differential rules* and *periodic rules*.

Two rules are set as differential rules in order to establish the humidity and temperature thresholds that have to be reached before a message is sent. In the case of temperature there must be a difference of two degrees with last temperature notified in order to send a new message. In the case of humidity, the threshold is a variation of two percent from the last message sent (see table 3). Either, humidity and temperature thresholds and periodic rule time can be configured by changing its values in MultiHopEngineM.nc module. Implementation details will be explained later.

Table 3. Rules description

Rule	Value	Details
Differential	2 ºC	Sends a message when exist a humidity difference of 2 degrees since the last sending.
Differential	2 %	Sends a message when exist a humidity difference of 2% since the last sending.
Periodic	½ hour	Sends a message every half an hour.

The implementation includes a periodic rule. It forces the routing protocol to send a message if no message has been sent in half an hour. Moreover, this periodic rule allows to know the network lifetime too as the gateways nodes will stop sending messages.

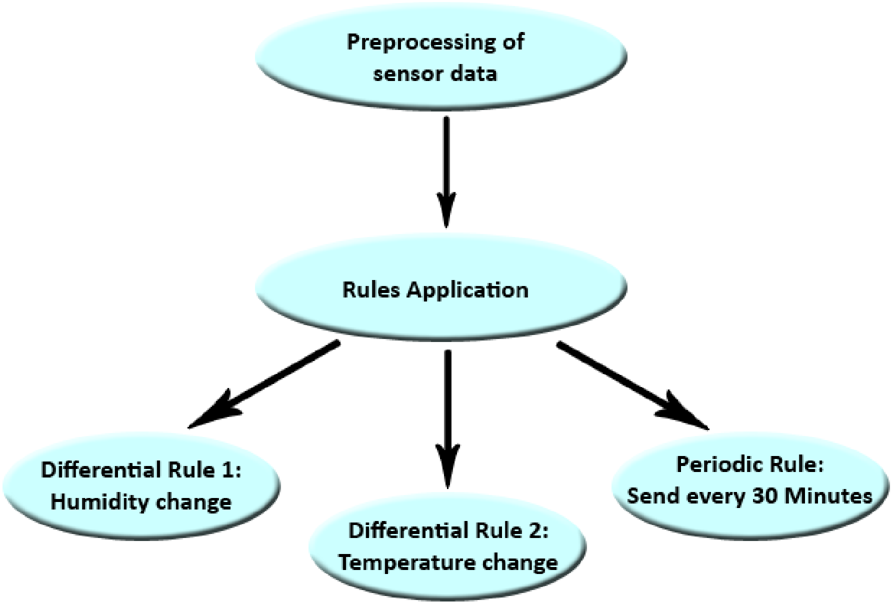

Figure 3. Rules application

SendRMR interface: The implementation begins with the creation of a new sending interface. It gives the routing protocol a new send method which accepts as parameters the humidity and temperature values used in the differential rules. The interface keeps the method described in the generic Send interface.

RMR implementation: The routing implementation is located inside the Xbow contribution. It is based, as was pointed out earlier, on the Basic Improved Routing and the rules to give the protocol the ability to decide about the significance of the sensed information were developed over its implementation. This implementation is made up of a library file, called Multihop.h, which describes the format of multihop packets; a configuration file, MultiHopRouter.nc, which carries out the wiring between routing modules; MultiHopLEPSM.nc module, entrusted with route calculation, neighbor tables management and father searching; and MultiHopEngineM.nc module, which implements the processing of the sensed information, the applying of the rules and the sending of messages depending on their importance. This module gets the humidity and temperature sensed values through the SendRMR interface. To avoid unnecessary processing, the periodic rule is checked. If no differential rules are checked or if any of the two values reaches the rule thresholds the message is sent.

The algorithm strategy is reflected in the pseudo code below (Remember the IntellBuilding application collects data each 8 seconds and tries to send the message to the base station):

```
each 20 seconds
    update neighbor table
    choose parent
    broadcast neighbor table

...

RMRSend(message, temperature, humidity)
    if counter > 224 // 225 periods of 8 seconds = 30 minutes
          send(message)
          counter=0
    elif(humidity – oldHumidity)^2 > thresholdHumidity
          send(message)
          counter=0
          oldHumidity = humidity
    elif(temperature – oldTemperature)^2 > thresholdTemperature
          send(message)
          counter=0
          oldTemperature = temperature
    else
          counter = counter + 1
          counter = counter % 255
    endif

end RMRSend
```

4 Outcome results

An earlier research of our group contributed with information about the performance of the IntelBuilding application provided with B-MAC for micaZ and a non optimal routing algorithm, Reliable Routing [6]. Two tests were carried out and network lifetime was increased to 235 hours with one of the configurations tested.

Table 4. First results with B-MAC

Test	Sleep period	Wake up period	Network's lifetime
Reliable routing + B-MAC (configuration 1)	1200 ms	1100 ms	165 hours
Reliable routing + B-MAC (configuration 2)	2500 ms	900 ms	235 hours

Due to the transcendence of the increase in the lifetime (remember the XMTS400 application only achieved 70 life hours over the network), it was decided to integrate our improvements at routing level with the MAC algorithm in order to check the benefits achieved by using a cross-layer frame.

Two tests were carried out: the first of them was a comparison between the earlier first configuration and B-MAC with the Basic Improved Routing. It gave us a first measure of the improvement over our base algorithm without including yet our optimizations; the second test adds these optimizations to the Basic algorithm. The test results are detailed below.

Table 5. Optimizations results

	Routing Type	Sleep Period	Wake Up Period	Network's Lifetime
Test 1	Basic Improved	1200 ms	1100 ms	183 hours
Test 2	RMR	2500 ms	900 ms	335 hours

While tests were running, several measurements of power consumption were made in the network nodes. As a result, network lifetime could be estimated before the end of the test. Figure 4 shows the instantaneous consumption for a node running the most relevant applications tested. From top to bottom, the high consumption of the XMTS400 application can be observed. This application is the less power-efficient in all our tests. The second consumption belongs to the IntellbuildApp with Basic Improved Routing and the standard MAC. It caused a clear improvement over the previous application. Next applications rised the network lifetime due to the incorporation of BMAC. This allows the nodes to sleep and this way the power consumption is strongly reduced.

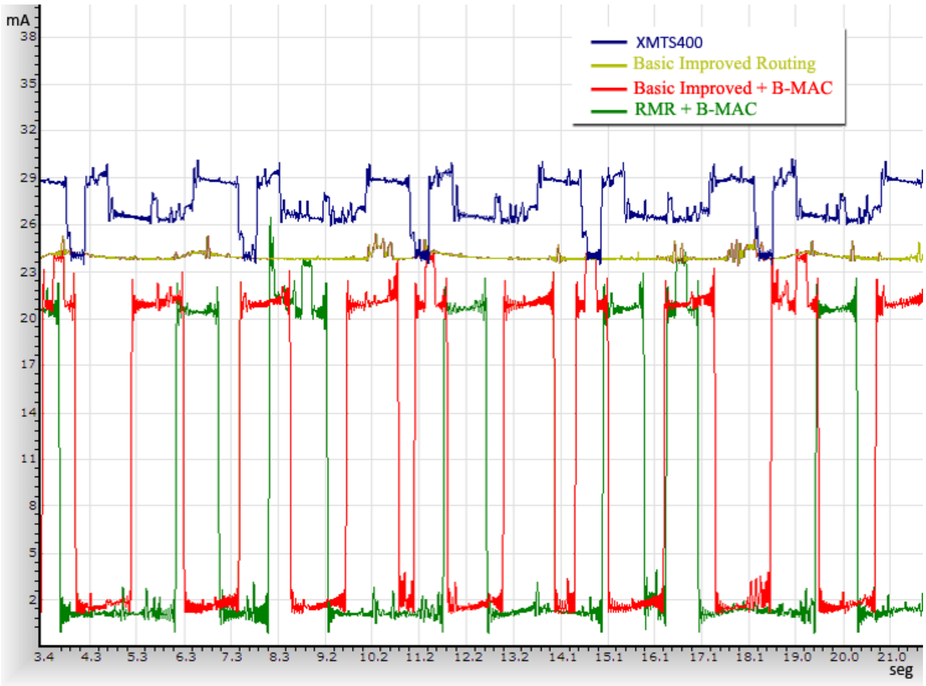

Figure 4. Instantaneous power consumption (node 2)

It is important to be aware of modes lifetime. Next graphic shows how lifetime is distributed along network nodes (gateway nodes and common nodes). The incrise in nodes lifetime is obvious in the tests. Nodes 9 and 10 were ruled out because its sensorboards were faulty. However, some nodes lifetime almost reached 350 hours. As the nodes with the longest lifetime did not reach 90 hours in the first tests, the obtained results must be explicitly pointed out.

	XMTS400	Xmesh	Reliable	Basic	Basic Improved	Reliable + B-Mac (config 1)	Basic Improved + B-MAC	Reliable + B-MAC (config 2)	RMR + B-MAC
Node 1	42	90	79	100	97	166	192	236	340
Node 2	43	84	90	93	93	165	181	225	331
Node 3	60	90	65	91	101	148	179	244	318
Node 4	83	88	94	97	99	166	170	248	344
Node 5	83	84	88	95	100	164	205	232	329
Node 6	42	25	94	27	93	166	183	238	319.5
Node 7	68	86	92	94	102	164	200	244	302.5
Node 8	80	90	80	97	102	165	194	242	333
Node 9	25	57	26	94	29	79	31	126	32
Node 10	39	31	28	28	29	82	33	123	35.5

Table 6. Nodes lifetime

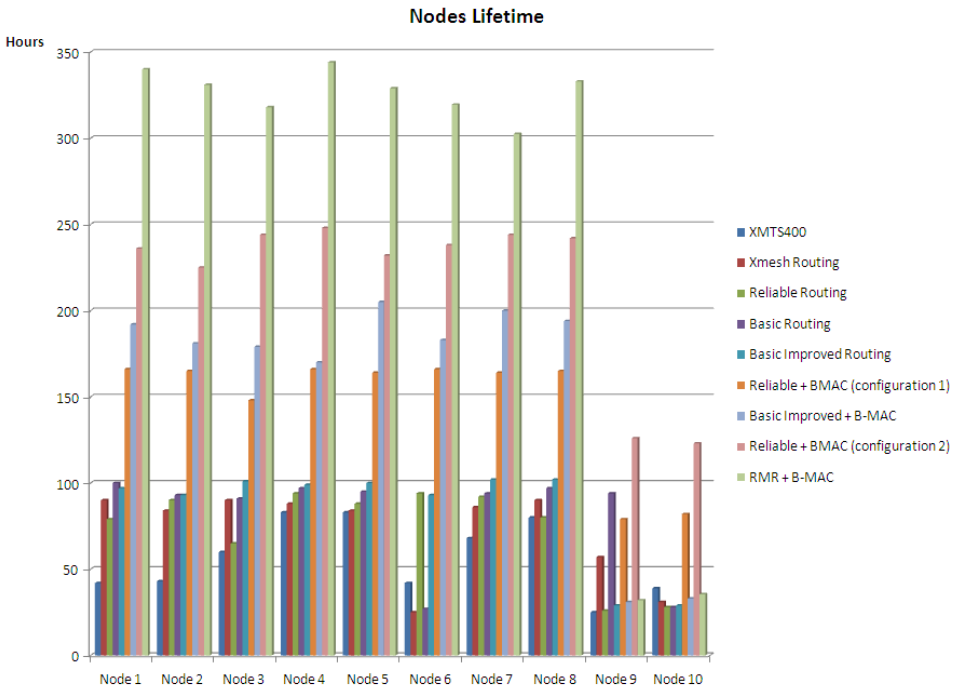

Figure 5. Nodes lifetime

Figure 6 shows the network lifetime in all tests within the specified metric. The network lifetime reach 335 hours, which means an improvement of 480% over the initial lifetime with the XMTS400 application.

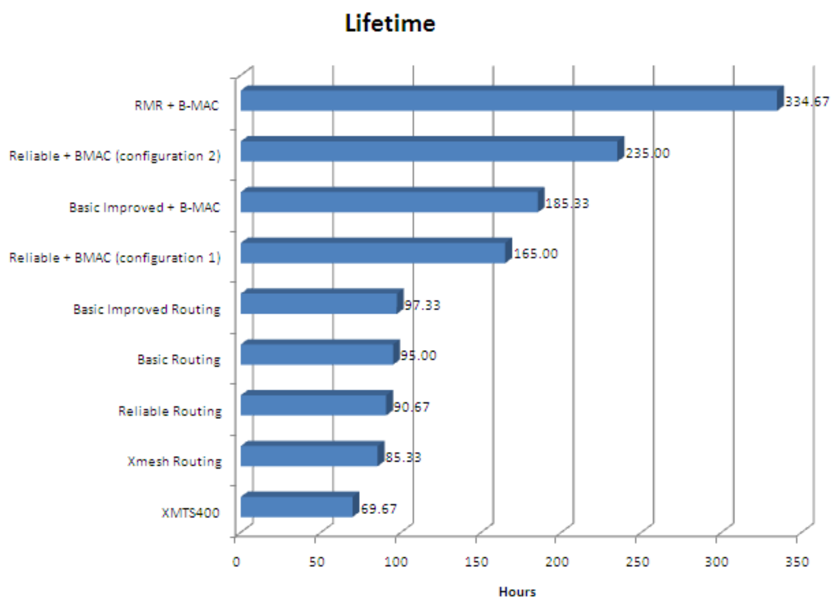

Figure 6. Tests results

References

1. Jamal N. Al-Karaki, Ahmed E. Kamal, "Routing Techniques in Wireles Sensor Networks: A Survey". Dept. of electrical and computer Engineering Iowa State University, Ames, Iowa
2. [Spanish] Vicente López Camacho, "Integración de protocolos en redes de sensores inalámbricas". Trabajo de investigación Universidad de Castilla-La Mancha, Mayo 2006.
3. H. Karl y A. Willig, "Protocols and Architectures for Wireless Sensor Networks". Editorial John Wiley & Sons, Ltd, ISBN 13978-0-470-09510-2, 2006
4. [Spanish] Óscar Alonso Blanco, "Evaluación de una red de sensores con protocolo AODV y tecnología radio IEEE 802.15.4". PFC Universidad Politécnica de Cataluña Septiembre 2005
5. T. Olivares, P.J. Tirado, F. Royo, J.C. Castillo and L. Orozoco-Barbosa, "*IntellBuilding: A wireless sensor network for intelligent buldings". Po*ster. Fourth European Conference on Wireless Sensor networks (EWSN 2007), Parallel and Distributed Systems Report Series, report number PDS-2007-00. ISSN: 1387-2109 . Delft (Ned) , January 2007.
6. [Spanish] T. Olivares y Fernando Royo Título: B-MAC y otros protocolos de Control de Acceso al Medio Publicación: Technical Report DIAB 06092, Departamento de Sistemas Informáticos. Universidad de Castilla-La Mancha, Septiembre 2006, http://www.info-ab.uclm.es/trep.php
7. Geoff V. Merrett, Nick R. Harris, Bashir M. Al-Hashimi, Neil M. White, "Energy Controlled Reporting for Industrial Monitoring Wireless Sensor Networks", School of Electronic and Computer Science, University of Southampton, UK.
8. http://www.i3a.uclm.es

9. [Spanish] José Carlos Castillo Montoya, Teresa Olivares Montes, “RMRRouting: Un algoritmo de encaminamiento basado en reglas para Redes de Sensores Inalámbricas. Desarrollo y prueba en la red IntellBuilding”. Universidad de Castilla-La Mancha, Escuela Politécnica Superior de Albacete, Departamento de Sistemas Informáticos. Diciembre 2006.