MeshMAC: Enabling Mesh Networking over IEEE 802.15.4 through Distributed Beacon Scheduling

Panneer Muthukumaran, Rodolfo de Paz, Rostislav Spinar, and Dirk Pesch

Center for Adaptive Wireless Systems
Cork institute of Technology
Rossa Avenue, Cork Ireland
{panneer.muthukumaran,rodolfo.depaz,rostislav.spinar, dirk.pesch}@cit.ie

Abstract. Although IEEE 802.15.4 is being considered as a promising standard for low-cost low-power Wireless Sensor Networks (WSNs), several issues in the specification are still open. One of those open issues is how to build a synchronized multi-hop mesh network for power efficient, scalable, and robust networking. In fact, while the current specification supports multi-hop networking using peer-to-peer topology, it restricts its use to the beaconless mode. This paper presents an extension to the current IEEE 802.15.4 standard in which mesh networking in the low power beacon mode is enabled through the use of distributed beacon scheduling. The work presented in this paper has been implemented on top of the Open-ZB IEEE 802.15.4 implementation within the TinyOS operating system. The feasibility of our proposal is demonstrated and evaluated through both physical test-bed and computer based emulation.

Keywords: Wireless Sensor Networks, Medium Access Control, Mesh Networking, IEEE 802.15.4 Standard.

1 Introduction

Wireless Sensor Networks (WSNs) have gained much interest over the past few years due to their ability to provide interaction between computer networks and their physical environment by sensing (or controlling) physical parameters. WSNs can enable a wide range of applications such as intelligent buildings, environmental control, health care or disaster relief operations. Although, there is no single set of requirements that classifies all WSN applications, most require a large number of battery powered nodes to operate for long periods of time without human intervention.

To facilitate large scale deployments, the sensing nodes must cooperate to efficiently route data over long distances from source to destination. In these environments where not every device can communicate directly with the destination, mesh network topologies are seen as a flexible and robust manner to provide multi-hop communication. Mesh topologies offer flexibility and robustness by facilitating path formation from any source to any destination within the network. Mesh topologies also enable network coverage extension, create redundancy, improve link quality and

J. Zheng et al. (Eds.): ADHOCNETS 2009, LNICST 28, pp. 561–575, 2010.

general network reliability simply by adding devices. The deployment of wireless sensor networks in mesh topology is therefore desirable.

Due to the battery powered nature of WSN nodes, energy efficient operation and data transmission is of paramount importance to afford long term operation without supervision. At the MAC layer a balance needs to be struck between achieving high quality radio resource allocation and energy expenditure. Several sources contribute to energy inefficiency in MAC layer protocols. The main sources are medium access collisions, idle listening, control packet overhead and overhearing. Traditionally, low-power MAC designers try to overcome them following either asynchronous or synchronous duty cycling strategies. Synchronous MACs reduce energy consumption by coordinating nodes into efficient sleep/wakeup schedules [1]. Their premise is that the cost of coordination is minimal compared to the energy savings achieved. In contrast, asynchronous MACs do not require coordination among the nodes in the network since they use preamble sampling methods to minimize idle listening [5, 6]. When a node transmits data, a long preamble is attached to the front of a packet. On the other side, receiver nodes remain in the sleep state and periodically turn the radio on to check whether the channel is busy. These schemes however suffer from long and variable preambles at the transmitter side and are best suited for bit streaming transceiver chipsets. However, the latest trend is toward packetized radios in which the preambles are of a fixed length such as for the TI CC2420 IEEE802.15.4 compliant chipset [12].

The IEEE 802.15.4 standard [2] specifies the currently most significant commercially adopted MAC protocol for sensor networks. The IEEE802.15.4 MAC provides two modes of operation, the asynchronous beaconless and the synchronous beacon enabled mode. The beaconless mode requires nodes to listen for other nodes' transmission all the time, which can drain battery power fast. The beacon enabled mode is designed to support the transmission of beacon packets between transmitter and receiver providing synchronization among nodes. Synchronization allows devices to sleep between coordinated transmissions, which results in energy efficiency and prolonged network lifetimes. However, current IEEE802.15.4 based standards such as Zigbee [3] or 6lowpan [4] restrict the beacon mode to star or cluster tree topologies only. While the former is one-hop limited, which reduces network coverage, the latter does not provide the scalability and robustness enabled by mesh topologies. This paper presents the MeshMAC, a distributed beacon scheduling mechanism that enables mesh networking over the IEEE802.15.4 beacon mode. Results of both, a physical test-bed and a computer emulation of micaZ motes [18], show that the proposed strategy enables collision free and energy efficient multipath mesh networking in a distributed manner.

The rest of the paper is organized as follows. Section 2 presents related work on beacon enabled medium access control and beacon scheduling. Section 3 presents the MeshMAC protocol and describes its implementation. Section 4 presents detailed results of a functional, data transmission and energy efficiency evaluation of the proposed approach. Section 5 concludes the paper and discusses future directions of this work.

2 Related Work

An IEEE 802.15.4 network is composed of nodes that take different roles during the operational phase of the system. Full Function Devices (FFD), also called beacon enabled devices, can operate either as Personal Area Network Coordinators (PANc), Cluster Heads (CLH) or routers. On the other hand, Reduced Function Devices (RFD) also called non beacon devices can only operate as end devices. A network includes at least one FFD, operating as the PANc. Other devices can either act as end devices forming a star around the coordinator, as routers creating a mesh network or as a combination of CLHs and end devices creating a cluster-tree topology.

The IEEE 802.15.4 MAC protocol specifies a beacon enabled and a beaconless mode of operation. In the beaconless mode, devices communicate asynchronously, requiring nodes to be continuously in receive mode, awaiting reception of data transmissions from other devices. Devices compete for channel access using an unslotted non-persistent CSMA/CA protocol. In the beacon enabled mode, devices synchronize their actions and coordinate data transmission with each other. FFDs periodically transmit beacon frames to synchronize wake up/sleep schedules with neighboring nodes. Channel access and data transmission are carried out using a superframe structure. As shown in Figure 1, the superframe is bounded by network beacons and is divided into 16 equally sized slots, the first of which is dedicated to the transmission of the beacon frame. The main purpose of the beacon is to synchronize devices in the network, to identify the PANc, and to describe the superframe structure.

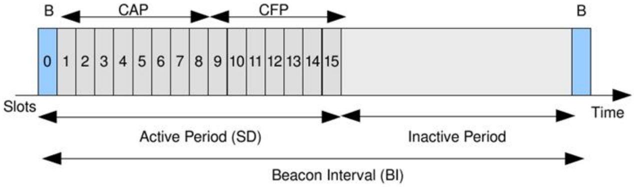


Fig. 1. Superframe structure in the 802.15.4 beacon enabled mode

The format of the superframe is decided by the PANc and is constructed from the Beacon Interval (BI), which defines the time between two consecutive beacon frames, and the Superframe Duration (SD), which defines the nodes' active period in the BI. The superframe duration provides a contention access period (CAP) in which all devices use a slotted CSMA/CA protocol to gain access and compete for time slots, followed by a contention free period (CFP) for low latency applications which is divided in guaranteed time slots (GTSs) to be allocated by the PANc. In order to reduce energy consumption, the coordinator can introduce an inactive period by choosing BI > SD. The inactive period defines a time period during which all devices, coordinator included, go into a sleep mode. BI and SD are determined by two parameters, the Beacon Order (BO) and the Superframe Order (SO), respectively, as

$$\mathrm{BI} = \mathrm{aBaseSuperframeDuration}{*}2^{\mathrm{BO}} \tag{1}$$

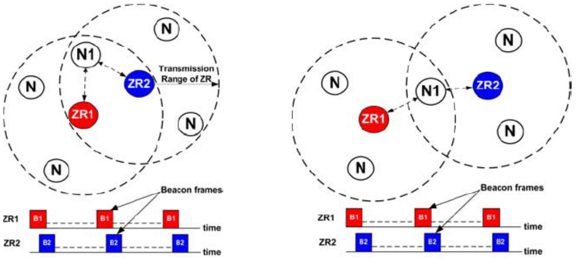
$$\mathrm{SD} = \mathrm{aBaseSuperframeSuration}{*}2^{\mathrm{SO}} \tag{2}$$

$$\text{for } 0 \le \mathrm{SO} \le \mathrm{BO} \le 14$$

where aBaseSuperframeDuration = 15.36 ms (assuming 250 kbps in the 2.4 GHz frequency band) denotes the minimum duration of the superframe, corresponding to SO = 0.

While the IEEE802.15.4 beacon enabled MAC provides energy efficient operation through sleep periods, it does not define how to schedule beacon and active periods to avoid overlapping of the periods among neighboring devices in multihop topologies. By neighbors we mean any group of devices that are within radio range of each other and can thus “hear” each others beacon. A new beacon enabled device must select a beacon transmission offset time that is not the same as its neighbors so as to avoid direct beacon collision – this is the simplest case and the most intuitive (see Figure 2a). The second cause for collision is known as indirect collision and occurs when a device hears two beacons from non-neighboring devices that overlap (see Figure 2b).

In spite of the importance of collision free operation, the standard does not specify how this can be achieved. In a multihop mesh topology each neighbor can synchronize and send/receive data from any other neighbor, so in order to avoid direct and indirect collisions a device should not transmit beacons at the same time of any of its neighbors or its neighbor’s neighbors.



a. Direct beacon frame collision

b. Indirect beacon frame collision

Fig. 2. Possible beacon frame collisions in IEEE 802.15.4

A number of proposals have been presented in the literature trying to achieve collision-free beacon scheduling. Koubaa et. al. present a beacon scheduling algorithm for IEEE 802.15.4/Zigbee Cluster-Tree Networks [7]. The proposed approach, called Superframe Duration Scheduling (SDS), builds upon the requirement for beacon scheduling outlined in the Zigbee specification for Cluster-Tree multi-hop topologies. The SDS algorithm defines a new beacon negotiation command packet that is encapsulated within a data frame. A device first associates with a coordinator node acting as an end device. If this new end device is required to transmit beacons it sends a beacon negotiation command to the coordinator. The coordinator uses the SDS algorithm to determine if all currently associated devices and the additional requesting device can be scheduled. A list of beacon enabled devices can be scheduled if the sum of the duty cycles is less than or equal to unity. If the device list is still schedulable, it determines an available time slot in which the new device can transmit its beacon and maintain a contention access period (CAP) in which data can be sent. The timeslot is determined so as not to overlap with existing timeslots. If successful, the beacon schedule time is sent to the device in a negotiation response packet.

A number of problems exist with the SDS algorithm. The algorithm only functions within the PANc or in a powerful node which has knowledge of the node locations for the entire network. Although this centralized control reduces the computational overhead and information flow between distributed devices, it can result in excessively data flow of control traffic towards the coordinator, which results in devices close to the coordinator potentially being overloaded in relaying this data. A distributed approach is more attractive here and is the core of our proposal. Furthermore, since the SDS algorithm builds upon the Zigbee specification for beacon scheduling, the approach is specifically designed for Cluster-Tree topologies, which is not as reliable and likely less power efficient than a mesh topology.

The IEEE 802.15.5 Task Group 5 is discussing a proposal for beacon scheduling for mesh topologies [8]. The proposal involves making fundamental changes to the superframe structure of the IEEE802.15.4 MAC to provide a beacon-only period (BOP) in which beacons of neighbors and neighbors' neighbors will be transmitted. This proposal however involves changing the MAC superframe structure, which affects the interoperability with the current MAC standard.

The work presented in [9] is also proposing distributed beacon scheduling for IEEE802.15.4 beacon enabled mesh networks. However, the algorithm requires every node to send the association request to the PANc by tree routing where every node forwards the request to their parent node until the PANc is reached. The PANc then replies within the association response with a random number. This random number is used to select an unused random slot as its active superframe. Then the allocated active superframe is validated for beacon collisions during an integrity phase. Although this solution minimizes the setup time, the integrity check may take a long time. The approach still requires a central node to process the association request and to generate the random numbers, and beacon collisions are detected in a reactive manner.

3 The MeshMAC

The MeshMAC protocol addresses the issues identified above through a distributed beacon scheduling strategy in which a beacon enabled device calculates its schedule to transmit beacons and specifies the active superframe duration based only on locally available information without the need to change the standardized superframe structure. The MeshMAC protocol assumes that the entire PAN operates on a fixed beacon interval (BI) and active superframe duration (SD). Hence, the superframe structure for all beacon enabled devices is the same. In doing so, the MeshMAC adopts a similar approach as the SDS algorithm [7] by scheduling the active superframe durations during the inactive periods of neighbour nodes.

Following subsections define the novel MeshMAC distributed beacon scheduling specification, the unicast and broadcast mesh data transfer and the detailed protocol operation.

3.1 Distributed Beacon Scheduling Specification

The MeshMAC defines a distributed beacon scheduling scheme for IEEE 802.15.4 based mesh networks. The scheduling algorithm assumes that the entire network operates on the same Beacon Interval (BI) and active Superframe Duration (SD). The durations of these two time intervals are determined by BO and SO as per equation (1) and (2). The proposed algorithm requires, in order to schedule a node i that the sum of the duty cycles (DC) of its N beacon enabled neighbors and neighbors' neighbors devices is less than or equal to unity. This can be expressed as:

$$\sum_{i=1}^{N} DC_i = \sum_{i=1}^{N} \frac{SD_i}{BI_i} \leq 1 \tag{3}$$

The beacon scheduling algorithm thus limits the scheduling to every node's 2 hop neighbors since nodes further than two hops do not cause collisions. The MeshMAC protocol therefore schedules devices based on the following approach:

1. Let N be the maximum number of beacon enabled devices operating on a $2r$ transmission range of device i with range r and fixed BO and SO. This number corresponds to $2^{\text{BO-SO}}$ active superframe durations that can be scheduled on the beacon interval BI=2^{BO} satisfying condition (3). This number is constant for every node's duty cycle and equal to $2^{\text{SO}}/2^{\text{BO}}$.

$$2^{(BO-SO)} \cdot \frac{2^{SO}}{2^{BO}} = 1 \tag{4}$$

$$N = 2^{(BO-SO)} \tag{5}$$

2. Let the number of time slots be equal to the number of nodes N where each slot is equal to one active superframe duration SD=2^{SO}.
3. Reserve one active superframe duration exclusively for broadcast communication in the PAN. Hence, the maximum number of devices N_s that can be scheduled within a $2r$ range is

$$N_S = 2^{BO-SO} - 1 \tag{6}$$

4. Every node within the $2r$ transmission range collects the active superframe duration slots occupied by all beacon enabled devices. In order to know all the beacon enabled devices within the $2r$ range, the node needs to collect its two hop node's superframe duration slots.
5. Select the first empty slot among the N slots and advertise the selection decision.
6. Return "not schedulable" if there is no empty slot.

Using this specification, the MeshMAC implements a distributed beacon scheduling strategy in which the selection of the beacon transmission offset of a beacon enabled device is not determined by the PANc in a centralised manner but in a distributed fashion based only on local information available in its two hop transmission range. The two hop neighbor approach is important in determining the beacon transmission offset in order to avoid direct and indirect collisions (see Figure 2). The selection of the appropriate slot could differ from the first available one, as established in the current approach, since it may depend on data latency, network load or other metrics. On the other hand, the reservation of one slot for broadcast communication is essential since each beacon enabled device is sleeping during other devices' active periods.

3.2 Mesh Data Transfer

The MeshMAC facilitates two types of data transmission: broadcast and unicast. Broadcast transmission takes place in the reserved broadcast active superframe duration slot. The broadcast slot is used as the time reference to compute the beacon offsets of beacon enabled nodes. Every node embeds its beacon transmission time offset in their beacon frames. The unicast data transmission takes place during a destination node's active period. For non beacon enabled devices (End devices), the data transfer occurs in the beacon enabled source/destination node's active period. The data to be transmitted is queued until the appropriate time slot in the superframe is active.

3.3 MeshMAC Operation

The formation of a mesh network topology has two phases, the initialization phase, where the device acts as an end device and the maintenance phase, where the device operates in beacon enabled mode. During the initialization phase, every node establishes itself as an end device, in a similar way to the cluster tree topology by association method. The device then performs a coordinator scan, which is the period in which a new device scans the radio channel for beacon packets transmitted by existing beacon enabled devices in the PAN. In this way scheduling information is not required to be sent to the PANc each time a FFD requests a beacon schedule time, hence reducing the control traffic overhead towards the coordinator.

The proposed protocol operation is shown in Figure 3. A node joins the network by associating with a beacon enabled device, based on the strongest signal strength. During this procedure the new device is assigned a network address using the network layer address assignment scheme. For the distributed beacon scheduling, each beacon enabled node (FFD) requires the list of its two hops neighbor nodes. This is obtained by two new primitives *MLME-NEIGHBOUR_SCAN* and *MLME-NLIST*. Using *MLME-NEIGHBOUR_SCAN* the node obtains its neighbor's beacons (neighbor list) and records the beacon transmission time of each neighbor in the neighbor table. With the *NLIST* procedure the device requests the neighbors' neighbor list in the broadcast active superframe duration by sending the *NLIST_REQ* command. Upon receipt, the neighbor nodes will transmit their direct neighbor list in their active superframe duration using direct data transmission. The list will contain the short address and slot offset from the broadcast slot. To facilitate easy access and resolving the index in the neighbor table the broadcast slot is used as the first slot in the neighbor table. If there is a conflict in the neighbor table, the node sends an error message in broadcast slot.

Upon completion of this step the node will have a complete list of its neighbors and neighbors' neighbors. With this information, each FFD can determine its own schedule period by finding an empty entry in the neighbor table. The method to define the best empty slot is implementation and application specific as it could be influenced by routing, data latency and other network layer related functionality. The approach used here is to select the first available empty entry from the broadcast slot.

Figure 4 depicts transceiver modes, beacon scheduling and data transmission for a wireless sensor network with 4 routers (FFDs). It can be seen that nodes become active during their own active superframe periods, when receiving and processing beacon frames from neighbor nodes, and when transmitting data frames. The data that may be available to transmit during the inactive period is queued and transmitted at the start of the active period. In Figure 4 this would be the case for the transmission TX_2 from R4 to R2. Every router becomes also active during the broadcast slot, e.g. R4 broadcasting a packet (TX_b).

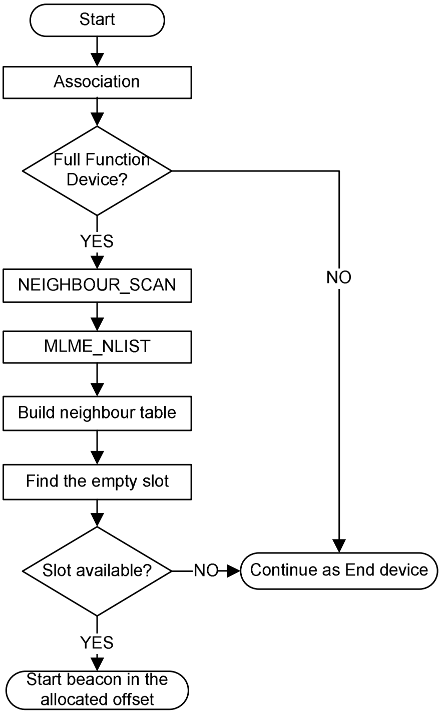


Fig. 3. MeshMAC protocol operation

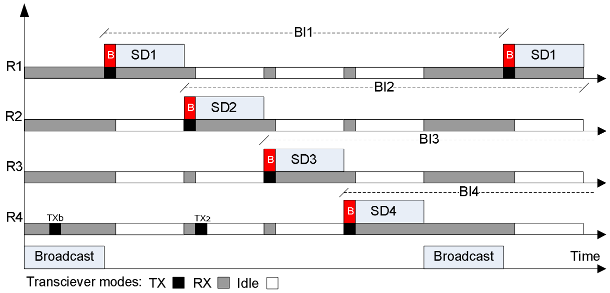


Fig. 4. Transceiver modes and data transmission example for 4 routers

4 Evaluation

The MeshMAC protocol aims at providing distributed beacon scheduling on top of IEEE 802.15.4. The MeshMAC is designed therefore for low power standard compliant radios such as TI CC2420 [12] that are integrated with a low power microcontroller. As the MAC layer protocol is tied with hardware parameters such as timer, radio and battery power, it is essential to verify its operation on a suitable hardware platform.

Hence, the MicaZ [13] platform was selected to create a wireless sensor network test-bed where to evaluate the MeshMAC protocol. We use the Re-Mote framework [11] to provide remote access to every MicaZ mote in the test-bed. Furthermore, a CC2420 packet sniffer hardware [15] with a Daintree Protocol Analyzer software [16] is also used to interpret and analyze the protocol's operation.

In addition to the physical test-bed, we use the AvroraZ [14] emulator, which is an instruction level virtual machine based emulator for the MicaZ architecture, executing the same code that runs on the real hardware nodes. The advantage of using AvroraZ is that we can study a larger scale network and accurately evaluate energy consumption thanks to its extended AEON (Accurate Prediction of Power Consumption) model [10]. The CC2420 radio in the MicaZ nodes requires different levels of current draw depending on its state of operation. In order to conserve battery power, the radio transceiver can be switched to idle mode and only turned back to transmit or receive mode when necessary since these modes consume higher amounts of energy. The current draw in idle mode is 426μA whereas in receive mode consumes 18.8mA. MeshMAC thus switches the transceiver to idle mode during inactive periods. The AvroraZ battery monitor computes the energy consumption from the time spent in each power mode and their corresponding current draws. The configuration shown in Table 1 has been used for the evaluations presented here. Specific configurations are explained with each scenario.

The MeshMAC protocol is evaluated in three different ways, first the distributed beacon scheduling operation is verified, then the multipath mesh networking is demonstrated and finally the energy efficiency is evaluated.

Table 1. Evaluation configurations

Parameter	Value
Frequency channel	18 (2.443 Ghz)
PAN ID	0x1234 (4660)
Transmission Power	-7dBm (0.19mW)
Beacon Order (BO)	8
Superframe Order (SO)	4
Beacon Interval (BI)	245760 symbols
Superframe duration (SD)	15360 symbols

4.1 Distributed Beacon Scheduling

The distributed beacon scheduling is verified and analyzed by deploying MicaZ nodes in the Re-Mote test-bed and also by evaluating the operation within the AvroraZ emulator. In this evaluation we demonstrate the operation of a FFD joining the network and determining its beacon transmission offset independently through distributed beacon scheduling algorithm. The purpose of the evaluation is to verify that the calculation of beacon transmission offset and the active superframe duration schedule is distributed without using a centralized node. The scenario uses values of BO=8 and SO=4 being 15 the maximum number of devices that can be scheduled by MeshMAC in a two hops region (cf. equation (6)).

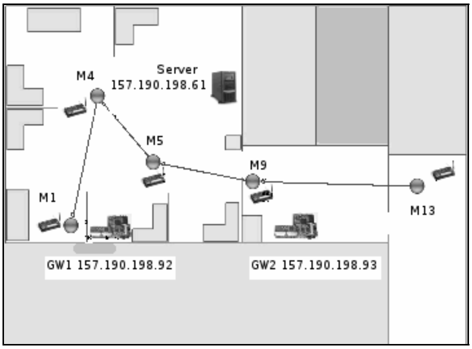


Fig. 5. Re-mote test-bed at CIT lab

Wireless sensor nodes have been deployed in our lab as per Figure 5, where node M1 is programmed to act as PAN coordinator and nodes M4, M5, M9 and M13 are programmed to act as routers. Each device's one hop neighbors are determined by broadcasting packets with fixed transmit power and observing which devices receive them. The results are summarized below in Table 2. The distributed beacon scheduling algorithm specification states that nodes within a two hops neighborhood should not use the same active superframe duration to avoid frame collisions. From Table 2, it can be seen that node M1 and M13 are more than two hops away. Therefore, it is possible to use the same active superframe duration of M1 and M13 to transmit beacons.

Table 2. Node's Neighborhood

Node	One hop direct neighbor
M1	M4,M5
M4	M1,M5,M9
M5	M1,M4,M9
M9	M4,M5,M13
M13	M9

The PAN coordinator (M1 in Figure 6) starts the network formation by reserving a scheduling period for its active superframe duration and another period for broadcast transmissions. As shown in Figure 6, M1 reserves period 1 for broadcast and period 2 for its own superframe duration, its direct neighbors M4 and M5 select distinct periods 2 and 3 respectively. On the other hand, node M9 is not a direct neighbor of M1, but an indirect neighbor through node M4 and M5. To avoid indirect frame collisions, M9 which has the two hops neighbor list acquired from the NLIST procedure, selects the next available empty scheduling period that is not used by node M1, M4 and M5. This can be seen in the results shown in Figure 6 as node M9 selects the non overlapping period 5 as its own active superframe duration period.

In the coordinator scan, node M13 finds M9 as its only beacon neighbor and selects and associates with this node. In the *NLIST REQ* and *NLIST RESP* negotiations node M9 sends the list of its direct neighbors and their respective scheduling periods. In our

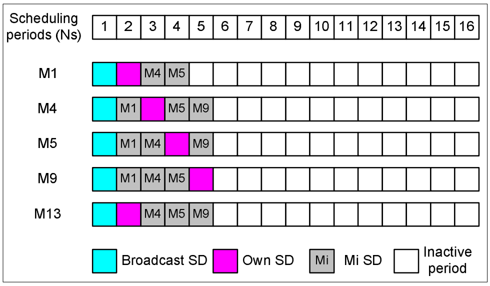


Fig. 6. Distributed Beacon Scheduling example

example, it sends node M4, M5 and M9 with period numbers 2, 3 and 4 respectively. Then node M13 selects the first available empty period from its neighbor table that is the number 1 as shown in Figure 6.

In order to verify the MeshMAC operation for a larger mesh network, we evaluated the algorithm in AvroraZ with 25 nodes that were arranged in a mesh topology as per Figure 7a, where the distance between nodes is about 6 meters. Each node's transmission range is approximately 10 meters with -7dBm transmit power. So, every node can communicate with its horizontal, vertical and diagonal immediate neighbors.

Node0 is programmed to act as coordinator and all other nodes are programmed to act as router nodes, with nodes started in 60 seconds intervals. Using AvroraZ's serial debug monitor and packet sniffer, the beacon schedule is verified by debug messages and also using the Daintree protocol analyzer. This evaluation verified the MeshMAC capability to either select non overlapping active periods within two hops or to reuse them when routers are more than 2 hops away.

4.2 Multipath Mesh Networking

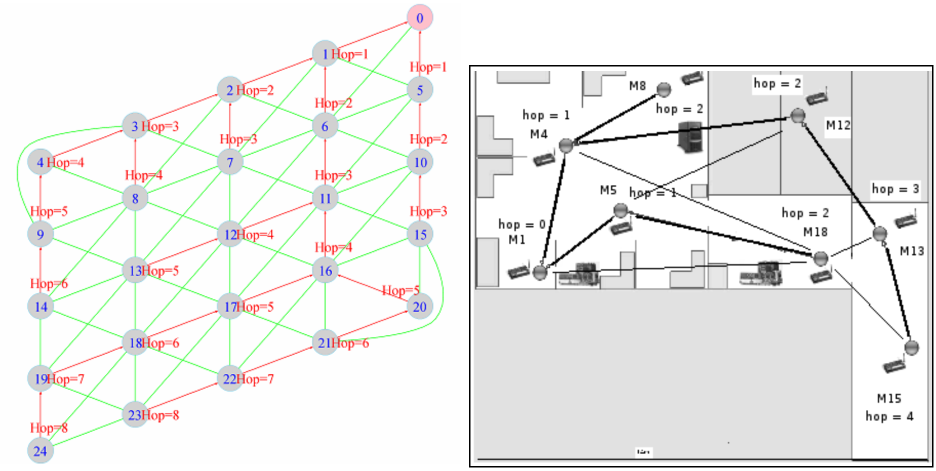
The evaluation of multipath mesh networking is carried out based on a simple routing strategy, where the next route in the path is selected based on a hop count. We created a RoutingExample test application to demonstrate this strategy and evaluated it in our physical test-bed and in the AvroraZ emulator.

In the test one node acts as the source and the PANc acts as the destination. All nodes in the network, apart from the coordinator, are full function devices acting as routers. The PANc starts the network and the other nodes join later using the association procedure. Nodes find their beacon transmission time offset with the distributed beacon scheduling algorithm. Beacon frames are embedded with a hop count that starts from the PANc. Every router increments its parent hop count during the association procedure and transmits the computed value in the beacon frame. Routers obtain their neighbor's hop counts from received beacons and add the hop count field to their neighbor table. This way, the hop count is distributed throughout the whole network. Routing decisions are taken in each node with the rule that, every node forwards data to its neighbor who has lower hop count, which avoids loop problems.

We evaluated this approach in the CIT's Re-Mote test-bed shown in Figure 7b during a 7hours 30minutes. The RoutingExample application collected the possible neighbor nodes that could be used as next hop. The gathered information was analyzed for the number of paths between the source node M15 and the coordinator M1. Since node M1 acts as PAN coordinator, all other nodes M4, M5, M8, M12, M13, and M18 are programmed to act as router nodes. Node M15 is the source node. The coordinator node collects the data and prints the path each packet traversed. The data path shown with the thick line in Figure 7b represents the association connectivity and the thin line shows the mesh data path.

From the collected data, the number of unique paths available to the coordinator based on hop count is plotted in Figure 8b. It can be seen that there exist 8 different paths to reach the coordinator from node M15 to the PANc (node M1) using our simple routing example. In contrast, the standard cluster tree routing approach with beacon scheduling can only provide one route to the PAN coordinator in an energy efficient manner.

We repeated the same multipath evaluation in the AvroraZ emulator using the network of 25 nodes depicted in Figure 7a. The number of paths available to the sink (node0) from each node is shown in Figure 8a. We can see that, compared to the physical test-bed results, the number of possible routes is increased in a larger network[1].

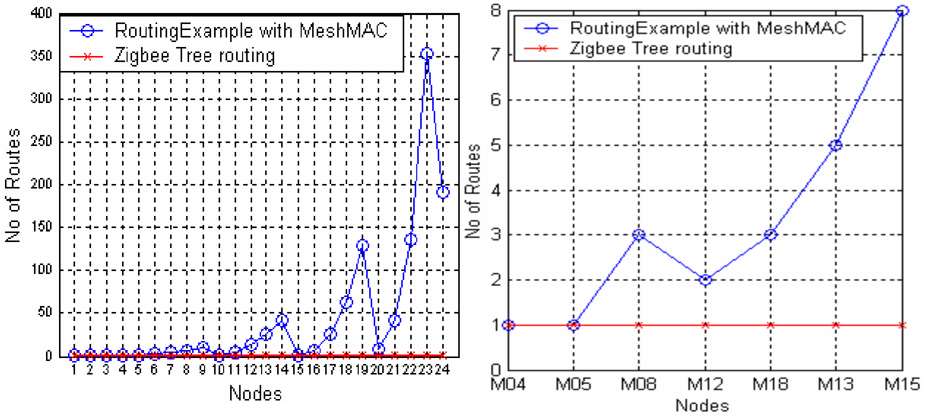


a. AvroraZ WSN emulated network topology

b. CIT WSN physical test-bed network topology

Fig. 7. Wireless sensor network mesh topologies

[1] The following mathematics can be used to check the number of possible loop-free routes r_i that a node i can follow to the destination. Let x be the number of neighbor nodes of i with hop count h_x and let $y \leq x$ be the number of neighbor nodes of i with hop count $h_y < h_i$. The number of routes to the destination r_i can be computed as the sum of the routes r_y available from every neighbor node j with lower hop count than i.



a. AvroraZ emulated network evaluation

b. Re-Mote physical test-bed evaluation

Fig. 8. Number of paths to coordinator node

4.3 Energy Efficiency

The energy efficiency of the proposed MeshMAC is evaluated using the AvroraZ emulator. We ran two different tests: the first evaluates the energy consumption of the MeshMAC with respect to the network node density and the second test evaluates the energy efficiency for the sample network topology shown in Figure 7a.

The influence of network density on energy efficiency is evaluated by monitoring the energy performance when increasing the number of neighbor nodes. The number of neighbors within the transmission range of the node under test is increased every 6000 seconds and the energy consumption of the MeshMAC during that period is measured.

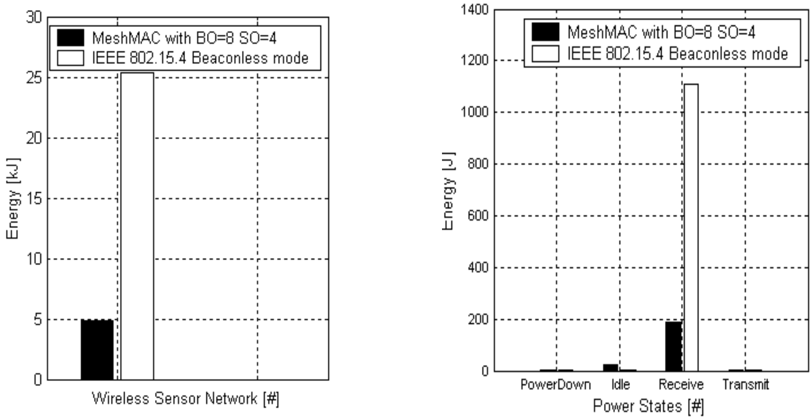
The dependency of the energy spent by the radio on network density is evaluated for the IEEE802.15.4 beaconless mode, the SDS algorithm [7] and the proposed MeshMAC. The results are shown in Table 3. The two synchronous protocols show a vast energy reduction when compared to the standard IEEE802.15.4 beaconless mode. It can also be seen that the MeshMAC energy consumption varies slightly with the network density. This dependency stems from the energy spent for tracking neighbor beacons which is crucial for the distributed beacon synchronization. This, however, is a negligible increase considering that the MeshMAC facilitates broadcast communication through a dedicated SD, while the SDS algorithm does not, computes beacon transmission offsets in a distributed manner, while SDS is centralised, and allows mesh networking which has been shown in the literature [17] to substantially increase network lifetime compared to tree topologies, used by SDS.

Table 3. Radio energy dependency on number of neighbors (N)

Beaconless	SDS	MeshMAC	
Energy (J)	Reduction (%)	Reduction (%)	
		N=1	N=14
338.37	84.23	84.23	83.03

For the second test, we ran the RoutingExample application with the MeshMAC in AvroraZ for the topology shown in Figure 7a and evaluated the MeshMAC's energy performance for the 24 nodes in the network. The total amount of energy spent is calculated by summing up the individual node's energy. The MeshMAC consumes 4831 joules compared to 25377 joules for the beaconless mode. As shown in Figure 9a, the energy saving is more than 80% for the whole network.

The amount of energy spent in each router's radio is also analysed for both the MeshMAC and the beaconless mode. Results for node 8 are shown in Figure 9b. As can be seen, the MeshMAC spends 82% less energy in receive mode than the standard IEEE802.15.4 beaconless MAC. This is because the devices running the beaconless mode are always active, while devices running the MeshMAC algorithm are sleeping during other nodes' active periods. This can be checked through the computation of the duty cycle of MeshMAC devices, by using the configuration test parameters (see Table 1) and equation (3), showing that devices are only active the 6.25% of the time. On the other hand, it can also be seen in Figure 9b how the amount of energy spent in the MeshMAC idle mode increases compared to the beaconless mode. This is because the beaconless mode only switches between transmit and receive modes while the MeshMAC also switches its transceiver to idle mode during the inactive period to save more battery power.



a. Network energy efficiency

b. Radio energy states for node 8

Fig. 9. Energy efficiency evaluation

5 Conclusions

We have proposed the MeshMAC, a distributed beacon scheduling based MAC for the IEEE802.15.4 beacon mode. The objective of the work was to develop a low power IEEE802.15.4 compliant protocol that can enable low power mesh networking through distributed beacon scheduling. The motivation to focus on mesh topologies was to improve reliability compared to the standard IEEE802.15.4/ZigBee cluster tree topology, which offers only a single communication path between source and destination, resulting in single points of failure that could create potential bottlenecks and network partitions.

We have demonstrated that the MeshMAC's operation achieves reliable distributed beacon scheduling and results in energy efficient and highly scalable mesh networking compared to the standard IEEE802.15.4 MAC. Furthermore, routing protocols can utilize the multiple paths enabled by a mesh network to achieve load balancing resulting in prolonged network lifetime compared to tree topologies. Future work will study aspects of slot allocation strategies, variable superframe durations, and the problem of potential slot allocation collisions when two or more nodes join the network at the same time.

References

1. Ye, W., Heidemann, J., Estrin, D.: An energy-efficient MAC protocol for wireless sensor networks. In: Proc. of IEEE INFOCOM, New York, USA, June 2002, pp. 1567–1576 (2002)
2. IEEE 802.15.4 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks, LR-WPANs (2006)
3. Zigbee Alliance, Zigbee specification. Document 053474r17, http://www.zigbee.org
4. IETF RFC 4944, Transmission of IPv6 Packets over IEEE 802.15.4 Networks
5. El-Hoiyi, A., Decotignie, J.-D., Hernandez, J.: Low power MAC protocols for infrastructure wireless sensor networks. In: Proc. of 5th EW conference, Barcelona, Spain (February 2004)
6. Polastre, J., Hill, J., Culler, D.: Versatile Low Power Media Access for Wireless Sensor Networks. In: Proc. ACM SenSys 2004, Baltimore, MD, USA, November 2004, pp. 95–107 (2004)
7. Koubâa, A., Cunha, A., Alves, M., Tovar, E.: TDBS: a time division beacon scheduling mechanism for ZigBee cluster-tree wireless sensor networks. Real-Time Systems 40(3), 321–354 (2008)
8. Jeon, H.-I., Kim, Y.: BOP Location Considerations and Beaconing Scheduling for Backward Compatibility to Legacy IEEE 802.15.4 Devices, in the IEEE 802.15.5 Task Group 5
9. Burda, R., Wietfeld, C.: A Distributed and Autonomous Beacon Scheduling Algorithm for IEEE802.15.4/ZigBee Networks. In: Proc. of IEEE MASS 2007, Pisa, Italy (October 2007)
10. Landsiedel, O., Wehrle, K., Götz, S.: Accurate Prediction of Power Consumption in Sensor Networks. In: Proc. of 2nd IEEE EmNetS, Sydney, Australia (May 2005)
11. Fonseca, J., Spinar, R.: Re-Mote Testbed Framework, http://code.google.com/p/remote-testbed/
12. CC2420 radiochip datasheet: 2.4Ghz IEEE 802.15.4 Zigbee Ready RF Transceiver. Texas Instruments, http://focus.ti.com/lit/ds/symlink/cc2420.pdf
13. Low Power 2.4Ghz MicaZ mote for Wireless Sensor Networks. Crossbow Technology, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAZ_Datasheet.pdf
14. De Paz, R., Pesch, D.: AvroraZ: Extending Avrora with an IEEE 802.15.4 Compliant Radio Chip Model. In: Proc. of the 3rd ACM PM2HW2N Workshop, Vancouver, Canada, October 2008, pp. 43–50 (2008)
15. Chipcon, Texas Instruments Incorporated, Chipcon Packet Sniffer for IEEE802.15.4, http://focus.ti.com/docs/toolsw/folders/print/cc2430db.html
16. Daintree Networks, Sensor Network Analyser (2008), http://www.daintree.net
17. Shah, R.C., Rabaey, J.M.: Energy Aware Routing for Low Energy Ad Hoc Sensor Networks. In: Proc. IEEE Wireless Com. and Network Conf., March 2002, pp. 350–355 (2002)
18. Crossbow, Inc., The MICAz 2.4 GHz, IEEE/ZigBee 802.15.4 compliant, wireless module, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf