An Energy-efficient Real-Time Communication Framework for Wireless Sensor Networks

Emanuele Toscano, Orazio Mirabella, Lucia Lo Bello
RETISNET Lab
Department of Computer Engineering and Telecommunications
University of Catania
Catania, ITALY
{omirabel, lucia.lobello}@diit.unict.it

Abstract— **This paper addresses the architecture, protocol stack and routing algorithm of a framework, called RTPAW (Real-Time Power-Aware) devised to support energy-efficient real-time communication over Wireless Sensor Networks (WSNs) used in monitoring applications. The aim of RTPAW is to provide soft real-time traffic with an appropriate QoS while reducing the energy consumption of the nodes, which have to work for long periods without the possibility of replacing their batteries. The proposed framework exploits the features of an Aggregation level introduced between the MAC and Routing layers. This layer mainly deals with reducing the amount of energy dissipated, while the Routing layer is entrusted with achieving the desired QoS, in terms of delivery speed, to support the transmission of soft real-time traffic. The paper presents the RTPAW performance and discusses the way there are affected by changes in the operating parameters and network load.**

I. Introduction

A Wireless Sensor Network (WSN) for monitoring applications typically consists of nodes which process their data and exchanging it amongst themselves as well as with a base station via a Sink node. As WSN nodes are generally located in the proximity of or inside the phenomenon they are monitoring, and the environments involved are often remote or hostile to humans, they should be able to function without human intervention for as long as possible. In order to meet the long-lasting autonomy requirement, low-power consumption is the main issue to be tackled. For this reason, and to allow for deployment of WSNs at affordable production costs, low-power processors and very small memories are typically used. This, however, is not sufficient, as the amount of energy consumed by communications in WSNs is usually much greater than that used for processing. There is therefore a major need for protocols able to optimize power consumption, so as to prolong the lifetime of the nodes and thus that of the network as a whole. However, the requirement on power consumption clashes with the need for real-time support, which comes out as WSNs used for monitoring applications mostly feature periodic soft real-time traffic and thus require a way to enforce a minimum data delivery speed so as to meet delay constraints.

The communication protocols for WSNs existent in the literature aim either at minimizing power consumption (e.g., [1], [2] and [3]) or at providing soft real-time traffic with the desired QoS (e.g., [5] and [6]). This paper describes the Real-Time Power Aware Framework (RTPAW), which targets a trade-off between power consumption and delivery speed by exploiting the features of both categories of protocols. An earlier version of RTPAW was sketched in [10]. Here we give a more detailed description. In addition, a performance evaluation of RTPAW performance, obtained by ns-2 simulations, is presented. The sensitivity of RTPAW performance to changes in the operating parameters and network load is also discussed.

II. Related work and Motivation

A. Related work

In order to minimize power consumption, cluster-based routing protocols, such as LEACH [1] and MECH [3], adopt a hierarchical routing strategy. A limited number of always active nodes, called cluster heads, form a backbone, while the other nodes can remain asleep and only wake up when data is being sent. The cluster heads are elected in rotation and remain cluster heads for a certain period of time, called a round. Intra-cluster communication uses Time Division Multiple Access (TDMA). A super-frame is created, in which each node has its own time slot. Once data is acquired, the cluster heads transmit it directly to the base station. Code Division Multiple Access (CDMA) is used in order to reduce the impact of radio interference between different clusters. This approach enables energy saving, but suffers from scalability problems which make it unsuitable for large networks, as it requires clock synchronization at a network level, which is only possible for small networks. Moreover, in LEACH cluster heads communicate directly with the base station. On the other hand MECH supports hierarchical message forwarding, but does not guarantee any QoS.

Another class of routing algorithms has been developed with the aim of providing WSNs with a given QoS. Among them, SPEED [5] and MMSPEED [6]. Based on geographical routing, which is particularly efficient in networks covering a large geographical area, both approaches try to guarantee a minimum speed in data delivery. However, these algorithms were developed on 802.11 and do not target power consumption.

Conversely, the RPAR [4] protocol targets real-time applications and at the same time tries to optimize power consumption, constantly regulating the transmission power. This approach is, however, affected by anomalous behavior in heavy traffic conditions, which tends to favor network congestion. The reason for this behaviour is that, when a node is congested, due to high contention, it has to undergo a large number of retries before transmitting a packet correctly, due to high collision probability. Hence, RPAR increases the transmission power, worsening the situation. In addition to this problem, it has to be highlighted that energy saving is limited, as nodes never go to sleep.

B. Motivation

Our proposal derives from the need to find a communication technique for WSNs that is efficient as regards power consumption and able to support soft real-time traffic. Another highly desirable characteristic is the ability to use, where possible, standard protocols or established protocols that have been widely studied (e.g. in [9]). For this reason in this work we chose to use the 802.15.4 standard [7],[8] for the MAC layer, whereas for the routing layer we envisaged an adapted version of SPEED.

III. THE RTPAW FRAMEWORK

A. Network architecture proposed

As geographical routing is not based on the physical address of a node, but on its position, if there are several nodes geographically very close to each other, not all of them have to be active at the same time. This allows for energy saving. To achieve an alternation between activity and sleep periods, a proper network architecture has been devised.

The RTPAW architecture inherits the main features of cluster-based protocols, but introduces a set of new concepts. The nodes are grouped into clusters, which we call *Aggregated Units* (AUs). The AU structure is different from that of the clusters in the protocols currently proposed in the literature. Here, the nodes in an AU belong to three different categories:

- *Cluster Head* (CH);
- *Relay Node* (RN);
- *Cluster Node* (CN).

In each AU there is one CH, one RN and a varying number of CNs, as shown in Fig. 1. The CH has the task of collecting data from the sensor nodes belonging to the cluster (the CNs) and periodically transmitting it to the RN. The task of the latter is to forward the data to other RNs or the Sink node. In this architecture, therefore, the CH handles transmission within the cluster, while the RN handles transmission outside the cluster.

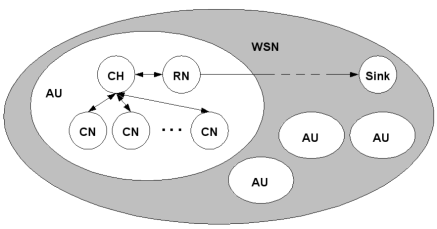

Fig. 1. RTPAW Network architecture

There are three different types of traffic: communications between the CH and the CNs, the ones between the CH and the RN, and the ones between RN and RN or RN and Sink. The first and second types of traffic are periodic and, as we will explain later, mainly aim at the functioning of the AU (and are managed by the Aggregation Layer). The third type of traffic is not periodic, and is handled at a higher layer, as it is relevant to the single AUs (and is managed by the Routing Layer).

Splitting the RN and CH roles implies several benefits. Firstly, the RN is able to perform full time packet forwarding, thus we have better routing performance: if RN and CH roles were unified in the CH, packet forwarding could be performed only when there is no data from CNs. This would require network-wide clock synchronization and reduce the bandwidth utilization (CH would be a bottleneck). The parallelism between RN and CH operations achievable splitting the roles provides a better bandwidth exploitation, and reduces latencies and chances of congestion. Furthermore, having RN and CH roles, in conjunction with the use of different radio channels for nearby AUs, allows for isolation between contention-free intra-cluster communications and contention-based inter-cluster communications, to the benefit of both performance and network scalability.

B. Protocol architecture of the RTPAW Framework

The RTPAW protocol architecture proposed here features an Aggregation Layer which acts as a mediator between the MAC and Routing layers for the combined handling of energy awareness and real-time support. The Aggregation Layer deals with creating and managing the AU and transmitting the first two types of traffic described before. The Routing Layer lies above the Aggregation Layer and forwards packets between AUs, thus handling the third type of traffic. In this architecture, the MAC layer closely collaborates with the Aggregation layer to provide the Routing layer with a uniform view of the set of sensor nodes making up the AU. The basic addressable entity in the Routing layer is therefore not the single WSN node, but the single AU.

The Aggregation layer is split into two sub-layers, with the lower part (called MAC-dependent) which strongly depends on the MAC protocol used and represents an extension of the basic functions needed to implement the level above. This sub-layer has to provide primitives in order to set the radio channel, put nodes to sleep and wake them up, query the battery charge status, perform channel scans (i.e. Energy Detection scans), send and receive frames. Using this set of primitives it is possible to create the MAC- independent sub-layer. The upper layers primitives depend on the protocol used; however, a set of basic primitives should be provided for every protocol. For example, the MAC-independent part of the Aggregation Layer should always provide primitives to create the AU, set up the AU (i.e. beacon period), manage the AU (i.e. CH or RN election), send and receive data (i.e. CN to

CH or CH to RN). While the Routing layer should always provide primitives to send and receive data (i.e. RN to RN), and to send control packets whenever needed.

The main task of the Aggregation Layer is to create and handle the cluster and the aim is to reduce consumption by scheduling periods of activity and sleep periods. The Aggregation Layer may also perform some data processing if it is not single CN data, but some aggregated quantity obtained from multiple CN samples, that has to be forwarded in the WSN. As mentioned previously, the MAC level and the MAC-dependent part of the Aggregation Layer work closely, as the activity periods may coincide with certain states of the MAC Layer. For example, if TDMA is used for transmission inside a cluster, it is possible to make nodes go to sleep during time slots other than their own. Above the Aggregation Layer virtually any routing algorithm providing a certain QoS can be used. The algorithm will operate viewing the whole AU as a single node.

The expected advantages of the proposed architecture are:

- Reduced power consumption, depending on the efficiency of the Aggregation protocol used;
- Advanced QoS management, depending on the efficiency of the Routing protocol used.

Moreover, depending on the aggregation protocol used, as the routing unit is the whole AU, rather than the single node, the AU will continue to live even if several of its nodes cease to function. In addition, the distance between two aggregated units is much greater than that between the single nodes in the network. Therefore, if a geographical routing algorithm is used, the system is less sensitive to the inaccuracy of location mechanisms.

IV. THE PROTOCOLS USED

A. Physical and MAC Layer

At PHY and MAC layers the 802.15.4 standard [7],[8] has been adopted; the non-beacon enabled mode has been chosen to guarantee greater scalability and fault tolerance. In order to avoid inter-AU interference, we create a cell-based architecture using the 16 different radio channels on 2.4GHz. In this manner it is possible to make the radio cells at the Physical layer coincide with the AUs at the Aggregation Layer. Selection of the transmission channel can be automatic during initialization of the nodes, using the Energy Detection scan (ED scan) procedure defined in [7] and [8], or set according to the position of a node. In the latter case, we can create the cellular radio architecture by manually setting transmission channels with the aim of minimizing the interferences among nodes on different AUs.

B. Aggregation Layer

The Aggregation Layer handles data transmission in a single AU. In every AU a super-frame structure is created, and each CN belonging to the AU sends data to its own CH, during the assigned timeslot. It should be noted that the Aggregation Layer super-frame, which is shown in Fig. 2, is not mapped on the 802.15.4 super-frame, but is created at a higher level using the 802.15.4 non-beacon enabled mode. An important difference between RTPAW and the other cluster-based protocols existent in the literature is that, whereas the latter ones usually require network-wide clock synchronization, our protocol requires synchronization at the AU level only.

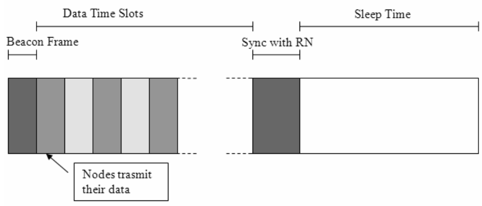

Fig. 2. Aggregation Layer super-frame.

As mentioned previously, in our approach there are not only cluster heads (CHs) and nodes belonging to a cluster (CNs), but also relay nodes (RNs). A CH and an RN are elected in each cluster. The former collects data from the other nodes (except the relay node), whereas the latter forwards packets from one cluster to another. It is necessary to provide a small period of time in which the CH and RN nodes synchronize their data. The RNs must always be active, while the CHs can go to sleep only after synchronization with the RN. All the others can go to sleep and only wake up to receive synchronization signals from the CH or to transmit their data during the assigned time slot. As CHs and especially RNs consume more power than the other nodes, they have to be elected in rotation, in such a way as to balance the power consumption.

The normal functioning of the protocol is divided into three different phases: initialization, election and data transfer. The initialization phase is executed when a node is first activated, whereas election and data transfer alternate, not necessarily at regular intervals. The following is a brief description of the three phases.

1) Initialization: The main aim of the initialization phase is definition of the cellular architecture. We assume that all the nodes know their own position and that they have been randomly arranged with a relatively uniform density. It is therefore possible to create a homogeneous cellular structure, as a grid subdividing the area being monitored into a number of small uniform regions, each hosting a cell. The next step is the first election, during which any of the nodes equipped with the greatest amount of energy can be elected as the CH of his AU. Then the CH elects the RN (as described below) and sends the transmission schedule to the CNs.

2) Election: In cluster-based protocols integrating a cluster head rotation mechanism, whenever a CH is elected it is necessary to reconstruct the whole cluster. In the presence of tight deadlines, or when constant updating of the variables being monitored is needed, this may degrade the QoS. It was therefore decided to separate the distributed algorithm for the first election from the one used later on, which is centralized. In the latter case, at a certain point (after a pre-established time or because its remaining power has dropped beneath a given threshold) the CH autonomously decides which node is to be its successor and notifies the node involved. From the next transmission cycle onwards, the new CH will start operating. The decision regarding the next CH is based on the residual energy of the nodes in the cluster, signalled in the frame that nodes send during normal transmission phases.

Election of the RN is different. The CH elects the RN autonomously when it is requested. An RN whose power has dropped beneath a certain threshold notifies the CH during their synchronization phase. The CH consequently chooses as the next RN whichever of the nodes with the greatest amount of energy has the strongest signal. The former information can be directly devised by the hardware, while the latter can be obtained with a negligible overhead, inserting it in the packets that CN nodes send to the relevant CH.

3) Data transfer: Intra-AU data transfer follows a pre-established synchronized sequence which emulates a super-frame structure in the Aggregation Layer. In this way, it is possible to avoid collisions. The super-frame starts with a beacon frame from the cluster head, used for transmission synchronization in the AU. After the beacon, there are time slots during which the CNs can transmit their data to the CH, using TDMA. During all the time slots assigned to the other nodes, a CN can go to sleep. It must, however, wake up again on time to receive the next beacon frame. The last section of the super-frame is for synchronization between the CH and the RN. In the meanwhile, the RNs form a backbone of nodes that are always active in forwarding packets to the Sink node. They communicate over a single dedicated channel, so during the synchronization phase it is necessary to switch channels temporarily. When the RN acquires the updated CN data from the CH, it forwards it as defined by the Routing Layer. Only RNs can forward data, so they are the only nodes that run the routing algorithm.

C. Routing Layer

As the Routing layer is located above the Aggregation layer, packets are not addressed to single nodes, but to single AUs. So, the only task of the routing algorithm is to forward packets from a source AU to their final destination, usually the Sink node. The scenario RTPAW was devised for is one in which the WSN comprises a large number of nodes and may cover a wide area. For this reason, although the underlying Aggregation layer contributes towards increasing the scalability of the network, the algorithm used for routing between the AUs has to be able to handle a large network without any difficulty. In addition, it is advisable to use a routing algorithm that is as fault-tolerant as possible. As said before, the presence of an underlying Aggregation layer mitigates the system-wide impact of faults occurring in single nodes. Finally, the routing algorithm has to make it possible to achieve the desired QoS, which in our case is delivery speed. A routing algorithm which possesses all these features is SPEED [5]. For this reason a SPEED-inspired approach is used in RTPAW. There are a few differences between the RTPAW adapted version of SPEED and the one described in [5], which are given below.

In RTPAW, the forwarding of packets does not involve singol nodes, but whole AUs (through RNs); therefore the address used here to route data packets is not constituted by the real geographical coordinates of the current RNs, but on the *virtual coordinates* of the whole AU, which are an approximation of the AU centroid coordinates. Another difference is that hop-to-hop transmissions require Acks, and the per-hop delay is calculated according to the formula

$$delay = W_q + (T_{ack} - T_s)/2 \quad (4.1)$$

where W_q is the time elapsed waiting in a queue, T_s is the arrival time of a packet and T_{ack} is the time when the Ack is received.

Finally, as the RNs periodically change, we need some means to keep the network in the steady state even after the election of new RNs. When a new RN is elected, the old one sends the new RN its neighbouring table. As soon as an RN becomes active, it immediately sends a broadcast beacon, so that its neighbours can update their neighbouring tables with the MAC address of the new RN. A second beacon is sent after a short time, in order to minimize the chance that any neighbours will fail to update their table. Then the node can start to send periodic beacons normally, as described in [5].

V. Simulations and Evaluation

In order to evaluate the effectiveness and performance of the RTPAW framework, we simulated the network architecture and the protocol stack of the framework with the *ns-2* [11] tool. For the physical parameters of the simulated nodes, the datasheet of COTS devices, i.e., the MaxStream XBee modules [12], were taken into account. We performed several simulations, with different network loads and number of nodes. In the following subsections, the results obtained in terms of energy consumption, e2e delay and delivery speed are discussed.

A. Energy efficiency of RTPAW

The energy efficiency evaluation was initially performed upon a small-sized network, to avoid excessively long simulation run-times. In this evaluation it is important to have a long simulated time, as our aim is to estimate the average node consumption in the long term. For this reason, we considered a scenario made up of 135 sensor nodes grouped in 9 AUs, each with 15 nodes. The monitoring area is set to 900 m^2 (a square with 30 *m* sides), while the area covered by a single AU is 100 m^2 (a square with 10 *m* sides). Each sensor node sends its data every 10 seconds towards the Sink node. The payload of a CN data packet is only a 4bytes integer, but considering the overhead due to the 802.15.4 and RTPAW headers, the frame length for a CN data packet is 15bytes. The *setpoint speed* [5] (that is, the minimum forwarding speed) is set to 1 *km/s*. Twelve hours of network functioning were simulated.

The efficiency index adopted here is the mean power consumption of a node inside an AU. We evaluated this

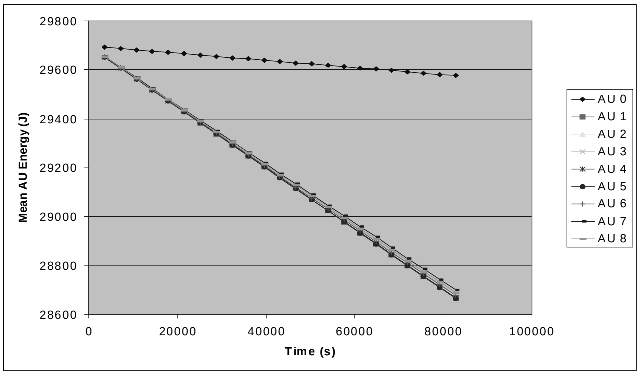

Fig. 3. Mean AU residual energy.

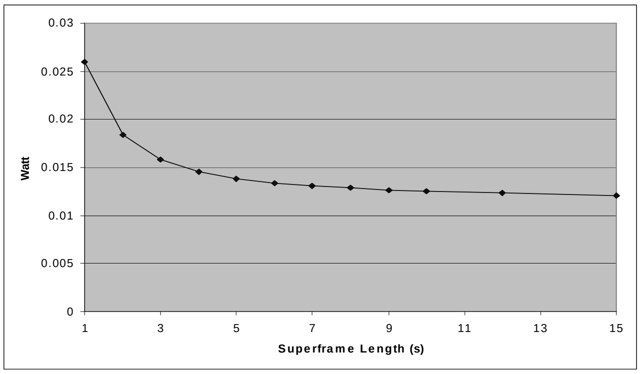

Fig. 5. Mean AU power consumption vs. varying super-frame length.

parameter measuring the residual energy power in each sensor node at regular time intervals (every hour) and then calculating the arithmetical mean of the residual energy obtained from each node belonging to the same AU.

The mean residual energy of each AU as a function of time is shown in Fig. 3. Looking at the figure, we notice that the mean consumption for AU_0 is the lowest. This is because in AU_0 nodes communicate directly with the Sink node. Here the Sink node replace the RN, and no other RN is needed in this AU. The energy consumption of the Sink node is not taken into account in the figure, because we assumed that the Sink node is directly connected to a power source. Fig. 3 highlights two important features of the RTPAW framework. The first is the energy consumption balance among different AUs, which all, except for AU_0, maintain very close energy values along the time axis. The second is the linearity of the AU mean residual energy curves.

Both features are obtained thanks to the Aggregation Layer, which schedules transmission and sleep times in a constant and fair way among the CNs. The only not-constant (and not-linear) part of the AU energy consumption is due to the RN. However, as the RN never goes to the sleep state and for sensor nodes the difference in the energy consumption of transmit and receive states is small, this part could also be approximated to a constant value.

Thanks to the linearity of the residual energy curves, it is possible to estimate the AU mean energy consumption per time unit. As a result, energy consumption over an arbitrary time interval or an approximation of the overall network duration can be calculated (but, in this case, the battery capacity as well as the mean energy consumption have to be considered). The computation of the mean AU energy consumption is equivalent to the computation of the angular coefficient of the line representing the mean AU energy. Making use of the linearity of the AU mean energy consumption, we can approximately estimate the power consumption in very large networks, by simulating only a few minutes (e.g., some dozen) of network functioning. Fig. 4 shows a graph obtained from a simulation of 25 minutes of network functioning. In this scenario we have 1500 nodes grouped into 100 different AUs. The monitoring area is 10000 m^2 (a square with 100 *m* sides), while the area covered by a single AU remains set to 100 m^2. The setpoint speed also remains set to 1 *km/s* too. Fig. 4 shows that the mean consumption of a node in any AU (other than AU_0) in this simulation is just above 12 *mW*. Without RTPAW, assuming that nodes are in the receive state for the 90% of their time and the remaining 10% are transmitting data, the mean consumption is about 163 *mW*. Thus, by lowering the duty cycle of the nodes through the Aggregation Layer, RTPAW reduces the power consumption by an order of magnitude.

What mostly affects power consumption is the length of the super-frame. In fact, with a longer super-frame, nodes can stay asleep for a longer time. As the super-frame becomes smaller, the CH and CN duty cycles increases, so we necessarily have an increase in power consumption, as shown in Fig. 5. The energy consumption obtained in our simulations is, however, much lower than the estimated 163 *mW* without node duty

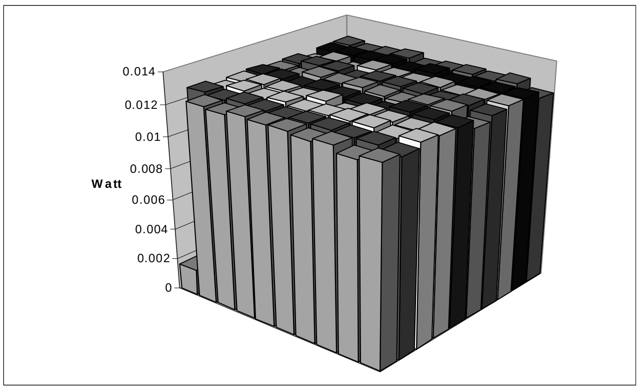

Fig. 4. Mean AU power consumption per time unit (1500 nodes grouped in 100 Aus; each node sends data every 10 *s*).

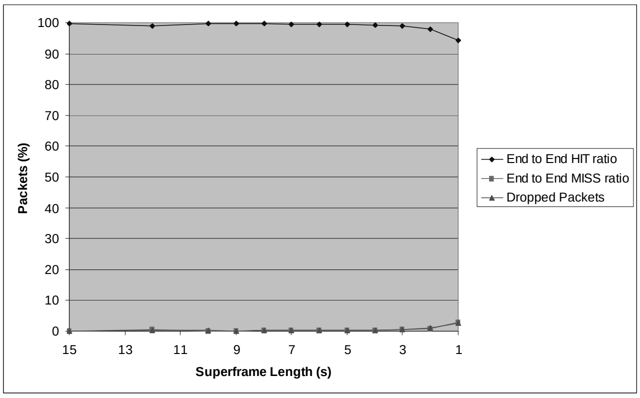

Fig. 6. End-to-end deadline hit ratio, miss ratio and dropped packets vs. varying super-frame length.

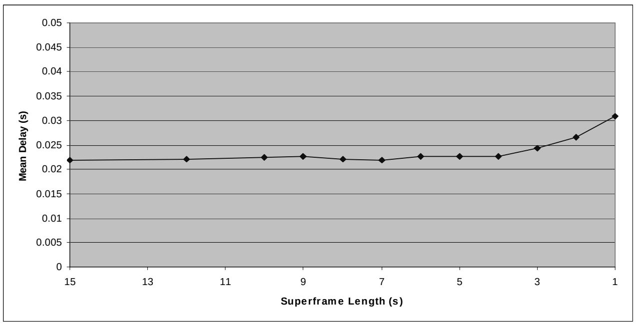

Fig. 7. End-to-end mean packet delay vs. varying super-frame length.

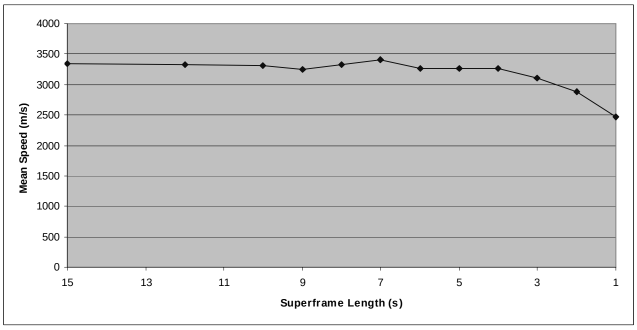

Fig. 8. End-to-end mean packet delivery speed vs. varying super-frame length.

cycles (with the assumptions explained before). Finally, we notice that the plot in Fig. 5 shows an asymptote slightly lower than 12 *mW*. This is due to the RN, which always stays awake, while all the other nodes reducing their power consumption by lowering their duty cycles. So, when the super-frame length increases greatly, the mean power consumption of a node inside an AU converges to the sum of the power consumption of the RN plus the power consumption of the other AU nodes in the sleep state, divided by the number of AU nodes.

B. QoS offered by RTPAW

To assess the QoS support offered by RTPAW the second scenario used in the previous paragraph was adopted. Here, the packet generation period ranges from a minimum of 10 seconds (with an overall network injection rate of 150 packets per second) to a maximum of 1 second (with an overall network injection rate of 1500 packets per second). The payload of the CN data packet is a 4bytes integer, which, considering the overhead due to the 802.15.4 and RTPAW headers, results in a frame length for the CN data packet of 15bytes. The final destination of every packet is the Sink node. The simulated time for this scenario was set to 25 minutes.

The graph in Fig. 6 summarizes the QoS performance in terms of end-to-end (e2e) speed hit ratio, miss ratio and dropped packets obtained by RTPAW. Here we highlight that there is a speed hit every time a packet reaches its final destination with a delivery speed greater or equal to the setpoint speed, and that end-to-end refers to the path from the source RN to the Sink node. In the opposite case, there is a speed miss. Dropped packets are due to network congestion. Referring to Fig. 6, the hit ratio always remains very close to 100% with almost any super-frame length, and decreases to about 95% with a 1-second super-frame (with an overall packet injection rate of 1500 packets per second). The percentage of late and dropped packets are both negligible in almost every simulation, and they increase equally at about 3% in the case of a 1-second super-frame.

The results in Fig. 7 and Fig. 8 represent the mean e2e delay and the mean e2e packet delivery speed, respectively, with a varying super-frame length (and therefore with a varying network load). Both delay and speed values remain almost unchanged until a 3-seconds super-frame (corresponding to a packet injection rate of 500 packets per second) is reached. When the network load increases, QoS slightly worsen, however both delay and speed values remain (considering the setpoint speed set to 1*km/s*) satisfactory.

VI. Conclusions and Future Work

Performance results obtained through ns-2 simulation showed the good behaviour, in terms of both QoS support and energy consumption, of the RTPAW framework. Future work will address implementation on COTS ZigBee modules and the development of novel routing protocols for RTPAW.

References

[1] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-Efficient Communication Protocols for Wireless Microsensor Networks (LEACH)". Proc. of the 33rd Hawaii International Conference on Systems Science-Volume 8, 2000, pp. 3005-3014.

[2] A. Manjeshwar, D. Agrawal, "TEEN: a Routing Protocol for Enhanced Efficient in Wireless Sensor Networks", Proc. of the 15th Internat Parallel and Distributed Processing Symp, 2001, pp. 2009-2015.

[3] Ruay-Shiung Chang and Chia-Jou Kuo, "An Energy Efficient Routing Mechanism for Wireless Sensor Networks", Proc. of the 20th Int. Conf. on Advanced Information Networking and Applications, 2006.

[4] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, T. Abdelzaher, "Real-time Power-Aware Routing in Sensor Networks", 14th IEEE Internat. Workshop on Quality of Service, 2006, pp. 83-92.

[5] T. He, J. Stankovic, C. Lu, T. Abdelzaher, "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks", Proc. IEEE Int'l Conf. Distributed Computing Systems, 2003, pp. 46-55.

[6] E. Felemban, C.G. Lee, E. Ekici, "MMSPEED: multipath Multi-SPEED protocol for QoS guarantee of reliability and. Timeliness in wireless sensor networks", IEEE Trans. on Mobile Computing, 5,6, June 2006.

[7] IEEE 802.15.4 Standard Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Standard for Information Technology, IEEE-SA Standards Board, 2003.

[8] IEEE 802.15.4-2006 Standard Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Standard for Information Technology, IEEE-SA Standards Board, 2006.

[9] B. Bougard, F. Catthoor, D.C. Daly, A. Chandrakasan, W. Dehaene, "Energy efficiency of IEEE 802.15.4 standard in dense wireless microsensor networks: modeling and improvement perspectives", Proc. of Design, Automation and Test in Europe. IEEE, 2005,V1, pp.196-201.

[10] E. Toscano, G. Kaczynski, L. Lo Bello, "RTPAW: a Real-Time Power Aware Framework for Wireless Sensor Networks", WIP Proc. of the 13th IEEE Real-Time and Embedded Technology and Applications Symposium, Bellevue, USA, Apr. 2007, pp. 60-63, available at http://cse.seas.wustl.edu/Research/FileDownload.asp?733.

[11] The Network Simulator – http://www.isi.edu/nsnam/ns.

[12] Maxstream XBee datasheet – http://www.maxstream.net.