Dynamic Data Prioritization for Quality-of-Service Differentiation in Heterogeneous Wireless Sensor Networks

Majid Nabi*, Milos Blagojevic*†, Marc Geilen* and Twan Basten*†
*Department of Electrical Engineering, Eindhoven University of Technology, the Netherlands
†Embedded Systems Institute, Eindhoven, the Netherlands
Email: {m.nabi,mblagojevic,m.c.w.geilen,a.a.basten}@tue.nl

***Abstract*—In many applications of Wireless Sensor Networks (WSNs), heterogeneity is a common property in terms of different sensor types and different circumstances like node location, link quality, and local node density. In many applications, there are several different sensor types with entirely different Quality-of-Service (QoS) requirements. The requirements may also vary over time according to the application scenario and also due to network dynamics. Different requirements appeal different approaches while forwarding sensed data through a multi-hop communication network. This paper proposes a dynamic priority assignment strategy to be used for data routing in heterogeneous WSNs aiming to fairly propagate information according to its importance and requirements. To cope with heterogeneity and dynamics, nodes in the routing path dynamically compute priorities for individual data items according to the attached QoS requirements. We apply the proposed strategy for a healthcare monitoring application scenario which consists of an ambient network and several mobile clusters of nodes in the form of Wireless Body Area Networks (WBANs). The nodes have very different requirements and WBANs show a high mobility in the network with more stringent demands. The results show a large improvement in the achieved QoS for more demanding information.**

I. Introduction

In many applications of Wireless sensor networks (WSNs), there is a large variety between different sensor nodes in the network in terms of Quality-of-Service (QoS) requirements. Moreover, different environment situations for different nodes cause more heterogeneity in the network. The relative sensor node position, different distances to the sink nodes, nonuniform network density, different interference levels, and varying quality of wireless links are some sources of environment heterogeneity in WSNs.

Both QoS requirements and the surrounding situation for a sensor node are prone to vary over time. Mobility is one important source of environment variations and it sometimes can entirely change the network topology and density depending on the mobility level. The QoS requirements of a sensor node can also change over time according to the application scenario. Context-aware data propagation is an example of such changes in which the requirements may change considering the sampled data. Multi-scenario applications can be another example in which the behavior of the network changes over time based on the selected scenario. For instance, in an ambient intelligence application, different scenarios might be used during day and night time.

The spatial and temporal diversity in requirements and environment should be considered while designing communication protocols. Data routing and information dissemination protocols specifically should take this into account to meet the different QoS requirements. Data routing without attention to the heterogeneity may lead to very poor services for important information and an unnecessarily good service for information of lower importance. Regardless of the type of routing protocol, a relaying node in a multi-hop routing path may have several data items waiting to be forwarded at any time. On the other hand, the node may have a limitation in the amount of data that it can transmit in a given time duration. The limitation can be caused because of lower layer constraints like Medium Access Control (MAC) schedules or power consumption limitations. This situation may happen more often for highly congested nodes like nodes closer to the sink nodes. So at any time, the node has to select a subset of the data items in its queue to transmit, and postpone the rest for transmission in the future.

In this paper, a dynamic priority assignment strategy is proposed to be used for data routing in such networks. Nodes in the routing path calculate priorities for the existing data items in the queue according to the relative requirements and the history of each data item. QoS requirements are defined individually for each data item at the time of initiating the item at the source node. This way, the QoS requirements are not labeled to the sensor nodes and so it can be changed over time allowing to handle temporal variations.

As a concrete application with a considerable diversity in the QoS requirements, we consider continuous patient monitoring. In such an application, WBANs are responsible for sensing and propagation of biological signals of patients. On the other hand, an ambient sensor network is used to monitor the ambient parameters like temperature and humidity as well as relaying both WBAN and ambient information to the sink nodes in the network. The WBANs form clusters of wireless sensor nodes that are mobile and also exhibit mobility within the cluster (human body movements).

In [1], we proposed MCMAC, an optimized TDMA-based MAC protocol for mobile clusters. We developed this protocol to be used as the MAC layer for communication in a healthcare application. However, because of high diversity between sensor nodes in this application, special attention

978-1-4577-0093-4/11/$26.00 ©2011 IEEE

should also be given to upper layers like data dissemination. Data items from body sensor nodes are of higher importance than ambient information. Moreover, different body sensor nodes have different requirements. Further, based on the health status of the patient, the QoS requirements can vary. Note that in such an application, the sensor nodes with more important data items basically have high mobility. We applied our priority calculation mechanism for data dissemination on top of the MCMAC layer to show how it behaves in a highly heterogeneous network and improves QoS metrics with different constraints. The scope of the proposed service is not limited though to this specific scenario and can be used for many heterogeneous networks.

The next section reviews related work for priority-based data routing. Section III describes our healthcare application scenario, and the used protocol stack. The dynamic priority-based routing mechanism and priority assignment strategy are presented in Section IV. The protocol evaluation results are given in Section V. Section VI concludes.

II. Related Work

Data prioritization has been used for data routing in several protocols for WSNs. This section reviews the existing approaches in using priorities in WSNs and states the goals and the contribution of this paper.

In [2], a priority-based routing path selection mechanism is exploited for a proposed multi-path routing protocol (PRIMP) which is based on the directed diffusion [3]. Actually, sampled data items are not prioritized in this protocol. Instead, each gradient is given a priority tag based on its accumulated hop count to the sink or the remaining energy source of nodes in that particular routing path. The source node then uses the priority tags of all received gradients to select the best.

The Priority-based Dynamic Adaptive Routing (PDAR) protocol is proposed in [4] aiming to balance the energy consumption while providing better service for significant information. The protocol is based on a former routing protocol for multi-hop wireless ad hoc networks called Dynamic Source Routing (DSR) [5] with the emphasis on congestion prediction and priority scheduling for data routing. Data packets are categorized into two classes of vital and common packets. Accordingly, every node in the routing path maintains two separate data queues, each dedicated to a certain class of packets. The packets in the higher-priority queue (vital packets) are always sent before packets in the lower-priority queue (common packets). Data priorities are supposed to be determined by the application.

In a recent work, presented in [6], the Priority-based Hybrid Routing (PHR) mechanism is proposed in which the characteristics of the sensed data determine its priority. An abrupt change in the data stream reveals the importance of the new data. Consequently, a multi-path diffusion-based mechanism is used for forwarding the packets of high importance to provide a more reliable and faster data delivery. A single-path routing mechanism based on the known Ad-hoc On-demand Distance Vector (AODV) [7] approach, that is prone to data loss, is exploited for normal packets.

Each work has a specific criterion for assigning the priorities and a particular means for providing proper services according to the priorities. In this paper, we propose a mechanism to dynamically assign priorities to data items waiting to be forwarded at any node regardless of the type of routing structure with a focus on considering dynamic heterogeneity in the network. The mechanism aims to provide differentiated services for data items according to their QoS requirements.

We also consider scenarios in which the requirements change over time. To provide such a flexibility, instead of attaching priority values to the data packets, relative QoS requirements are attached to the individual data items by the source nodes. Then the priorities are calculated at each relay node in the routing path taking the attached QoS requirements, and the history of the data item into account. Doing so, firstly, a source node can change the requirements for its data items at any time. Changes in the requirements can be the consequence of changing the running scenario or, for example, based on tracking the sampled data itself for detecting special events or situations [6]. Second, as the history of the data item (for instance the time it spent on the path) is taken into account, dynamic priority calculation provides appropriate services for nodes farther away from the sink node. This is specifically interesting for mobile nodes for which the hop-distance to the sink node varies over time.

III. Motivating Application

In this section, we explain an illustrative healthcare application scenario, the intended network architecture, and the used communication protocol stack. This application has been indeed our initial motivation for exploiting our dynamic priority calculation mechanism, as it shows high diversity in the requirements of different sensor nodes, high mobility of sensor nodes in the face of cluster mobility, and a high demand for multiple scenarios with different QoS requirements.

A. Application Scenario

We consider healthcare monitoring applications such as elderly care or monitoring patients with special chronic diseases like COPD (Chronic Obstructive Pulmonary Disease). In this scenario, the patient's home is equipped with sensors to measure ambient parameters like temperature and humidity. Sensors may be installed on the walls or also on the furniture such as chairs, beds, and electronic equipment to monitor the activity pattern of the person. Sensor nodes are wireless for ease of installation and support of limited mobility, like refurnishing the house.

On the other hand, the patient is equipped with a WBAN. The sensors are placed on several positions on the body to measure vital biological signals. Temperature sensors (for fever), ECG (heart status), blood pressure sensors, SpO2 respiration sensors, GSR (Galvanic skin response) sensors, and accelerometers are some common biological sensors in a WBAN. A subset of these sensors may be selected based on the condition that patients are facing.

Some nodes (sinks) in the the network collect and process data from body sensors as well as ambient information. Sink nodes can also send information to a medical center through a wired or wireless network, receive feedback, and inform the patient. A reason for having more than one sink is to have better services like better latency and reliability. In a typical deployment, we can put for example a sink on every floor of the building.

A high heterogeneity and dynamism is observed in this application scenario. First, body sensor information is much more important than ambient information. Moreover, within a WBAN, sensor nodes form a heterogeneous mix with quite different requirements. An ECG sensor may need a high sampling rate while a few samples per day is sufficient for an SpO2 sensor. Further, patient movement is actually a group mobility of several sensor nodes which makes the network topology quite dynamic.

There also many changes in QoS requirements over time. The sampling rate, the required latency, and the importance of sensed data change according to the patient's condition. In an emergency situation, a very fast and reliable data delivery is needed. For instance, precise information from the ECG sensor might also be required whereas just transmission of the heart beat rate once per 10 seconds is enough in a normal situation. Ambient information can also have varying requirements. It may be reasonable that the ambient data sampled by sensor nodes closer to the patients location are of more interest. Therefore, different sampling rates may be used based on the relative distance to the WBANs.

B. Network Architecture

According to the application scenario, we categorize the sensor nodes in the network into two classes: *static* nodes and *mobile cluster* (MC) nodes. Suppose that $S = \{s_1, s_2, \cdots, s_{N_s}\}$ is the set of N_s static sensor nodes. These nodes are normally static but they can be relocated (limited mobility). Moreover, a small subset of the set of static sensor nodes is considered as the sink nodes ($Sinks \subset S$). The number and location of the sink nodes are determined according to the circumstances in the real deployment considering the overall QoS requirements.

On the other hand, we approach the WBANs as mobile clusters of sensor nodes. These nodes show a high mobility in the sense of both group mobility and individual mobility within the cluster. In a deployment, we assume that we have N_{mc} mobile clusters which can have different numbers of nodes. Assume that $MC_i = \{c_1^i, c_2^i, \cdots, c_{N_i}^i\}$, $1 \leq i \leq N_{mc}$ is the set of N_i sensor nodes of the i^{th} mobile cluster. The total number of sensor nodes in the network (network size N) is then $N = N_s + \sum_{i=1}^{N_{mc}} N_i$.

The information of all sensor nodes in the network should reach a sink node through multi-hop communication. The static sensor nodes are responsible for such data forwarding in the network for data from both mobile cluster nodes and static nodes. In contrast, the MC nodes just transmit sampled data from their own cluster and do not participate in forwarding the information of other nodes outside their cluster. This is because of tighter power consumption restrictions for the sensor nodes in WBANs. The intra-cluster communication is not relevant here. In one scenario, every cluster can have a gateway node that gathers sampled data over the cluster and forwards it to the static network. In an alternative mechanism, every MC node can directly send its data to the static network. In any case, static nodes route that information to sink nodes using multi-hop communication.

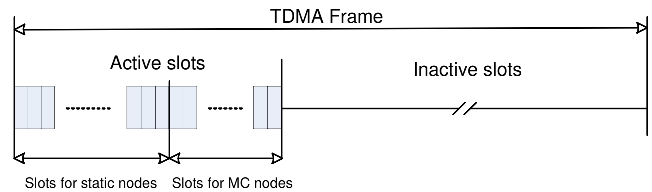


Fig. 1. The structure of a TDMA frame in the MCMAC protocol.

C. Medium Access Control

In [1], we developed the MCMAC protocol to support cluster mobility in WSNs. The protocol is based on the Time Division Multiple Access (TDMA) strategy and is specifically designed to be used as a bottom layer of the dissemination mechanism that we use in our application. A TDMA frame consists of equal size time slots from which a small subset are active slots and the rest are inactive slots. All transceivers are off in the inactive slots to reduce energy consumption. A specific subset of the active slots in every frame is dedicated to the mobile clusters. MC nodes perform a hybrid TDMA with fixed slot assignment and Carrier Sense Multiple Access (CSMA) mechanism for using these slots. Static nodes dynamically occupy one slot of the remaining active slots for their transmission which is unique in their 2-hop neighborhood. Thus, it supports the limited mobility of the static nodes which our scenario may have. Fig. 1 depicts the structure of a TDMA frame in the MCMAC protocol.

Essentially, every node in the network listens to the channel in the active slots of the frames to receive information of both static nodes and MC nodes. We included an optimization mechanism to reduce the power consumption of listening though. Static nodes just listen to the active slots dedicated to mobile clusters when they expect a cluster in their vicinity.

The MAC protocol is equipped with a time synchronization mechanism which is a prerequisite for implementing a TDMA-based protocol. The mechanism provides a global time t as well as a method to align the TDMA frames. As the nodes can transmit their packets in their transmit slot, the possible amount of information that a node can send in a round (TDMA frame) is constrained by the size of the slots.

D. Data Dissemination

There are two different data flow directions in the network. First, all sampled data from all the sensor nodes in the network should be gathered at the sink node. This is actually an all-to-one dissemination scheme. The other data flow direction is broadcasting information from the sink node to all sensor nodes. This information can be control commands or information of different communication layers including the application itself. This is a one-to-all dissemination scheme. We use a data dissemination mechanism on top of the TDMA-based MAC layer to establish both types of data flows.

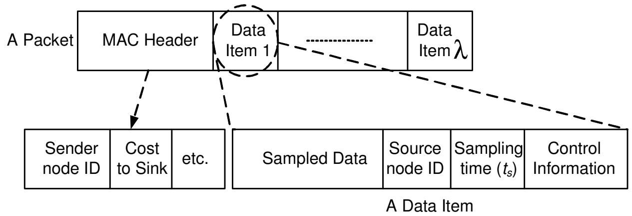


Fig. 2. The structure of a transmission packet in data dissemination.

The high cluster mobility in the network and the different data dissemination schemes suggest a non-deterministic routing strategy like epidemic mechanisms. Static deterministic routing structures like a tree are not appropriate for such a mobile network. We use a directed flooding strategy specifically adapted for our application to forward sensed information toward a sink node. On the other hand, a pure gossip based data dissemination is exploited for sink data to be disseminated to all nodes in the network.

In our terminology, we use *packet* to refer to what a node broadcasts in its transmit slot. As mentioned before, the maximum packet length is determined by the slot length. Packet length limitations can also come from the radio chip specifications or power optimization issues. A packet consists of the MAC header (including the ID of the transmitter) as well as data items. A *data item* is a single sample of data of a node. It includes the sampled data itself, the ID of the source node, the sample time, and other required control information. The number of data items per packet λ (≥ 1) depends on the size of the packet, the MAC header size, and the size of each data item. Fig. 2 illustrates the structure of a transmission packet in this dissemination mechanism.

Every static sensor node maintains a pool of data items that it has received and has to forward. We assume that the nodes have sufficient memory to store one data item (the newest received one) of each node (maximum N entries). The maximum might be reached in very congested nodes like nodes close to the sink nodes. An entry of the data item pool includes the received data item as well as some control information like the arrival time (t_a), and transmitted (*TXed*) and acknowledged (*Acked*) flags. The *TXed* flag shows if the data item has been transmitted so far. Accordingly, the *Acked* flag shows if an acknowledgement from a node closer to the sinks has been received for this data item. This does not apply for sink data items that are going to be broadcast to the whole network. Sink data items are recognized by the source ID existing in the structure of the data item.

To the aim of directed flooding of sensed data items towards the sink nodes, an underlying minimum cost computation service is used. In a simple case, the cost can be the hop-count to the sink node. Using such a service, static nodes know their *cost* to reach the sink and use that for data forwarding. The nodes also include their cost in their transmission packets so that the receiver node knows the status of its neighbors to make decisions about the arriving packets. Because of the high mobility of mobile clusters in the network, the cost to the sink will change frequently over time and so is not reliable. Therefore, it is not calculated and not used for the MC nodes.

Algorithm 1: Processing the received packet pkt.

Data:
$x.t_s$: Sample time of the data item x
$x.src$: Source node of the data item x
TxNode: Direct transmitter of pkt
Cost: Cost to the sink (minimum hop-count)

RECEIVEPACKET(pkt)
foreach *RxItem* $\in pkt$ **do**
    $CItem$ = RetrieveFromDataPool($RxItem.src$);
    if IsSink($RxItem.src$) $\vee$ IsMCNode($TxNode$) **then**
        if $RxItem.t_s > CItem.t_s$ **then**
            StoreInDataPool($RxItem$);
        end
    else
        if ($TxNode.Cost \geq Cost$) **then**
            if $RxItem.t_s > CItem.t_s$ **then**
                StoreInDataPool($RxItem$);
            end
        else
            if ($CItem \neq NIL \wedge RxItem.t_s \geq CItem.t_s$) **then**
                StoreInDataPool($RxItem$);
                SetAcked($RxItem.src$);
            end
        end
    end
end

Every static node may receive several packets at each round, each including several data items. Algorithm 1 shows the process that a node performs upon receiving a packet. The data items directly received from an MC node as well as data items originated by a sink node are processed without considering the cost to the sink. For other data items, if the cost of the current node to the sink is less than the cost of the direct transmitter of the packet, the item is considered for storing in the pool. Otherwise, it means that the item is transmitted from a node closer to the sink and so the current node will not participate in forwarding that item. However, it is considered as an acknowledgement for the existing data item in the pool. Note that a received data item originating from node *src* will be stored in the pool only if it is newer than the possibly existing one from node *src*. Besides this, the node inserts its own data into the pool whenever it samples new data.

At every round, the node makes a packet by selecting λ data items from its pool and delivering that to the MAC layer to be transmitted in its dedicated transmit slot. The selection mechanism has a very strong impact on the network performance. In the next section, we propose a priority assignment strategy to provide appropriate services according to the QoS demands. Here we use this strategy to assign a relative priority value to each data item existing in the pool and then select the λ items with the highest priority.

IV. EXPLOITING DYNAMIC PRIORITIES

In this section, we present our approach for data routing taking the desired individual QoS requirements of the data items into consideration. First, the overall approach for data item selection is presented. We then define the QoS requirements that we consider here and explain how these requirements are announced by the sensing nodes. The strategy used for priority assignment is then presented.

A. Priority-based Data Forwarding

To have a better understanding of the proposed method, we recall the main goals of exploiting such a mechanism. The first goal is to disseminate or route data in a dynamic and heterogeneous WSN in order to provide proper services aiming to meet all QoS requirements to the extent possible with the available capacity. Note that one can state QoS requirements which cannot be met considering the situation in the deployed network. The point here is that given a network deployment with heterogeneity in the requirements and a communication protocol stack, we extract useful information to detect relatively more important data items at any time to be forwarded earlier. The second goal is to provide the flexibility to have time variant QoS requirements. The sensor nodes generating data items can then change their requirements based on the situation.

Assume that sensor node s_j has k data items waiting to be forwarded. It is able to transmit a limited number of items (λ items) in a given time duration. If $k \leq \lambda$, node s_j can forward all items. However, there may be congestion in many nodes in the routing path. The data traffic depends of the position of the node and the network data load. In the case that $k > \lambda$, a selection mechanism must be applied.

A uniformly random selection strategy has been used in [8]. Doing so, a statistically equal chance is given to all data items waiting to be forwarded. Besides that, in [8], the nodes always send their own data items. Moreover, if a data item is going to be removed from the local memory because of lack of space, it will be transmitted hoping that some other nodes store it. However, this is not the best option when there are different QoS requirements for different data items.

Our approach here is to assign a priority value to the k data items waiting to be forwarded and then select λ items with the highest priority values. The priority values are assigned dynamically according to the QoS requirements of individual data items, and their history in the routing path so far.

B. QoS Requirements

Any source node has particular requirements for its sampled data. The requirements can be specified based on the type of the sensor, the current situation, and also the observation of the data stream to detect sudden changes revealing events, like in [6]. In general, we assume that $\overline{\mathbf{Q}} = (Q_1, \cdots, Q_d)$ is the vector of requirement values of d different QoS requirements that are of interest in the running application and are specified for each data item. Suppose that the sensor node s_i wants to initiate a new data item at time t_s referred to as $\Gamma_i^{t_s}$. The node s_i decides about its QoS requirements and includes the specified vector $\overline{\mathbf{Q}}_i^{t_s}$ into the data item. For priority calculation by the nodes in the routing path, some application dependent parameters might also be needed to be attached to the data items. Sampling time of the source node (t_s) in our application scenario is an example.

The overhead of adding QoS requirements $\overline{\mathbf{Q}}_i^{t_s}$ and other parameters to the data items depends on the level of heterogeneity in the requirements existing in the network. A specific quantization method is used for representing this information which does not have to be uniform. For instance, if we have four different levels for Q_1 in a scenario, it can be represented by just two bits in each data item. All nodes in the routing path are then able to calculate the absolute value of Q_1. Moreover, some of the required parameters may overlap with the existing payloads in the routing mechanism. For instance, we require the sampling time of each data item for priority calculation which already exists in the payload of the data items in our application scenario and is used by the data dissemination protocol (Section III.D).

The QoS metrics and their interpretation strongly depend on the expectations of the running application and can not be unique for all applications. To better illustrate the mechanism, we specifically consider *reliability* and *latency* requirements as two common QoS metrics related to the individual data items, which are also of interest in our motivating healthcare application. The reliability requirement $0 < R_i^{t_s} \leq 1$ of a particular data item is defined as the probability that the data item will reach a sink node. The time distance between initiating the data item $\Gamma_i^{t_s}$ at the source node and the arrival time at the nearest sink is the latency $Lat_i^{t_s}$. Accordingly, the latency requirement $L_i^{t_s}$ is a constraint on the achieved latency. If this constraint cannot be met, the value of $Lat_i^{t_s} - L_i^{t_s}$ should be as low as possible. For other applications, a packet may become useless as soon as the latency constraint is not met.

The node s_i includes $\overline{\mathbf{Q}}_i^{t_s} = (L_i^{t_s}, R_i^{t_s})$ as the required latency and reliability into the data item that it samples at time t_s. Here, we also include the sampling period T_s^i of node s_i. This parameter actually informs the relaying nodes in the path about the time of initiating the next data item by node s_i. This parameter is required for routing mechanisms like ours in which an older data item of node s_i will be overwritten by the received newer item even before transmission because just one data item of node s_i can be stored in the node s_j in the routing path at any time.

C. Dynamic Priority Assignment

The priority values, in our mechanism, are not assigned to the data item at the source node. Instead, the requirements of individual data items are included into the messages. Dynamic priorities are then calculated at each node in the routing path according to the requirements, the time that the data item has been in the path and the current node s_j. Doing so, dynamic change of the requirements is also supported.

The importance of the data items waiting to be forwarded in a node s_j can be ordered in different ways considering different QoS metrics. Therefore, d priority values are calculated taking the requirement of each metric separately into account. Thus, we obtain a vector $\overline{\mathbf{P}} = (p_1, \cdots, p_d)$ called *partial priority* values for each data item in which value p_l is the calculated priority value related to the QoS requirement Q_l where $1 \leq l \leq d$. The partial priorities are calculated considering the type of the metric and its interpretation in the running application. Finally, we have k partial priority vectors for k existing data items in node s_j denoted by $\overline{\mathbf{P}}_m$ where $1 \leq m \leq k$.

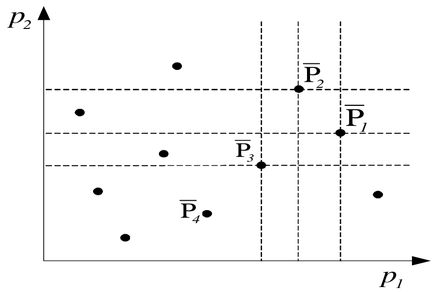


Fig. 3. An example of the obtained partial priority vectors in a two-dimensional space for nine data items waiting to be forwarded.

The next step is to extract a single priority value for each data item according to its calculated partial priority vector as follows.

$$P_m = f(\overline{\mathbf{P}}_m) \ , \ 1 \leq m \leq k \qquad (1)$$

The exact function can be plugged in according to the expected behavior of the protocol or desired QoS. However, the function $f(.)$ should obviously be monotone by providing higher priority values for *Pareto dominating* [9] partial priority vectors. The partial priority vector $\overline{\mathbf{P}}_{m_1}$ dominates vector $\overline{\mathbf{P}}_{m_2}$ denoted $\overline{\mathbf{P}}_{m_1} \succeq \overline{\mathbf{P}}_{m_2}$, if none of the individual priority values in $\overline{\mathbf{P}}_{m_1}$ is less than the corresponding one in $\overline{\mathbf{P}}_{m_2}$. Monotonicity means that if $\overline{\mathbf{P}}_{m_1} \succeq \overline{\mathbf{P}}_{m_2}$ then $f(\overline{\mathbf{P}}_{m_1}) \geq f(\overline{\mathbf{P}}_{m_2})$ for all $1 \leq m_1, m_2 \leq k$.

Figure 3 depicts an example of the obtained partial priority vectors with two QoS metrics. In this example, $\overline{\mathbf{P}}_3 \succeq \overline{\mathbf{P}}_4$ and so using the function $f(.)$, it is expected to come to priority values such that $P_3 > P_4$. Accordingly, we should have $P_1 > P_3$ and $P_2 > P_3$. But the comparison between vectors $\overline{\mathbf{P}}_1$ and $\overline{\mathbf{P}}_2$ is not straightforward as none of them dominates the other one. The relative priority values in these cases should be calculated according to the criteria in the running application and the nature of the QoS metrics, for example as a weighted average of prioritizing criteria.

For the latency requirement of the data item in our application, we consider at time t the remaining time to expiration of the latency requirement, which is calculated by $L_i^{t_s} - (t - t_s^i)$. Less remaining time to the latency deadline inspires a higher priority for the data item. Nevertheless, there is a chance that the data item is not selected due to high congestion at node s_j and the presence of other items of higher priority; eventually the deadline may expire. In this case, we continue increasing the priority of the data item to give it a higher chance to be selected as soon as possible. Even if the latency deadline has expired, the data item still has value and we try to decrease the final latency of delivering the data item to the sink. In another scenario, which we do not work out here, the ratio of data items reached to the sink meeting the latency constraint can be of higher importance for the running application. In such a scenario, once the latency constraint expired for a data item, its priority should be reduced to create a higher chance for other items that may still make their deadline.

Parameter T_c^i (used for priority calculation) is computed as follows.

$$T_c^i = \begin{cases} L_i^{t_s} - (t - t_s^i) & L_i^{t_s} > (t - t_s^i) \\ \frac{L_i^{t_s}}{(t - t_s^i)} & otherwise \end{cases} \qquad (2)$$

Once the latency deadline expires, the value of T_c^i goes below one and keeps decreasing over time. The partial priority of data item $\Gamma_i^{t_s}$ regarding the latency requirement is then set as $p_1 = 1/T_c^i$.

Accordingly, for the reliability requirement, we consider the expected time for the data item of node s_i to be overwritten by the next data item of this node. $T_s^i - (t - t_a^i)$ can be used as an estimation of arrival of a newer data item. T_s^i is the time of initiating the next data item by node s_i (sampling period).The parameter t_a^i denotes the arrival time of the data item $\Gamma_i^{t_s}$ to the current node s_j. However, this can happen either earlier or later based on the dissemination of the current data item $\Gamma_i^{t_s}$ and the successor items $\Gamma_i^{t_s + n.T_s}$ in the routing path to the current node. Eqn. 3 provides a value regarding the expected time of $\Gamma_i^{t_s}$ being overwritten.

$$T_o^i = \begin{cases} T_s^i - (t - t_a^i) & T_s^i > (t - t_a^i) \\ \frac{T_s^i}{(t - t_a^i)} & otherwise \end{cases} \qquad (3)$$

If a newer data item is not received after T_s^i, the T_o^i goes below one and it keeps decreasing. It means that we are going to give higher priority to this item as it might be closer to being overwritten. The partial priority related to the reliability requirements is then obtained by $p_2 = \frac{R_i^{t_s} \times (1 + N_o^i)}{T_o^i}$, where N_o^i denotes the number of times that data items from node s_i have been subsequently overwritten without being transmitted. Once a data item from node s_i is selected for transmission, the value of N_o^i will be reset.

Both partial priority values calculated so far are related to the time. Higher priority means that there is less time to expiration the deadline if it has not been reached yet. So taking the maximum priority value is a reasonable choice that forces the mechanism to select the items closer to their deadline either related to the latency or reliability. Therefore, the basic component of the priority value for the data item is then obtained as follows which is a monotone function.

$$P'_i = max\{p_1, p_2\} \qquad (4)$$

Based on the strategy of the routing protocol, priority values will be adapted after the transmission or the reception of an acknowledgement for the data item. The final priority value for the data item of node s_i is calculated by Eqn. 5.

$$P_i = P'_i \times (\omega_1 \cdot \overline{TXed_i} \cdot \overline{Acked_i} + \omega_2 \cdot TXed_i \cdot \overline{Acked_i} + \omega_3 \cdot Acked_i) \qquad (5)$$

where $TXed_i$ and $Acked_i$ show if the existing item from node s_i has been transmitted, and acknowledged, respectively. Three coefficients $0 \leq \omega_1, \omega_2, \omega_3 \leq 1$ determine the behavior of the protocol for transmitted and acknowledged items. For instance the value $\omega_3 = 0$ gives the acknowledged items the lowest priority. If the routing mechanism does not support an acknowledgment mechanism, this part is simply removed from

Eqn. 5. The value of ω_2 is set based on the strategy of the routing mechanism for retransmitting data items. Moreover, a dynamic value can also be used based on the reliability requirement of the data item and the outgoing links quality of the current node s_j.

V. Performance Evaluation

To investigate the behavior of data dissemination using the dynamic prioritization of data items, we run extensive simulations with various network setups. We implemented the protocol stack described in Section III as well as the priority assignment mechanism in the MiXiM [10] framework on top of the OMNeT++ 4 [11] simulator. We used the specification of *MyriaNed* [12] sensor nodes for the physical layer characteristics. These nodes use a *Nordic* chip as transceiver working in the 2.4 GHz frequency band with a data rate up to 2Mbps. The radio chip has fixed size packets of 32 bytes that can include three data items ($\lambda = 3$) in our protocol stack. The TDMA frame size of the MAC layer is set to one second.

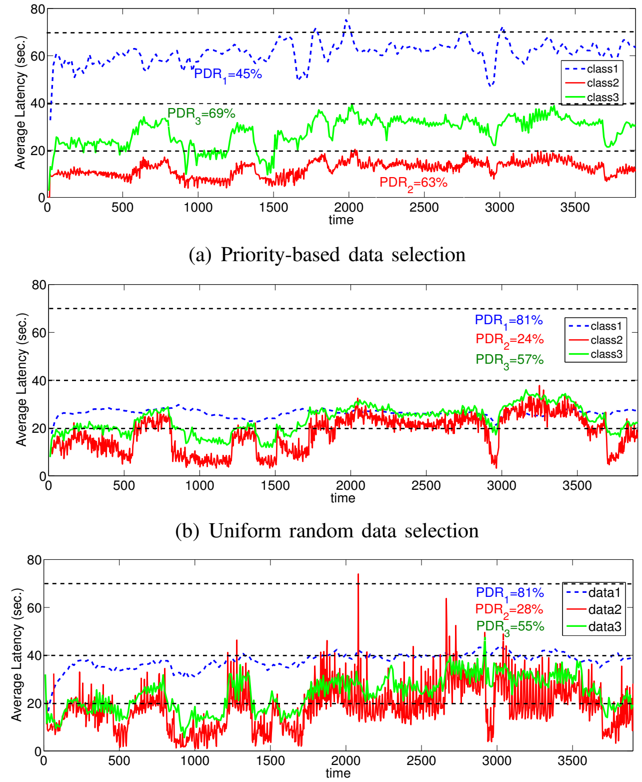
Each simulation was run for 4000 rounds (TDMA frames). To have statistically more reliable results, every experiment was repeated 20 times with different seeds for the random generator. All results shown in this section are the average over all runs with different seeds. To have proper group mobility of mobile clusters as well as individual node mobility within the clusters in our simulation, we used a particular configuration of MoBAN [13] which is a mobility model for WBANs. The same mobility pattern has been used for all simulation runs to have a more controlled comparison.

The data prioritization mechanism is worth considering for networks with a relatively high data load in which the bandwidth of the network should be carefully shared among requesting data items according to their demands. Thus, the simulations have been specifically set up for creating a reasonable level of data load in the network. However, the setups and the QoS requirements are representative for the situation in many applications. To show the effect of the mechanism, we also run simulations using other item selection strategies. We used the strategy used in [8] which is based on a uniformly random data item selection as well as a First-In First-Out (FIFO) strategy. Note that the underlying MAC and routing protocols are exactly the same and the only difference is the item selection strategy.

A. Setup 1: Data Prioritization Behavior

The first simulation setup mainly aims to investigate the effect of using the data prioritization mechanism by comparing to runs without using priorities. We randomly distributed 100 static sensor nodes in a square area. To ensure an even distribution across the area, the nodes are placed with a 10% variance around fixed grid points. The sink node is placed somewhere around the middle of the area. Four mobile clusters ($N_{mc} = 4$) are considered, each containing five sensor nodes ($N_i = 5$), that move using the mobility parameters of a human.

The simulation was run with three classes of QoS requirements which remained fixed for the whole simulation run. The first class includes all static nodes (ambient sensors) that are more delay tolerant with the lowest packet delivery requirements and the lowest rate of data generation. MC nodes are categorized into two classes. Two nodes of each cluster (WBAN) have the highest sampling rates with tightest requirements while the rest require less service. Table I presents the exact values of sampling rates and QoS requirements.



(a) Priority-based data selection

(b) Uniform random data selection

(c) FIFO-based data selection

Fig. 4. The average latency of received data items from all nodes in each class generated at the same time and their average PDRs in setup 1.

TABLE I
QoS requirements of sensor nodes in simulation setup 1
($1 \leq i \leq 100$, $1 \leq k \leq 4$)

class	nodes	T_s	L $Req.$	R $Req.$
Class 1	s_i (static nodes)	20s	70s	0.5
Class 2	c_1^k , c_2^k	3s	20s	0.8
Class 3	c_3^k , c_4^k , c_5^k	10s	40s	0.7

Figure 4 depicts the simulation results for setup 1. The average latency of all received items generated at the same time from the nodes in each class is separately shown for the whole simulation time. Horizontal dashed lines in the figure show different latency constraints. The average achieved Packet Delivery Ratio (PDR) values for each class over the whole simulation time are also shown next to the corresponding latency curves. Note that a part of the variations in the latency is caused by the movement of mobile clusters which are sometimes far away or close to the sink node.

Figure. 4(a) shows the results when our priority-based data selection is used. It is observable that the bandwidth has been shared according to the requirements of each sensor class. Latency and PDR values are quite close to the desired level of each class. In contrast, the result obtained from the

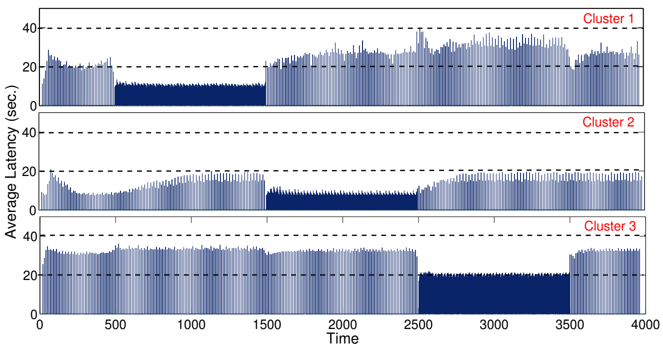


Fig. 5. The average latency of data items over nodes c_3, c_4, and c_5 of each cluster with a time frame of more stringent QoS requirements.

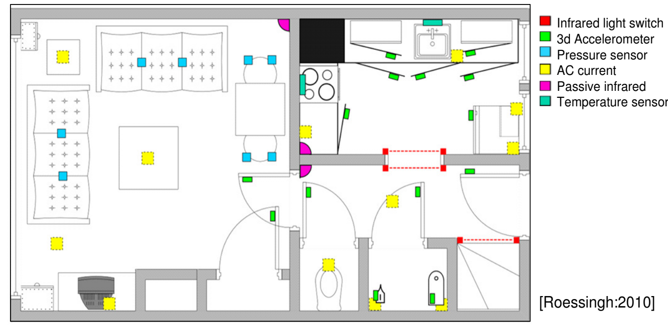


Fig. 6. The floor plan and node placement in the COPD experiment.

random item selection (Fig. 4(b)) shows a very low PDR for class 2 (highest demanding nodes). Although the shown latency values are still close to the constraints, these values are just for the received data items. The majority of items have been lost (PDR=24%). Good PDR (81%) and latency values (almost around 10 sec.) have been achieved for class 1 which are not necessary at all. This happens because all items from different nodes are treated in the same way when a uniform random selection is applied. Since the sampling rate of nodes in class 1 is the lowest, the chance of being overwritten is low. That causes a good PDR for those data items. This kind of bandwidth sharing is actually what our priority-based item selection tries to avoid. Fig. 4(c) shows that a similar result (even worse) is obtained when a FIFO-based item selection strategy is applied. Furthermore, in this case the latency requirement of the nodes in class 2 is very often violated.

B. Setup 2: Dynamic QoS Requirements

We run a second simulation setup aiming to observe the behavior of the mechanism in a network with varying QoS requirements. The setup is mostly the same as for the first setup with the same node classes having the requirements shown in Table I. Here, for some periods, MC nodes in class 3 decide to increase their sampling rates and switch to the same requirements as nodes in class 2 ($T_s = 3s,\ L = 20s,\ R = 0.8$). In a healthcare monitoring application, this can happen for certain patient conditions. Three clusters in this experiment make such a change in separate time frames of length 1000 rounds starting from simulation round 500, 1500, and 2500, respectively. Clusters are static in this experiment so that the reaction of the mechanism to the changes in the requirements is separated from the effect of mobility. Fig. 5 exhibits the average latency of received items over three sensor nodes c_3, c_4, and c_5 of each cluster, separately. The figure reveals how the QoS requirement changes are tracked by our dynamic priority mechanism.

C. COPD Setup: A Realistic Heterogenous Setup

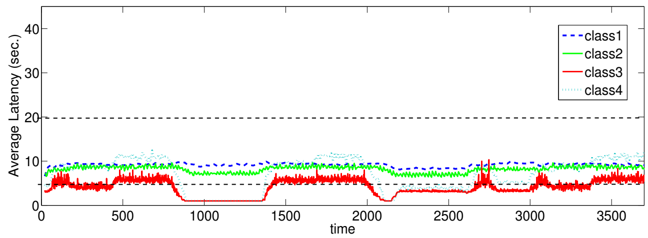
The last set of simulations is done based on the deployment of a real healthcare application. The Roessingh Research and Development center is planing an experiment for COPD patient monitoring. Patients suffering from COPD should be careful about the amount of activity they perform during a day and they have to distribute their energy over the whole day. The main goal of the planned experiment is to investigate the possibility of non-intrusive Activities of Daily Living (ADL) recognition by analyzing the information gathered by a sensor network. The patient's home is equipped with *MyriaNed* [12] wireless nodes with appropriate sensors. Fig. 6 shows the node placement and sensors type of the first floor of the house. Moreover, the patient is equipped with several bio-sensor nodes (ECG sensors, accelerometers, temperature sensors).

The experiment matches with our intended application scenario. There is also a noticeable diversity in the sensors types and QoS demands in this network. Further, the deployment itself exhibits heterogeneity in the network density over the experiment's area. We use the setup of this experiment with exactly the same node placement as shown in Fig. 6 as a case study to test our proposed mechanism. There are 42 static nodes as well as one WBAN including five sensor nodes. Although there are several kinds of sensors in the static part, to simplify presentation of the results, we assume all static nodes to have the same requirements except for the accelerometers (s_{29-42}) that have more stringent QoS requirements. We also have two classes of requirements for WBAN nodes. Table II presents the sampling period and QoS requirements of the four sensor classes in this experiment. Although the setup comes from ADL recognition, we here assume a scenario including medical monitoring of the patient with stringent requirements for data from some of the WBAN sensors. ECG sensors (c_1 and c_2) of the WBAN have the highest possible sampling rate (equal to the TDMA frame length) with highest QoS requirements. This is the most important information which should reach the sink very fast. Other sensor nodes on the body (class 4) have less strict requirements.

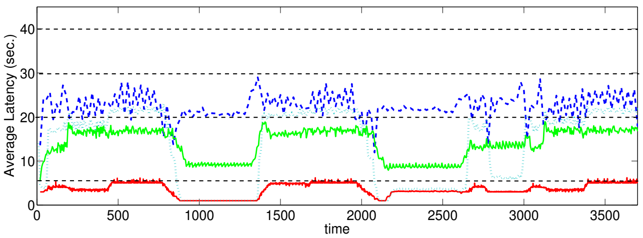
TABLE II
QOS REQUIREMENTS OF SENSOR NODES IN COPD EXPERIMENT SETUP

class	nodes	T_s	L *Req.*	R *Req.*
Class 1	s_i $(1 \le i \le 28)$	20s	40s	0.5
Class 2	s_i $(29 \le i \le 42)$	5s	20s	0.7
Class 3	c_1 , c_2	1s	5s	0.8
Class 4	c_3 , c_4 , c_5	10s	30s	0.7

Figure 7 shows the average latency of received data items over all nodes in each class in a simulation of the setup. Major variations in the latency values are caused by the variations in the relative hop-distance of WBAN nodes to the sink node due to the mobility of the patient. In the time frame between rounds 900 and 1400, the WBAN was one-hop away from the



(a) Uniform random data selection



(b) Priority-based data selection

Fig. 7. The average latency of received data items for each class of nodes in COPD experiment setup.

sink. So the latency is one second (one TDMA round) using either the random or the priority-based item selection.

The results obtained by performing priority-based item selection (Fig. 7(b)) show that the mechanism properly distributes the network capacity among data items according to the requirements of each class. All classes satisfy the required latency level in Fig. 7(b). In contrast, Fig. 7(a) reveals that the latency constraint for class 3 (which has the tightest requirements) is sometimes violated using the random item selection. It happens even though the achieved PDR value for this class is very low (38%) and not acceptable. Again we emphasize that the latency values are just for the data items that reach the sink node. To gain a better understanding of the situation, consider the achieved PDR values of this experiment shown in Fig. 8. Using random item selection, good latency and PDR values are obtained for class 1 which is not necessary. But the PDR value of high reliability demanding items of class 3, for instance, is quite low. By applying our priority-based item selection, the PDR values are much closer to the required values. It means that the PDR of data items in classes 2 and 3 has been increased at the expense of worse values (but still acceptable) for classes 1 and 4. We reach the same conclusion as for setup 1 about the behavior of our mechanism. As future work, we are implementing our mechanism in *MyriaNed* [12] sensor nodes to test it in a real life setup.

VI. Conclusion

In this paper, a dynamic priority assignment mechanism is proposed for data dissemination in wireless sensor networks (WSNs) with heterogeneity in the Quality-of-Service (QoS) requirements among different wireless nodes. The goal is to distribute network bandwidth among the data items according to their relative QoS demands. The mechanism is specifically vital for multi-hop WSNs with high data loads. The method also supports varying QoS requirements which will be useful for multi-scenario applications. As the priority values are

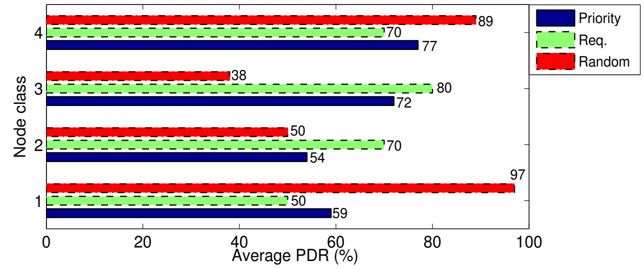


Fig. 8. The requested and achieved PDR values in COPD experiment setup.

calculated dynamically in every node in the routing path according the QoS requirement and the history of the data item, mobility of nodes is properly taken into account. A healthcare application scenario was used as a representative case study. Extensive simulations using several setups are performed to observe the behavior of our mechanism. The results clearly show that using dynamic data prioritization, the data items with more stringent QoS demands receive better service at the cost of less but sufficient service for less demanding items. As future work, we are trying other applications with different protocol stacks and QoS metrics. We also plan to run large scale experiments using *MyriaNed* [12] wireless sensor nodes to further evaluate the performance of the proposed mechanism.

Acknowledgment

This work was supported by the Dutch innovation program Point-One, through project ALwEN, grant PNE07007.

References

[1] M. Nabi et al, "MCMAC: An optimized medium access control protocol for mobile clusters in wireless sensor networks." in *Proc. of 7th IEEE Conf. on Sensor, Mesh and Ad Hoc Communications and Networks (SECON).* IEEE, June 2010, pp. 28–36.
[2] Y. Liu and W. W. Seah, "A scalable priority-based multi-path routing protocol for wireless sensor networks," *International Journal of Wireless Information Networks*, vol. 12, no. 1, pp. 23–33, 2005.
[3] C. Intanagonwiwat et al., "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 2–16, 2003.
[4] J. Chen, M. Zhou, D. Li, and T. Sun., "A priority based dynamic adaptive routing protocol for wireless sensor networks." in *Proc. of int'l Conf. on Intelligent Networks and Intelligent Systems (ICINIS).* IEEE, November 2008, pp. 160–164.
[5] D. B. Johnson, D. A. Maltz, and J. Broch., "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks." in *In Ad Hoc Networking, Chapter 5.* Addison-Wesley, 2001, pp. 139–172.
[6] S. Kim, S. Lee, H. Ju, D. Ko, and S. An., "Priority-based hybrid routing in wireless sensor networks." in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC).* IEEE, April 2010, pp. 1–6.
[7] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing." in *Proc. of 2nd IEEE Workshop on Mobile Computing Systems and Applications.* IEEE, February 1999, pp. 90–100.
[8] D. Gavidia and M. van Steen, "A probabilistic replication and storage scheme for large wireless networks of small devices," in *Proc. 5th IEEE Int'l Conf. Mobile and Ad Hoc Sensor Systems (MASS).* IEEE, 2008.
[9] V. Pareto, "Piccola biblioteca scientifica," *Manual of Political Economy*, pp. 795–825, 1906, translated into English by Ann S. Schwier (1971).
[10] A. Kopke et al, "Simulating wireless and mobile networks in OMNeT++ - the MiXiM vision," in *Proc. 1st Int'l Conf. on Simulation Tools and Techniques (SIMUTools).* ICST, Brussels, 2008.
[11] "OMNeT++ website. http://www.omnetpp.org."
[12] F. van der Wateren, "The art of developing WSN applications with MyriaNed," Chess Company, the Netherlands, Tech. report, 2008.
[13] M. Nabi, M. Geilen, and T. Basten., "MoBAN: A configurable mobility model for wireless body area networks." in *Proc. of 4th Int'l Conf. on Simulation Tools and Techniques (SIMUTools).* ICST, March 2011.