Traffic-Differentiation-Based Modular QoS Localized Routing for Wireless Sensor Networks

Djamel Djenouri, Ilangko Balasingham

***Abstract*— A new localized quality of service (QoS) routing protocol for wireless sensor networks (WSN) is proposed in this paper. The proposed protocol targets WSN's applications having different types of data traffic. It is based on differentiating QoS requirements according to the data type, which enables to provide several and customized QoS metrics for each traffic category. With each packet, the protocol attempts to fulfill the required data-related QoS metric(s) while considering power-efficiency. It is modular and uses geographical information, which eliminates the need of propagating routing information. For link quality estimation, the protocol employs distributed, memory and computation efficient mechanisms. It uses a multi-sink single-path approach to increase reliability. To our knowledge, this protocol is the first that makes use of the diversity in data traffic while considering latency, reliability, residual energy in sensor nodes, and transmission power between nodes to cast QoS metrics as a multi-objective problem. The proposed protocol can operate with any medium access control (MAC) protocol, provided that it employs an acknowledgment mechanism (ACK). Extensive simulation study with scenarios of 900 nodes shows the proposed protocol outperforms all comparable state-of-the-art QoS and localized routing protocols. Moreover, the protocol has been implemented on sensor motes and tested in a sensor network testbed.**

***Index Terms*— wireless sensor networks, quality of service, geographical routing, distributed protocols.**

I. Introduction

Many applications of wireless sensor networks (WSN), such as vehicular and biomedical, have diverse data traffic with different quality of service (QoS) requirements. This paper focuses on these applications, for which it proposes a localized QoS routing protocol. Traffic-differentiation, while simultaneously considering latency, reliability, residual energy, and transmission power in a localized way represents the key features of our contribution.

We consider a general scenario typical for many of the targeted WSN applications, where sensors collect different kinds of data and transmit them towards fixed sinks via other sensors in a multi-hop, ad hoc paradigm. We define two kinds of sinks; primary sink and secondary sink, to which a separate copy of each message that requires high reliability is sent. A typical example of such a scenario is patient monitoring in a hospital room, where different health parameters are to be captured and forwarded to health care servers accessible by the medical staff. Traffic is diverse and may have different QoS requirements, depending on the monitored parameter and its value, the patient health situation, the patient context, e.g. regular monitoring vs. monitoring in operating room, etc. Duplication towards a secondary sink may be useful if high reliability is required. Three different classes of QoS requirements are used in the proposed protocol: i) energy efficiency (including both residual energy and the required transmission power), ii) reliability, and iii) latency. The first requirement is traffic-unrelated, contrary to the other ones. It can be viewed as application-related QoS metric that must be taken into account for all types of traffic, since ensuring a long network lifetime is essential for all applications.

With these requirements data traffic may be split into: i) regular traffic that has no specific data-related QoS need, ii) reliability-sensitive traffic, which should be delivered without loss but can tolerate reasonable delay, e.g., file transfer for long term data, iii) delay-sensitive traffic, which should be delivered within a deadline but may tolerate reasonable packet loss, e.g. video streaming, and finally iv) critical traffic of high importance and requiring both high reliability and short delay (delivery within a deadline), e.g safety alarms in vehicular applications, physiological parameters of a patient during a surgery, etc. Following this classification the proposed protocol is designed using a modular approach, aiming to ensure exactly the required QoS for each packet. A module is devoted to each QoS requirement, in addition to the queuing module and neighbor manager. The queueing module is responsible for implementing a priority multi-queuing strategy that gives more priority, and it consequently ensures shorter latency, to critical and delay-sensitive packets. The neighbor manager runs the HELLO protocol that enables exchanging information between neighboring nodes and implements the link reliability and latency estimators. It uses light-weight estimators based on EWMA (Exponential Weighted Moving Average), which have small memory footprint and are suitable for WSN. These estimators enable the other modules to locally select the most appropriate neighboring node amongst the available candidates offering positive advance towards the destination. Figure 1 shows the solution framework.

The rest of the paper is structured around the following sections: Section II discuses the related work. Section III gives the network model, notations, and assumptions. The solution and its different modules is presented in Section IV. Section V presents a mathematical analysis of the solution, while SectionVI is devoted to performance evaluation through a comparative simulation study. Section VII describes the implementation on sensor motes and shows some results of

Djamel Djenouri is with CERIST Research Center, Algiers, Algeria, and a former post-doc fellow at the Department of Electronics and Telecommunications, NTNU, Trondheim, Norway

Email: ddjenouri@mail.cerist.dz

Ilangko Balasingham is with Interventional Center, Oslo University Hospital and Institute of Clinical Medicine, and University of Oslo, Norway

Email: ilangko.balasingham@medisin.uio.no

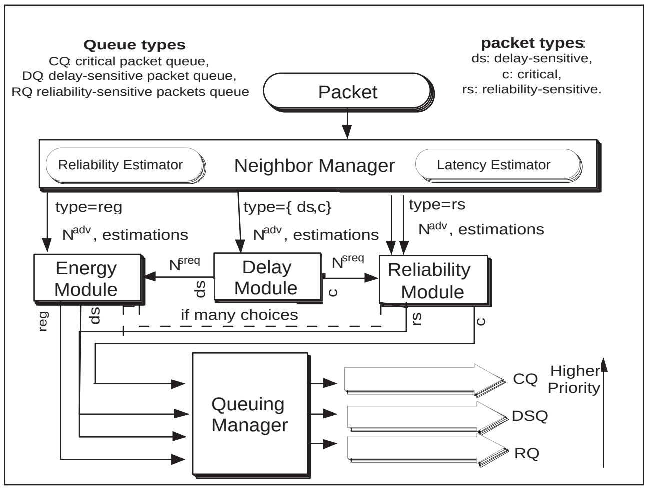


Fig. 1. Solution Framework

the tests. Finally, Section VIII draws the conclusion.

II. Related Work

Geographic or localized routing is a promising approach for WSN. It has the advantage of being scalable and efficient compared to global information routing (reactive and proactive), notably in networks where nodes are stationary or with low mobility. A common feature of all localized routing protocols is the use of localization information, to select next router amongst neighboring nodes that are geographically closer to destination. However, the selection strategy differs from a protocol to the other depending on the constraints to consider. In this paper route selection is based on QoS objectives. Many research efforts have been devoted to multi-objective QoS routing in WSN using localization information, resulting in several routing protocols. SPEED [1] attempts to route packets through routes that ensure a given fixed speed, and uses EWMA (Exponential Weighted Moving Average) for link latency estimation. The aim of this protocol is basically to reduce delay, but it probabilistically chooses the node amongst the ones supposed to fulfill the required speed, which is energy-efficient and balances the network load. However, SPEED does not consider reliability. The protocol proposed in this paper is different from SPEED in many aspects. It considers reliability and uses dynamic speed that can be adjusted for each packet according to the required deadline for reception. It addresses energy using a deterministic approach and balances the load only amongst nodes estimated to offer the required QoS. MMSPEED [2] attempts to improve SPEED and defines multi-speed routing, where several routing layers -each ensuring a given speed- are used. Packets are associated to appropriate layers according to their required speed, and reliability is ensured through probabilistic multi-path towards a single sink, which may results in congestion at nodes near the sink. To prevent such congestion our protocol uses different approach to address reliability, namely single-path multiple-sink, which will be described later. Moreover, link reliability is considered in the routing metrics.

DARA [3] considers reliability, delay, and residual energy in the routing metric, and defines two kinds of packets: critical and non-critical packets. The same weighted metric is used for both types of packets, where the only difference is that a set of candidates reached with a higher transmission power is considered to route critical packets. For delay estimation, the authors use queueing theory and suggest a method that, in practice, needs huge amount of sample storages. The proposed protocol differs from DARA with respect to many aspects. First, in addition to residual energy, the transmission energy is also considered. The protocol also makes a more comprehensive traffic differentiation and defines several categories according to the QoS requirement instead of using only two classes. Furthermore, it uses a modular approach with different metrics for each category instead of a single weighted metric. This enables customized and dynamic QoS provision according to the packet type. Finally, it uses the memory-efficient EWMA for link and delay estimation and considers both queuing time and transmission delay, whereas DARA only considers queuing time.

Residual energy, reliability, and geographic advance are used in [4]. However, the authors consider the use of harvesting energy for estimating residual energy of neighboring nodes, assuming environmentally powered sensors. We do not consider such kind of power and assume the only available power is through batteries. Chipara et al. [5] did not consider residual energy and reliability, but they took into account delay and required transmission power in their localized QoS routing. For delay estimation they used Jacobson's algorithm that has the advantage of considering both average and variation of samples, but like [3], it requires past observations are stored [6]. Mahaptara et al. [7] use only last observations to estimate latency in their QoS routing protocol that combines delay and residual energy in a weighted routing metric. Some other localized routing protocols such as [8], [9], [10], [11], [12], [13] deal only with energy efficiency, and most of them combine residual energy and transmission power in a weighted routing metric.

All the protocols proposed thus far do not make a clear differentiation in route selection between traffic with respect to QoS requirements. They define either the same combined metric (of all the considered QoS metrics), or several services but with respect to only one metric. Our main contribution is the design of a routing protocol enabling to provide different QoS services regarding latency, reliability, and energy all together according to the traffic type. The proposed protocol is the first that makes such differentiation and considers all the above mentioned QoS metrics.

III. Network Model and Assumptions

We assume nodes are aware of their positions, either through an internal GPS (global positioning system) device or a separate distributed localization service [14]. Each node v_i is supposed to be aware of its current battery state B_{v_i} (also termed *residual energy*). This can be obtained at any time from the current voltage and the discharge curve that features the battery [4]. We assume that nodes have the same transmission power range, P_{range}, and that each node can control its transmission power. The set of nodes in v_i's vicinity denoted by, N_{v_i}, consists of v_i's neighboring nodes, defined by: $N_{v_i} = \{v_j : dist_{v_i,v_j} \leq P_{range}\}$, where $dist_{v_i,v_j}$ is the Euclidean distance between nodes, v_i and v_j. In addition to N_{v_i}, $N^{advc}_{v_i,v_d}$ is defined as the set of neighboring nodes providing positive advance for node, v_i, towards final destination, v_d. It consists of neighboring nodes that are closer to the destination than v_i. That is, $N^{advc}_{v_i,v_d} = \{v_j \in N_{v_i} : dist_{v_j,v_d} \leq dist_{v_i,v_d}\}$. Like all geographic routing protocols, each node needs to know its neighboring nodes and their current parameters, e.g., position, battery state, etc. This can be ensured via the execution of a HELLO protocol like in [1], [2], [4], [7], [8], [15].

For localized routing to be effective, nodes are supposed to be either stationary or having low mobility. Otherwise, positions will change very frequently, and a high frequency of HELLO packets exchange will be needed to keep the node up-to-date about the neighboring nodes, which is resource consuming. The proposed protocol does not deal with void situation. We say there is a void situation between two *non-neighboring* nodes in the particular case where there is no node in the network closer to one of them than the other. Node density is supposed to be high enough to prevent such situation. The target WSN application where the protocol may be used should gather different traffic with different QoS requirements. The proposed solution would not be effective in WSN of homogeneous traffic such as automatic irrigation in agriculture. Without loss of generality, two sinks acting as primary and secondary sinks are supposed to be used and located at geographically divergent position. It is possible to extend the model to use multiple sinks and associate each sensor node with two geographically divergent sinks.

To transmit one bit from a source to a destination over a distance, d, the total consumed energy (by both nodes) is given as [9]:

$$E = 2E_{elec} + \beta d^{\alpha}, \tag{1}$$

where E_{elec} is the energy utilized by transceiver electronic, which is independent of the distance. βd^{α} accounts for the radiated power necessary to transmit over distance d, where α is the path loss ($2 \leq \alpha \leq 5$) and β is a constant given in $Joule/(bits \times m^{\alpha})$. Equation (1) is for unicast packets. For broadcasting messages from a given node, v_i, the energy can be written as

$$E = (\parallel N(v_i) \parallel +1)E_{elec} + \beta d^{\alpha}, \tag{2}$$

where $\parallel N(v_i) \parallel$ denotes the number of v_i's neighboring nodes, and d is the distance traversed by the packet (usually set to the power rang for broadcast packets).

IV. Protocol Overview

A. Neighbor Manager

This module is the first that receives packets from the higher and the lower layers. It is responsible for executing the HELLO protocol, managing neighbor table, implementing estimation methods, running the other modules, and providing them with the required information according to the packet type. It uses a neighbor table that assigns an entry for each neighboring node, which includes all the information related to the node such as its position, residual energy, estimated waiting time for each queue, estimated transmission delay, required transmission energy towards it, and estimated packet delivery ratio. The three latter parameters are estimated by the neighbor manager, while the others are estimated by the neighboring nodes themselves using their own neighbor managers. They are updated upon each reception of a HELLO packet. Periodically, or upon observing significant change in some parameters, each node broadcasts a HELLO packet including its current position, residual energy, and its estimation of the other local parameters. Obviously, high frequency (short period) of HELLO packets exchange provides relevant and up-to-date information, but it would become resource consuming. This means it is required that this period should be carefully selected to maintain proper balance between information freshness and cost.

Neighboring nodes use HELLO packets to update existing entries, add new entries when new nodes move within the node's vicinity, and delete entries when neighboring nodes move away or break down, which can be detected in case of not receiving HELLO packets after a defined period of time (timeout). The HELLO protocol implemented by this module is not much different from state-of-the art localized routing, such as [2], [7], [4], [8], [15], [1]. It just adds some more estimation information to the exchanged packet, as mentioned above. In the following, the estimators implemented by the neighbor manager are described.

1) Reliability Estimation: Exponential Weighted Moving Average (EWMA) [6] estimation is more suitable for WSN compared to the other estimation methods such as flip-flop estimator, Kalman filter, and linear regression. These methods use *statistically meaningful median* upon previous estimates' variation, which is variance-calculation based and requires important storage resources. EWMA has the advantage of being simple and less resource demanding compared to other methods [4], [15]. Still, it can react quickly to significant changes, while being stable and less influenced by sporadic, large deviated measurements. WMEWMA (Window Mean Exponential Weighted Moving Average) is very similar to EWMA but updates the estimated parameter in regular time intervals instead of doing it for every packet, which is appropriate for estimating link latency.

Algorithm 1 describes the WMEWMA-based link reliability estimation of the proposed protocol. prr_{v_i,v_j} denotes the packet reception ratio of the link relaying node v_i to node v_j. It indicates the probability of successful delivery over the link (estimated link reliability). This parameter is updated by v_j at each time window, w, and inserted into the HELLO packet for usage by node, v_i, in the next window. Therefore, the given algorithm is run by node, v_j, and not, v_i, contrary to the other algorithms given later. The time window is expressed in terms of number of packets transmitted by node, v_i. Upon receiving each packet, the node updates the current window, cw, the number of packets received, r, as well as the number of known missed packets, f, where $packet.sc$ is the sequence number of the current received packet, and sp is the one of the last received packet. At the end of time window, prr_{v_i,v_j} is updated as indicated in instruction 8, and parameters are re-initialized. a is a tunable parameter of the moving average. Appropriate values for a and w for a stable WMEWMA are $w = 30$ and $a = 0.6$ [6].

Algorithm 1: Reliability Estimator (v_i, v_j) run at v_j

1 **Initialization:** sp=f=r=0; $prr_{v_i,v_j} = 1$;
2 **for** *each reception of packet p from* v_i **do**
3 cw=cw+1;
4 r=r+1
5 f= f+packet.sc-(sp+1)
6 sp=packet.sc;
7 **if** cw=w **then**
8 $prr_{v_i,v_j} = aprr_{v_i,v_j} + (1-a)\frac{r}{r+f}$
9 f=r=cw=0

2) Latency Estimation: Algorithm 2 illustrates the estimation of parameters used by the delay-sensitive module. As in [2], the EWMA approach is employed, but it is used for both transmission delay and queueing delay. Each node, v_i, estimates transmission delay, dtr_{v_j}, of outgoing link for each neighbor, v_j, as well as its queueing delay, w_{v_i}, and broadcasts the latter in the HELLO packets. Therefore, for v_i, every w_{v_j} is obtained from v_j. It needs to estimate w_{v_i} and dtr_{v_j}. We will show later that several queues are used, i.e. critical packets and delay-sensitive packets are inserted in different queues. Therefore, each packets' type has a separate estimation of w_{v_i} for the queuing delay, say $w_{v_i}[packet.type]$. This delay represents the time between packet insertion into the queueing system and when it becomes at the position of transmission. Instruction 6 of Algorithm 2 shows the EWMA update, where ω is the exact waiting time of the packet that can be calculated through a local time stamp. It represents a sample for w_{v_i} estimation. The same approach is adapted for estimating dtr_{v_j}, (instructions 7 to 11 of Algorithm 2), where t_0 the time the packet is ready for transmission and becomes the head of transmission queue, t_{ACK} the time of the reception of acknowledgment packet (ACK), bw the bandwidth, and $size(ACK)$ the size of the ACK packet. This way, dtr_{v_j} includes estimation of the time interval from the packet becomes head of v_i's transmission queue until its reception at node, v_j. This takes into account all delays due to contention, such as channel sensing, channel reservation (RTS/CTS) if any (depending on the used MAC protocol), time slots, etc. Note that for a given packet, computation complexity of both estimators is constant ($O(1)$)

B. Energy Module

This module is responsible for routing regular packets as well as the other packets, when more than one candidate satisfy the required QoS criteria. Both power transmission cost and residual energy of routers should be considered to achieve power efficiency. A non-aggregated min-max approach is used for this trade-off [16]. The problem is to select at node, v_i, the most power-efficient node for destination, v_d, from the set of neighboring nodes offering positive advance, $N^{advc}_{v_i,v_d}$. Considering Eq. (1), the cost that can be managed

Algorithm 2: Latency Estimator (v_i, v_j)

1 **Initialization:** $w_{v_i}[packet.type] = dtr_{v_j} = 0$;
2 **for** *each packet transmission* **do**
3 **if** $w_{v_i}[packet.type] = 0$ **then**
4 $w_i[packet.type] = \omega$
5 **else**
6 $w_{v_i}[packet.type] = aw_{v_i}[packet.type] + (1-a)\omega$
7 **for** *each ACK reception from* v_j **do**
8 **if** $dtr_{v_i} = 0$ **then**
9 $dtr_{v_i} = t_{ACK} - size(ACK)/bw - t_0$
10 **else**
11 $dtr_{v_i} = adtr_{v_i} + (1-a)(t_{ACK} - size(ACK)/bw - t_0)$

when routing is only the radiated power for transmission. That is, for every candidate node, v_j, the required energy related to routing is given by $\beta(dist_{v_i,v_j})^\alpha$ - called hereafter the transmission power. The other criterion is the battery state, B_{v_j}, of every candidate node, v_j.

Algorithm 3 describes the min-max approach of the energy module. v_T denotes the node that has the minimum transmission power cost, which represents the optimum with respect to the first criterion, while the second criterion's optimum, denoted by v_B, is the node having the highest amount of energy in its battery. For every candidate, v_j, its relative deviation for each metric's optimum is calculated (instructions 1 to 3). Note that battery states, B_{v_j}, B_{v_B}, are always positive, contrary to transmission powers that may be less than 0, which explains the removal of max function from instruction 3 (also note that $B_{v_j} < B_{v_B} \forall j$). After that the set, S_0, of nodes minimizing the maximum deviation with respect to the two criteria, is calculated as shown in instruction 4. If $|S_0|$ contains merely one element, then obviously it is the selected optimum. Otherwise, instruction 8 indicates the case where the metric, k, for which $\{Z_k(v_j)\}$ in instruction 4 reaches the maximum is not unique for all elements of S_0, i.e., some nodes have maximum deviation in Z_T and others in Z_B. Note that $M[max()]$ stands for the metric for which the maximum is obtained. In this case, the node offering the best advance from S_0 will be selected. Otherwise, the final solution is the set of nodes from S_0 that minimizes the deviation for the metric other than l, where l represents the metric for which $\{Z_k(v_j)\}$ reaches the maximum. Note that the computation complexity of this algorithm is linear. It is $O(N^{advc}_{v_i,v_d})$, which is $\leq O(N_{v_i})$

C. Reliability Module

The reliability module is presented by Algorithm 4. First, reliability is addressed by sending a copy to both primary and secondary sinks, respectively denoted by PS and SS. This multi-sink single-path approach is selected instead of the single-sink multi-path approach used in [2], which results in data packets convergence near or at the sink, and thus increases traffic contention and collisions [3]. For each copy, candidate offering the highest reliability, prr, is selected (instruction 3). prr_{v_j} is estimated by the neighbor manager as shown in Section IV-A.1. If several nodes have the maximum reliability, then the most energy efficient is selected using the energy module. Note that the computation complexity of this algorithm is $O(N_{v_i})$.

Algorithm 3: Energy Module $v_i, v_d, N^{advc}_{v_i,v_d}$

for *each* $v_j \in N^{advc}_{v_i,v_d}$ **do**
  $Z_T(v_j) = \max(\frac{|\beta(dist_{v_i,v_j})^\alpha - \beta(dist_{v_i,v_T})^\alpha|}{|\beta(dist_{v_i,v_T})^\alpha|}, \frac{|\beta(dist_{v_i,v_j})^\alpha - \beta(dist_{v_i,v_T})^\alpha|}{|\beta(dist_{v_i,v_j})^\alpha|})$
  $Z_B(v_j) = \frac{B_{v_B} - B_{v_j}}{B_{v_j}}$
$S_0 = \{v_x : \max_{m \in \{T,B\}} \{Z_m(v_x)\} = \min_{j \in N^{advc}_{v_i,v_d}} \max_{k \in \{T,B\}} \{Z_k(v_j)\}\}$
if $(|S_0| = 1)$ **then**
  $S = S_0'$s element
else
  if $(\exists x, y \in S_0 : M[\max(Z_T(x), Z_B(x))] \neq M[\max(Z_T(y), Z_B(y))])$ **then**
    $S = \{v_x : (dist_{v_i,v_d} - dist_{v_x,v_d}) = \max_{j \in N^{advc}_{v_i,v_d}} (dist_{v_i,v_d} - dist_{v_j,v_d})\}$
  else
    $l = M[\max(Z_T(x), Z_B(x))]$, $k = \{T, B\} - l$
    $S = \{v_x : Z_k(v_x) = \min_{j \in S_0} \{Z_k(v_j)\}\}$

Algorithm 4: Reliability Module $(v_i, v_d, N^{advc}_{v_i,v_d})$

$D_{set} = \{PS, SS\}$
for *each* $vd \in D_{set}$ **do**
  $S_{rel} = \{v_x \in N^{advc}_{v_i,v_d} : prr_{v_x} = \max_{j \in N^{advc}_{v_i,v_d}} (prr_{v_j})\}$
  if $(|S_{rel}| = 1)$ **then**
    $S = S_{rel}$ element
  else
    call energy module with S_{rel} as parameter, instead of $N^{advc}_{v_i,v_d}$

D. Latency Module

Packet velocity approach given in [5] is used by this module. It has the advantage of not requiring any synchronization between nodes as it uses relative times. The main difference between our approach and [5] is that the former uses a simple but memory and time-efficient estimation method instead of Jacobson's algorithm. Moreover, it considers queueing time at both the current and the next hop, together with link latency. We suppose every delay-sensitive packet has a delivery deadline, dd, specified by the upper layers. It indicates the time the packet should be delivered to the final recipient.

We define two velocities (speeds); required velocity, s_{req}, and velocity offered by node ,v_j, denoted by, s_{v_j}. The required velocity is proportional to the distance and the time remaining to the deadline, rt. At each hop and prior to each transmission at the MAC layer the transmitter updates this parameter and puts it in the header as

$$rt = rt_{rec} - (t_{tr} - t_{rec} + size/bw), \qquad (3)$$

where t_{rec} represents the reception time, t_{tr} the transmission time, bw the bandwidth, $size$ the packet size, and rt_{rec} the previous value of rt (at time of reception). $t_{tr} - t_{rec} + size/bw$ gives the whole delay from reception of the packet until transmission of the last bit. It includes both queueing delay $(t_{rec} - t_{tr})$ and data transfer delay $(size/bw)$. Upon reception of the packet, node v_i uses the rt value updated at the previous hop to calculate the required velocity, s_{req}, as illustrated in Algorithm 5. After that it estimates the velocity offered by neighboring nodes that provide positive advance, by taking into account waiting time at the queue of node v_i, say w_{v_i}, transmission time, dtr_{v_j}, and waiting time at the queue of the next hop, w_{v_j}. Remember that the waiting time estimations depend on the packet type, and there will be different w_{v_j} for each packet type $(packet.type)$. After computing velocities of all candidate nodes, the delay-sensitive module calculates the set of nodes supposed to meet the required deadline, S_{del}, and calls the energy module, or reliability module in case of critical packets. The called module selects the most appropriate router from the set, S_{del}, if it includes several nodes. Computation complexity of this algorithm is $O(N_{v_i})$.

Algorithm 5: Latency Module $(v_i, v_d, N^{advc}_{v_i,v_d})$

$s_{req} = \frac{dist_{v_i,v_d}}{rt}$
for *each* $v_j \in N^{advc}_{v_i,v_d}$ **do**
  $s_{v_j} = \frac{dits_{v_i,v_d} - dist_{v_j,v_d}}{w_{v_i}[packet.type] + dtr_{v_j} + w_{v_j}[packet.type]}$
$S_{del} = \{v_x \in N^{advc}_{v_i,v_d} : s_{v_x} \geq s_{req}\}$
if $(|S_{del}| = 1)$ **then**
  $S = S_{del}$ element
else
  if *packet.type=delay-sensitive* **then**
    call energy module with S_{del} as parameter, instead of $N^{advc}_{v_i,v_d,v_d}$
  else
    if *packet.type=critical* **then**
      call reliability module with S_{del} as parameter

E. Queuing Manager

To obtain low latency when routing critical and delay-sensitive packets, higher priority should be given to these packets in channel contention than the delay-insensitive packets (regular and reliability-sensitive packets). Also, critical packets need higher priority than delay-sensitive packets. This can be achieved through a multi-queue priority policy, which is implemented by the queueing manager module as described in Algorithm 6. We propose to use three queues and send packets from the highest priority queue to the lowest one. The highest priority queue, CQ, is used by critical packets, the second highest priority queue, DSQ, is used by delay-sensitive packets, and the least priority queue, RQ, is used by regular and reliability-sensitive packets. The number of critical and delay-sensitive packets is usually low, and there would be periods where their respective queues are empty. Otherwise, lower priority traffic may be indefinitely blocked by higher priority traffic. In this case, a timeout policy for each packet is used to remove it to the highest priority queue. This multi-queueing defines contention priority locally, i.e., between packets from the same node. It is possible to define priority for all traffic between neighboring nodes by modifying the MAC protocol slots and backoff time [2]. Nevertheless, such a mechanism requires important modifications in the MAC layer and may increase packet collision in dense networks. Computation complexity of this algorithm is constant, i.e., $O(1)$.

Algorithm 6: Queuing Manager (v_i)

```
for each packet reception, after routing do
    if packet.type = critical then
        insert packet in CQ
    else
        if packet.type = delay-sensitive then
            insert packet in DSQ
        else
            insert packet in RQ
        initialize timeout(packet)
for each timeout(packet) expatriation do
    move packet to CQ
for each packet transmission do
    cancel timeout(packet)
```

V. Analysis

A. Queueing Time

In this section an analysis using queuing theory of waiting time for the multi-queueing system presented in Section IV-E is given. This waiting time will be used to draw upper and lower bounds, respectively of end-to-end packet delivery probabilities and end-to-end delays. A general analysis with n queues is provided, where $n = 3$ in the proposed solution. It would be straightforward to extend the solution to define more priorities, even between packets of the same category. The following analysis is general and directly applies to any extension.

At node l, assume packets arrival to each queue of priority i, say Q_i^l, follows a Poison process of rate, λ_i^l. The service time, which represents the packet transmission delay, is unknown but can follow any general distribution. It is, however, independent of the queue and thus represented by a unique random variable for all queues, say τ. Also suppose queues' sizes are high enough to hold all the arriving packets. The model is thus a queueing system of type $M/G/1$[1][17]. The waiting time of queue, Q_i^l, is denoted by, W_i^l, and the number of packets at each queue, Q_i^l, by $N_{Q_i^l}$. Priority is decreasing with the queue number, i.e. Q_1^l has the highest priority and Q_n^l the lowest one. The following will be used in the analysis [17]:

- **Littles's formula:**
$$E[N_{Q_i^l}] = \lambda_i^l E[W_i^l] \quad (4)$$
- **Server Utilization** The server utilization for packets of priority i, say ρ_i^l, is
$$\rho_i^l = \lambda_i^l E[\tau]. \quad (5)$$

To make sure that all packets will be served within a finite delay it is assumed that, $\sum_{i=1}^{n} \rho_i^l < 1$.

For a packet P arriving to the k^{th} queue the expected waiting time, $E[W_k^l]$, is the sum of: i) the expected remaining time of the current packet in service, $E[R]$, ii) the service time of all packets in queues having a higher or the same priority, i.e. packets in $Q_i^l, i \leq k$, and iii) the service time of packets in higher priority queues arriving during P's waiting time, i.e., after P but prior to its service. Formally speaking

$$E[W_k^l] = E[R] + \sum_{i=1}^{k} E[N_{Q_i^l}]E[\tau] + \sum_{i=1}^{k-1} E[M_i^l]E[\tau], \quad (6)$$

where M_i^l is the number of arriving packets to queue, Q_i^l, during the waiting time of P, given by,
$E[M_i^l] = \lambda_i^l E[W_k^l]$. Using the latter expression along with Eq. (4) and (5), Eq. (6) becomes

$$E[W_k^l] = \frac{E[R] + \sum_{i=1}^{k-1} \rho_i^l E[W_i^l]}{1 - \sum_{i=1}^{k} \rho_i^l}. \quad (7)$$

Lemma 1: $\forall k \in \{1, ..., n\}$,
$$1 + \sum_{i=1}^{k} \frac{\rho_i^l}{(1 - \sum_{j=1}^{i-1} \rho_j^l)(1 - \sum_{j=1}^{i} \rho_j^l)} = \frac{1}{1 - \sum_{j=1}^{k} \rho_j^l}$$

Proof: By recurrence on k, see appendix:

Lemma 2: $\forall k \in \{1, ..., n\}$
$$E[W_k^l] = \frac{E[R]}{(1 - \sum_{j=1}^{k-1} \rho_j^l)(1 - \sum_{j=1}^{k} \rho_j^l)}$$

Proof: By recurrence on k and using Lemma 1, see appendix.

According to [17], $E[R]$ can be expressed as

$$E[R] = \frac{1}{2} \sum_{i=1}^{n} \lambda_i^l E[\tau^2], \quad (8)$$

where $E[\tau^2]$ in our case is the second moment of the service time of all packets given by
$E[\tau^2] = Var[\tau] + E[\tau]^2$.

Finally, using Eq. (8) in Lemma (2) we derive the final expression of $E[W_k^l]$ as

$$E[W_k^l] = \frac{E[\tau^2] \sum_{i=1}^{n} \lambda_i^l}{2(1 - \sum_{j=1}^{k-1} \rho_j^l)(1 - \sum_{j=1}^{k} \rho_j^l)}. \quad (9)$$

B. End-to-End Delay Upper Bound

Assume the longest loop-free geographical[2] route separating a source and a sink is h hops. The end-to-end delay is the sum of waiting times in queues and transmission times for all the hops from the source to the destination. For a packet of priority k, the upper bound, $\overline{delay_k}$, of the end-to-end delay can be given by

[1]Poisson arrival, general service, one server, no specification for the queue size (∞)

[2]By a geographical route we refer to a route constructed using geographical routing, in which the euclidian distance towards the destination keeps decreasing from hop to hop

$$E[\overline{delay_k}] = \sum_{l=1}^{h}(E[W_k^l] + E[\tau]).$$

Using Eq. (9) we obtain

$$E[\overline{delay_k}] = \sum_{l=1}^{h} \frac{E[\tau^2]\sum_{i=1}^{n}\lambda_i^l}{2(1-\sum_{j=1}^{k-1}\rho_j^l)(1-\sum_{j=1}^{k}\rho_j^l)} + hE[\tau].$$

If we note $\overline{\lambda_i}$ the maximum packet arrival rate for queue of priority i, and $\overline{\rho_i}$ the corresponding server utilization, $E[\overline{delay_k}]$ can be bounded by:

$$E[\overline{delay_k}] \leq \frac{hE[\tau^2]\sum_{i=1}^{n}\overline{\lambda_i}}{2(1-\sum_{j=1}^{k-1}\overline{\rho_j})(1-\sum_{j=1}^{k}\overline{\rho_j})} + hE[\tau]. \quad (10)$$

Note that $E[\tau]$ (the transmission time's expected value) greatly depends on the MAC protocol to be used. The proposed protocol and techniques are general and independent of the MAC protocol. Therefore, analyzing this parameter is beyond the scope of this work. An example of such analysis for CSMA/CA can be found in [18].

C. Reliability-sensitive Packet Delivery Lower Bound

The probability of successful delivery by the system is the probability of successful delivery of at least one copy. Let us denote each of the two events, successful delivery of each copy, by e_1 and e_2, respectively, and the system successful delivery by e. We have

$P[e] = P[e_1] + P[e_2] - P[e_1|e_2]P[e_2]$.

Since e_1 and e_2 are independent events, $P[e_1|e_2] = P[e_1]$, thus

$$P[e] = P[e_1] + P[e_2] - P[e_1]P[e_2]. \quad (11)$$

Both $P[\underline{e_1}]$ and $P[\underline{e_2}]$, the lower bounds of respectively $P[e_1]$ and $P[e_2]$, or the probability of delivery when selecting longest routes, can be given by

$$P[\underline{e_1}] = P[\underline{e_2}] = \prod_{i=1}^{h} E[prr] = E[prr]^h,$$

where $E[prr]$ is the expectation of the packet delivery ratio, or the probability of successful delivery, over a link. Using this, Eq. (11) becomes

$$P[\underline{e}] = 2P[\underline{e_1}] - P[\underline{e_1}]^2 = 2E[prr]^h - E[prr]^{2h}, \quad (12)$$

which is the lower bound of $P[e]$. Note that for reliability-unsensitive packets (regular and delay-sensitive packets), the bound is trivially $E[prr]^h$.

D. Computation and communication complexity

We have seen that the communication complexity for each routing module of node v_i is linear with N_{v_i}. Moreover, if we note $N_{v_{max}}$ the maximum network density, i.e. the maximum number of neighboring nodes in the network, then the worst case complexity of each module becomes $\theta(N_{v_{max}})$. Routing a packet is simply the sequential execution of one or two modules, and therefore the overall complexity is additive, which is still linear $\theta(N_{v_{max}})$. Like all localized routing, the only communication overhead of the proposed protocol is limited to the HELLO packet exchange, which consists in periodic transmission by each node to its neighbors. Its complexity is $\theta(n)$ transmission and $\theta(nN_{v_{max}})$ reception. This computation and communication complexity is typical for localized protocols.

VI. Performance Evaluation

A. Comparative Study

To evaluate the performance of the proposed protocol, we carried out a simulation study using GloMoSim [19]. In this study, the proposed protocol - called hereafter LOCALMOR (LOCalized Multi-Objectives Routing) - is compared with state-of-the-art localized routing, namely, DARA [3], SPEED [1], MMSPEED [2], EAGFS [10], and a simple geographical greedy forwarding (GFW) [20]. The last protocol is not QoS protocol. It was chosen as a benchmark representing basic geographic routing, which indicates the improvement provided by the other QoS protocols. Inevitably, energy-efficiency of QoS protocols will be affected when data-related QoS requirement increases. A QoS protocol is energy-efficient if it is less affected and effectively responds to the requirements by balancing data-related QoS and energy efficiency. EAGFS considers only power-efficiency. It is expected to achieve the maximum performance in terms of energy balancing, and thus serves as a benchmark of energy-efficiency. All the other protocols are QoS protocols. Protocols like [4] have completely different assumptions, e.g. environmental powered sensors, and therefore are out of scope. Table I provides a qualitative comparison among the QoS protocols involved in the simulation study, where E_{Tx} and $Eres$ respectively stands for transmission and residual energy.

The simulation configuration consists of 900 nodes located in a 1800 m^2 area, and 1000 s of simulation time. This high number of nodes permits to investigate scalability of the protocols. The nodes were uniformally distributed in a grid topology, with approximatively 100 m of power range, resulting in an average density of 8 (each node has seven neighboring nodes on average). Two sinks located at the corners of the grid, i.e. $(0, 0)$ and $(1800, 1800)$, were used as the primary and secondary sinks, respectively. Traffic was generated from a node located in the center of the simulation area, which makes routes long and equal distance towards both sinks. Constant bit rate traffic (CBR) was used to generate traffic, with 1 kB/s in all scenarios. Some amendments have been made to CBR to include packet type generation with stochastic distribution, following tunable rates for each type. Critical packets and regular packets were used in the simulation. These two classes allow to test all the modules since both delay-sensitive and reliability-sensitive modules are employed to route critical packets. Moreover, it does not make sense to compare the protocols in diverse QoS requirement scenarios. None of the adversaries makes a comprehensive traffic differentiation like LOCALMOR, which would obviously outperform all the protocols in this case. The next section

TABLE I
QUALITATIVE COMPARISON

Protocol	Considered metric	Estimation method	Traffic-differentiation	Considered delay	Duplication
EAGFS	E_{Tx} & E_{res})	no	no	no	no
SPEED	delay&E_{res}	EWMA	no	link	no
MMSPEED	delay&E_{res}&reliability	EWMA	delay requirement	queueing+link	towards the same sink
DAARA	delay&E_{res}&reliability	variance-based	critical vs. no critical	queuing	towards different sinks
LOCALMOR	delay&E_{Tx}&E_{res}&reliability	EWMA	4 classes	queueing+link	towards different sinks

separately investigates the performance of the protocol with each packet class. Critical packet rate was varied from 0.1 to 1, where the performance metrics were measured. Note that for each setting the remaining rate to 1 represents regular packet rate, and the overall traffic load was constant and fixed to 1 kB/s in all scenarios. Table II summarizes the simulation parameters. The performance metrics used are: the packet reception ratio (prr) of all packets and critical packets, the end-to-end delay (of all packets and critical packet), the ratio of packets arriving within the deadline, the standard deviation of the power consumption, and the network lifetime. The deadline was fixed in this simulation to 0.3 s for all critical packets. The simulation results presented in this section were obtained after 9 days of runtime on two-laptops. Each point of the plots is the average of 9 measurements with a 95% of confidence interval.

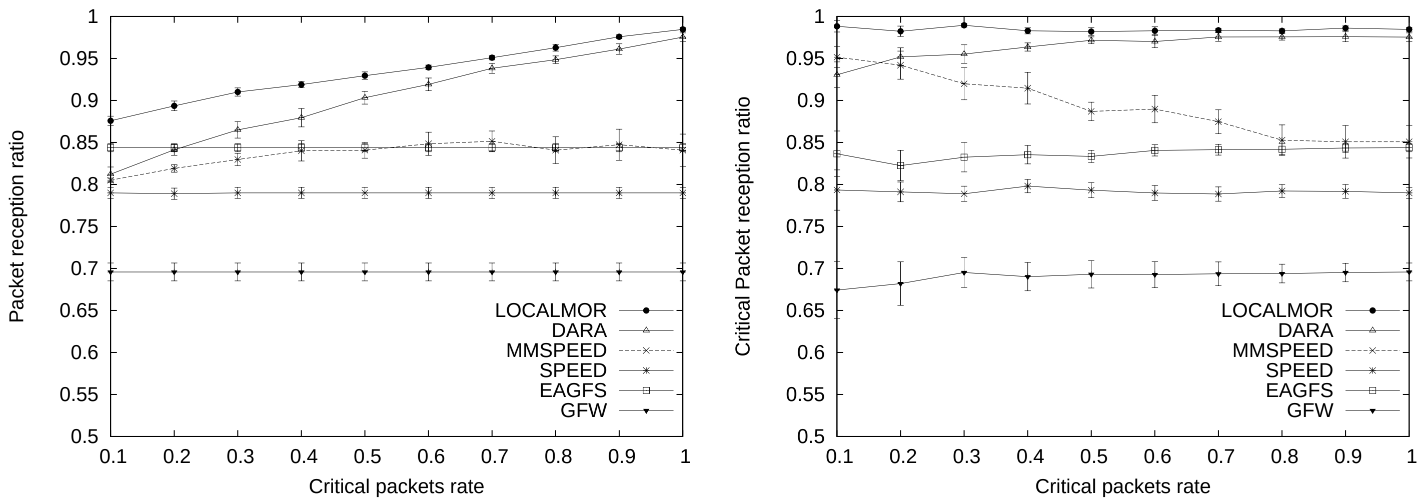
TABLE II
SIMULATION PARAMETERS

Parameter	Value
Number of nodes	900
Simulation area	$1800 * 1800\ m^2$
Traffic	CBR (1kbyte/sec), with regular and critical packets
Critical packet rate	from 0 to 1
Regular packet rate	1-Critical packet rate
Deadline of critical packets	0.3sec
Hello Period	5sec
EWMA factor (a)	0.6
EWMA window	30
Required total prr (used by MMSPEED & DARA)	100 %
Tx power range	$\approx 100\ m$
Initial battery charge	40 Joule
Physical Layer	RADIO-ACCNOISE
MAC layer	802.11 with DCF
Bandwidth	200KB/S
Propagation model	Free space

Figures 2 (a and b), 3 (a and b), and 4 (a) show that LOCALMOR clearly outperforms all protocols with respect to the corresponding metrics. In the second position, DARA shows good performance compared to other protocols. LOCALMOR and DARA linearly increase their performance as a function of critical packet rate, while all other protocols' performances are relatively stable. For high critical packet rates, LOCALMOR halves the delay compared to the majority of protocols and provides 25% better performance compared to DARA. It increases the packet reception ratio up to 13% compared to MMSPEED and to 20% compared to SPEED. For high critical packet rates DARA's packet reception ratio is closer to that of LOCALMOR but it is still inferior. The difference becomes more important for low rates and achieves 7%. The high reliability of LOCALMOR and DARA is due to the use of efficient duplication towards different sinks, contrary to MMSPEED that uses a multi-path single-sink strategy. This kind of duplication results in packet congestion either at the final sink or intermediate nodes. The other protocols do not consider reliability as QoS metric. The linear increase of the packet reception ratio for LOCALMOR and DARA with the increasing critical packet rate can be explained by the subsequent increase of duplications (applied only to critical packets). This means, larger number of critical packets that we have more they are duplicated, which subsequently increases their reception ratio. On the other hand, low delays in both protocols are due to the consideration of queuing waiting time. In addition to this, LOCALMOR considers transmission delay, which justifies its superiority vs. DARA. MMSPEED also considers queueing and transmission delays, but on the other hand, the use of multi-path single-sink transmissions causes congestion and thus results in several retransmission of packets before successful reception, which explains the relatively higher delay for MMSPEED. The linear decrease of the end-to-end delay for LOCALMOR and DARA (Figure 3 (a)) is due to the same reason as of the packet reception ratio, i.e. packets are routed through more delay-efficient link as the critical packet rate increases.

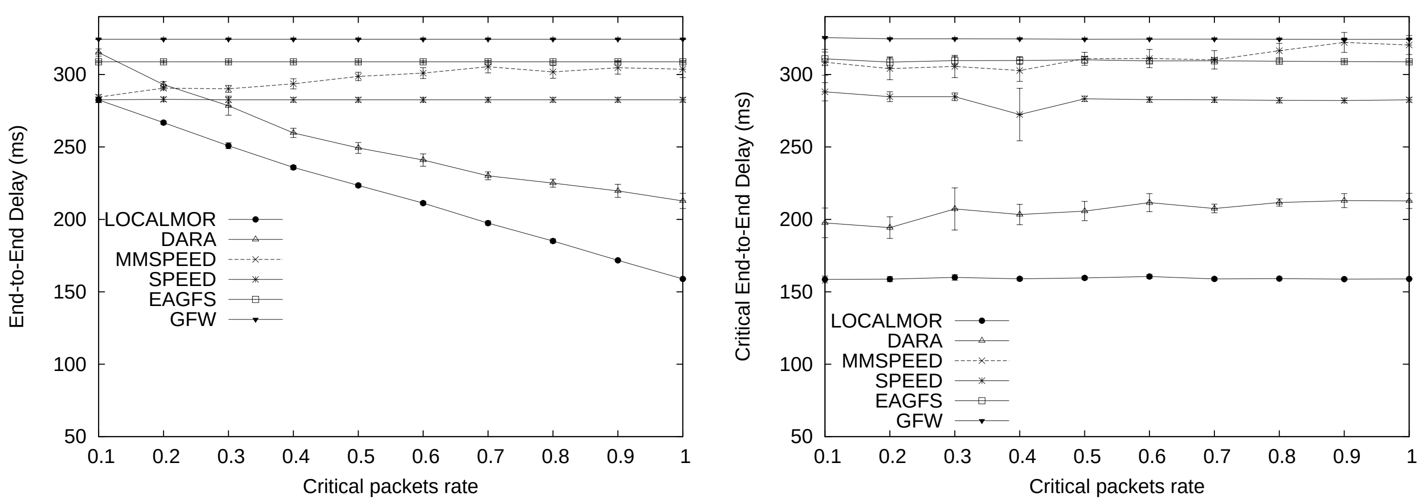
In Figure 3 (b) -depicting the end-to-end delay of critical packets vs. critical packet rate- we notice a stable end-to-end delay for all protocols, which reflects the stability of the routes selected for critical packets that are obviously not affected by the critical packets rise. EAGFS balances the traffic and shows the lowest battery-deviation, as depicted in Figure 4 (b). This protocol considers only energy consumption, which explains its power-efficiency but also low efficiency with respect to the data-related QoS metrics (Figures 2 and 3). SPEED also has low battery-deviation, as it always selects the next router randomly amongst the available candidates. This enables balancing the power consumption well. However, it does not make any distinction among packets and does not consider reliability. This resulted in low performance with respect to the other metrics.

LOCALMOR ensures a trade-off between traffic related QoS metrics and energy. For low and average rates of critical packets, it has the lowest battery-deviation like EAGFS (Figure 4 (b)). Then, its energy deviation smoothly increases as rates become higher, but it is still better than the majority of QoS protocols. This gradual rise is due the decrease of possible choices. LOCALMOR balances the load only among nodes estimated to ensure delivery within the deadline and having the highest reliability. SPEED on the other hand uses a fixed traffic-unrelated speed and ignores reliability. Therefore, as the number of critical packets increases LOCALMOR inevitably balances packet amongst fewer nodes than SPEED. Unlike



(a) (b)

Fig. 2. Packet Reception Ratio



(a) (b)

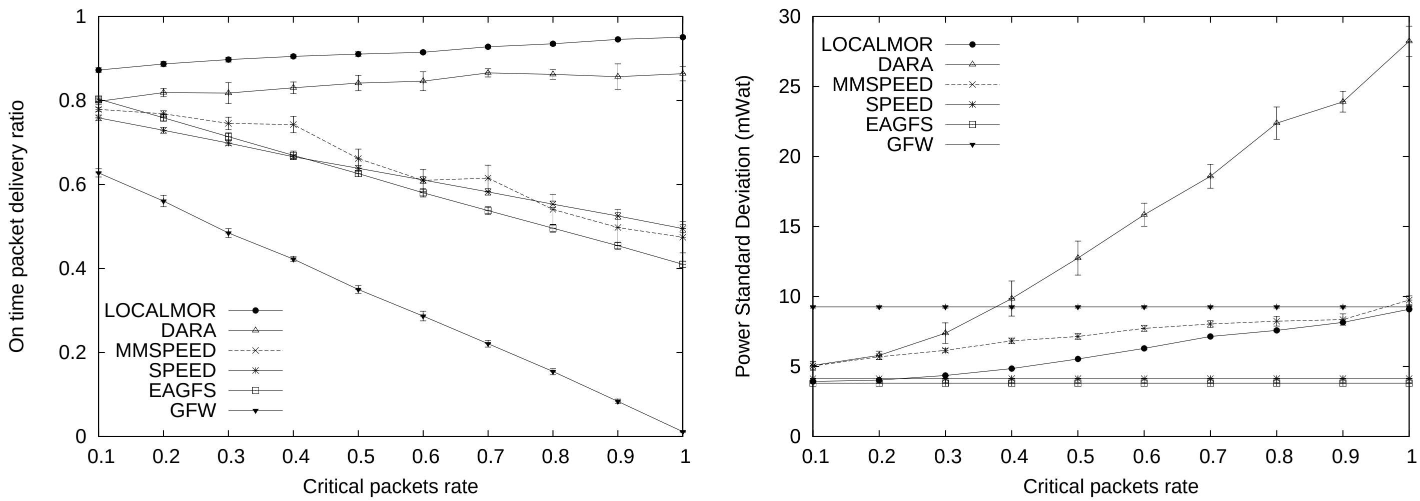
Fig. 3. End-to-End Delay

the other metrics, DARA performs poorly in terms of energy, as it does neither use any traffic balancing technique nor any probabilistic selection. This traffic balancing intuitively affects the network lifetime, defined as the time until the first node runs out of its battery.

The impact of traffic balancing on network lifetime is shown in Figure 5, where protocols having low deviations ensure long lifetime and vice versa. In other words, protocols ensuring stable traffic balancing (EAGFS, SPEED, etc.) have the highest network lifetime and are unaffected by the increase of critical packets. LOCALMOR has the highest lifetime along with EAGFS until 50% critical packet rate, After that its lifetime smoothly decays but remains higher than 600 sec, which is higher than the majority of the compared protocols. This decrease is due to the number of nodes used to balance the traffic, which inevitably decreases with critical packet rate (that yields more QoS requirements. Note that LOCALMOR uses only nodes estimated to ensure the required QoS, contrary to traffic-unrelated energy-efficient protocols (SPEED and EAGFS). In conclusion, LOCALMOR's strategy was clearly demonstrated to be effective in ensuring low latency and high reliability, while always considering the energy and balancing the load amongst nodes providing the required QoS.

B. Traffic Differentiation Analysis

Traffic differentiation is the main feature that distinguishes the proposed protocol. In the previous subsection only critical and regular packets were used, as none of the compared protocols considers the other classes (delay-sensitive and reliability-sensitive traffic). This subsection analyzes the traffic differentiation of the protocol. We use the same configuration as described previously. Nonetheless, this part of the study focuses on LOCALMOR and uses all types of traffic.



(a) (b)

Fig. 4. a: Delivery Within Deadline, b: Power Consumption Standard Deviation

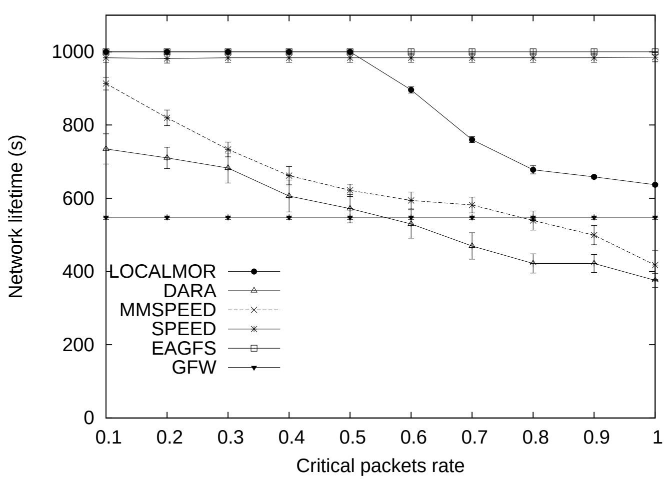


Fig. 5. Network Lifetime (time to first battery drain)

QoS traffic, including delay-sensitive, reliability-sensitive, and critical packets were varied in the same way as critical packets were varied in the previous section, i.e. each QoS traffic varies from 0.1 to 1 and the remaining rate is set to regular packets. Figure 6 depicts the obtained results, where the x-axis represents the rate of QoS traffic (delay-sensitive, reliability-sensitive, and critical for each plot respectively). The difference regarding the end-to-end delay (Figure 6 (a)) between the traffic sensitive to this metric (critical and delay-sensitive packet) and the traffic unsensitive to it (regular-sensitive packets) is clear and becomes more important as the QoS traffic rate increases.

The difference increases linearly until the end-to-end delay of delay-sensitive and critical traffic almost becomes halved compared to reliability-sensitive traffic. This increase was expected due to large delay-sensitive and critical traffic, where packets are routed through more delay-efficient links, while with reliability-sensitive traffic the protocol considers only link reliability. This explains the constant and relatively high end-to-delay for the reliability sensitive traffic. But it also justifies its superiority vs. delay-sensitive traffic with respect to packet reception ratio (PRR) as shown in Figure 6 (b), since link reliability is not considered for delay-sensitive traffic.

PRR of the traffic sensitive to the reliability (critical and reliability-sensitive classes) increases linearly with its rate, from 86% (87% for critical packets) to 98%, whereas it is stable for delay-sensitive packets at the interval of 80% to 83%. This can be explained by the same reasons as with the delay, i.e. more critical and reliability-sensitive traffic results in giving more consideration to link reliability, which is not considered for delay-sensitive traffic. Critical traffic is sensitive to both metrics, which explains the obtained high performance for this class. The small difference between critical traffic and delay sensitive traffic with regard to end-to-end delay (Figure 6 (a)) is due to the queuing priority that is higher for the former. The difference is, however, little and does not exceed 19 ms. The difference of PRR between critical and reliability-sensitive traffic (Figure 6 (b)) is flipping but insignificant since it never exceeds 2%. This because there is no priority between the two classes with respect to this metric. However, for the delay metric the critical packets are more prioritized.

VII. Implementation on Sensor Motes and Experimentation

The proposed protocol was implemented and tested using real sensor motes. TelosB[3] motes were used, along with Contiki operating system [21]. TelosB (TPR2420) enables USB programming and data collection, and it includes an IEEE 802.15.4 radio (CC2420) with integrated antenna. It has low power consumption and uses two AA batteries. The radio is a ZigBee compliant RF transceiver that operates in the Industrial scientific and medical (ISM) band, at 2.4 to 2.4834 GHz. TellosB's microcontroller is an 8 MHz TI MSP430, with

[3] www.xbow.com

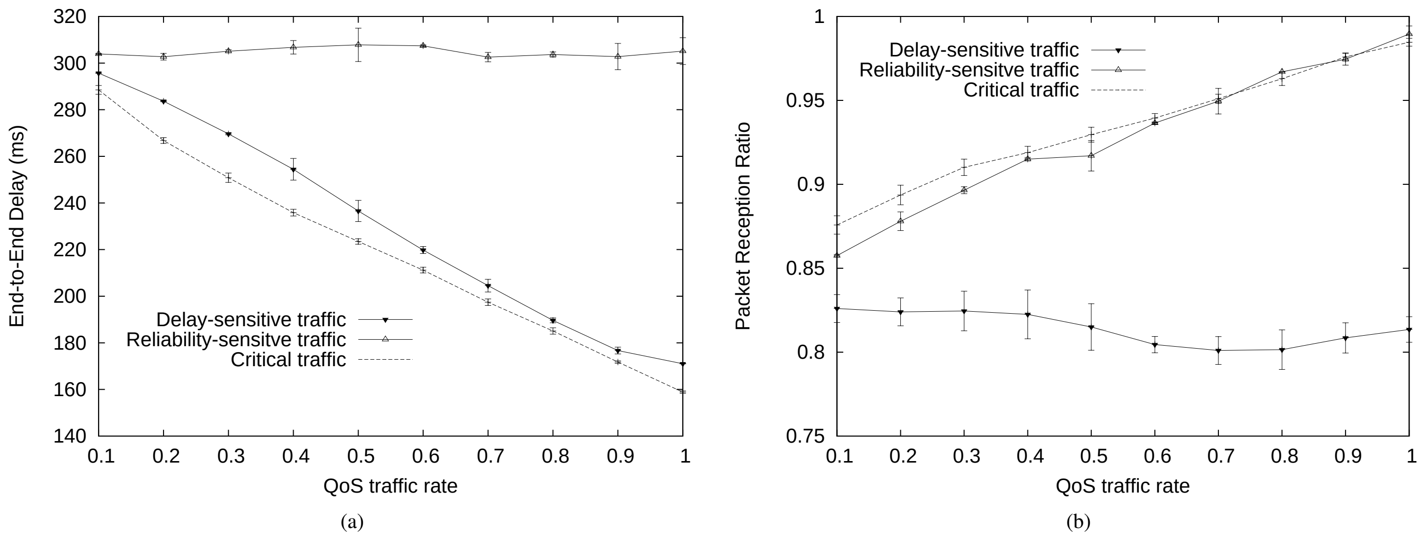


Fig. 6. a: End-to-End Delay, b: Packet Reception Ratio

a 10 KB RAM and 1MB of external flash memory. All these features make TelosB an appropriate research platform suitable for our experiments. Contiki is the first sensor and embedded systems platform that supports TCP/IP, which is important for real life deployment in many applications. It facilitates the integration of sensor networks with other existing network such as Internet. Contiki is written in C programming language, and uses the protothread programming paradigm that is more natural and makes the coding of the protocol easier compared to pure event-based systems [22]. It also includes a lightweight communication stack called RIME, providing a useful programming interface for communication protocol implementation. The interface is composed of many modules such as the multi-hop module used in this implementation. The size of the binary code of the implementation was 24 KB. This includes code for routing protocol, traffic generation module, the operating system kernel and libraries, all uploaded as a single file. The protocol code is a simplified implementation of all the modules except the queuing manager. Embedded systems like Contiki implement packet transmission calls as non-blocking procedure, which prevents any queuing management implementation. Due to physical limitations of sensor motes and the difficulties in real deployment, it is extremely difficult to perform as extensive evaluation as done in the simulation study. The aim of this experiment is to practically investigate the feasibility of the protocol. The motivation is neither to evaluate the scalability of the protocol nor to compar it with other protocols, which was already carried out in the previous section.

An experimental network of 15 nodes were deployed with one source and two sinks, where one acts as primary and the other as secondary. Every sink was connected to a laptop computer through a USB port. To control the topology on a small surface, we used the power control mechanism provided by the CC2420 driver that enables 31 discrete values (from 1 to 31). We fixed the maximum transmission power to 2, resulting in a power range of few tens of centimeters (less than 1 m). As shown in Figure 7, the resulting network topology offers an acceptable connectivity and has several multi-hop routes separating the source and the sinks, which is needed to test a routing protocol. At the MAC layer, null-MAC was used to eliminate the effects of duty-cycling-related delay. The source node generated a 20 bytes packet each second and transmitted it to the primary sink and possibly also to the secondary sink. This depends on the packet type, decided upon each transmission. 40% of the packets were regular, 20% were delay-sensitive, 20% were reliability-sensitive, and 20% were critical. Serial port was used by each sink to construct log files, which were used for collecting the results.

The results are very encouraging and show that more than 96% of packets were correctly delivered with reasonable delay. More importantly, they show that the protocol's traffic differentiation strategy provides different QoS for the different packet types as depicted in Figure 8. Figure 8 (a) shows the packet reception ratio of critical vs. regular packets, while Figure 8 (b) shows the end-to-end delay of delay-sensitive vs. regular packets. It becomes clear that the protocol ensures higher reliability and lower latency to packets requiring such performances than regular packets. In more complex scenarios, e.g. with more nodes, routes, etc., the route selection strategy (routing protocol) would have a much more important impact on the performance metrics, as shown in Section VI.

VIII. Conclusion

We have proposed a new localized routing protocol for wireless sensor networks (WSN). The proposed protocol takes into account the traffic diversity, which is typical for many applications, and it provides a differentiation routing using different quality of service (QoS) metrics. Data traffic has been classified into several categories according to the required QoS, where different routing metrics and techniques are used for each category. The protocol is built up around a modular design following the traffic classification. For each packet, it

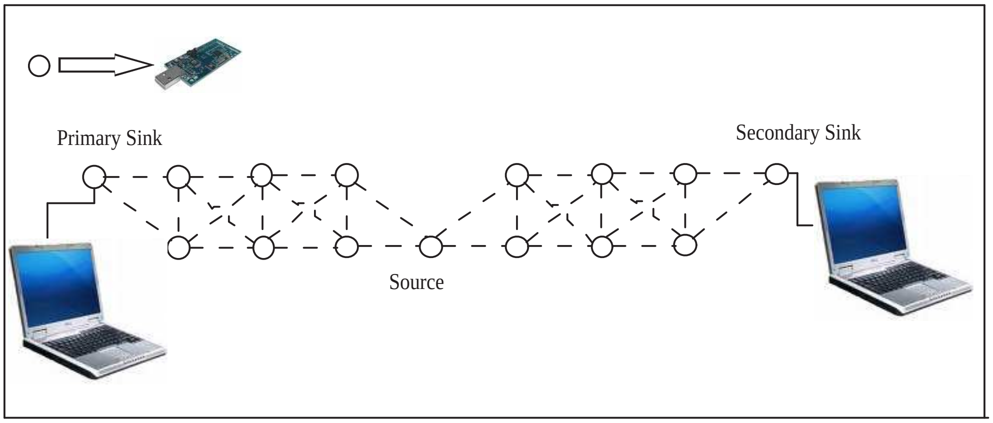


Fig. 7. Experimental Network Topology

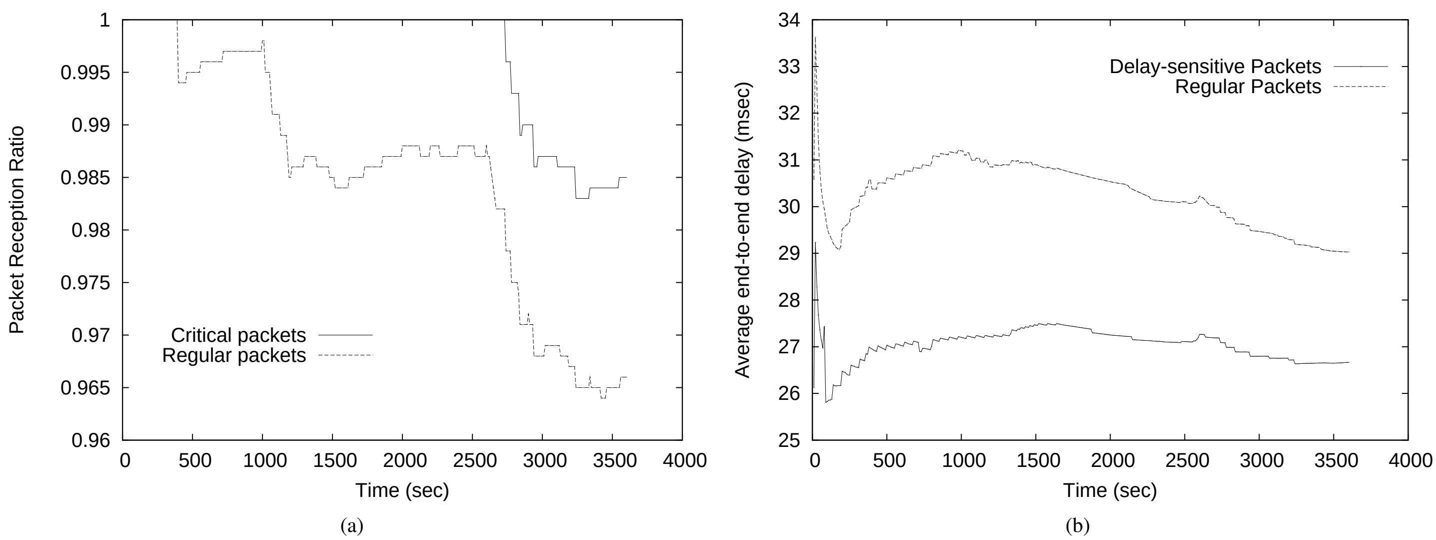


Fig. 8. a: Packet Reception ratio, b: End-to-End Delay

tends to ensure exactly the required QoS metrics in a power-aware way. It employs memory and computation efficient estimators, and it uses a multi-sink single-path approach to increase the reliability. Energy has been considered for all packets and achieved by selecting always the most power-efficient candidate among those offering the required data-related QoS (delay and/or reliability). The consideration and differentiation of both delay and reliability requirements distinguish the proposed protocol from the state-of-the-art protocols. The protocol uses multi-queuing with priority, giving more priority and hence lower latency to critical and delay-sensitive packets. This multi-queuing may be implemented at the network layer without any modification at the lower layers, and overall the protocol does not depend on a specific medium access control (MAC) protocol and requires only minor modifications at the MAC layer for calculating estimates. It can operate with any protocol as long as it employs an acknowledgment (ACK) mechanism. Simulation results show that the proposed protocol outperforms all compared state-of-the-art routing protocols by offering the best QoS while considering energy. Moreover, it has been implemented and tested in a sensor network tested. The obtained results are very encouraging and show that the protocol ensures good QoS and makes traffic differentiation based on QoS requirements. This makes the protocol suitable for WSN with heterogeneous traffic, such as medical and vehicular applications.

ACKNOWLEDGEMENTS

This work was carried out at the Norwegian University of Science and Technology (NTNU) as a part of the MELODY project funded by the Research Council of Norway, during the tenure of a Post-doc fellowship from the European Research Consortium for Informatics and Mathematics (ERCIM). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions, which significantly helped to improve the quality of the manuscript. The first author thanks CERIST research center, which is supported by the Algerian Ministry of Higher Education and Scientific Research, for the facilities during the final steps of the work.

APPENDIX

Proof of Lemma 1:
By recurrence on k:

1) for $k = 1$: the equality can be checked by replacement in the two expressions.
2) Assume the equality holds for k-1, and try to prove it for k.

$$1 + \sum_{i=1}^{k} \frac{\rho_i^l}{(1-\sum_{j=1}^{i-1}\rho_j^l)(1-\sum_{j=1}^{i}\rho_j^l)} =$$

$$1 + \sum_{i=1}^{k-1} \frac{\rho_i^l}{(1-\sum_{j=1}^{i-1}\rho_j^l)(1-\sum_{j=1}^{i}\rho_j^l)} +$$

$$\frac{\rho_k^l}{(1-\sum_{j=1}^{k-1}\rho_j^l)(1-\sum_{j=1}^{k}\rho_j^l)}$$

Using the recurrence hypothesis we get

$$1 + \sum_{i=1}^{k} \frac{\rho_i^l}{(1-\sum_{j=1}^{i-1}\rho_j^l)(1-\sum_{j=1}^{i}\rho_j^l)} =$$

$$\frac{1}{1-\sum_{j=1}^{k-1}\rho_j^l} + \frac{\rho_k^l}{(1-\sum_{j=1}^{k-1}\rho_j^l)(1-\sum_{j=1}^{k}\rho_j^l)} =$$

$$\frac{1-\sum_{j=1}^{k}\rho_j^l + \rho_k^l}{(1-\sum_{j=1}^{k-1}\rho_j^l)(1-\sum_{j=1}^{k}\rho_j^l)} =$$

$$\frac{1-\sum_{j=1}^{k-1}\rho_j^l}{(1-\sum_{j=1}^{k-1}\rho_j^l)(1-\sum_{j=1}^{k}\rho_j^l)} = \frac{1}{1-\sum_{j=1}^{k}\rho_j^l}\square$$

Proof of Lemma 2:
By recurrence on k.

1) for $k = 1$: by replacing in Eq. (7) and Lemma 2, we can check that they produce the same expression.
2) Assume the equality holds for $k-1$, and try to prove it for k .
Using the recurrence hypothesis, Eq. (7) gives

$$E[W_k] = \frac{E[R]}{1-\sum_{j=1}^{k-1}\rho_j^l}(1 +$$

$$\sum_{i=1}^{k} \frac{\rho_i^l}{(1-\sum_{j=1}^{i-1}\rho_j^l)(1-\sum_{j=1}^{i}\rho_j^l)}).$$

Using Lemma 1 we get

$$E[W_k^l] = \frac{E[R]}{(1-\sum_{j=1}^{k-1}\rho_j^l)(1-\sum_{j=1}^{k}\rho_j^l)}\square$$

REFERENCES

[1] T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher, "A spatiotemporal communication protocol for wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 16, no. 10, pp. 995–1006, 2005.
[2] E. Felemban, C.-G. Lee, and E. Ekici, "Mmspeed: Multipath multi-speed protocol for QoS guarantee of reliability and timeliness in wireless sensor networks," *IEEE Transaction on Mobile Computing*, vol. 5, no. 6, pp. 738–754, 2006.
[3] M. M.-O.-R. Md.Abdur Razzaque, Muhammad Mahbub Alam and C. S. Hong, "Multi-constrained QoS geographic routing for heterogeneous traffic in sensor networks," *IEICE Transactions on Communications*, vol. 91B, no. 8, pp. 2589–2601, 2008.
[4] K. Zeng, K. Ren, W. Lou, and P. J. Moran, "Energy aware efficient geographic routing in lossy wireless sensor networks with environmental energy supply," *Wireless Networks*, vol. 15, no. 1, pp. 39–51, 2009.
[5] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher, "Real-time power-aware routing in sensor networks," in *In Proceeding of the IEEE International Workshop on Quality of Service (IWQoS)*, 2006.
[6] A. Woo and D. Culler, "Evaluation of efficient link reliability estimators for low-power wireless networks," University of California, Berkeley,, Tech. Rep., 2003.
[7] A. Mahapatra, K. Anand, and D. P. Agrawal, "QoS and energy aware routing for real-time traffic in wireless sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 437–445, 2006.
[8] X. Wu, B. J. d'Auriol, J. Cho, and S. Lee, "Optimal routing in sensor networks for in-home health monitoring with multi-factor considerations," in *PERCOM '08: Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 720–725.
[9] T. Melodia, D. Pompili, and I. F. Akyildiz, "On the interdependence of distributed topology control and geographical routing in ad hoc and sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 520–532, 2005.
[10] T. L. Lim and M. Gurusamy, "Energy aware geographical routing and topology control to improve network lifetime in wireless sensor networks," in *IEEE International Conference on Broadband Networks (BROADNETS'05)*, 2005, pp. 829–831.
[11] S. Wu and K. S. Candan, "Power-aware single- and multipath geographic routing in sensor networks," *Ad Hoc Network*, vol. 5, no. 7, pp. 974–997, 2007.
[12] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transsaction on Networking*, vol. 12, no. 4, pp. 609–619, 2004.
[13] C.-P. Li, W. jen Hsu, B. Krishnamachari, and A. Helmy, "A local metric for geographic routing with power control in wireless networks," in *Second Annual IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, September 2005, pp. 229 – 239.
[14] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher, "Range-free localization and its impact on large scale sensor networks," *ACM Transaction on Embedded Computer Systems*, vol. 4, no. 4, pp. 877–906, 2005.
[15] T. Roosta, M. Menzo, and S. Sastry, "Probabilistic geographical routing protocol for ad-hoc and sensor networks," in *International WorkShop on Wireless Ad-Hoc Networks (IWWAN)*, London, UK, 2005.
[16] C. A. Coello, "An updated survey of GA-based multiobjective optimization techniques," *ACM Comput. Surv.*, vol. 32, no. 2, pp. 109–143, 2000.
[17] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, 2nd ed. Wesley, 1994.
[18] a. M. D. Athanasios Gkelias, V. Friderikos, and A. H. Aghvami, "Average packet delay of csma/ca with finite user population," *IEEE Communications Letters*, vol. 9, no. 3, pp. 273– 275, 2005.
[19] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: A library for the parallel simulation of large-scale wireless networks," in *The 12th Workshop on Parallel and distributed Simulation. PADS'98*, Banff, Alberta, Canada, May 1998, pp. 154–161.
[20] B. Karp and H. T. Kung, "Gpsr: greedy perimeter stateless routing for wireless networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 2000, pp. 243–254.

[21] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *LCN '04: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*. IEEE Computer Society, 2004, pp. 455–462.

[22] A. Dunkels, O. Schmidt, T. Voigt, and M. Ali, "Protothreads: simplifying event-driven programming of memory-constrained embedded systems," in *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM, 2006, pp. 29–42.

Djamel Djenouri obtained the engineer degree, the master degree, and the PhD in computer science from the University of Science and Technology USTHB Algiers, Algeria, respectively in 2001, 2003 and 2007. During his PhD he visited John Moors university in Liverpool, UK, where he carried out collaborative work with researchers of the Distributed Multimedia Systems and security group. From 2008 to 2009 he was granted a post-doctoral fellowship from the European Research Consortium on Informatics and Mathematics (ERCIM), and he worked at the Norwegian university of Science and Technology (NTNU), in Trondheim, Norway, where he participated in the MELODY project supported by the Norwegian Research Council. Currently, Dr Djamel Djenouri is a permanent full-time researcher at the CERIST research center in Algiers. His researches focus on ad hoc and sensor networking, especially on the following topics: quality of service, security, power management, routing protocols, MAC protocols, fault tolerance, sensor and actuator networks, and vehicular applications. Dr Djamel Djenouri participated in many international conferences and published more than 30 papers in international peer-reviewed journals and conference proceedings. He is a member of the ACM and served as TPC member and chair of many international conferences and workshops. He also served as reviewer for many international Journals. In 2008, Djamel Djenouri was granted the best publication award from ANDRU, supported by the Algerian government.

Ilangko Balasingham received MSc and PhD degrees from the Department of Telecommunications, the Norwegian University of Science and Technology (NTNU), Trondheim, Norway in 1993 and 1998, respectively, both in signal processing. He performed his Master's degree thesis at the Department of Electrical and Computer Engineering, the University of California Santa Barbara, USA. From 1998 to 2002, he worked as a Research Scientist developing image and video streaming solutions for mobile handheld devices at Fast Search & Transfer ASA, Oslo, Norway, which is now part of Microsoft Inc. Since 2002 he has been with the Interventional Centre, Oslo University Hospital, Oslo, Norway as a Research Scientist, where he heads the Wireless Sensor Network Research Group. He was appointed as an Adjunct Professor in Signal Processing in Medical Applications at NTNU in 2006. His research interests include super robust short range communications for both in-body and on-body sensors, body area sensor network, microwave short range sensing (non invasive, remote and continuous estimation of blood pressure), and short range localization and tracking sensors, catheters, and micro robots.