QoSNET: An integrated QoS network for routing protocols in large scale wireless sensor networks

Therence Houngbadji *, Samuel Pierre

Department of Computer Engineering, École Polytechnique de Montréal, P.O. Box 6079, Centre-Ville Station, Montreal, Quebec, Canada H3C 3A7

ARTICLE INFO

Article history:
Received 6 October 2009
Accepted 15 March 2010
Available online 20 March 2010

Keywords:
Sensor networks
Iterated function system
Routing
Quality of service
Percolation

ABSTRACT

Numerous QoS routing strategies focus on end-to-end delays to provide time constrained routing protocols in wireless sensor networks (WSNs). With the arrival of wireless multimedia sensor networks, traffic can be composed of time sensitive packets and reliability-demanding packets. In such situations, some works also take into account link reliability to provide probabilistic QoS. The trade-off between the guarantee of the QoS requirements and the network lifetime remains an open issue, especially in large scale WSNs. This paper proposes a promising multipath QoS routing protocol based on a separation of the nodes into two sub-networks: the first part includes specific nodes that are occasionally involved in routing decisions, while the remaining nodes in the second sub-network fully take part in them. The QoS routing is formulated as an optimization problem that aims to extend the network lifetime subject to QoS constraints. Using the percolation theory, a routing algorithm is designed to solve the problem on the respective sub-networks. Simulation results show the efficiency of this novel approach in terms of average end-to-end delays, on-time packet delivery ratio, and network lifetime.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Wireless sensor networks (WSNs) consist of an emergent technology deployed for a large range of solutions, spanning military, civilian, environmental and commercial applications. They consist of a large number of low-cost sensing devices equipped with wireless communication and computation capabilities. Given the recent advances in WSNs, it is expected that video sensors will be supported in such networks, for applications such as battlefield intelligence, security monitoring, emergency response, and environmental tracking. For example, multimedia sensors may monitor the flow of vehicular traffic on highways and retrieve aggregate information such as average speed and number of cars. Given the physically small nature of sensors, and since multimedia applications typically produce a huge volume of data that require high transmission rates and extensive processing, power consumption becomes a fundamental concern in WSNs. More particularly, transmitting a video stream using the shortest path will drain the node of its energy along the path, thereby shortening the network lifetime, for example. Most proposals for routing in WSNs are based on a flat, homogenous architecture in which every sensor has identical physical capabilities and can interact only with its neighbor-sensors. However, flat topologies are not always best suited to handle the amount of traffic generated by multimedia applications, including audio and video [1]. Therefore, developing new routing strategies to maximize the network lifetime, while satisfying the QoS requirements, represent a critical problem to be addressed.

Some works attempt to address the issue through single path routing strategies which only guarantee delay constraints. Compared to multipath routing, single path routing algorithms in WSNs are simpler and they consume less energy. Moreover, aside from meeting delay constraints, multipath routing is also reliable. Reliability can be characterized by way of a packet delivery ratio, which is defined as the ratio between the number of unique packets successfully received by the sink over the number of packets generated by source nodes. In WSNs, multipath routing is used to establish multiple paths between all source–sink pairs in order to increase the likelihood of reliable data delivery. Multiple copies of data along different paths [2] are sent to reduce data delivery delays by sharing transmission delays among the different paths available from the source to the destination.

Thus, end-to-end delays, reliability requirements as well as the residual energy of nodes in WSNs are considered to build a new multipath routing strategy over a discretized network area. This perspective considers how underlying communication networks can achieve hard QoS constraints while efficiently utilizing network resources. This work complements our previous work [3] on addressing systems in large scale WSANs. It extends routing...
capabilities by providing QoS for large scale WSNs that possibly include powerful devices such as actors or mobile robots. In a first step, the network area is discretized into cells using the fractal theory iterated function system (IFS) [4]. In each cell, a cell controller (CC) is elected to control the signaling operation (address allocation, statistics collection, etc.). These CCs form the cell controller network (CCN) which is also expected to integrate powerful devices such as actors and mobile robots. The remaining nodes compose the simple sensor network (SSN) which maintains the primary data routing role of sensor nodes. The hard QoS constraints are fulfilled by using a switching routing mechanism between the two sub-networks in order to determine a forwarding set. The latter consists of the selected nodes where the incoming packet will be forwarded. This switching mechanism is accomplished by applying a percolation theory [5] to the discretized network area. The remainder of this paper is organized as follows: Section 2 discusses related works and Section 3 provides an in-depth look at the proposed routing strategy. Performance evaluations are detailed in Section 4 followed by a conclusion in Section 5.

2. Related work

Routing techniques are divided into three categories based on their underlying network structures: flat, hierarchical, and location-based [6]. Furthermore, these protocols can be classified into multipath-based, query-based, negotiation-based, QoS-based, and coherent-based. This section presents an overview of the existing QoS-based routing protocols used in WSNs. QoS-based protocols allow sensor nodes to balance energy consumption and certain predetermined QoS metrics before they deliver data to the sink node. One of the first routing protocols for WSNs, sequential assignment routing (SAR) [7], introduces the notion of QoS in routing decisions according to three factors: energy resources, QoS on each path, and the priority level of each packet. In order to avoid single route failures, a multipath approach is coupled with localized path restoration. SAR maintains multiple paths from nodes to a base station. Although, this ensures fault-tolerance and easy recovery, the protocol suffers from high overhead to maintain tables and states for all sensor nodes, especially in the presence of a large number of nodes. SPEED [8], another QoS routing protocol for WSNs, provides soft real-time end-to-end guarantees, yet it also maintains a desired delivery speed across sensor networks with a two-tier adaptation method included to divert traffic at the networking layer, and locally regulating packets sent to the MAC layer. MMSPEED [9], an extension of SPEED, supports service differentiation and probabilistic QoS guarantee. For delivery timeliness, multiple network-wide packet delivery speed options are provided for different types of traffic according to their end-to-end deadlines. As is the case with SPEED, since all MMSPEED mechanisms work locally without global network state information and end-to-end path setup, they become scalable and adaptive to network dynamics. Both SPEED and MMSPEED share a common deficiency: they fail to consider energy consumption metrics. Several delay-aware routing techniques have been proposed [10–13] to improve delays. A routing algorithm proposed by [14] uses the global knowledge of node queue sizes by the gateway in order to calculate end-to-end delays, lowest-cost paths and to generate routing tables. A heuristic solution to find energy-efficient paths for delay-constrained data in WSNs has been designed in [15]. They employ topology control and a model of contention delays caused by the MAC layer. Paths between source and sink nodes are identified and indexed according to the amount of energy they consume. End-to-end delays are estimated along each of the ordered paths and the one with the lowest index that satisfies the delay constraint is selected. Multi-constrained multipath routing is proposed in [16,17], where a node to sink packet delivery is achieved based on the so-called Soft-QoS constraints expressed in terms of reliability and delays. This model addresses the issue of multi-constrained QoS in WSNs by considering the unpredictability of network topology and attempting to minimize energy consumption. However, with these approaches, packets may get routed to a node that is highly congested and/or whose energy is critical. Simulation results conducted for a small network size do not guarantee the same performance for large scale WSNs in which the number of selected paths can increase and become a source of wasted energy. A real-time power-aware routing (RPAR) protocol [18] is proposed to achieve application specified communication delays at low energy cost, by dynamically adjusting transmission power and routing decisions. In [19], a protocol called RelNForm is proposed to deliver packets with the desired reliability level by sending multiple copies of each packet along multiple paths between the source and the sink. The number of paths used is dynamically determined depending on the channel error probabilities. Instead of using disjoint paths, GRAB [20] uses a path interleaving technique to achieve a high level of reliability.

To date, all multipath routing techniques address the QoS routing in terms of the end-to-end delay and reliability in small WSNs, made up of homogeneous nodes whose capabilities are almost identical. Furthermore, it seems that there is a need to address the same issues in large scale WSNs. The novelty of our proposal is then to provide QoS in large scale WSNs, and to deal with the heterogeneous properties of the deployed nodes by assuming that powerful nodes such as actors or mobile robots can cohabitate with resource impoverished sensor nodes. Most of the multipath routing protocols that aim to provide QoS in WSNs do not have the ability to support the disparity among nodes in terms of capability. Our proposal fills that gap by taking into account these differences through the integrated QoS network that allows the nodes to switch the routing decision between two separated networks. Moreover, some works especially those in [16,17] which are closer to ours use probabilistic QoS routing to meet the QoS requirements with predefined probabilities. Instead of using the same approach, our work breaks the tie by meeting hard QoS requirements.

3. The system model

WSNs with randomly deployed nodes in a two-dimensional network area are being considered. Nodes are powered by battery and they are assumed to be aware of their own location. Indeed, nodes can find their own approximate location coordinates using triangulation or multilateration methods. A grid network architecture is considered, as it discretizes the network into two-dimensional micro-scale areas that favor operation distributed over the network. Other forms such as triangles or hexagons could be used to discretize the network area; yet the grid representation is selected for its simplicity. Current network representations using grid structures [21] usually lead to an underutilization of the radio coverage areas, as pointed out by [22]. This is due to the fact that all nodes within a grid must be able to reach all nodes in adjacent grids. Actually, nodes are forced to cover less than half the distance allowed by the radio range. To deal with this issue, the fractal grid representation, using a square form, discretizes the network area by using the entire radio coverage of nodes. The proposed system model follows the same IFS design strategy used in [3] for WSN addressing systems. IFSs are simple methods to construct fractals built by uniting several copies of themselves ad infinitum [4]. Each copy is transformed by a set of functions which stand for a finite number of M transformations \( \gamma_j \), \( 0 \leq j \leq M - 1 \).

\[
\gamma_j(x, y) = (a_jx + b_jy + e_j, c_jx + d_jy + f_j)
\]  (1)
where \( a_j, b_j, c_j, d_j, f_j \in \mathbb{R} \). For a given initial form \( S \), the small affine copies \( \gamma_j(S), \gamma_{2j}(S), \ldots, \gamma_{2^j}(S) \) are produced and lead to a new form \( w(S) = \bigcup \gamma_j(S) \). The mapping process starts with a set \( S_0 \) and computes recursively \( S_k = w(S_{k-1}), \forall k > 1 \). The sequence \( \{ S_k \} \) converges to a final set called the IFS attractor. Node coordinates are used by the IFS to determine the cell where the node is located. In order to apply IFS to the network area, four linear affine transformations that represent the mapping functions are defined as follows:

\[
\begin{align*}
\gamma_0(x, y) &= \left( \frac{x}{2}, \frac{y}{2} \right) \\
\gamma_1(x, y) &= \left( \frac{x + 1}{2}, \frac{y}{2} \right) \\
\gamma_2(x, y) &= \left( \frac{x}{2}, \frac{y}{2} + 1 \right) \\
\gamma_3(x, y) &= \left( \frac{x}{2}, \frac{y + 1}{2} \right)
\end{align*}
\]

The initial set \( S_0 = \{(0, 0), (0, 1), (a_0, a_1), (0, a_0)\} \) represents the network area and these affine transformations represent a contraction of the area by a factor \( s = 1/2 \), followed by a translation. The indexes of these transformations define a code space \( \Gamma = \{00, 01, 10, 11\} \). Indeed, \( \phi = \phi_0 \phi_1 \phi_2 \phi_3 \cdots \) denotes an element of the attractor \( \phi \in \Gamma \). Fig. 1(c) illustrates the resulting attractor. Sub-squares represent the cell and the dots show the deployed sensor nodes. The sensor node recursively computes the cell to which it belongs by applying the defined transformations and it then compares its coordinates with the obtained sub-square bounds.

Fig. 1(a) shows a one-order attractor while Fig. 1(b) provides an overview of the same attractor at a four-order level, in sub-square "00". Note that nodes are aware of the dimensions of the deployment area, as such information can be gathered by means of network-wide flooding. Since sensor nodes cannot use an infinite alphabet sequence of the code space \( \Gamma \) to determine their cell address, the \( k_{\text{order}} \) value that defines the number of iteration of the IFS, as well as the length of the cell address chosen, are finite and bounded. An address \( \phi_{\text{cell}} \) of any cell is a finite sequence of elements of \( \Gamma \) such as:

\[
\phi_{\text{cell}} = \phi_0 \phi_1 \phi_2 \phi_3 \cdots \phi_{k_{\text{order}}} \quad \phi_i \in \Gamma
\]

To allow communication between a sensor node and other nodes in its cell, a worst-case scenario is considered, in which the distance between two nodes in any cell equals the cell’s diagonal so that the cell’s side \( l_{\text{cell}} \) is bounded as \( l_{\text{cell}} \leq \frac{r}{\sqrt{2}} \). Since \( l_{\text{cell}} = a/2^{k_{\text{order}}} \), the number of sequences \( k_{\text{order}} \) in the cell address can be chosen so that:

\[
k_{\text{order}} \geq 1 + \log_2 \left( \frac{a}{r} \right)
\]

where \( r \) denotes the sensor node radius. Thus, the cell address consists of the first part of the node address. Within each cell, nodes elect a dedicated entity that represents the Cell Controller (CC), to maintain the global states within the cell. The only function of CCs is to be signaling points within the cell. To discriminate sensors in the same cell, the CC assigns an ID to each of them. Consequently, the complete address of a node can be written as:

\[
\phi : \phi_{\text{id}} , \quad 0 \leq \phi_{\text{id}} < n_{\text{cell}}
\]

where \( n_{\text{cell}} \) denotes the number of nodes in the cell of address \( \phi \) and \( \phi_{\text{id}} \), the node’s identifier in that cell. A periodic refresh, issued by the CC, triggers an update of the address pool by removing the IDs of unavailable nodes.

### 3.1. The QoS network model

With the assigned address, each sensor node can participate in the network routing operations. As indicated above, each cell is equipped with a CC that manages the cell operations. Numerous studies address QoS routing for WSNs that aims to allow each node to select a path based on QoS requirements. Although this seems to be an interesting approach, we believe that, in general, the large scale of WSNs must allow using certain nodes to build a separate network used to signal operations, and allow others to perform regular network operations, such as routing. The resulting sub-network that includes the CCs can also integrate powerful entities such as the actor overlay networks design presented in [3] or any other resource-rich nodes. Doing so should almost free each node to periodically perform a dedicated role, as is the case in most WSN clustering schemes. Such network is called cell controller networks (CCNs). The remaining nodes in the network, i.e., those which are neither elected CCs nor categorized as an actor or mobile robot, make up the second type of sub-network, which mainly performs routing operations.

Fig. 2 illustrates both layers of the network plan. The lower level plan represents the entire network, while the higher level plan is exclusively composed of CCs. Our conjecture is that by performing only network signaling operations and occasional routing decisions, the CCs will support the same burden as a simple sensor node whose only role consists of performing data forwarding. Obviously, given the large number of nodes deployed, it is assumed that it is not wasteful for sensor nodes to be exclusively dedicated to performing signaling tasks. Before delving into the details of how QoS requirements are guaranteed, let us explicitly define the parameters selected as QoS metrics.

#### 3.2. End-to-end QoS parameters

Traditional networks specify QoS in a rigid manner, using metrics such as the average bandwidth, bounded delays and jitter requirements. In dynamic environments such as mobile wireless networks, it is difficult to provision and support rigid QoS. Works by [16,17], which are similar to our proposal, deal with the so-called Soft-QoS to fulfill QoS requirements by selecting end-to-end delays and path reliability as metrics.

In our proposal, we still strive to achieve hard-QoS constraints. For that purpose, and in order to integrate the energy consumption

---

**Fig. 1.** (a) One-order attractor. (b) Four-order cells of attractors. (c) Sensor nodes spanned in the attractor cell.

**Fig. 2.** Two-layer overview of the WSN.
issue, the End-to-End Path Battery Cost (EEPBC) is introduced as additional metric. In fact, our hypothesis is that, extending the lifetime of individual nodes will prevent early failures, thus helping to maintain the hard QoS constraints. As a result, the proposed routing scheme, in addition to the end-to-end delay and path reliability, considers the residual energy of the forwarding nodes. Let $b_i(t)$ denote the battery capacity of a sensor $s_i$ at time $t$. The EEPBC $B_{\text{EEPBC}}(\psi)$ between the source node and the destination node $s_d$ is defined as:

$$B_{\text{EEPBC}}(\psi) = \sum_{\psi \in \psi} \sum_{t=1}^{T} b_i(t)$$

(6)

The rationale for choosing the residual energy in this proposal is that the information provided by the battery is easy to acquire and it offers an accurate level of energy consumed in the network. Since battery capacity is directly incorporated into the routing protocol, this metric prevents nodes from being overused, thereby increasing their lifetime.

Data can be reported at different rates depending on the type of application used or the heterogeneity of the nodes in the network. The readings generated by detecting nodes can be subject to various time deadline constraints. Let $d(s_i, s_{i+1})$ denote the delay between the sensor node $s_i$ and neighbor $s_{i+1}$. The end-to-end delay between any source node $s_s$ and the destination node $s_d$ over the set of paths $\psi$ is given by:

$$D_{\text{req}}(\psi) = \min_{\psi \in \psi} \left\{ \sum_{s_i \in \psi} d(s_i, s_{i+1}) \right\}$$

(7)

Indeed, $D_{\text{req}}(\psi)$ is the minimal achievable delay when the generated data are routed along the set of paths between $s_s$ and $s_d$. The delay $d(s_i, s_{i+1})$ between two nodes consists of the elapsed time for successfully transmitting a packet after having received it. Thus, it includes queuing time, contention time, as well as transmission, retransmission and propagation times. Since our goal aims to allow CCs to participate in occasional routing decisions, one can expect that the delay experienced to route packets in CCNs will be lower than in SSNs, as they are subject to less congestion.

Reliability, often computed to assess the probability of transmission failures, can be characterized by a packet delivery ratio, defined as the ratio of the number of unique packets successfully received at the destination over the number of packets generated by source nodes. The multipath end-to-end reliability between a source node $s_s$ and a destination node $s_d$ on a set of paths $\psi$ can be written as follows:

$$R_{\text{req}}(\psi) = 1 - \prod_{p \in \psi} (1 - r(p))$$

(8)

where $r(p)$ is the reliability of a path $p$, equal to $\prod_{s_i \in p} r(s_i, s_{i+1})$ and $r(s_i, s_{i+1})$ denotes the reliability of the link $(s_i, s_{i+1})$. Since reliability is multiplicative, a variation in any links on $p$ or an increasing number of hops on $p$ will markedly alter end-to-end reliability. That explains why single path routing cannot always meet reliability requirements. Once QoS parameters have been identified, one must define how such requirements are met. The following section defines the strategy used to put together these QoS parameters in an optimization problem.

### 3.3. Switching QoS routing

In heterogeneous WSNs, a wireless node may also contain different sensors such as audio, video and scalar sensors. As the priority of such heterogeneous traffic differs, considering service differentiation appears to be an additional issue. There are two types of traffic: real-time traffic has hard time constraints, such as delays, although it is more tolerant of packet losses; non-real-time traffic usually exists in networks and produces thousands of packets which are generated synchronously or asynchronously. In order to remain generic and not delve into the details of priority levels assigned to the different types of traffic, a general formulation of the QoS routing problem is considered. The joint goal of reducing energy consumption and end-to-end delays, while maximizing path reliability, appears to be very challenging. The routing objective thus consists of finding a set of paths $\psi(s_s, s_d)$ between source and destination that meet the specified QoS requirements at data source, while maximizing the EEPBC. The QoSNet routing problem can be formulated as follows:

**QoSNet:** Given the tuple $(D_{\text{req}}, R_{\text{req}})$ of QoS requirements specified by the source node, that represents the end-to-end delays and the reliability level, find the set of paths $\psi(s_s, s_d)$ from the source node $s_s$ to the destination node $s_d$ that maximizes the end-to-end battery power, subject to deadline and reliability constraints.

$$\max B_{\text{EEPBC}}(\psi)$$

Subject to:

$$D_{\text{req}}(\psi) \leq D_{\text{req}}$$
$$R_{\text{req}}(\psi) \geq R_{\text{req}}$$

Expression (9) maximizes the residual energy of the selected nodes, while constraints (9a) and (9b) express the delay and reliability requirements of the data source node, respectively. The problem of determining a QoS route that satisfies multiple constraints is known to be NP-hard [16]. The problem definition requires exact information regarding nodes on their paths to the destination, which is almost impossible to obtain in WSNs. Hence, to solve this challenge, each source node periodically collects information from its one-hop neighborhood. One-hop link metrics are much easier to acquire. Furthermore, such a strategy is scalable to the network size. By uniformly partitioning the QoS requirements at all downstream hops, the overall QoS requirements can be met. A node can satisfy the hop requirement by selecting next hop nodes based on the link condition. As in [16], the link delay $L_i^t$, the reliability $R_i^t$ requirements and the local battery costs $L_i^b$ for each node, considering the hop count $h_i$ are expressed as follows:

$$L_i^t = (D_{\text{req}} - D_i)/h_i$$
$$L_i^b = b_i(t)$$

(10)

(10a)

(10b)

where $D_i$ is the actual delay experienced by a packet at node $s_i$ from the source node, and $R_i$ indicates the portion reliability requirement assigned to the path through node $s_i$ as decided by the upstream node of $s_i$. Moreover, $F_w(s_i)$ denotes the forwarding set to be determined by the node. Based on the above definitions, the QoSNet routing problem can be reformulated for each node as follows:

**QoSNet:**

$$\max L_i^b$$

$$\sum_{j \in N(s_i)} x_{ij}(1 - R_{ij}) \leq \log(1 - L_i^t)$$
$$x_{ij}D_{ij} \leq L_i^b$$

(11a)

(11b)

Expression (11) maximizes the residual energy of the forwarding set. Constraints (11a) express the fact that the selected links in the forwarding set must all together satisfy the local reliability requirement. Constraint (11b) expresses that the delay of the selected links must be lower than the local delay requirement. The forwarding set $F_w \subseteq N(s_i)$ which consists of the solution to this prob-
lem, is determined by the set of neighbors $N_{\text{ssn}}(s_i)$ in the SSN, or the set $N_{\text{ccn}}(s_i)$ in the CCN, so that:

$$\begin{align*}
N_{\text{ssn}}(s_i) & \cup N_{\text{ccn}}(s_i) = N(s_i) \\
N_{\text{ssn}}(s_i) & \cap N_{\text{ccn}}(s_i) = \emptyset
\end{align*}$$  \hspace{1cm} (12)

The following section describes how nodes select the set that is subsequently considered to solve QoSNet. A typical forwarding scenario is described in Fig. 3. Feasible paths are found in the SSN when the detected event is forwarded from the source node $S$, while node $A$, when solving the problem, finds feasible paths only in the CCN. When the node fails to find a feasible solution either in $N_{\text{ssn}}(s_i)$ or $N_{\text{ccn}}(s_i)$, the packet is dropped. Each forwarding node solves QoSNet and the packet is forwarded until it reaches its destination. Intermediate nodes might receive duplicate packets for the same session. For that purpose, each node maintains a table that records the tuple $(\phi, \phi_{\text{seq}}, D_i)$ of a data reporting session. As mentioned above, $\phi : \phi_{\text{seq}}$ identifies the reporting node and seq the associated sequence number. This table is emptied periodically and the node keeps a time stamp of its last clean out in order to maintain coherent routing timing. The rationale behind this is guided by the fact that a forwarding node which receives a packet twice, forwards the latter only when the previous experienced delay is twice, forwards the latter only when the previous experienced delay is greater than the actual delay. Indeed, let $D_{\text{exp}}$ denote the delay experienced by the packet previously received by node $s_i$ for the same session. The node solves QoSNet to determine $F_w$ only when the inequality $D_i < D_{\text{exp}}$ holds. So far, we have shown how a packet is forwarded, but how the QoSNet problem is actually solved has yet to be explained. In the following section, a promising strategy is developed in order to solve this issue, by applying site percolation to the discretized area.

3.4. Mapping QoSNet resolution to site percolation

The resolution of QoSNet is limited to the node’s neighborhood $N(S_i)$, and can be achieved in polynomial time $O(|N(S_i)|)$. However, if the nodes in all cells determine their forwarding set by considering both their SSN and CCN neighbors, they will spend additional energy resources to those of the nodes in the CCN as they are already involved in the network signaling operations. To occasionally allow nodes in certain cells to consider both their SSN and CCN neighbors to solve QoSNet in such a situation, should help save energy resources and extend the network lifetime. One way of addressing this issue is to take a probabilistic approach based on the percolation theory. Introduced by Broadbent and Hammersley [5], the percolation theory has been widely applied in various fields, including economics, biology, sociology and communication. In the field of wireless networks, this theory has been used in the past to study node connectivity.

Site percolation is a kind of random graph in which the edges are formed only between open neighboring sites. In site percolation, each site can be either open, with probability $p_c$, or closed, with probability $1 - p_c$. An edge exists only where two adjacent open sites are connected. All sites are connected and become a cluster if any pair of open sites is connected. A cluster that spans from one side of the plane to the other is said to be an infinite cluster. A rigorous mathematical study of site percolation has been conducted in [4]. To model the QoSNet resolution to site percolation, consider the square grid in Fig. 4, in which each cell represents a site. A site is considered open when the nodes in that site solve the QoSNet problem by considering their neighbor in the SSN, and in the CCN when they fail to find a solution with the SSN neighbors. A closed site corresponds to the situation when the node that has a packet to forward solves the QoSNet problem by using only its SSN neighbors. Obviously, when the value of the percolation probability is $p_c = 1$, all network sites are open to solve the QoSNet problem in both SSN and CCN networks. Fig. 4 illustrates the way network nodes solve the problem. A dark square represents the cells that determine their forwarding set by considering both their SSN or CCN neighbors exclusively, while the blank boxes represent the squares that solve it using the SSN neighbors only. First, the node solves the problem by considering its SSN neighbors, and if a feasible solution is not found, the node attempts to solve it by considering the CCN neighbors if their cell is open.

So far, we have shown how to model the QoSNet resolution by way of site percolation. However, the way the various cells are selected has yet to be defined. Indeed, let $n(t)$ depict the number of instances the nodes in a given cell must solve QoSNet, to determine the forwarding set $F_w$ during a period of time $t$: $n(t)$ the number of successful attempts to find a feasible solution when the nodes consider their SSN neighbors. These parameters are collected by the CC on a periodic basis. The CC activates its cell by notifying its members to also consider the CCN neighbors for QoSNet resolution. The probability $p_c$ for a CC to be activated is computed as follows:

$$p_c = 1 - \frac{n(t)}{n(t)}$$  \hspace{1cm} (13)

Illustration: assume nodes in cell $\phi = 100001$ altogether have successfully found feasible solutions $n(t) = 100$ during the elapsed period of time $t$, while receiving $n(t) = 120$ to forward. The CC of that cell will activate its cell with the probability $p_c = 0.16$ to allow the nodes in that cell to consider the neighbors in $N_{\text{ccn}}(s_i)$ for the subsequent time period. If the cell is activated, its CC will notify its member, in order to use this new setting to process the incoming packets to be forwarded. Table 1 shows the QoSNet pseudo-code executed by the node when it has packets to forward.

4. Performance evaluation

This section shows the effectiveness of the proposed routing scheme in terms of on-time packet delivery ratio, average end-to-end delays, and the network lifetime. For that purpose, extensive simulations were conducted in Qualnet [23] to assess the performance of QoSNet. Baseline comparisons were made with the MCMP [16] and the God routing. These schemes are selected for the following reasons: (a) the God routing indicates the higher achievable performances of an ideal QoS routing protocol; (b) the MCMP relies on Soft-QoS constraints by setting the values of predefined parameters. Instead, our goal in this comparison is to show that, hard QoS constraints can still be met when using our scheme. Moreover, the MCMP protocol is more tied to our work in term of routing strategy. Indeed, in the God routing, each node knows the
state of all network links and nodes. This scheme can be viewed as an upper bound of a QoS routing algorithm with end-to-end delays and reliability requirements. The average delivery delay equals the latency experienced by successfully received packets. On-time packet delivery ratio means the number of successfully received packets that satisfy the QoS requirements over the total number of generated packets.

Our simulations assess the worst-case performance where link delays and reliability levels always change suddenly at any transmission instant, making them unpredictable. The success probability of each transmission is randomly picked from $[0.8, 0.9]$, which implies that the link reliability ranges from 0.8 to 0.9. Link delays are also randomly distributed over the range of $[1, 50]$ ms. Link delays consist of the time elapsed to successfully transmit a packet after it was received. It thus includes queuing time, contention time, as well as transmission, retransmission and propagation times. The sink is located on the upper right corner of the sensor field. Over time, energy is dissipated at regular intervals until a connection time or the network lifetime. Let $T_i$ denote the disconnection time of an individual node. The network lifetime equals the time when the residual energy at any one of the sensor node reaches zero, and is given as:

$$T = \min \left( \left\{ t_i \right\}_{i=1}^n \right)$$  \hspace{1cm} (14)

The simulation implemented in Qualnet follows the parameter settings shown in Table 2. The delay constraints are taken in the range of $[60, 120]$ ms and the default reliability threshold is set at 0.7. For the MCMP routing, both parameters $\alpha$ and $\beta$ are set at 95%. A data packet contains 128 bytes. Different random seeds were applied to generate different network configurations and all simulations lasted 1000 s. The results show an average of over 15 simulation runs.

To highlight how percolation affects the performance indexes, a scenario was simulated where each CC activates its cell with a given percolation probability. Figs. 5–7, respectively, illustrate the on-time packet delivery ratio, the average end-to-end delay, and the network lifetime for various percolation probabilities. Indeed, allowing the participation of CCN nodes in routing decisions helps improve the QoS requirements. However, there is a price to pay for such performance, as evidenced by the network lifetime which decreases when many CCN nodes are involved in routing decisions. If the nodes in CCN are made of powerful devices like actors or mobile robots, as described by the AON in our previous work [3], that will result in better performances; in such scenario, wireless links between such devices are more reliable and are subject to short transmission delay.

Figs. 8 and 9, respectively, present simulation results by comparing other routing schemes for on-time delivery ratio and the average latency experienced by the delivered data packets. The

### Table 1
QoSNet routing algorithm.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1.   | $T(i) = N_{init}(i); 
      Forwarding set $F_{in}(i) = \emptyset$ |
| 2.   | $\text{candidates set } F_{in}(i) = \{h_j, b_k, \text{ and } \text{all } \text{nodes } j \in \text{set } T(i)\}$ |
| 3.   | If $C_{in}(i) \neq \emptyset$ |
| 4.   | Compute $l_i^s = \frac{1}{\sqrt{E_i}}$ |
| 5.   | Do |
| 6.   | Select the node $v \in C_{in}(i)$ with the maximum residual energy $b_v$ |
| 7.   | $F_{in}(i) = F_{in}(i) \cup \{v\}$ |
| 8.   | Until constraint (9a) is met |
| 9.   | Else if $(C_{in}(i) = \emptyset$ or $F_{in}(i) = \emptyset)$ then |
| 10.  | If the cell is activated then |
| 11.  | $T(i) = N_{init}(i)$; goto 2 |
| 12.  | Else |
| 13.  | Update statistics |
| 14.  | Discard the packet and return |
| 15.  | End if |
| 16.  | End if |

(A) When the node receive a packet to forward

(B) When the node is a Cell Controller

1. At the expiration of the timer $T_{percolation}$ do
2. Collect the statistics from cell members
3. Compute the percolation probability $p_c$
4. Activate the cell with probability $p_c$ and send notification to members

End

### Table 2
Simulation parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network area</td>
<td>200 m x 200 m</td>
</tr>
<tr>
<td>Sensors</td>
<td>256</td>
</tr>
<tr>
<td>Transmission range</td>
<td>40 m</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>250 kbps</td>
</tr>
<tr>
<td>Packet size</td>
<td>128 bytes</td>
</tr>
<tr>
<td>Simulation time</td>
<td>1000 s</td>
</tr>
<tr>
<td>Reliability</td>
<td>[0.8, 0.9]</td>
</tr>
<tr>
<td>Reporting rate</td>
<td>1 packet/s</td>
</tr>
<tr>
<td>Percolation period</td>
<td>50 s</td>
</tr>
</tbody>
</table>

![Percolation stages for QoSNet resolution.](image)

![Effect of percolation on the delivery ratio.](image)
QoSNet routing scheme performs better in terms of delivery ratio when compared to the MCMP scheme, since it takes the residual energy of the routing nodes into account. This performance can be explained by the fact that some nodes in the MCMP scheme are frequently chosen and soon run out of energy. Obviously, the God routing scheme outperforms all of the studied schemes due to the fact that all network nodes are aware of the state of all network links.

The QoSNet scheme is more akin to the God routing as the nodes know the state of the local links, while the knowledge is global in the God routing scheme. Fig. 10 illustrates the performance of the respective schemes in terms of network lifetime when the delay requirement varies. As expected, since the QoSNet scheme takes into account the residual energy of the nodes for path selection, it seems that the network lives longer than that of the MCMP scheme, which only minimizes the number of paths in order to save energy. The QoSNet scheme outperforms the God routing scheme for low delay requirements as in God routing scheme, the node selects links with the least delays, potentially including nodes with low residual energy. However, the QoSNet scheme only selects links with nodes endowed with high residual energy; this explains why it outperforms God routing in terms of low delay requirements.

Figs. 11 and 12 illustrate the results of the respective schemes when the value of the reliability requirement varies. As for the QoSNet scheme, the delivery ratio is tied to the God routing scheme and does not influence the performance indexes. This situation can be explained by the fact that the paths selected satisfied...
together the reliability requirements. However, the gap between the respective schemes in terms of the delivery ratio is due to the fact that when the nodes do not find a feasible solution, they drop the packet to be forwarded; MCMP thus drops more packets than other schemes.

Figs. 13–15, respectively, illustrate the average end-to-end delay, the on-time packet delivery ratio, and the network lifetime when the number of deployed nodes varies. It clearly appears that using the QoSNet routing protocol offers a good benefit compared to the MCMP scheme in terms of delay and delivery ratio. As expected, the node redundancy increases, so that the network partition time is prolonged. The nodes in the paths selected by QoSNet live longer as evidenced in Fig. 15. This resilience allows a diversity of choices among the nodes in the vicinity of the routing node, and then favors the lower delay links to be considered as forwarding links. The QoSNet scheme only selects links with nodes endowed with high residual energy and then outperforms the God routing in term of network lifetime. Indeed, with the God routing scheme the node selects links with the least delays, potentially including nodes with low residual energy. Consequently, the same node transmits more packets overall with God Routing than QoSNet, and so the network depletes in advance. Meanwhile, the God routing scheme performs better in terms of delay and delivery ratio, as it models the ideal case where each node is aware of the states of every node and every link in the network, and then it selects the path accordingly.

When the number of nodes increases, the path diversity is enhanced, and the multiplicity of the links that satisfy the QoS requirements allows the schemes to select the best possible path. As it is impossible in reality for a node to apply the God routing...
scheme, the QoSNet scheme is a good alternative because it is closer to the God routing scheme than MCMP. Fig. 16 illustrates the number of alive nodes over the time. The simulation runs 2000 s and the number of alive nodes is taken every 200 s. The nodes in MCMP are depleted more quickly because that scheme reduces the number of paths in order to prolong the network lifetime, while QoSNet relies on the nodes residual energy and the switching mechanism of percolation to achieve the same goal. The energy hole created by this depletion dramatically reduces the lifetime of the network in MCMP. This justifies the results depicted in Figs. 15 and 16 on the network lifetime and the number of alive nodes in the network.

5. Conclusion

This paper presents a new routing strategy to provide QoS in large scale WSNs. Existing routing algorithms that aim to provide QoS, focus only on end-to-end delays and link reliability in relatively small wireless networks. To deal with the scale factor, the proposed scheme splits WSNs into two sub-networks composed of nodes spread over a discretized area. The first includes cell controllers and possibly other powerful nodes such as actors, while the second is composed of the remaining sensor nodes. Considering the QoS metrics such as end-to-end delays and reliability, the proposed routing strategy is designed for large scale WSNs whose goal consists of extending network lifetime. By using the percolation theory, a switching mechanism allows the node that has a packet to forward to select forwarding nodes among its neighbors in the two sub-networks. Simulation results validate our scheme by showing its efficiency in terms of mean end-to-end delays, on-time packet delivery ratio, and most importantly, extended network lifetime.

References