Stochastic Time Evolving Routing Protocol based on Energy and Delay Metrics

Nesrine Ouferhat & Abdelhamid Mellouk
Image, Signal and Intelligent Systems Lab-LISSI & Dept. of Networks and Telecoms, IUT C/V
University of Paris-Est (UPEC), Creteil, France.
Email: {ouferhat, mellouk}@univ-paris12.fr

Abstract—**In this article, we introduce a Quality of Service (QoS) routing algorithm based on dynamic state-dependent policies. The proposed algorithm uses a bio-inspired approach based on trial/error paradigm to optimize two QoS different criteria: Energy and end-to-end delay. Our proposal, called EDEAR "Energy and Delay Efficient Adaptive Routing", uses a paradigm based on routes' exploration. In this phase, we collect information in terms of energy and delay by using continuous learning parameters on the network and update routing table maintained at each node of the network. This phase of exploration has been optimized by proposing a new algorithm based on multipoint relay for energy consumption, thus reducing the overhead generated by the packets exploration. Numerical results obtained with NS simulator for different levels of traffic's load and mobility show that EDEAR gives better performances compared to traditional approaches.**

***Index Terms*—State dependent algorithm, Multi criteria routing optimization, Delay, QoS, Energy consumption.**

I. INTRODUCTION

Wireless sensor networks (WSN) are a particular type of Delay Tolerant Networks that exhibit intermittent connectivity. Sensors are deployed in WSN and exchange environmental information to establish an overview of the region monitored on the network. This information is made accessible to external users through one or more intermediate (s) node(s) [1].

A sensor is a physical component able to accomplish three tasks: identify a physical quantity, treat any such information, and communicate with other sensor nodes [1], [2], [3].

Therefore, QoS measures should be introduced to the network so that quality of real time services can be guaranteed. So, one of the most promising directions underlying WSN networks is routing mechanisms, that can guarantee new value added services by addressing the integration of dynamic criteria supported by the network, and not only the static ones. The most popular formulation of the optimal distributed routing problem in a data network is based on a multicommodity flow optimization whereby a separate objective function is minimized with respect to the types of flow subjected to multicommodity flow constraints. Given such complexity due to the diversity of the QoS constraints, we focus our attention in this paper in bio-inspired QoS routing policies based on Reinforcement Learning paradigm.

In fact, a lot of work is actually done concerning the optimization of the energy consumption constraint which directly affects the lifetime of the network [1], [2], [4]. In the case of routing function, several protocols have been also proposed to conserve energy consumption in sensor networks, moreover, connectivity and coverage of sensor nodes allow to communicate among themselves in order to send data to the sink [2], [3]. Other routing protocols have been also proposed in WSN to improve other QoS constraints such as delay.

Our aim in this paper is to propose a new adaptive routing protocol that improves two dynamic criteria simultaneously: energy consumption and end-to-end delay based on a continuous time evolving mechanism, called EDEAR (Energy and Delay Efficient Adaptive Routing). The integration of more than one QoS parameters in routing protocol increases the complexity of current routing algorithms. In fact, the problem to determine a QoS route that satisfies two or more path constraints (for example, delay and cost) is known to be NP-complete [5]. One major difficulty is that the time required to solve the Multi-Constrained Optimal path problem exactly cannot be upper-bounded by a polynomial function. Hence much of the focus over the last few years has been on the development of pseudo-polynomial time algorithms, heuristics, and approximation algorithms for multi-constrained QoS paths.

Our paper is organized as follow: In next section, we present routing approaches in WSN. We shall present state dependent policies in routing paradigm in section 3. Our proposal is detailed in section 4. Simulations are given in section 5 and we will finish by conclusion in last section.

Manuscript received Mars 1, 2009; revised September 1, 2009; accepted November 1, 2009.
Corresponding author : mellouk@univ-paris12.fr

doi:10.4304/jcm.5.2.144-151

II. ROUTING IN WSN

Goal aspects that are identified as more suitable to optimize in WSN are QoS metrics. Sensor nodes essentially move small amounts of data (bits) from one place to another. Therefore, equilibrium should be defined in QoS and energy consumption, to obtain meaningful information of data transmitted. Energy efficiency is an evident optimization metric in WSN. More general, several routing protocols in WSN are influenced by several factors:

A minimum life of the system: In some cases, no human intervention is possible. Batteries of sensors can not be changed; the lifetime of the network must be maximized.

Fault tolerance: Sensor network should be tolerant to nodes failures so as the network route information to other nodes.

Delay: Delay metric must be taken into account for real-time applications to ensure that data get on time.

Scalability: Routing protocols must be extendable, even with several thousand nodes.

Coverage: In WSN, each sensor node obtains a certain view of the environment. This latter is limited both in range and in accuracy (fig. 1); it can only cover a limited physical area of the environment. Hence, area coverage is also an important design parameter in WSN. Each node receives a local view of its environment, limited by its scope and accuracy cover a large area composed by the union of much small coverage.

Connectivity: Most WSN have high density of sensors, thus precluding isolation of nodes. However, deployment, mobility and failures varied the topology of network, so connectivity is not always assured [6].

Quality of Service: In some applications, data should be delivered within certain period of time from the moment is sensed; otherwise the data will be useless. Therefore bounded latency for data delivery is another condition for time-constrained applications. However, in many applications, conservation of energy which is directly related to network lifetime is considered relatively more important than the quality of data sent. As the energy gets depleted, the network may be required to reduce quality of results in order to reduce energy dissipation in nodes and hence lengthen network lifetime. Hence, energy-aware routing protocols are required to capture this requirement.

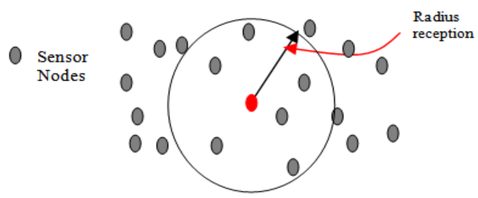

Figure 1. Radius of reception of a sensor node.

To ensure these aspects, one can find two methods to routing information from a sensor network to a sink:

Reagent (requested): If you want to have the network status at a time T, sink broadcasts a request throughout the network so that sensors back their latest information reported to the sink. The information is then forwarded in multi-hop manner to sink [7].

Proactive (periodical): The network is monitored to detect any changes, sending the sink periodically broadcasts, following an event in the network such as a sudden change in temperature and movement. Sensors near the event back then the information recorded and route it to the sink [8].

Otherway, depending on the network structure, routing can be divided into flat-based routing, hierarchical-based routing, and location-based routing. All these protocols can be classified into multi-path based, query-based, negotiation-based, QoS-based, or coherent-based routing techniques depending on the protocol operation.

Below, we shall present some QoS routing protocols in WSN.

A. Sequential Assignment Routing (SAR)

SAR manage multi-paths in a table routing which attempts to achieve energy efficiency and fault tolerance. SAR considers trees QoS criteria during the exploration of routing paths, the energy resource on each path and priority level of each packet. By using these trees, multiple paths of the sensors are well trained. One of these paths is chosen depending on energy resources and QoS of path. SAR maintains multiple paths from nodes to sink [9]. High overhead is generated to maintain tables and states at each sensor.

B. QoS routing protocol with energy consideration

The link cost used in this protocol is formulated as a function which takes into account node's energy consumed and error rate. Based on extended version of Dijkstra's algorithm, the protocol establishes a list of paths with minimal cost. Then, the one with the best cost is selected [10]. This protocol uses hierarchical routing; it divides the network into clusters. Each cluster consists of sensor nodes and one cluster head. The cluster head retrieves data from the cluster and sends it to the sink. QoS routing is done locally in each cluster.

C. SPEED

This protocol provides a time limit otherwise the information is not taken into account. It introduces a concept of "real time" in wireless sensor networks. Its purpose is to ensure a certain speed for each packet in network. Each application considers the end to end delay of packets by dividing the distance from the sink by the speed of packet before deciding to accept or reject packet. Here, each node maintains information of its neighbors and uses geographic information to find the paths. SPEED can avoid congestion when the network is heavy [9].

III. WSN STATE DEPENDENT ROUTING APPROACHES

Learning process allows improving the performance of system basis on past experience. This improvement is based on the ability to extract information from experiments, and on exploiting this information to improve a given performance function.

In general, learning is the automatic modification of system parameters or, more rarely, the number and organization of system, to adapt the treatment to a particular task.

There are three families of learning approaches depending on the available information nature and aim research:

- supervised learning,
- unsupervised learning,
- and reinforcement learning.

In the following, we call agent, entity or system subject to learning [11].

A. Supervised learning

This type of learning is commonly used, it is necessary to have a set of data pairs (input/desired output). The difference between the network output and the desired output provides a quantitative error measure in the network calculation; this error is used for adaptation. The most used learning method is the gradient propagation.

B. Unsupervised Learning

Unlike supervised learning, only input information is supplied to the system. The choice of output is based on similarities found between the different inputs (according to auto – organizing rule). The system will discover the patterns in these configurations that can be divided into similar configurations groups.

C. Reinforcement learning

In this case, there is often only qualitative information for evaluate the response calculated. Reinforcement learning uses this assessment to improve system performance. This form of learning is a method of learning by trial and error.

Reinforcement learning (RL) is learning for which only a qualitative measure of the error is available. In this case, the agent receives stimuli from its environment and responds by choosing an appropriate action for his behavior. His reaction is judged according to a predefined objective in a form of note, called a "reward". Agent receives the reward and modifies its future actions to reach an optimal behavior. An action noted negatively will use under the same conditions less then a positive note. RL is to learn for an autonomous agent a behavior to be adopted in its interaction with its environment to achieve objectives without any external intervention. RL is based on two simple principles:

1. For a given state, if an action causes immediately something bad, then the system learns to not make this action in the same state.

2. If in a given state, all possible actions lead to something bad, then the system will learn to avoid being in this state.

The agents used in adaptive routing are in majority based on RL paradigm. The basic idea is to improve current path optimization policy due an interaction with the time evolving parameter's network to take into account their dynamicity without requiring local assessment of a deployed strategy. However, most algorithms for reinforcement learning use a value function of Markov Decision Process (MDP) which constitutes the formal model of RL. There are two main strategies for solving RL problems. The first one is to search in space of behaviours in order to find one that performs well in the environment. The second one is to use statistical techniques and dynamic programming methods to estimate the utility of taking actions in states of the environment.

On each step of interaction, the agent receives an input, *i,* some indication of the current state *s* of the environment; the agent then chooses an action *a* to generate an output. The action changes the state of the environment, and the value of this state transition is communicated to agent through a scalar reinforcement signal. Fig. 2 represents the agents learning the environment [5].

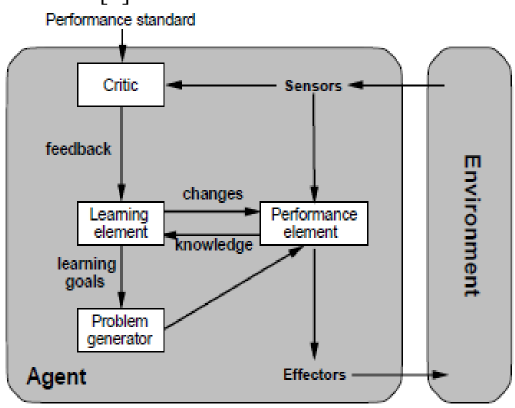

Figure 2. Agent's learning from the environment.

IV. EDEAR PROPOSAL

As delay tolerant networks are characterized by dynamic traffic conditions, this requires taking into account the level of QoS routing process. Therefore, each approach of adaptive routing must be robust and reactive enough to accommodate changes in traffic conditions while minimizing the packet delivery time. Approaches of routing based on reinforcement learning have been proposed recently. This type of approach is well suited to the problem of QoS routing, as the environment model where node is located is a priori unknown. However, the effectiveness of these approaches in routing decision depends highly on conditions of current traffic on the network and its evolution.

Our proposal is based on adaptive approaches; its objective is to find the best path in terms of energy consumed and end-to-end delay. It assumes that at each node, information about the residual energy, the energy consumed and the average delay links are available even before the route is requested. We assume that these estimates are made during the discovery of neighbor by an external mechanism independent of routing process.

Our proposal is also considered as hybrid protocol; it combines the concept of on demand searching routes, and proactive exploration concept. This joint mechanism in the exploration phase allows to EDEAR to find alternative routes missed in the first phase. For this, EDEAR is based on two agent explorers: *Int-E-Route* (Interest Exploration Route) and *Resp-E-Route* (Response Exploration Route). The first one, generated at the sink when this one demands a request, is available on the network to find routes to the source node. To limit overhead generated by packets discovery, we use an optimized distribution mechanism based on Multi Point Relay (MPR) OLSR protocol [12].

The arrival agent *Int-E-Route* to the source node initiates the creation of the second agent, *Resp-E-Route*. This latter takes the reverse path followed by the first Int-E-Route agent.

A. Int-E-Route Agent

We use the mechanism of periodic "Hello" messages to discover the neighbors. Each node broadcasts Hello messages, containing a list of its neighbors and the links state with its symmetrical neighbors, asymmetrical or lost. "Hello" messages are received by its one hop neighbors and will not be relayed; each node is able to know its one and two hop neighbors. A message is sent in broadcast way and will not be relayed (the distribution is local). This latter is composed by:

- A list of addresses of its symmetrical neighbors.
- A list of addresses of its asymmetrical neighbors.

When a "Hello" message is received by a neighbor, if a node finds its own address in the asymmetrical list, then, this link is valid.

A node is considered as a neighbor if and only if the collision rate of "Hello" packets sent to it, is below a certain threshold fixed initially.

Whenever, Int-E-Route gets through intermediate node, it retrieves its address and the information of its residual energy, energy consumed on link and the average delay.

There is a neighbour table in each node of the network. This table, shown in fig. 3, contains:

- N-id: ID of its one-hop neighbours.
- N-states: link state with its one hop neighbors (symmetric or asymmetric).
- List-N-2 hops: its two-hop neighbors.
- E_r : Residual energy of node.
- E_c : Energy consumed on each link by one-hop neighbors.
- D_{link} : Link delay.
- QE_c : Cost energy consumed on the link.
- QD_{link} : Cost delay of the link.

Id-N	N-states	List N-2 hops	E_r	E_c	D_{link}	QE_c	QD_{link}

Figure 3. Neighbors Table.

Packets *Int-E-Route* are sending periodically as the route to this node is requested. This period is defined by the interval exploration δt which is calculated by simulation (for our scenarios, this value started with 10 seconds). Thus, each δt times, a new packet *Int-E-Route*, with new sequence number, is created and sent in the same manner as first *Int-E-route* is generated.

This agent is sent in broadcast to all one-hop neighbors nodes. To reduce overhead generated by packets exploration, only MPRE nodes will be allowed to relay this packet.

Int-E-Route recovers energy consumed and delay with its neighbors and stores it in list memory. The sequence number of packet does not change. This process continues until *Int-E-Route* arrives to the source node.

The cost path from the source to sink is calculated as follow:

Let's the residual energy of each node must be exceeds a parameter ε fixed before (ε corresponds to the energy required to send a fixed volume of information):

$$E_r \geq \varepsilon \qquad (1)$$

Where:

E_r: represent the residual energy and ε the threshold of minimum energy for active node before sending all data.

The cost of a node *i* for a link between nodes *i* and *j* is computed as:

$$Cost\ of\ node_i^j = Cost\ of\ link_i^j = \gamma QE_{c_i}^j + \theta QD_i^j \qquad (2)$$

Where:

QE_C is the cost of energy consumed on link *(i, j)*, and QD represent the cost of mean delay on link *(i, j)*, γ *and* θ are tuneable parameters, with $\gamma \succ \theta$ to give more importance to the energy metric.

The cost function can be completed as the cost of path as follows:

$$f_{cost} = \sum_{j=1}^{K} h_j C_j \qquad (3)$$

C is a scalable vector for QoS metrics, and K represent the number of QoS criteria considered in routing process.

The cost of the whole path constructed with N nodes between source node and sink is:

$$\text{Cost of path} = \gamma \sum_{i=1}^{N-1} QE_{ci} + \theta \sum_{i=1}^{N-1} QD_i \qquad (4)$$

B. EDEAR's exploration mechanism

In our EDEAR proposal, we used the same mechanism as OLSR [12] by replacing the cost function by the energy cost stored in the neighbors table (fig. 3). So, the choice of MPRE for each node will be based on the links that offer the best cost estimated locally. An example of MPRE process is illustrated in fig. 4. The algorithm for selecting MPRE should be summarized as follows:

• Each node N_i of the network as MPRE chooses among its neighbors for one-hop, all the nodes up to a neighbor with two-hops.

• Each node selects as MPRE node that has the best link cost. In the case of equality between two nodes, the one that covers more than two-hop neighbors is chosen. This step is repeated until all neighbors at two hops are covered.

In this example, the node *N1* searches their neighbors MPRE that will route the packets *Int-E-Route*. First, *N1* will choose the node *N2* because *N6* is an isolated node. Then, it will choose the node *N3* MPRE which has the minimum energy. It will choose after the node that has the minimum energy *N3* or *N4*. Such as *N3* and *N4* have the same energy, *N1* will choose as MPRE node *N4* due the fact that is covers a maximum of nodes. *N1* has found all its neighbors MPRE once all the nodes in two hops are covered.

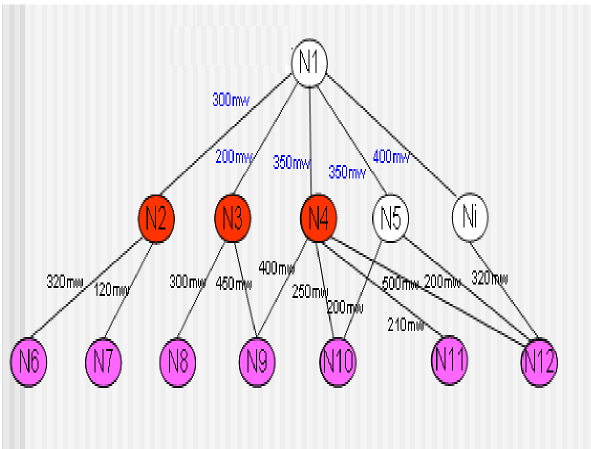

Figure 4. MPRE algorithm

C. EDEAR's Resp-E-Route Agent

Agent *Resp-E-Route* updates routing tables based on information in the neighbors tables over all nodes crossed. This update consists of readjustment of link costs associated with each entry in routing table and generated by source node upon reception of *Int-E-Route* packet. It retrieves all information gathered by the packet *Int-E-Route* at the sink. The Packet *Resp-E-Route* is sent in unicast mode and takes the opposite route borrowed by the packet *Int-E-Route*. The source node generates packets *Resp-E-Route* on the basis of information on total cost of path followed by the packets *Int-E-Route*. When the arrival of the first packet *Int-E-Route* to source node is happen, the variable *"best cost path"* is used to store the cost of the path. At each arrival *Int-E-Route* packet to the source node, the cost of path calculated for this route is compared with the lowest cost path saved in the variable "best cost route". If there is a better path, the traffic will switch to this one. Each *Resp-E-Route* packet retraces exactly the same path taken by the *Int-E-route* packet which is at the origin of the path. At each passage trough the intermediate node, it compares its address with the destination address of sink *Resp-E-Route* packet to verify if it reaches the destination.

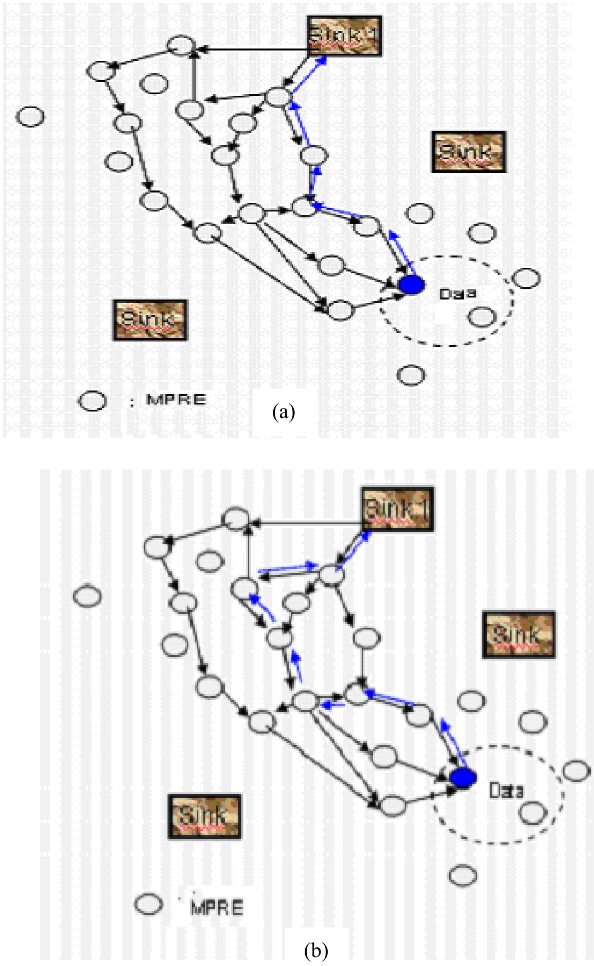

Figure 5 (a) and (b). Data sent over the best path.

Fig. 5 shows a scenario where the information is routed on the best path (fig. 5a). If a better path is present during the evolution of the information (fig. 5b), EDEAR switches this information to the new best path. This latter is chosen based on the "*best cost path*" variable. In the case where the new value is less than the previously value stored, the value is updated and a *Resp-E-Route* packet is then generated. Otherwise, no packet *Resp-E-Route* will be generated. This technique reduces the number of packets exploration *Resp-E-Route* and restricts the exploration of routes to those offering the lowest cost path to the sink node containing data. *Resp-E-Route* packets are sent in unicast mode and each one retraces exactly the same path taken by packet *Int-E-route*. At each passage through intermediate node part of the route, it compares its address with the destination address of sink.

If the node under consideration is the recipient of *Resp-E-Route*, a calculation of routing table is done by using information regarding the cost of path recorded in packet *Resp-E-Route*. Otherwise, the node calculates the new cost of the route already travelled and sends packet *Resp-E-Route* to the next neighbor in the list of intermediate nodes.

D. EDEAR's updating and calculating routing table

Routing process consist to find the path with minimum cost in order to minimize simultaneously node energy consumed on all paths and the end-to-end delay between sink and source node. The fundamental problem is to update the link cost; the algorithm that we propose is adaptive and is based on routing tables maintained at each node. When a route is requested, the node controls its routing table if there is an entry that corresponds to requested node. If no entry appears for that node, the node creates an entry and initializes the cost of links; evenly distributed over all its neighbors MPRE.

These costs are updated during transition from the agent *Resp-E-Route* based on information of links cost retrieved by visiting all nodes on the route. The updated link cost is done as follows (fig. 6):

- Replaces in each link:

$$Cost\ of\ node_i^j = Cost\ of\ link_i^j = \gamma QE_c + \theta QD \quad (5)$$

The new costs at $(t + \delta t)$ in the node i on the link *(i, j)* is:

$$QE_{c_i}^j(t+1) = \frac{E_{c_i}^j(t)}{E_{c_i}^j(t+\delta t)} * QE_{c_i}^j(t) \quad (6)$$

and

$$QD_{c_i}^j(t+1) = \frac{D_{c_i}^j(t)}{D_{c_i}^j(t+\delta t)} * QD_{c_i}^j(t) \quad (7)$$

Where:

$E_{c_i}^j(t)$: Energy consumption in the node i on the link *(i, j)* at a time t.

$E_{c_i}^j(t + \delta t)$: Energy consumption in the node i on the link *(i, j)* at a time $(t+\delta t)$.

$D_{c_i}^j(t)$: Link delay in the node i on the link *(i, j)* at a time t.

$D_{c_i}^j(t + \delta t)$: Link delay in the node i on the link *(i, j)* at a time $(t+\delta t)$.

We refresh then the new cost of the path based on equation (4) and continues iteratively the update process.

The variable "Best cost route" is updated as follow:

$$\text{'Best cost route'} = \text{Max}\left(\gamma \sum_{i=1}^{N-1} QE_{ci} + \theta \sum_{i=1}^{N-1} QD_i\right) \quad (8)$$

```
Start
  Do
     Every δt  sec
       Do
         Generate Int-E-Route

           If there is no entry in the routing table
             for the 'sink'
             Then
             Create an entry in the routing table
           AND
               broadcast the agent Int-E-Route
            Else
               Broadcast agent Int-E-Route

                If Id_Node ≠ Id_sink
                  If Int-E-Route is already past
                  AND
                  Current 'best cost route' ≤ Past
                  'best cost route'
                  Then
                  Recover the cost link AND
                  recover Id_ node
                  Calculate the total cost AND
                  broadcast  Int- E-Route
                Else
                    Send Resp- E-Route
              End If
            End If
End
```

Figure 6. Int-E-Route algorithm

V. SIMULATION AND RESULTS

To evaluate our proposal, we choose to compare with SAR protocol [9]. This one is in the same family as EDEAR, it represent also a QoS routing protocol which takes into account energy and delay according to the end-to-end delay. Simulation on NS was done according different criteria: energy consumption, end-to-end delay, loss rate and mobility. Simulations have been performed for different network topologies. Nodes, whose number varies from 10 to 100, are randomly distributed in network area.

In fig. 7, we calculate the end-to-end delay (sec) in the Y-axis depending on flow rate in the X-axis (packets/sec). This allows improving the end-to-end delay even in a loaded network. EDEAR gives better performance compared to SAR in the case where the network has a traffic volume. As EDEAR is an adaptive routing protocol, it takes into account the actual state of the network and learns from past experiences in the selection process of new paths.

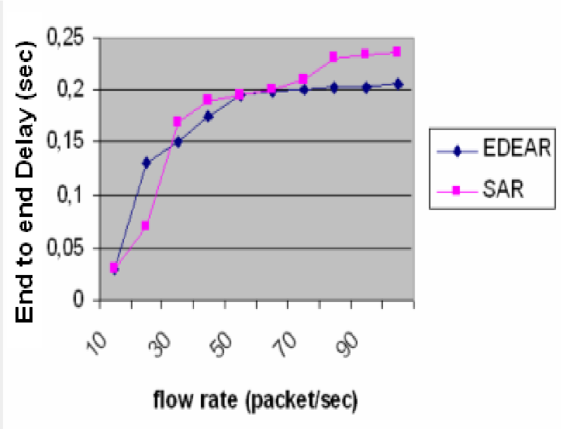

Figure 7. End-to-end delay regarding flow rate.

Fig. 8 gives network energy consumption results for EDEAR and SAR protocols in terms of number of connexions in the deployed network. Based on the cost function to select routing paths, our protocol chooses the path with the least energy consumption, conserve energy of the network and increase the lifetime of the network. We can note also that EDEAR consumes energy less than SAR both in network with low and high traffic volume.

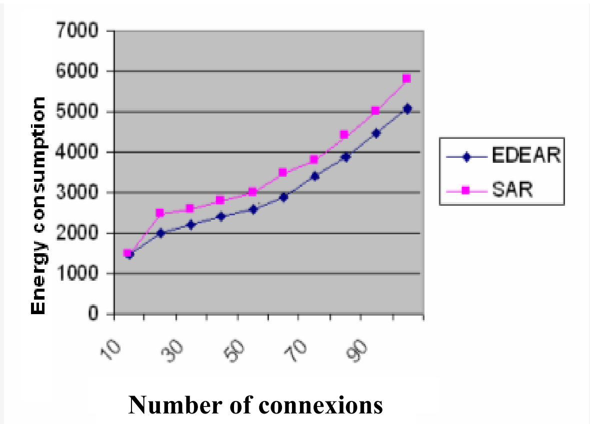

Figure 8. Energy consumption regarding number of connexions.

Fig. 9 compares the loss rate for EDEAR and SAR protocols in terms of number of connexions. Results confirm our analytical study; EDEAR gives better performance than SAR in high load traffic conditions.

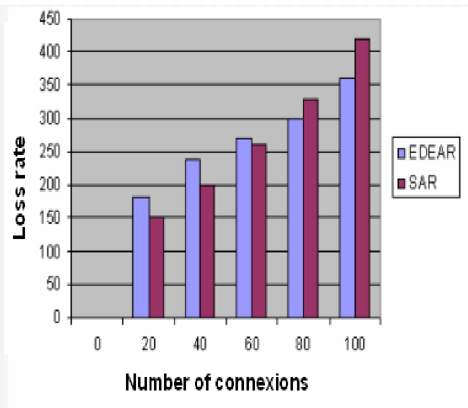

Figure 9. Loss rate regarding number of connexions.

A last scenario used in our simulations concerns mobility of the deployed sensors. Fig. 10 and 11 represents a scenario with low and high mobility. We can note that EDEAR gives better performance than SAR in high load traffic conditions in the both cases: low and high mobility of sensors. Routing tables are probabilistic in EDEAR; several paths may be route to destination. Therefore, if neighbors change, the mechanism of adjusting probabilities can be adapted to almost instant to degradation performance on followed path.

EDEAR can route on the best path and don't need to reinitialized time as SAR protocol. Indeed, SAR protocol requires much time reaction for topology changing. Exploration network by EDEAR offers a better adaptation to topology changing. When the volume of the traffic is high, the performances difference between the two protocols in terms of energy and delay increase.

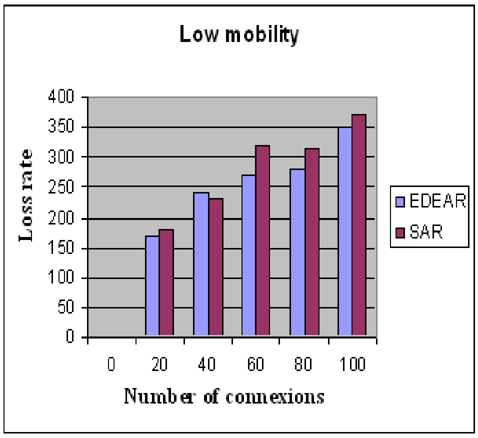

Figure 10. Loss rate results in low mobility scenario

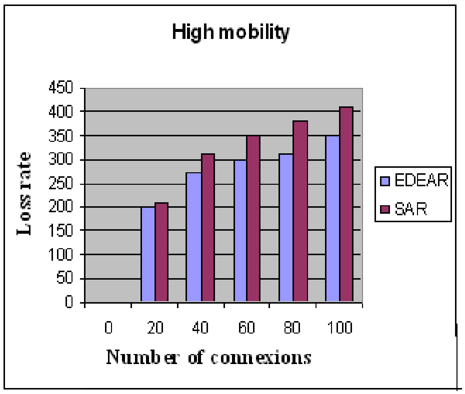

Figure 11. Loss rate results in high mobility scenario

VI. CONCLUSION

In this paper, we presented a new way to make adaptive routing with quality of service in order to improve the end-to-end delay and increase the lifetime of a delay tolerant network. Our protocol, called EDEAR, explores the network and chooses the best path routing in terms of energy and delay to route information based on reinforcement learning paradigm. For that, an explorer agent is used and learns from past experiences to choose the next path; data is always sent over the best path.

Our approach offers advantages compared with other classical approaches, such as SAR protocol. In particular, it reduces much better the energy consumed required for information sent, updates routing and network exploration more reactively. To reduce the overhead packets in the exploration process, our protocol used an enhanced version of OLSR algorithm based on energy to construct MPRE nodes.

Simulation results have shown the efficiency of the adaptive approaches compared to traditional approaches. In fact, the proposed algorithm takes into account the network state in a better way than the classical approaches do. Also, our protocol is well-suited for a dynamic mobile environment. In particular, an adaptive algorithm that simultaneously processes an observation and learns to perform better is a good way to consider time evolving networks. Such adaptive algorithms are very useful in tracking a phenomenon that evolves over time, especially in Delay Tolerant Networks.

Finally, our work in progress consists to use other metrics to find the best optimal paths (residual bandwidth, loss ratio, etc.) and to split traffic under all the considered paths using cognitive approaches.

REFERENCES

[1] A. Kumar, D. Manjunath, J. Kuri, D. Manjunath, “Ad Hoc Wireless Sensor Networks (WSNs)”. *Wireless Networking*, 2008, pp. 337-37.

[2] J-F. Buford, H.Yu, E-K.Lua, “Mobility and Heterogeneity”, *P2P Networking and Applications*, 2009, pp. 299-317.

[3] R. Olfati-Saber, A. Fax, and R.M. Murray, “Consensus and cooperation in networked multi-agent systems”, *Proceedings of the IEEE Proceedings*, vol. 95, no. 1, 2007, pp. 215-233.

[4] S. Farahani, “Battery Life Analysis ZigBee”, *Wireless Networks and Transceivers*, 2008, pp. 207-224.

[5] A. Mellouk, S. Hoceïni, M. Cheurfa, “Reinforcing probabilistic selective Quality of Service routes in dynamic irregular networks*”, Computer Communications*, vol. 31, no. 11, 2008, pp. 2706-2715.

[6] Tran-Xuan and al. “an efficient RSS-Based localization scheme with calibration in wireless sensor networks”, *IEICE Transactions on communications*, vol. 91, no. 12, 2008, pp. 4013-4016.

[7] F. Shearer “System-Level Approach to Energy Conservation”, *Power Management in Mobile Devices*, 2008, pp. 217-259.

[8] Y-J. Lu, T-L. Sheu, “An efficient routing scheme with optimal power control in wireless multi-hop sensor networks”, *Computer Communications*, vol. 30, no. 14-15, 2007, pp. 2735-2743.

[9] C. Ok, S. Lee, P. Mitra, S. Kumara, “Distributed energy balanced routing for wireless sensor networks”, *Computers & Industrial Engineering*, vol. 57, no. 1, 2009, pp. 125-135.

[10] L. Yuan, W. Cheng, X. Du, “An energy-efficient real-time routing protocol for sensor networks”, *Computer Communications*, vol. 30, no. 10, 2007, pp. 2274-2283.

[11] R.S. Sutton and A. G. Barto, "Reinforcement Learning" MIT Press, 1997.

[12] D-Q. Nguyen, P. Minet, “Anlysis of MPR selection in the OLSR protocol”. *Proc. Of PAEWN*, Niagara Falls, Ontario, Canada, 2007.

Nesrine Ouferhat-Allou obtained his PHD degree from University of Paris Est, France in 2009. Her research interests include dynamic Routing and Quality of Service in Wireless Sensor Networks. He is currently a Post Doctoral researcher in LiSSi Laboratory, UPEC University.

Abdelhamid MELLOUK (Senior Member IEEE) is Full Professor and Head of Networks and Telecommunication (N&T) Department of Paris-Est University, France, He graduated in Computer Network Engineering from the University of Paris Sud 11 Orsay (with distinction) and received his PhD in informatics Engineering from the same university and the “Habilitation” of Sciences Diploma in Networks Telecommunication from the University of Paris 12. His current research interests include quality of service; routing and switching optimization in dynamic traffic networks, artificial neuronal intelligence together with biologically inspired techniques.

Prof. Mellouk has authored or co-authored three books and several international publications in major journals and conferences and served or is serving also as Associate Editor, Editorial Board, Guest Editor, or Reviewer for major journals.