A Novel Cooperative Communication Protocol for QoS Provisioning in Wireless Sensor Networks

Xuedong Liang [*1,2], Min Chen [#3], Yang Xiao [†4], Ilangko Balasingham [*2,5], Victor C.M. Leung [#3]

[1] *Dept. of Informatics, University of Oslo, Oslo, Norway N-0316*
[2] *The Interventional Center, Rikshospitalet University Hospital, Oslo, Norway N-0027*
[3] *Dept. of Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada V6T 1Z4*
[4] *Dept. of Computer Science, University of Alabama, Tuscaloosa, AL 35487, USA*
[5] *Dept. of Electronics and Telecommunications, Norwegian University of Science and Technology, Trondheim, Norway N-7491*

[*] xuedongl,ilangkob@medisin.uio.no
[#] minchen,vleung@ece.ubc.ca
[†] yangxiao@ieee.org

***Abstract*—Cooperative communications have been demonstrated to be effective in combating the multiple fading effects in wireless networks, and improving the network performance in terms of adaptivity, reliability, data throughput and network life time. In this paper, we investigate the use of cooperative communications for quality of service (QoS) provisioning in resource-constrained wireless sensor networks, and propose *MRL-CC*, a Multi-agent Reinforcement Learning based multi-hop mesh Cooperative Communication mechanism for wireless sensor networks. In order to disseminate data reliably in *MRL-CC*, a multi-hop mesh cooperative structure is first constructed. Then a cooperative mechanism with cooperative partner assignments, and coding and transmission schemes is implemented using a multi-agent reinforcement learning algorithm. We compare the network performance of *MRL-CC* with *MMCC* [1], a Multi-hop Mesh structure based Cooperative Communication scheme, and investigate the impacts of network traffic load, interference and sensor node's mobility on the network performance. Simulation results show that *MRL-CC* performs well in terms of a number of QoS metrics, and fits well in large-scale networks and highly dynamic environments.**

I. Introduction

Wireless sensor networks (WSNs) have numerous potential applications, e.g., battlefield surveillance, medical care, wildlife monitoring and disaster response. In mission-critical applications, a set of QoS requirements (e.g., end-to-end delay, packet delivery ratio, and communication bandwidth) on network performance must be satisfied. However, due to the dynamic topology, time-varying wireless channel, and severe constraints on power supply, computation power and communication bandwidth of sensor nodes, quality of service (QoS) provisioning is challenging in WSNs.

Recently, a number of QoS support communication protocols have been proposed for WSNs [2], [3]. Most of these protocols are based on network traffic engineering, i.e., sensor nodes maintain network state information and use various algorithms to perform QoS routes' computation and maintenance. However, the network state information is inherently imprecise due to the dynamic wireless channel, node mobility and varying duty cycles. Thus, research on distributed, lightweight and highly adaptive communication protocols with QoS support is still needed.

In recent years, cooperative communications have been proposed to exploit the spatial diversity gains in wireless networks [4], [5]. Users in cooperative communication systems work cooperatively by relaying data packets for each other, and thus forming multiple transmission paths or virtual multiple-input-multiple-output (MIMO) system to the destination without the need of multiple antennas at each user. Cooperative mechanism is the key to the performance of cooperative communication protocols, however it is challenging to find the optimal cooperative policies in dynamic wireless networks, where reinforcement learning algorithms can be used to find the optimal control policy without the need of centralized control.

In this paper, we investigate the use of cooperative communications for QoS provisioning in resource-constrained WSNs, and propose *MRL-CC*, a Multi-agent Reinforcement Learning based Cooperative Communication protocol. In *MRL-CC*, a multi-hop mesh cooperative structure is constructed for reliable data disseminations, where the cooperative mechanism that defines the cooperative partner assignments, and coding and transmission schemes is implemented at each node using a multi-agent reinforcement learning algorithm. The cooperative nodes, regarded as multiple agents in the context of reinforcement learning framework, learn the optimal cooperative policy through experiences and rewards. Thus, by considering the interactions among each others, multiple agents can cooperatively learn the optimal policy by using locally observed network information and limited information exchange. Therefore, optimal network performance can be achieved without the need of maintaining precise network state information and centralized control.

The rest of the paper is organized as follows. Section II presents the related work. Section III describes the architecture overview, design issues and Q-learning algorithm implemen-

tations of *MRL-CC*. The performance analysis is presented in Section IV. Finally, Section V concludes the paper and gives future research discussions.

II. Related Work

Various cooperative diversity protocols have been proposed for wireless networks recently [4], [5]. Cooperation diversity gains, receiving and processing overheads, are investigated in [6]. A scalable, energy efficient and error-resilient routing protocol, REER [7], is proposed for dense WSNs. Based on geographical information, REER's design harnesses the advantages of high node density and relies on the collective efforts of multiple cooperative nodes to deliver data, without depending on any individual ones. In *MMCC* [1], a mesh structure is established for reliable data dissemination, random based and distance based values are used as the forwarding-node-election criteria. However, the random timer based criterion incurs extra delay, and the distance based value criterion is not always effective in dynamic WSNs.

Reinforcement learning provides a framework in which an agent can learn control policies based on experiences and rewards. In the standard reinforcement learning model, an agent is connected to its environment via perception and action, as shown in Fig. 1. On each step of interaction, the agent receives an input, i, some indication of the current state, s, of the environment; the agent then choose an action, a, to generate as an output. The action changes the state of the environment, and the value of the state transition is communicated to the agent through a scalar reinforcement learning signal, r. The agent's behavior, B, should choose actions that tend to increase the long-term sum of values of the reinforcement signal [8].

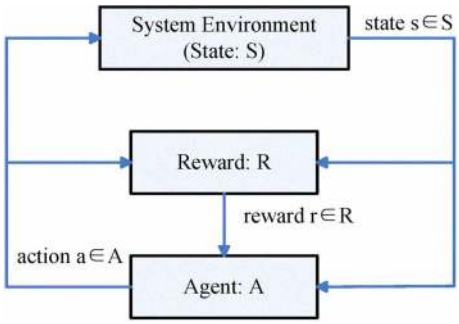

Fig. 1. A reinforcement learning model

The underlying concept of reinforcement learning is Markov Decision Process (*MDP*). A *MDP* models an agent acting in an environment with a tuple (S,A,P,R), where S is a set of states, A denotes a set of actions. $P(s^{'}|s,a)$ is the transition model that describes the probability of entering state $s' \in S$ after executing action $a \in A$ at state $s \in S$. $R(s,a,s^{'})$ is the reward obtained when the agent executes a at s and enters $s^{'}$. The goal of solving a MDP is to find an optimal policy, $\pi : S \mapsto A$, that maps states to actions such that the cumulative reward is maximized. Detailed information on reinforcement learning can be found in [8].

In WSNs, data packets are usually routed to the destination node through multi-hop communications. The QoS performance of the route relies on the overall routing procedures, i.e., each node, which involves in the routing procedure, contributes to the end-to-end QoS performances. It is worth noting that, nodes which are not directly involved in the routing procedure but are within the communication range of the forwarding nodes, may take actions (e.g., packet originating, forwarding) and have impacts on the route's QoS performance as well, due to the shared and contention nature of the wireless channel. WSNs can be characterized as multi-agent systems, where sensor nodes can be considered as agents, and the wireless channel and packet flows are regarded as the environment. In the multi-agent reinforcement learning algorithm, by exchanging local state values with immediate neighboring agents, an agent can consider both the rewards of neighboring and non-neighboring agents when it chooses actions, thus global cooperation can be achieved [9].

III. Cooperative Mechanism Design and Algorithm Implementations

In this section, we present the architecture and design issues of *MRL-CC*. First, an architecture overview of the network organization is presented. Then we describe the three phase operations of *MRL-CC*, namely mesh cooperative structure construction, Q-learning initialization and data dissemination phases. Finally, the design and implementations of the Q-learning algorithm are illustrated.

A. Architecture Overview

As shown in Fig. 2, *MRL-CC* employs a multi-hop mesh cooperative structure for reliable data dissemination in WSNs, i.e., data packets originated from the source are forwarded to the sink node by groups of cooperative nodes (denoted as CNs) relaying [7], [1]. In each group of CNs, a node will be elected as a forwarding node to forward the data packet to the adjacent group of CNs towards the sink node, and other nodes play as cooperative partners and will help in the packet forwarding in case the forwarding-node-election fails or the packet is corrupted in the transmissions.

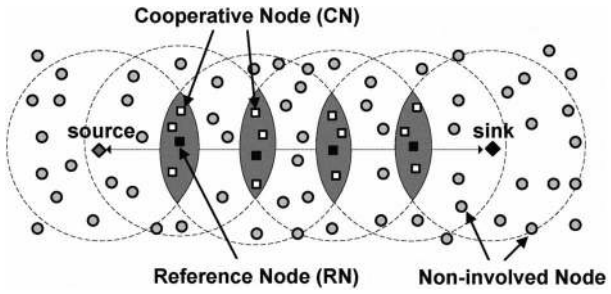

Fig. 2. Multi-hop mesh cooperative structure for data dissemination in WSNs

The forwarding-node-election in the CNs is based on a multi-agent learning algorithm, i.e., each node is implemented with a Q-learning algorithm, a model-free method which learns the value of a function $Q(s,a)$ to find an optimal decision

policy. Each node maintains the Q-values of itself and its cooperative partners, which reflect the qualities (e.g., delay, packet delivery ratio) of available routes to the sink. When a packet is received by a group of CNs, each node will compare its own Q-value with those of other nodes in the CNs; the node which determines it has the highest Q-value will be elected to forward the data packet to the adjacent CNs towards the sink.

Each time a packet is forwarded, all the nodes in the group of CNs will receive an immediate reward from the environment, which represents the quality of packet forwarding in terms of delay and packet loss rate. Nodes then use the rewards to update the Q-values, which will influence their future decisions of forwarding-node-election.

The algorithm will reach convergence after a certain amount of time, depending on the network size, node mobility and density. Nodes can simply use the learned policy to take appropriate actions, i.e., node with the highest Q-value will forward the packet to the adjacent groups of CNs towards the sink. To adapt to the dynamic nature of WSNs, *MRL-CC* explores the environments with a certain probability ε, namely ε-greedy method [10]. That is, with the probability of $1-\varepsilon$, the node with the the highest Q-value will forward the packet to the adjacent CNs; and with the probability of ε, a randomly chosen node will forward the packet to the adjacent CNs.

Thus, without using complicated prediction techniques, or explicitly frequent updating and maintaining of precise network state information, nodes can find the optimal cooperative policies through experiences and rewards in dynamic environments.

B. Multi-hop Cooperation Structure Construction Phase

To construct a multi-hop mesh cooperative structure, a set of nodes, termed as reference nodes (denoted as RNs) between the source and the sink (the source and the sink are also RNs) is first selected. The RNs are determined sequentially starting from the source node to the sink node, and the distance between two adjacent RNs is an application specific value, which is a trade-off between reliability and energy efficiency. Once the RNs are determined, a set of nodes around each RN will be selected as cooperative nodes (denoted as CNs), and thus, a multi-hop mesh cooperative structure is constructed in this phase. Data packets originated from the source will be forwarded to the sink by groups of CNs relaying.

A part of the mesh structure is shown in Fig. 3, where the set of n_{th} cooperative group is denoted by V_n, and its adjacent groups are denoted as V_{n-1} and V_{n+1}, which are one hop farther and closer towards the sink than V_n, respectively. Ideally, each node in V_n is connected with all the nodes in V_{n-1} and V_{n+1}; however, the links are unreliable and the qualities are varying over time and space due to the time-varying wireless channels and dynamic network topology.

The number of cooperative nodes in each CNs, and the number of CNs in the network, depend on the network size, node density and the trade-off between reliability and energy efficiency. Details of mesh cooperative structure construction and parameter selection can be found in [7], [1].

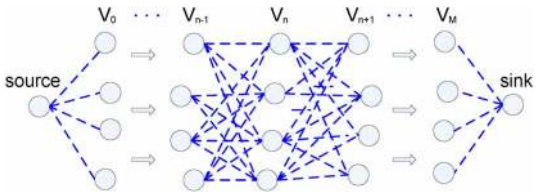

Fig. 3. Cooperation between adjacent groups of cooperative nodes

C. Q-learning Initialization Phase

In the initialization phase, each node is assigned with an initial Q-value. For node $i \in V_n$, its initial Q-value (denoted as $Q_{n_i}^{ini}$) is calculated based on the relative distance (compared with its cooperative partners in V_n) from node i to the nodes in V_{n+1}, as shown in (1).

$$Q_{n_i}^{ini} = d_{V_n,V_{n+1}}/d_{i,V_{n+1}} \tag{1}$$

where $d_{V_n,\mathrm{V_{n+1}}}$ is the average distance between V_n and V_{n+1}, which can be calculated as (2).

$$d_{V_n,V_{n+1}} = \frac{1}{N}\sum_{i=0,i\in V_n}^{N} d_{i,V_{n+1}} \tag{2}$$

where N is the number of cooperative nodes in the V_n (for simplicity, we assume N is identical for each group of CNs). $d_{i,\mathrm{V_{n+1}}}$ is the average distance between node i and V_{n+1}, which can be calculated as (3).

$$d_{i,V_{n+1}} = \frac{1}{N}\sum_{j=0,j\in V_{n+1}}^{N} d_{i,j} \tag{3}$$

In the initialization phase, node i exchanges its initial Q-value with the nodes in V_{n-1}, V_n and V_{n+1}, by broadcasting initialization messages.

D. Data Dissemination Phase

When a data packet is received by a number of nodes in V_n, each node will compare its Q-value with those of other cooperative nodes. The node which determines it has the highest Q-value will forward the data packet to V_{n+1}, and other nodes in V_n will deduce whether the packet forwarding is successful or not, by overhearing the packet transmission from V_{n+1} to V_{n+2}.

If the data packet is received by V_{n+1}, nodes in V_n's task of the current round of data forwarding are finished. Thus, all the nodes in V_n will receive positive rewards and update their Q-values, accordingly. [1]

If the packet forwarding fails, all the nodes in V_n will receive negative rewards (i.e., get punishment) and their

[1]In the Q-learning algorithm, not only the data forwarding node m will receive positive reward, but also the other cooperative nodes will get the premium. It is because the other cooperative nodes make the correct decision of electing m as the data forwarding node.

Q-values will be updated. Then, another forwarding-node-election will be conducted in V_n for packet re-transmission based on the updated Q-values. There are two reasons may cause the failure of packet forwarding:

- *forwarding election failure*: in this case, the node elected to forward the data packet is not eligible due to the out-of-date Q-value stored in other nodes in V_n,
- *packet transmission failure*: the packet is corrupted or collided during the transmission from V_n to V_{n+1}.

To address the problem of packet forwarding failure issues, each node maintains a timer T_{rf} for packet re-forwarding. That is, if nodes in V_n do not overhear that the packet delivery from V_{n+1} to V_{n+2} before the timer expires, nodes in V_n deduce the packet is not successfully forwarded from V_n to V_{n+1} and another forwarding procedure will be restarted by nodes in V_n using the updated Q-values.

When the Q-learning algorithm reaches convergence, nodes can simply use the learned cooperative policy to take appropriate actions, i.e., node with the highest Q-value will be elected to forward the packet to V_{n+1}, and nodes with lower Q-values are monitoring the packet forwarding and will help the packet delivering if the packet forwarding from V_n to V_{n+1} fails.

E. Q-learning Algorithm Implementations

In the context of reinforcement learning, for node $i \in V_n$, we define the states, actions and rewards as follows:

a) State: $S_i = \{k\}$, $k \in \{V_{n-1}, V_n, V_{n+1}\}$.

b) Action:

$$A = \begin{cases} a_f \\ a_m \end{cases} \tag{4}$$

Execution of a_f means that node i's forwarding of the packet from V_n to V_{n+1}, and a_m denotes that node i is monitoring the packet's forwarding.

c) Reward Function: The reward function is defined as (5).

$$rwd(i) = \begin{cases} \left(\frac{d_{V_n,sink} - d_{V_{n+1},sink}}{d_{V_n,sink}}\right) \Big/ \left(\frac{T_{V_{n+1}} - T_{V_n}}{T_{rmn}}\right) & \text{(5a)} \\ -\frac{T_{rf}}{T_{rmn}} & \text{(5b)} \end{cases} \tag{5}$$

Eq. (5a) is used to calculate the reward when the packet forwarding is successful, where $d_{V_n,sink}$ is the average distance between V_n and the $sink$, which can be calculated as (6).

$$d_{V_n,sink} = \frac{1}{N} \sum_{i=0, i \in V_n}^{N} d_{i,sink} \tag{6}$$

$T_{V_{n+1}}$ and T_{V_n} are the packet forwarding time at V_{n+1} and V_n, respectively, observed at node i using the local clock. T_{rmn} is the maximum amount of time that can be elapsed in the remaining path to the sink to meet the QoS requirements on end-to-end delay. T_{rmn} is updated after each packet forwarding, and the value is encapsulated in the data packet. The positive reward reflects the quality of the packet forwarding, i.e., relative progress towards the sink over a time unit.

Eq. (5b) is used to calculate the reward when the packet forwarding fails, The negative reward reflects the delay caused by the unsuccessful packet transmission from V_n to V_{n+1}.

The updating of Q-value iterates at each node in each forwarding procedure, and distributed value function - distributed reinforcement learning algorithm (DVF-DRL) [11] is used in the updating iteration.

For 1-hop forwarding, at iteration t, node $i \in V_n$ forwards a packet to V_{n+1}, and then $j \in V_{n+1}$ is elected to continue packet forwarding. Node i updates its Q-value as (7).

$$\begin{aligned} Q_i^{t+1}(s_i^t, a_i^t) = (1-\alpha) Q_i^t(s_i^t, a_i^t) + \alpha(r_i^{t+1}(s_i^{t+1}) + \\ \gamma w(i,j) \max_{a_j \in A_j} Q_j(s_j^t, a_j^t) + \\ \gamma \sum_{i' \in I, i' \neq j} w(i,i') \max_{a_{i'} \in A_{i'}} Q_{i'}(s_{i'}^t, a_{i'}^t)) \end{aligned} \tag{7}$$

where α is the learning rate, which models the updating rate of Q-values. r denotes the immediate reward of execution of the action. The weight of future rewards is defined by γ. I is the set of i's cooperative partners in V_n. $w(i,j)$ models how strongly node i weights of j's rewards in average. Eq. 7 shows that node i's Q-value is a weighted sum of the action's reward, i's Q-value, the maximum Q-value of j, and those of all i's cooperative partners.

IV. Performance Evaluation

To study the network performance of *MRL-CC*, we compare it with *MMCC*, a multi-hop mesh structure based cooperative communication scheme. A random forwarding-node-election scheme is also implemented and its performance is used as a comparison baseline.

A. Simulation Environments

We simulate a WSN where 200 sensor nodes are randomly distributed in a 400m $\times$ 200m rectangular area. We assume nodes are stationary in the simulations, except in the mobile scenario where 50 nodes are randomly chosen as mobile nodes and other are stationary. The source and the sink nodes are chosen randomly in each simulation run. Constant packet arrival rate with $5p/s$, and varying packet arrival rate (the probability of packet arrival rate of each sensor node follows a Poisson distribution with average $\lambda = 5p/s$), are used in the simulations.

Castalia [12] wireless sensor network simulator, which is based on the OMNeT++ discrete event simulation platform, is used as the simulation environment.

Table I lists the detailed simulation parameters.

B. Comparison with MMCC

The average end-to-end delay to the sink node in different wireless channel conditions are shown in Fig. 4 and Fig. 5, respectively.

The simulation results show that when the wireless channel is in a perfect condition, i.e., no error occurs in transmissions,

TABLE I
SIMULATION PARAMETERS

Parameters	Value
Number of sensor nodes	200
Simulation area	400 m × 200 m
Wireless channel model	Log shadowing wireless model
Path loss exponent	2.4
Collision model	Additive interference model
Mobility model	Random waypoint model
Physical and MAC layer	IEEE 802.15.4 standard
Packet length	40 bytes
Communication range	50 m
Data transmission rate	250 kbps
Simulation time	400 s
Number of simulation runs	10
N	4
ε	0.1
α	0.1
γ	0.5
$w(i,j)$	0.5, if j is the forwarding node $\frac{1}{2N}$, if j is the cooperative partners

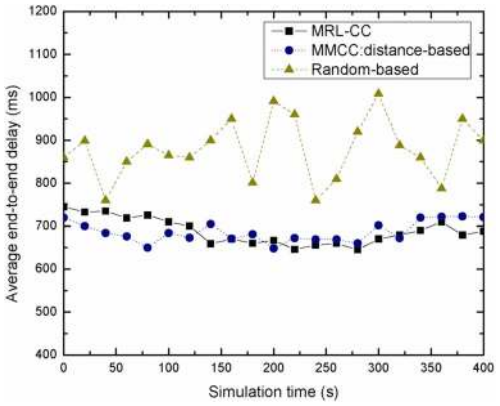

Fig. 4. Average end-to-end delay to the sink node (link failure ratio = 0)

MRL-CC and *MMCC* (distance based) have similar performances on the end-to-end delay. However, when the error-prone wireless channel is used in simulation, *MRL-CC* has better performance than *MMCC*. The reason is that in perfect wireless channel conditions, distance based protocols such as *MMCC*, are always effective, i.e., nodes which are closet to the sink are often the best forwarding candidates. However, in realistic channel conditions, it is not true that nodes closer to sink always have higher link qualities and should be elected as the forwarding nodes, and thus the use of distance based criterion in forwarding election is not always effective. For *MRL-CC*, by utilizing the knowledge learned from experiences and rewards, nodes with higher link qualities are more likely to be elected as the forwarding nodes in the CNs, and thus, the forwarding node assignments in *MRL-CC* is more adaptive than that in *MMCC*.

Fig. 6 and Fig. 7 illustrate the average packet delivery ratio from the source node to the sink node with constant and varying packet arrival rate, respectively.

We can observe that with constant packet arrival rate, *MRL-CC* and *MMCC* have similar performances on the packet delivery ratio. However, when the packet arrival rate varies, *MRL-CC* performs better than *MMCC*. The simulation results also verify that *MRL-CC* is more adaptive and flexible than *MMCC* in dynamic network conditions.

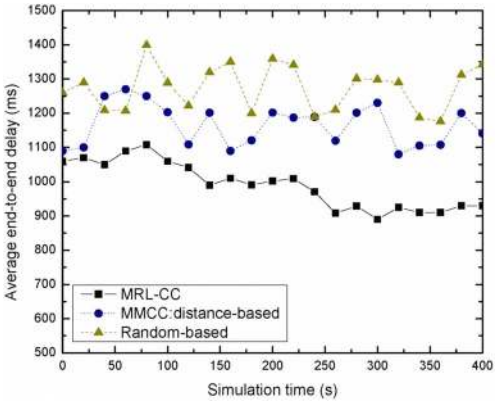

Fig. 5. Average end-to-end delay to the sink node (link failure ratio = 0.2)

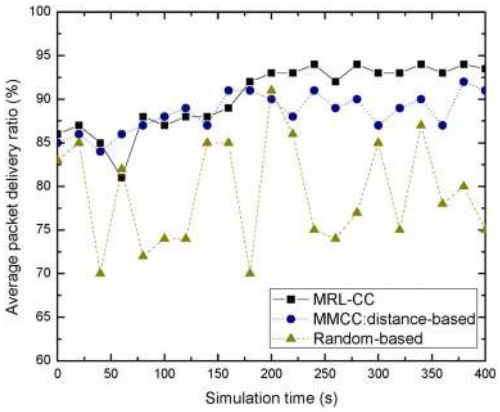

Fig. 6. Average packet delivery ratio to the sink node with constant packet arrival rate

The impact of network traffic load on average end-to-end delay, and the impact of node mobility on average packet delivery ratio are shown in Fig. 8 and Fig. 9, respectively.

The simulation results show that *MRL-CC* performs better than *MMCC*, especially when the network traffic becomes heavy and/or the network mobility level increases. It is because that *MMCC* selects data forwarding nodes either by a random value based criterion or a distance based criterion, and thus is lacking of flexibility to handle the network dynamics. In comparison, *MRL-CC* is much more intelligent in data forwarding-node-election since it learns the optimal cooperative policy through experiences and rewards. The flexible nature of computer machine learning allows it to adapt to the dynamic environment well, especially in networks with heavy traffic in highly dynamic scenarios.

We also notice that for all the QoS metrics in simulations, *MRL-CC* performs better after the simulation runs for a certain amount of time (i.e., around 50s). This is mainly because that there is a learning period in any learning based protocols, in

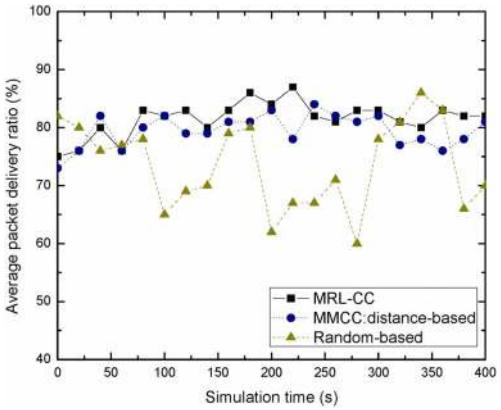

Fig. 7. Average packet delivery ratio to the sink node with varying packet arrival rate

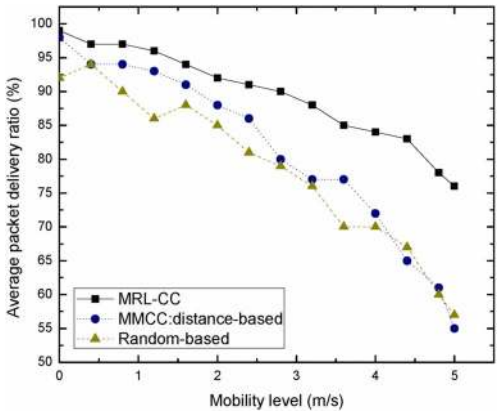

Fig. 9. The impact of node mobility on average packet delivery ratio

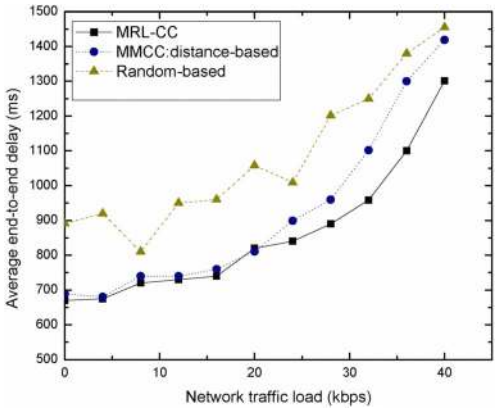

Fig. 8. The impact of network traffic load on average end-to-end delay

which agents (sensor nodes in this paper) explore all the available decisions (cooperative policy) and estimate the decision qualities, so that the network performance are improved over time. When the learning procedure is finished, nodes can take the optimal actions according to the state information.

V. Conclusions and Future Research

In this paper, we have investigated the use of cooperative communications for QoS provisioning in resource-constrained wireless sensor networks, and proposed *MRL-CC*, a multi-agent reinforcement learning based multi-hop mesh cooperative communication mechanism for wireless sensor networks. Simulation results show that *MRL-CC* performs well in terms of a number of QoS metrics and fits well in large scale networks and highly dynamic environments.

In future research, service differentiation and system fairness will be considered in the cooperative mechanism design. Moreover, we will examine the use of adaptive cooperative coding scheme (e.g., channel coding) and employ power allocation scheme to improve the network performance and prolong the network lifetime.

Acknowledgment

This research is in the context of the EU project IST-33826 CREDO: Modeling and analysis of evolutionary structures for distributed services (http://www.cwi.nl/projects/credo/). This work was supported in part by the Canadian Natural Sciences and Engineering Research Council under grant STPGP 322208-05. Professor Yang Xiao's work was partially supported by the US National Science Foundation (NSF) under the Grants No. CNS-0716211 and CCF-0829827.

References

[1] M. Chen, X. Liang, V. Leung, and I. Balasingham, "Multi-hop mesh cooperative structure based data dissemination for wireless sensor networks," in *Proc. The 11th International Conference on Advanced Communication Technology (ICACT'09)*, Phoenix Park, Korea, Feb. 2009, pp. 102–106.

[2] Q. Zhang and Y.-Q. Zhang, "Cross-layer design for QoS support in multihop wireless networks," *Proceedings of the IEEE*, vol. 96, no. 1, pp. 64–76, Jan. 2008.

[3] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, pp. 6–28, Dec. 2004.

[4] A. Nosratinia, T. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Communications Magazine*, vol. 42, no. 10, pp. 74–80, Oct. 2004.

[5] Y.-W. Hong, W.-J. Huang, F.-H. Chiu, and C.-C. J. Kuo, "Cooperative communications in resource-constrained wireless networks," *IEEE Signal Processing Magazine*, vol. 42, pp. 47–57, May 2007.

[6] A. Sadek, Y. Wei, and K. Liu, "When does cooperation have better performance in sensor networks?" in *Proc. IEEE The 3rd Sensor and Ad Hoc Communications and Networks (SECON'06)*, Reston, Virginia, USA, Sep. 2006, pp. 188–197.

[7] M. Chen, T. Kwon, S. Mao, Y. Yuan, and V. Leung, "Reliable and energy-efficient routing protocol in dense wireless sensor networks," *International Journal on Sensor Networks*, vol. 4, no. 12, pp. 104–117, Aug. 2008.

[8] L. P. Kaelbling, M. L. Littman, and A. P. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, May 1996.

[9] C.-K. Tham and J.-C. Renaud, "Multi-agent systems on sensor networks: A distributed reinforcement learning approach," in *Proc. The 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference (ISSNIP'05)*, Melbourne, Australia, Dec. 2005, pp. 423–429.

[10] R. S. Sutton and A. G. Barto, Eds., *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[11] J. Schneider, W.-K. Wong, A. Moore, and M. Riedmiller, "Distributed value functions," in *Proc. The 16th International Conference on Machine Learning*, Bled, Slovenia, Jun. 1999, pp. 371–378.

[12] H. N. Pham, D. Pediaditakis, and A. Boulis, "From simulation to real deployments in WSN and back," in *Proc. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'07)*, Helsinki, Finland, Jun. 2007, pp. 1–6.