QGeo: Q-Learning-Based Geographic *Ad Hoc* Routing Protocol for Unmanned Robotic Networks

Woo-Sung Jung, *Member, IEEE*, Jinhyuk Yim, *Student Member, IEEE*, and Young-Bae Ko, *Member, IEEE*

***Abstract*— This letter proposes a novel protocol that uses Q-learning-based geographic routing (QGeo) to improve the network performance of unmanned robotic networks. A rapid and reliable network is essential for the remote control and monitoring of mobile robotic devices. However, controlling the network overhead required for route selection and repair is still a notable challenge, owing to high mobility of the devices. To alleviate this problem, we propose a machine-learning-based geographic routing scheme to reduce network overhead in high-mobility scenarios. We evaluate the performance of QGeo in comparison with other methods using the NS-3 simulator. We find that QGeo has a higher packet delivery ratio and a lower network overhead than existing methods.**

***Index Terms*— Geographic routing, machine learning, Q-learning.**

I. Introduction

THE appearance of new mobile robotic devices, such as drones and remote controlled vehicles, offers opportunities for future services. At the same time, this revolution raises new challenges in unmanned robotic network design [1]. The most critical problem is the difficulty of routing in a network with easily disconnected features. Existing topology-based routing protocols tend to increase the routing cost due to their requirement of the entire route construction. Geographic routing is one of the promising solution for reducing routing control overhead.

Many geographic routing protocols intended to manage the routing overhead in mobile ad hoc networks have been proposed. For example, Location-aided routing (LAR) [2] controls overhead by restricting route request flooding. Although it reduces routing costs, a delay in the initial routing setup still exists because LAR begins data transmission after the complete route information is generated. To reduce route setup delays, greedy perimeter stateless routing (GPSR) [3] utilizes only the position information of neighbors. Despite the decrease in routing overhead, previous studies have shown that these methods limit the path selection due to the lack of understanding the dynamically changing environment.

Manuscript received September 21, 2016; revised November 17, 2016 and January 7, 2017; accepted January 8, 2017. Date of publication January 23, 2017; date of current version October 7, 2017. This work was supported by the "Location-based Tactical MANET Communication and Control S/W for Unmanned Ground Systems" project of the Agency for Defense Development (ADD), Korea. The associate editor coordinating the review of this letter and approving it for publication was X. Zhou. *(Corresponding author: Young-Bae Ko.)*

W.-S. Jung is with NeoReflection, Deajeon 34178, South Korea (e-mail: wsjung@neoreflection.net).

J. Yim and Y.-B. Ko are with the Department of Computer Engineering, Ajou University, Suwon 16499, South Korea (e-mail: jinhyuk@uns.ajou.ac.kr; youngko@ajou.ac.kr).

Digital Object Identifier 10.1109/LCOMM.2017.2656879

To overcome these drawbacks, machine learning concepts are adapted to routing algorithms. In this letter, we propose a QGeo, Q-learning-based geographic routing, for unmanned robotic networks. The QGeo can be executed in a distributed manner without global information. It can also provide the enhanced packet delivery ratio in high-mobility scenarios via a reinforcement machine learning algorithm. In the performance evaluation, we show that our QGeo generates less network traffic than the other comparable protocols.

II. Background and Related Work

A. Reinforcement Machine Learning

Reinforcement learning, a field of machine learning concerned with altering actions in a particular environment in order to maximize results, can be described as a Markov decision process (MDP) [4] consisting of (S, A, P, R) tuples, where S is set of states, A is a set of possible actions, P is state transition probability, and R is a reward function. Theoretically, we can determine the optimal machine state value using the MDP equation. However, this method is not suited to solve routing problems for mobile nodes because of the high computation cost.

Q-Learning [5], based on an approximated reinforcement machine learning algorithm, was introduced for running on resource-constrained mobile devices. The Q-Learning algorithm solves the routing problem in a distributed manner by utilizing the Q-value update equation as shown below:

$$Q(s_{t+1}, a_{t+1}) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha[f_R(s_t, a_t) + \gamma \max_{a'} Q(f_S(s_t, a_t), a')] \tag{1}$$

where $Q(s_t, a_t)$ is the Q-value of current state s_t when action a_t is selected at time t, $f_R(s_t, a_t)$ represents the reward function when state s_t selects action a_t, $\max_{a'} Q(f_S(s_t, a_t), a')$ is the maximum possible Q-value in the next state s_{t+1} when selects possible action a'. The learning rate α and discount factor γ are set between 0 and 1.

B. Q-Learning-Based Routing Protocols

There are several noteworthy studies of Q-learning-based routing. DACR [6] and QELAR [7] attempt to enhance quality of service (QoS) and energy efficiency using Q-learning based routing algorithms. EQR-RL [8] and RLGR [9] consider network lifetime enhancement. QoE Routing [10] utilizes reinforcement learning for enhancing performance, but not in high-mobility environments. MARS [11] and QGrid [12] propose machine-learning-based ad hoc routing schemes for vehicles. MARS predicts the movement of vehicles and then selects suitable routing paths between two roadside units.

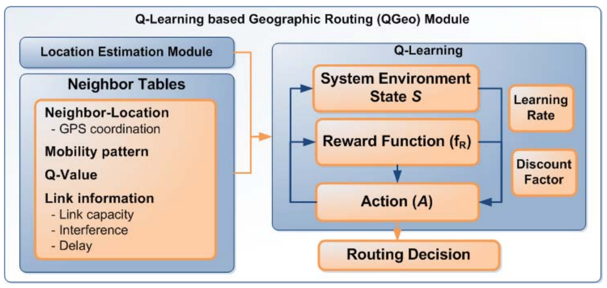


Fig. 1. Q-learning based Geographic Routing Module Architecture

While it provides low delay and high packet delivery ratio, it utilized a wired backbone network. QGrid considers the movement of vehicles between neighbor grids with low network overhead. It utilized a Q-learning based geographical grid routing algorithm. Although QGrid handles dynamic state change issues between each grid, the key limitation of this work is that QGrid does not consider link condition and location estimation error in the same grid. In addition, off-line Q-value calculation is limited when distribution of node is not stable.

III. Q-Learning Based Geographic Routing

In this section, we describe QGeo, a Q-Learning based Geographic routing protocol for autonomous robotic networks. The basic idea of QGeo is that mobile nodes make geographic routing decisions distributively, utilizing a reinforcement machine learning algorithm without knowledge of entire network. As shown in Fig. 1, the QGeo module consists of *location estimation*, the *neighbor table* and *Q-learning*.

Location estimation updates current mobile node location information that has been reported by GPS or other localization methods. Local information is exchanged by periodic HELLO messages that include location, current Q-value, link condition, and location error. The HELLO message interval can be adjusted based on the related velocity of neighbors.

The *neighbor table* manages the location of neighbors, their mobility pattern, link condition, location error and reported Q-value from received periodic HELLO messages. Neighbor location, Q-value, and link information are utilized in the Q-learning algorithm to generate a reward value. The mobility pattern is used to calculate link stability. Link capacity and interference information can be determined from MAC statistics.

In QGeo, *Q-learning* is the key component of routing decisions. We define each mobile node as representing a discrete node state s in the state set S. A node in the neighbor table is a transitional state of node s. Possible action a defines the state transition action from transmitter to neighbor node. In the exploration phase of learning, QGeo periodically exchanges state information. Each information is stored in the *neighbor table* of current state. In the exploitation phase, QGeo utilizes stored information to calculate reward function for deciding next state. Learning algorithm can be converged via the local maximum avoiding algorithm in the reward function.

NB: $\exists 1$ CdM $\forall$ singolo Packet.

The aim of previous work in geographic routing was to maximize transmission distance. However, forwarding node selection without concern for link condition or location error leads to frequent retransmission due to interference or obstacles. For supporting rapid and reliable transmission of unmanned robotic networks, we utilize a new concept of packet travel speed in the reward function, which is based on not only distance but also link status. The packet travel speed (f_{pts}) can be calculated as below:

DEFINIZIONE STANDARD DI VELOCITA', IL TRICK STA NEL TEMPO

$$f_{pts}(diff_{i,j}, P) = \frac{diff_{i,j}}{T_{i,j}(P)} \tag{2}$$

where $diff_{i,j}$ is the difference between the distances of nodes i and j to the destination. $T_{i,j}(P)$ is the packet travel time from node i to node j when the packet size is P. Packet travel time can be calculated by equation 3:

$$T_{i,j}(P) = (O + \frac{P}{r}) \times \frac{1}{1 - E_{link}} \times \frac{1}{1 - E_{loc}} \tag{3}$$

where O represents the channel access overhead and r is a data transmission rate. To calculate packet travel time, we take into account both link error(E_{link}) and location error(E_{loc}). E_{loc} is considered because mobile nodes share periodic GPS coordination with errors due to high mobility and measuring error [13]. The reward function can be defined as:

$$f_R(s_t, a_t) = \begin{cases} R_{max} & , when\ s_{t+1}\ is\ destination \\ -R_{max} & , when\ s_t\ is\ local\ maximum \\ f_{pts}/R_{pts} & , otherwise \end{cases} \tag{4}$$

i.e.

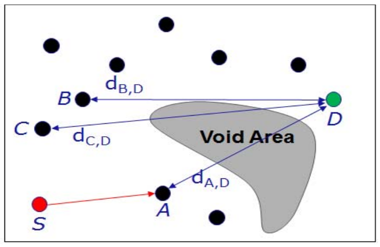
The first item in the equation 4 is the maximum reward value, R_{max}, if the next state is the destination. The second item is the negative reward value for avoiding the void area. In geographic grid routing, nodes can approach the void area due to utilizing local information without full network knowledge. This process leads to an increase in network cost. To solve this problem, we set a negative reward to prevent detours. The last item combines packet travel speed and the reference normalized factor R_{pts}. R_{pts} prevents excessive increment to the reward value because the Q-value is shared with neighbors via a HELLO message of a dedicated size. In this letter, we set R_{pts} as a $f_{pts}(d_{comm}, P)$ without an error condition when d_{comm} is communication range.

"SE IL MIGLIOR NODO SONO IO, C'È UN PROBLEMA"

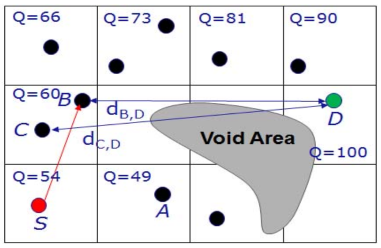
i.e. loops

In the Q-learning algorithm, discount factor γ represents the stability of future Q-value expectation and state transition. A high discount factor value indicates that the future state transition is stable, whereas a low discount factor value represents the expectation of a vulnerable node state transition. The basic discount factor is a constant value in the original Q-learning algorithm [5]. Our purpose is to select reliable link that has the high probability of connecting to a neighbor node. To reflect the dynamic mobile environment, QGeo manages dynamic discount factor γ according to distance and neighbor mobility pattern. The expected neighbor location is calculated via mobility patterns in the neighbor table. If the expected neighbor distance, $E[d_{i,j}]$, is shorter than d_{comm} after the next HELLO interval, we set γ to 0.6. Otherwise, we decrease the discount factor value. The reason for setting the mean value of γ to 0.6 is to compare relative routing schemes. γ is defined as below:

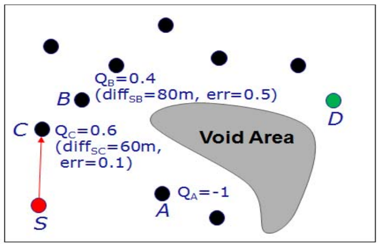
$$\gamma = \begin{cases} 0.6 & , when\ E[d_{i,j}] < d_{comm} \\ 0.4 & , otherwise \end{cases} \tag{5}$$



(a) GPSR



(b) QGrid



(c) QGeo

Fig. 2. Forwarding Node Selection Examples

The learning rate, α, relates to the convergence to a stable state. In this letter, we focus on the effects of the reward function and discount factor. Thus, we use fixed a learning rate value of 0.6.

Fig. 2 shows a comparison between GPSR [3], QGrid [12] and the proposed QGeo protocol. In each figure, S and D represent the source and destination nodes, respectively. Nodes A, B, and C are one-hop neighbors of node S. We can see in Fig. 2a that GPSR selects node A as a forwarding node because the remaining distance from $d_{A,D}$ is the shortest among candidate neighbor forwarding nodes. However, since GPSR only utilizes location information, node S cannot recognize the existence of the void area, which leads to additional packet forwarding to escape the void area. Fig. 2b shows the operation of the QGrid protocol, which operates by Q-learning-based grid routing. There are two steps for selecting the next forwarding node. In the first step, QGrid selects the optimal grid for avoiding the void area. Each Q-value is propagated from the grid where the destination node is located. After selecting the optimal grid, QGrid chooses the node in the selected grid that has the shortest remaining distance between the forwarding node and destination. In this example, node B is selected as the forwarding node. The QGrid approach is effective for avoiding the void area; nevertheless, it does not consider link status. If the link between nodes S and B has a higher error rate than other candidate links, node B is not the best solution and requires additional retransmission. To solve this problem, we utilize Q-learning with packet travel speed instead of the remaining distance. Fig. 2c illustrates our solution. In this example, node C has a higher packet transmission speed than node B owing to good link quality. As a result of Q-value calculation, node C is selected as a forwarding node.

IV. Performance Evaluation

We have implemented QGeo using NS-3 simulator and compared it with GPSR and QGrid. We summarize the detailed information about our experiment parameters in Table I. Initially, 25 nodes are deployed in a 250 m $\times$ 250 m region. Each node utilizes a 2.4 GHz single wireless interface for communication; we adopt 802.11g OFDM 6 Mbps for our MAC protocol and PHY rate. Furthermore, a Gaussian Markov mobility model is installed in each node to provide random mobility and velocity of each node is 0-15 m/s. A HELLO message is generated every 0.25 second and the lifetime for each neighbor entry is 0.4 second. For this application, 24 nodes try to report a 1 KB remote monitoring data packet to one control station node. Total simulation time is 100 seconds and we repeated our simulation 10 times with random seeds.

TABLE I
Experiment Parameters

Parameters	Settings
MAC & PHY	IEEE 802.11g OFDM 6 Mbps
Data Interval	0.25 ~ 1 Hz
Application Type	1KB UDP packet
Node Velocity	0 ~ 15 m/s
Mobility Model	Gaussian Markov Mobility Model
Propagation Loss Model	Log Distance Propagation Loss Model
HELLO interval	4 Hz
Neighbor Entry Lifetime	0.4 second
Link Error Rate	0 ~ 20 %
Location Error Rate	0 ~ 20 %

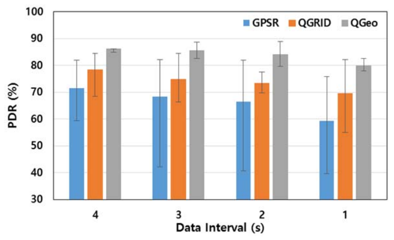


Fig. 3. Delivery Success Ratio.

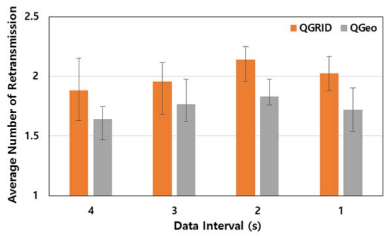


Fig. 4. Retransmission Comparison.

To evaluate QGeo, we selected PDR, number of retransmission, end-to-end delay, and network overhead as metrics. Fig. 3 represents the PDR comparison results of the three algorithms when the data interval changes. GPSR only utilizes location information to reduce the control overhead that causes the performance degradation due to the lack of path information. Using Q-Learning, however, QGrid shows higher performance than GPSR, though the absence of link and location error information still causes PDR degradation. In contrast, QGeo selects the node with the highest Q-value by regarding both the location and link condition of each node. The major reason for the PDR difference between QGrid and QGeo is also illustrated in Fig. 4. Fig. 4 shows the average retransmission count per flow. The lack of consideration of link and location error increases retransmission count, and thus leads to a decrease in PDR performance. In addition, PDR results demonstrate that QGeo provides a reliable network with less deviation than competing algorithms.

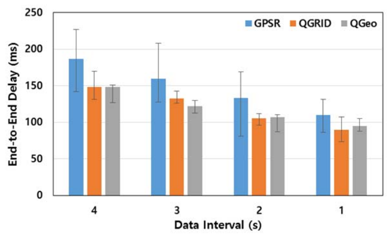


Fig. 5. Average End-to-End Delay

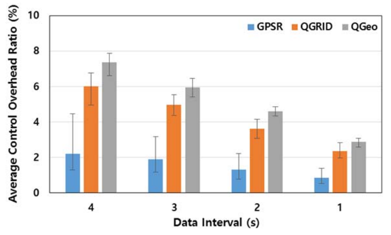


Fig. 6. Control Overhead Comparison.

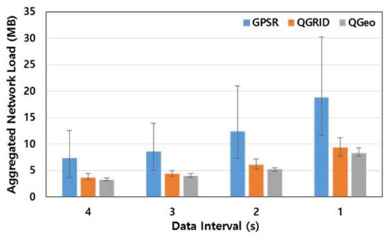


Fig. 7. Total Network Traffic w/o Overhead.

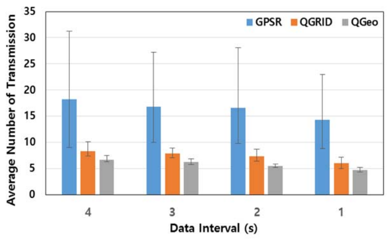


Fig. 8. Average Number of Forwarding per Flow

Fig. 5 illustrates the end-to-end delay comparison results. GPSR shows the longest delay because of the detour mechanism that bypasses the void region. As a result of the continual mobility of the experimental scenario, the void area is generated in simulation topology. To avoid this void area, GPSR frequently operates with perimeter forwarding rather than greedy forwarding, which increases delay. Unlike GPSR, QGrid and QGeo show similar delay patterns. QGrid is designed to support not real-time applications, but delay-tolerant network applications. Therefore, when the source node is unable to select a forwarding candidate node, it will hold the packet until the candidate node appears in its communication range. For this reason, we modify the QGrid algorithm to drop the packet when the application is not able to satisfy a certain delay requirement. In QGeo, each node calculates the reward value determined by packet travel speed and location error to every existing route and Q-value. Therefore, our algorithm shows the highest PDR and comparable delay result.

To examine the impact of protocol efficiency, we measured the control overhead and network traffic. We can see the average control overhead in Fig. 6. Because the QGrid and QGeo HELLO messages not only contain 16 bytes of location information, but also 4 bytes of Q-value information that increase the size of overhead. Furthermore, QGeo demands extra 4 bytes information, such as the link and location error for calculating reward values. Although QGeo shows a higher control overhead ratio than others, Fig. 7 indicates a different result, which represents the aggregated total network traffic generated by data communication without the size of the control overhead. These results denote that GPSR attempts to send the packet nearly two times more than QGrid and QGeo because of inaccurate and insufficient information. However, QGeo consumes little more network capacity for sharing network overhead; therefore, it reduces total network traffic load. Additionally, Fig. 8 illustrates the average number of transmissions to receive a packet. GPSR requires approximately 15 trials to transmit one packet successfully, but QGeo is able to transmit it within 5 trials. Lastly, The reason that GPSR has the high variation result is that forwarding node frequently meets the void area due to the high mobility.

V. CONCLUSION

$\xrightarrow{}$ AD UN CERTO PUNTO VI SIETE RESI CONTO CHE QGEO $\simeq$ QGRID

da cui queste conclusioni poco entusiaste

To fully realize the potential of unmanned robotic networks, routing problem caused by high mobility within these networks must be solved. This letter proposed Q-learning-based geographic routing that enhances network performance. Performance evaluation using the NS-3 simulator showed that QGeo achieves reliable transmission with low network overhead.

REFERENCES

[1] O. K. Sahingoz, “Networking models in flying ad-hoc networks (fanets): Concepts and challenges,” *J. Intell. Robot. Syst.*, vol. 74, nos. 1–2, pp. 513–527, 2014.

[2] Y.-B. Ko and N. H. Vaidya, “Location-aided routing (LAR) in mobile ad hoc networks,” *Wireless Netw.*, vol. 6, no. 4, pp. 307–321, 2000.

[3] B. Karp and H. T. Kung, “Gpsr: Greedy perimeter stateless routing for wireless networks,” in *Proc. ACM MobiCom*, 2000, pp. 243–254.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, vol. 1, Cambridge, U.K.: MIT Press, 1998.

[5] C. J. C. H. Watkins and P. Dayan, “Q-learning,” *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.

[6] M. A. Razzaque, M. H. U. Ahmed, C. S. Hong, and S. Lee, “Qos-aware distributed adaptive cooperative routing in wireless sensor networks,” *Ad Hoc Netw.*, vol. 19, pp. 28–42, Aug. 2014.

[7] T. Hu and Y. Fei, “QELAR: A machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks,” *IEEE Trans. Mobile Comput.*, vol. 9, no. 6, pp. 796–809, Jun. 2010.

[8] S. Z. Jafarzadeh and M. H. Y. Moghaddam, “Design of energy-aware QOS routing protocol in wireless sensor networks using reinforcement learning,” in *Proc. IEEE CCECE*, May 2014, pp. 1–5.

[9] S. Dong, P. Agrawal, and K. Sivalingam, “Reinforcement learning based geographic routing protocol for UWB wireless sensor network,” in *Proc. IEEE GLOBECOM*, Nov. 2007, pp. 652–656.

[10] N. Coutinho *et al.*, “Dynamic dual-reinforcement-learning routing strategies for quality of experience-aware wireless mesh networking,” *Comput. Netw.*, vol. 88, pp. 269–285, Sep. 2015.

[11] W. K. Lai, M.-T. Lin, and Y.-H. Yang, “A machine learning system for routing decision-making in urban vehicular ad hoc networks,” *Int. J. Distrib. Sensor Netw.*, vol. 2015, Jan. 2015, Art. no. 6.

[12] R. Li, F. Li, X. Li, and Y. Wang, “Qgrid: Q-learning based routing protocol for vehicular ad hoc networks,” in *Proc. IEEE IPCCC*, Dec. 2014, pp. 1–8.

[13] J. Yim, W.-S. Jung, and Y.-B. Ko, “Link quality based geographic routing resilient to location errors,” in *Proc. IEEE ICUFN*, Jul. 2015, pp. 95–96.