https://doi.org/10.7840/kics.2018.43.7.1227

무선 애드혹 네트워크 에서 강화학습을 이용한 동적 라우팅 경로 선택 알고리즘

양 흠◆, 유상조°

Dynamic Routing Path Selection Algorithm Using Reinforcement Learning in Wireless Ad-Hoc Networks

Qin Yang◆, Sang-Jo Yoo°

요 약

본 논문은 애드혹 네트워크에서 동적으로 변화하는 무선통신환경을 지원하기 위해 강화학습을 이용한 라우팅 프로토콜에 대해 제안한다. 제안된 방법의 목적은 전송률, 잔여 에너지, 그리고 종단간 지연과 관련한 라우팅 경로의 유틸리티 함수를 최대화하는 것이다. Q-learning 기반의 라우팅 경로 선택 알고리즘은 패킷 전송 성공률을 고려하여 제안되었다. Q-learning을 적용하는데 있어 각 노드를 상태로 정의하고 노드간의 링크를 행동으로 정의하였다. 패킷전송 및 에너지 소비의 신뢰성을 기반으로 패킷이 목적지에 도착하면 방문된 경로에 보상이 주어진다. 시뮬레이션 결과를 통해 제안된 방법이 다양한 통신상황에서 동적으로 환경에 적응하고 높은 유틸리티를 얻을 수 있음을 보인다.

Key Words : Q-learning, ad-hoc networks, node routing, quality of service, dynamic environment

ABSTRACT

This paper proposes a routing protocol scheme using reinforcement learning which supports dynamic wireless communication conditions in ad-hoc networks. The aim of this scheme is to maximize the utility value of routing path in terms of transmission rate, residual energy and end-to-end delay. Q-learning based routing path selecting algorithm is proposed with consideration of packet successful transmission ratio. Each node represents a state and the next packet transmission path link between nodes is called an action in Q-learning. A reward is given to visited path when a packet reaches to destination based on reliability of packet transmission and energy consumption. The simulation results show that our method can obtain dynamic environmental adaptivity and high utility in various communication situations.

Ⅰ. 서 론

Ad-hoc network is a collection of various devices such as laptops, vehicles, sensors and drones that are collectively known as nodes of ad-hoc network[1]. These nodes are organized in either homogeneous or

※ 본 연구는 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2017R1A2B4003512)

◆ First Author : Inha University Department of Information and Communication Engineering, daisyyang309@foxmail.com, 학생회원
° Corresponding Author : (ORCID:0000-0003-1533-0814) Inha University Department of Information and Communication Engineering, sjyoo@inha.ac.kr, 종신회원

논문번호 : 201806-0-099-SE, Received March 14, 2018; Revised July 4, 2018; Accepted July 5, 2018

heterogeneous manner. Every node of this network is communicated to each other wirelessly within a comparatively limited area. Machine learning allows an ad-hoc network to learn from previous experiences, make optimal routing actions and adapt to the dynamic environment. They are able to learn the optimal routing paths what will bring about energy saving and lifetime prolonging with reduction of complexity of a typical routing problem. Meanwhile, they meet quality of service (QoS) requirements in routing problem using relatively simple computational methods and classifiers[2]. In machine learning techniques, there are several ways to solve routing problem in ad-hoc networks. Guestrin et al.[3] introduced a general framework for sensors data modeling. This distributed framework relies on the network nodes for fitting a global function to match their own measurement. The nodes are used to execute a kernel linear regression in the form of weighted components. Barancho et al.[4] introduced "Sensor Intelligence Routing" (SIR) by using self-organizing map unsupervised learning to detect optimal routing paths. A slight modification on the Dijikstra's algorithm is proposed to form the network backbone and shortest paths from a base station to every node in the network.

One of the frequently-used machine learning techniques in path finding issue is reinforcement learning (RL), an efficient method for discovering policies in Markovian sequential decisions tasks. RL enables an agent (e.g., a ad-hoc node) to learn its dynamic environment conditions. The agent takes the best actions that maximize its long-term rewards by using its own experience. In general, RL schemes follow each agent based on the observation of current state selects an action from a predetermined set of actions and the action causes a state transition of the environment, and the agent receives a reward from it. The most well-known RL technique is Q-learning[5]. Q-learning related routing problem is an adaptive routing scheme for interconnection networks. In this paper, we propose a novel routing protocol by implementing Q-learning algorithm. We randomly allocate the nodes in a limited area and only nodes with communication range can be connected. Each node represents a state and a link for packet transmission between nodes called an action. Reward is given to visited path when a packet reaches to destination by Q-learning algorithm under the consideration of successfully transmitted packet ratio. One utility function is defined that consists of transmitted rate, residual energy and end-to-end delay to show the performance of the path. Q-learning will help this system to find the optimal path with highest utility.

The rest of this paper is organized as follows. Related work is presented in section Ⅱ. In section Ⅲ, we present the proposed Q-learning routing algorithm, along with its detailed conditions and utility function. The performance evaluation with simulation results is revealed in section Ⅳ. Conclusions are drawn in section Ⅴ.

Ⅱ. Related Works

Q-learning has been extensively and efficiently used in routing problem. There are several noteworthy studies of Q-learning-based routing which achieve good performance[6]. AdaR[7] is a self-configured routing protocol for wireless sensor networks based on reinforcement learning. In this protocol, packets are delivered from sensors to a centralized base station to learn the environment. When one node forwards a packet to another according to the current routing policy, both the action and associated reward are appended to the packet. When the base station receives the packet which contains samples along the routing path, the base station calculates new weights in the linear quality function and broadcasts them throughout the whole network to improve the forwarding policy. DACR[8] exploits a cooperative communication of enhancing of QoS guarantees in delay and reliability domains and QELAR[9] assumes genetic MAC protocols and aims at prolonging the lifetime of networks by making residual energy of sensor nodes more evenly distributed using Q-learning based routing algorithms. Other energy-aware QoS routing protocols called EQR-RL[10] and RLGR[11] consider network lifetime enhancement. QoE routing[12] utilizes RL for enhancing performance, but not in high-mobility environments. MARS[13] and QGrid[14] 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. This algorithm works well under dynamically changing conditions and then brings much possibility. Many other extensions of this algorithm have been proposed, e.g., Simulated Annealing Based Hierarchical Q-routing[15], Policy-Gradient Q-routing[16], Gradient Ascent Q-routing[17], K-Shortest Paths Q-routing[18], Enhanced Confidence Based Q-routing[19].

In this paper, probabilistic Q-learning algorithm is used to satisfy dynamic wireless environment conditions, for example end-to-end delay, transmitted rate and residual energy. The estimation of the environment condition combines previous status itself and prospective status. This makes it unlike many other conventional Q-learning approaches in ad-hoc network routing protocol.

Ⅲ. Q-learning for Ad-hoc Network Routing

3.1 Q-learning model

Q-learning as one of the simple and powerful RL algorithms is biologically inspired and acquires its knowledge by actively exploring its environment. As Fig. 1. shows, at each step t the agent, could be an ad-hoc node, executes action a_t, receives observation O_t and receives scalar reward R_t. Meanwhile, the environment receives action a_t and emits observation O_t and emits scalar reward R_t.

A reward is a scalar feedback signal, indicates how well agent is doing at step t. Then, the previous state s_t transfers to next state s_{t+1}. The agent's job is to maximize reward from the environment based on all previous experience. The fundamental idea of Q-learning algorithm as a routing protocol is to use Q-table as the routing table. Each state and action present routing information of node location. Fig. 2. gives an example of the proposed routing topology. Q presents its Q-value and a indicates its action. Every node has at least one neighbor stochastically and holds its own routing information. Data packet is transferred by choosing consecutive actions to the destination, and reward will be sent back through all visited nodes to the source and the routing information is updating during the process by proposed Q-learning algorithm. All the routing information is based on Q-learning table.

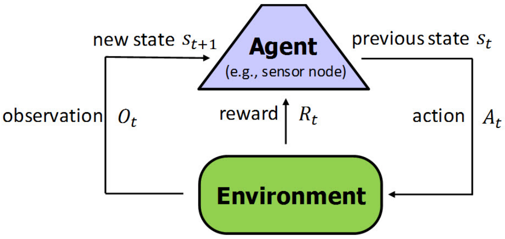


그림 1. 일반적인 강화학습 구조
Fig. 1. Basic reinforcement learning architecture

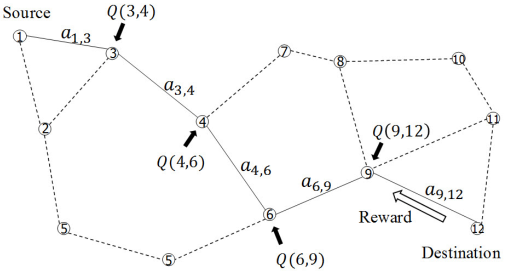


그림 2. Q-learning 라우팅 토폴로지의 예
Fig. 2. An example of Q-learning routing topology

Q-value is updated for representing optimal routing information of every iteration time. The proposed update rule is shown as following.

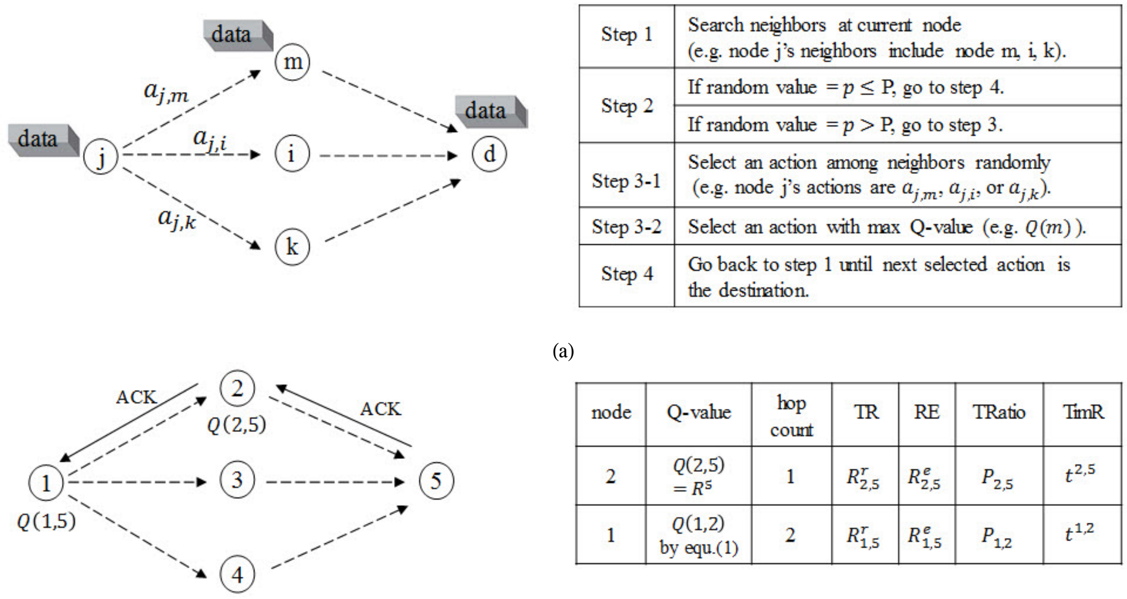
$$Q(j,a_m) = (1-\alpha)Q(j,a_m)' + \alpha\{\beta\times[R^r_{jd}+R^e_{jd}]+\gamma(P_{jm})^{t_{jm}}\times\max_{a*}Q(m,a*)\} \quad (1)$$

where node j denotes current state (i.e., the current node) and a_m shows the action to next state which indicates the next node m for relaying a packet; $Q(j,a_m)'$ is the previous Q-value on node j with a_m. R^r_{jd} represents average transmitted rate reward (TR) of node j to destination node d; R^e_{jd} represents average residual energy (RE) of current node j to destination d; P_{jm} is the packet success-transmitted ratio (TRatio) between j and m; t_{jm} is update time reliability (TimR) for link between node j to next node m. α, β is the scale factors and γ is the learning rate. If next node m is the destination,

$$\max_{a^*} Q(m,a^*) = R^d \tag{2}$$

where R^d is the terminal reward by destination which is a predefined value that should be greater larger than average transmitted rate reward and average residual energy. It is delivered to the previous visited node by using ACK.

In Fig. 3. (a), the data packet is needed to be delivered from source node (e.g., node j) to the destination node (e.g., node d). First, we need know how many possible neighbor nodes to transfer within the communication range. Then, we generate a random value p whose range is from 0 to 1 to compare with P, a predetermined probability of Q-learning choosing actions. If the random value $p \leq P$, we select the action (node) with the maximum Q-value node of all neighbor nodes. Oppositely, if the random value $p > P$, we select an action among all neighbors randomly. At the next node, we repeat these steps to reach the destination. The random value p is from the concept of Roulette Wheel Selection. Purpose of the random value p is giving more diversity to the node selection to avoid local optimal. Basically, the action selection is decided by predetermined probability P which is not a constant number but a log function with decreased trend. This can generate high possibility to selection random actions at first and gradually get high possibility to selection maximum Q value action till end. In Fig. 3. (b), once a packet arrives at the destination, an ACK within the terminal reward will be sent by destination to the previous visited node and update routing information. Taking node m as an example, node m is the last node before destination that the data packet goes through. Its Q-value equals the terminal reward and hop count is plus one. At the same time, we record its RT, RE, TRatio and TimR at data transmitting. After that, the previous node will transfer ACK to its former visited node and the former visited node updates its routing information till ACK is transferred to the source node. Then the routing path can be confirmed. In every iteration Q-table updates for providing next routing time.



Step 1	Search neighbors at current node (e.g. node j's neighbors include node m, i, k).
Step 2	If random value $= p \leq$ P, go to step 4.
	If random value $= p >$ P, go to step 3.
Step 3-1	Select an action among neighbors randomly (e.g. node j's actions are $a_{j,m}$, $a_{j,i}$, or $a_{j,k}$).
Step 3-2	Select an action with max Q-value (e.g. $Q(m)$).
Step 4	Go back to step 1 until next selected action is the destination.

(a)

node	Q-value	hop count	TR	RE	TRatio	TimR
2	$Q(2,5)$ $= R^5$	1	$R^r_{2,5}$	$R^e_{2,5}$	$P_{2,5}$	$t^{2,5}$
1	$Q(1,2)$ by equ.(1)	2	$R^r_{1,5}$	$R^e_{1,5}$	$P_{1,2}$	$t^{1,2}$

(b)

그림 3. 데이터 전송의 라우팅 과정과 보상에 대한 피드백 (a) 순방향 전파 (b) 역방향 전파
Fig. 3. The routing process of data transmission and reward feedback (a) forward propagation (b) backward propagation

3.2 Dynamic wireless conditions

As in (1) for Q-table updating, we take further consideration on realistic communication environment. Q-table update is not only based on terminal reward but also on the intermediate node's link reliability on the path, in which $\beta\times[R_{jd}^{r}+R_{jd}^{e}]\times\gamma(P_{jm})^{t_{jm}}$ term is used for Q-table update. Whenever network topology, node's energy level, or wireless channel condition is changed, R_{jd}^{r}, R_{jd}^{e}, or P_{jm} value at each node on the path is also changed.

Transmitted rate (R_{jd}^{r}) presents transmission efficiency and shows vital to the network performance. In the equation (1), we take transmitted rate to enhance the possible communication speed from the source to destination. Transmitted rate can be calculated as follow:

$$R_{jd}^{r}=r_{j,m}^{d}=(1-\delta_1)r_{j,m}+\delta_1 r_{m,d*}^{d} \tag{3}$$

where $r_{j,m}^{d}$ denotes the average transmitted rate from the current node j to the destination d through the next node m, $r_{m,d*}^{d}$ is the best path average transmitted rate from node m to destination d through $d*$ node. $d*$ is the next state (i.e., the next node of node m to the destination) for the best path in terms of

$$d*=\arg max_{a*}Q(m,a*) \tag{4}$$

This average transmitted rate includes the current state transmitted rate and the next state predicted transmitted rate to present a more comprehensive value and their influence adjusts under the scaling factor δ_1.

Residual energy (R_{jd}^{e}) in a routing problem, is a crucial factor to present energy efficiency of the network. Higher residual energy can bear more data transmission and prolong the network lifetime. Like transmitted rate, they both affect on updating current Q-value. Residual energy can be calculated by following equation

$$R_{jd}^{e}=e_{j,m}^{d}=(1-\delta_2)e_j+\delta_2 e_{m,d*}^{d} \tag{5}$$

where $e_{j,m}^{d}$ presents the average residual energy from j to destination d through m, $e_{m,d*}^{d}$ is the best path average residual energy from node m to destination d through $d*$ node.

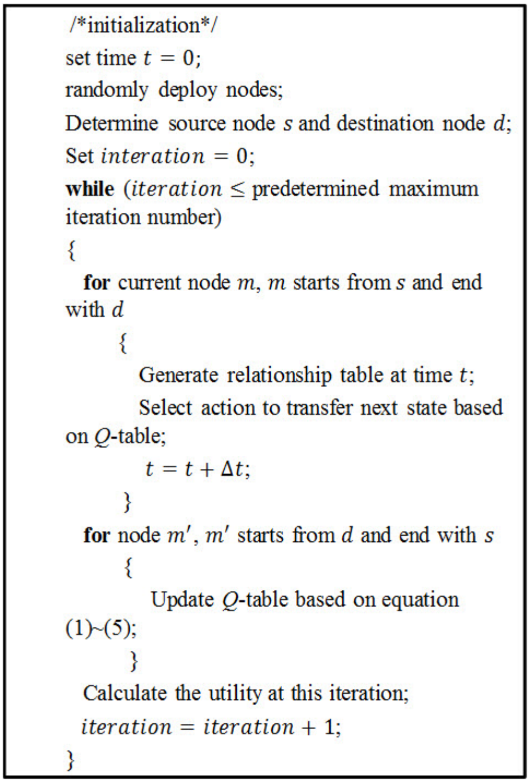
Successful packet transmission ratio (P_{jm}) indicates how well of the packet transmission every time, which guarantees the feasibility of the desired packet loss rate. Simultaneously, update time reliability (t_{jm}) is a scalar factor that represents the freshness of the successful packet transmission ratio. If the observed packet transmission statistics is not a recently observed value then the reliability of the value is low.

For performance evaluation in Section IV of the proposed Q-learning based routing algorithm, we define an utility function, which includes end-to-end delay, transmitted rate and residual energy for the selected routing path as in (6).

$$U=\omega_1\times(\frac{D_{\max}}{D})+\omega_2\times(\frac{R}{R_{\max}})+\omega_3\times(\frac{E}{E_{\max}}) \tag{6}$$

where D, R and E are the average end-to-end delay, average transmitted rate and average residual energy of the selected routing paths, respectively. $D_{\max}$, $R_{\max}$ and $E_{\max}$ are the predetermined maximum values. $\omega_1,\omega_2,\omega_3$ are weight factors and $\omega_1+\omega_2+\omega_3=1$.

Fig. 4. shows the pseudocode of our proposed routing algorithm. First we initialize time and node topology. Forward propagation is proceeded to generate the relationship table and select next node in terms of Q-table till reach to the destination. Then Q-table will be updated by equation (1)~(5) through backward propagation and utility can be estimated by equation (6). Repeat this process up to the maximum iteration time. The optimal path will be selected with the highest utility value.



```
/*initialization*/
set time $t = 0$;
randomly deploy nodes;
Determine source node $s$ and destination node $d$;
Set $iteration = 0$;
while ($iteration \leq$ predetermined maximum iteration number)
{
  for current node $m$, $m$ starts from $s$ and end with $d$
    {
      Generate relationship table at time $t$;
      Select action to transfer next state based on $Q$-table;
      $t = t + \Delta t$;
    }
  for node $m'$, $m'$ starts from $d$ and end with $s$
    {
      Update $Q$-table based on equation (1)~(5);
    }
  Calculate the utility at this iteration;
  $iteration = iteration + 1$;
}
```

그림 4. 제안된 Q-learning 알고리즘의 의사코드
Fig. 4. Pseudocode of the proposed Q-learning algorithm

Ⅳ. Simulation Results

Table 1 shows the simulation parameters used in this simulation study. We set the maximum range between nodes is 500 meters and the communication range is 200 meters. The transmitted rate value and residual energy value are set a range from 1 to 5. From the experience, the value of terminal reward should be much larger than transmitted rate and residual energy values. The successful packet transmission ratio between nodes is set a range from 0.9 to 0.95 and update time reliability for link is set a range from 0.9 to 1. The learning rate of Q-learning is 0.3 and scaling factor β, δ_1, δ_2 are 0.1, 0.5, 0.5 respectively.

First, we study the influence of network performance by increasing topology complexity. We generated different number of nodes from 20 to 60. Fig. 5 shows the program running time for increasing the number of nodes. Once increasing the node quantity, the running time increases as well but within very little difference. When the number of network nodes increases over 40, the growth rate is decreased. The reason is that after 40 nodes the average hop count to the destination is stable. With the difference of around 0.015 second, it almost can be seen number of network nodes does not highly influence our algorithm's running time. Dynamic topology change can guarantee efficiency of proposed method. Given the different values of terminal reward, the maximum Q-value of optimal path is shown in Fig. 6. With the growth of value of terminal reward, the variation tendency of maximum Q-value can be approximately considered

표 1. 시뮬레이션 파라미터
Table. 1. Simulation parameters

Simulation Parameters	Values
Q-learning iteration number	100
Number of action and state	20
Probability of selecting random state	0.3
Minimum selection probability	0.05
Decreasing rate	0.9994
Learning rate	0.3
Packet successfully-transmitted ratio	0.9~0.95
Update time reliability	0.9~1.0
Terminal reward	100~500
Transmitted rate	1~5
Residual energy	1~5
Scaling factor γ	0.7
Scaling factor β	0.1
Scaling factor δ_1, δ_2	0.5

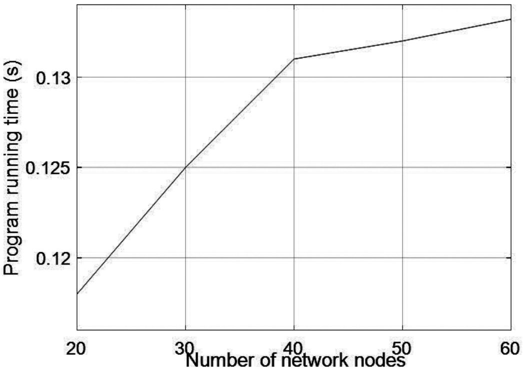


그림 5. 제안된 Q-learning 알고리즘의 의사코드 네트워크 노드 수에 따른 프로그램 실행시간
Fig. 5. Program running time among different number of network nodes.

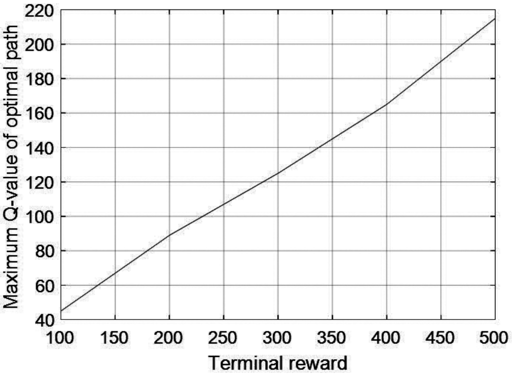


그림 6. 목적지 보상에 따른 최적 경로의 최대 Q값
Fig. 6. The maximum Q-value of optimal path with different values of terminal reward

as a linear function. In general, the terminal reward should be greater than transmitted rate and residual energy. However if it is too large, the impact of the transmitted rate and residual energy would be vanished. The larger terminal value results in finding the smaller hop count path in Q-learning. Over all the following simulations, the terminal reward and node quantity are chosen 200 and 40, respectively.

Fig. 7 shows the trend of the sum of Q-values of selected path within 100 iterations. As we can see, it takes on a growth trend and converges around 23th iteration. In order to keep randomness of action selection, the selected path may not always keep the maximum Q-value even after Q-table converges. Fig. 8 shows the maximum utility of the selected best routing path under the Q-table update. With the growth of the average Q-value, the trend of utility increases as well. It is worth noting that the Q-value based routing path selection results in good performance in terms of utility function.

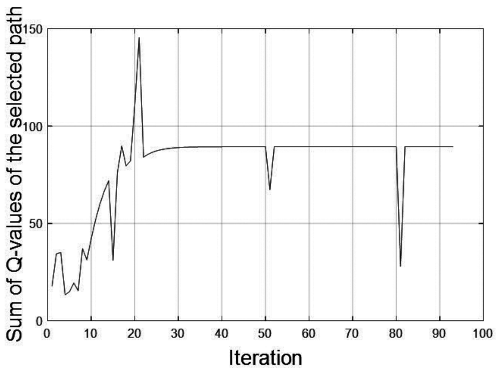


그림 7. 목적지 보상에 따른 최적 경로의 최대 Q값 선택된 경로에 대한 Q값의 합
Fig. 7. Sum of Q-values of the selected path

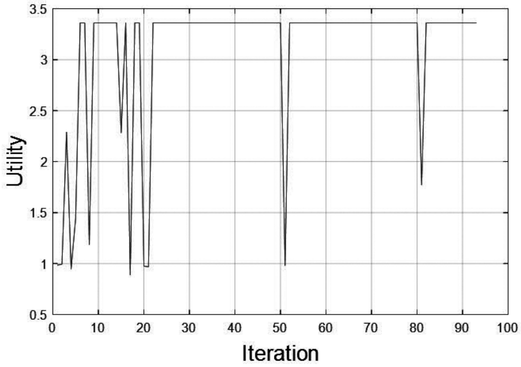


그림 8. 유틸리티 함수 기반의 네트워크 성능
Fig. 8. Network performance based on utility function

Fig. 9 gives a comparison using AODV (Ad hoc On-Demand Distance Vector)[20] and Q-AODV to our proposed method in terms of the number of control packet transmissions. As a typical routing protocol in wireless ad-hoc networks, AODV is widely used for unicast routing and multicast routing. During the routing path searching, route request message from the source node to the destination nodes is broadcasted to all its neighbor nodes so that any neighbor node should rebroadcast it. It increases control packet transmissions as increasing the number of nodes in as-hoc network also can cause broadcast storm problem. The

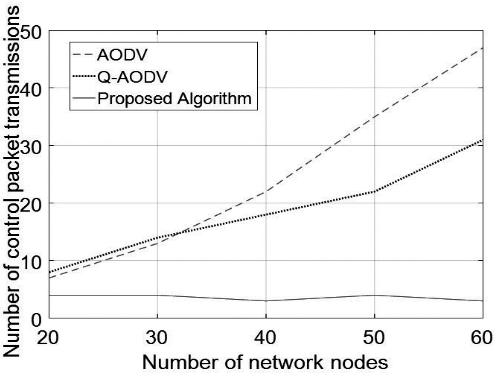


그림 9. 유틸리티 함수 기반의 네트워크 성능 제어패킷 전송의 수에 대한 AODV, Q-AODV와 제안된 방법의 비교
Fig. 9. A comparison among AODV, Q-AODV and our proposed method in terms of the number of control packet transmissions

control packets of AODV include RREQ (Route Request), RREP (Route Reply) and ACK (for the received data packet) while our proposed method only needs ACK control packet from the destination node with reward value. The second compared method is called Q-AODV protocol is a reactive protocol based on AODV protocol. As stated above, in AODV protocol, the processing time and transmission of control packets stored in nodes queues increases as the number of packets increases. Q-AODV captured the Q value as well can save the end-to-end delay then decrease the control packet transmissions in total. As Fig. 9 indicates, under the same simulation environment and communication conditions, the number of control packet transmissions of AODV increases dramatically with the growth of number of network nodes. While the number of control packet transmissions of the proposed method stays stable number of transmissions. With less control packet transmissions, our Q-learning based routing algorithm can reduce energy consumption and achieve higher efficiency.

Ⅴ. Conclusions

In this paper, we propose a probabilistic Q-learning algorithm for ad-hoc network routing under the consideration of various network requirements. Without any control packet transmissions, at each intermediate node based on Q-table it selects the next node to the destination that can maximize expected routing utility. Once a data packet is delivered to the destination node, the terminal reward is back-propagated through the intermediate nodes on the path. At each node Q-table update is performed not only using the terminal rewards, but also using the defined path and link reliability parameters including average transmission rate, residual energy and successful packet transmission ratio. The simulation results show that our method can obtain dynamic environmental adaptivity and high utility in various communication situations.

References

[1] G. Gankhuyag, A. P. Shrestha, and S.-J. Yoo, "Robust and reliable predictive routing strategy for flying ad-hoc networks," *IEEE Access*, vol. 5, pp. 643-654, Jan. 2017.

[2] M. A. Alsheikh, S. Lin, and D. Niyato, "Machine learning in wireless sensor networks: algorithm, strategies, and applications," *IEEE Commun. Surveys & Tuts.*, vol. 16, no. 4, pp. 1996-2018, Apr. 2014.

[3] C. Guestrin, P. Bodik, R. Thibaux, M. Pashkin, and S. Madden, "Distributed regression: an efficient framework for modeling sensor network data," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw.*, pp. 1-10, Apr. 2004.

[4] J. Barbancho, C. Leon, F. Molina, and A. Barbancho, "A new QoS routing algorithm based on self-organizing maps for wireless sensor networks," *Telecommum. Syst.*, vol. 36, no. 1-3, pp. 73-83, Nov. 2007.

[5] C. Watkins and P. Dayan, "Q-learning," *Machine Learning,* vol. 8, no. 3/4, pp. 279-292, May 1992.

[6] J. Dowling, E. Curran, R. Cunningham, and V. Cahill, "Using feedback in collaborative reinforcement learning to adaptively optimize MANET routing," *IEEE Trans. Syst., Man, and Sybernetics,* vol. 35, no. 3, pp. 360-372, May 2005.

[7] P. Wang and T. Wang, "Adaptive routing for sensor networks using reinforcement learning," in *Proc. IEEE Comput. and Inf. Technol.,* Sept. 2006.

[8] 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. 28-24, Aug. 2014.

[9] 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.

[10] 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,* pp. 1-5, May 2014.

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

[12] 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, Sept. 2015.

[13] 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, no. 6, Jan. 2015.

[14] R. Li, F. Li, and Y. Wang, "Qgrid: Q-learning based routing protocol for vehicular ad hoc networks," in *Proc. IEEE IPCCC,* pp. 1-8, Jul. 2014.

[15] A. M. Lopez and D. R. Heisterkamp, "Simulated annealing based hierarchical Q-Routing: a dynamic routing protocol," in *Proc. IEEE 8th Int. Conf. Inf. Technol.: New Generations,* vol. 1, pp. 791-796, Apr. 2011.

[16] N. Tao, J. Baxter, and L. Weaver, "A multi-agent policy-gradient approach to network routing," in *Proc. 18th Int. Conf. Machine Learning,* pp. 553-560, Jul. 2001.

[17] L. Peshikin and V. Savova, "Reinforcement learning for adaptive routing," in *Proc. Int. Joint Conf. Neural Netw.,* vol. 2, pp. 1825-1830, May 2002.

[18] S. Hoceini, A. Mellouk, and Y. Amirat, "K-shortest paths Q-routing: a new QoS routing algorithm in telecommunication networks," in *Proc. ICN 2005,* pp. 164-172, 2005.

[19] R. Desai and B. P. Patil, "Enhancement confidence based Q-routing for an ad hoc network," *Am. J. Educational Sci.,* vol. 1, No. 3, pp. 60-68, May 2015.

[20] *Ad hoc On-Demand Distance Vector (AODV) Routing, Request for Comments 3561*, IETF, Jul. 2003.

[21] S. J. Jang and S. J. Yoo, "Q-learning-based dynamic joint control of interference and transmission opportunities for cognitive radio," *EURASIP J. Wireless Commun. and Netw.,* vol. 2018, pp. 24, May 2018.

양 흠 (Qin Yang)

2016년 6월 : 중경우전대학교 통신정보공학과(공학사)

2016년 9월~현재 : 인하대학교 정보통신공학과

<관심분야> 무선센서네트워크, FANET, 머신러닝

유 상 조 (Sang-Jo Yoo)

1988년 2월 : 한양대학교 전자통신학과(공학사)

1990년 2월 : 한국과학기술원 전기및전자공학과(공학석사)

2000년 8월 : 한국과학기술원 전자전산학과(공학박사)

1990년 3월~2001년 2월 : KT 연구 개발 본부

1990년 3월~2000년 11월 : NIST(미국 표준기술연구원) 초빙연구원

2001년 3월~현재 : 인하대학교 정보통신공학과 교수

<관심분야> 무선 네트워킹 프로토콜, Cross-layer 프로토콜 설계, Cognitive Radio Network, 무선 센서네트워크, 미래인터넷