논문 20-45-12-13 The Journal of Korean Institute of Communications and Information Sciences '20-12 Vol.45 No.12
https://doi.org/10.7840/kics.2020.45.12.2099

주기적 Q-table 업데이트를 활용한 무선 라우팅 프로토콜

김 태 정◆, 정 진 우°

Wireless Routing Protocol with Periodic Q-Table Update

Taejung Kim◆, Jinoo Joung°

요 약

무선 라우팅 분야에 강화학습을 적용하여 네트워크의 안정성을 높이고 지연시간을 줄이려는 연구가 진행되고 있다. 기존 무선 라우팅 알고리즘들은 노드가 부모 노드를 선택할 때 부모 노드까지의 거리나 부모 노드까지 패킷 전달에 걸리는 평균 시간 등 주변 정보에 의존한다. 본 논문에서는 네트워크 전체 상황 정보를 주기적으로 갱신하고 이를 반영하여 강화학습의 일종인 Q-learning을 적용하는 방안을 제시하고, 이를 저전력 저손실 네트워크를 위한 표준 라우팅 프로토콜 RPL(Routing Protocol for Low power and Lossy network)에 적용한 라우팅 알고리즘을 제안한다. 제안하는 알고리즘은 네트워크의 트래픽 양이 많은 상황에서 기존 라우팅 알고리즘과 비교하여 네트워크의 부하가 집중되는 것을 막을 수 있다. 시뮬레이션을 통해 기존 Full echo Q-routing과 성능 비교를 진행하였고 제안하는 알고리즘이 PDR(Packet delivery rate)과 지연시간 측면에서 더 좋은 성능을 보이는 것을 검증하였다.

Key Words : IoT, Wireless sensor network, 802.15.4e, TSCH, RPL, Q-learning

ABSTRACT

Researches to enhance network stability and reduce latency by applying reinforcement learning in the wireless routing are being carried on. Existing wireless routing algorithms rely on neighbor information, such as the distance to the parent node or the average time taken to deliver the packet to the parent node when the node selects the parent node. In this paper, we propose to utilize the Q-learning, a kind of reinforcement learning, with periodically updating the whole network information and suggest to apply to RPL(Routing protocol for low power and lossy network), the standard routing protocol for low-power low-loss networks. The proposed algorithm can achieve the network load balancing, compared to the existing routing algorithm where the traffic load is high. Through simulation, we compared the performance with the existing Full echo Q-routing and verified that the proposed algorithm shows better performance in terms of PDR (Packet delivery rate) and latency.

Ⅰ. 서 론

무선 네트워크의 환경이 IoT로 변화함에 따라 IEEE 802.15.4e는 TSCH (Time Slotted Channel Hopping)를 새로운 표준으로 채택하였다[1,2]. TSCH의 표준 라우팅 프로토콜은 RPL(Routing Protocol for Low power and Lossy network)로, 저전력 손실 네트워크(LLN, Low power lossy network)를 위해 IETF에서 승인한 라우팅 프로토콜이다. 저전력 손실 네트워크의 문제는 많은 노드가 하나의 루트 노드로 연결

※ 이 성과는 과학기술정보통신부가 지원한 '정보통신방송연구개발사업'으로 지원을 받아 수행되었음."(과제고유번호: 2018-0-00846).
◆ First Author : Sangmyung Universit, 학생회원
° Corresponding Author : Sangmyung University, Department of Human-centered AI, jjoung@smu.ac.kr, 정회원
논문번호 : 202006-124-B-RN, Received June 10, 2020; Revised July 13, 2020; Accepted July 14, 2020

되어 혼잡이 발생하고, 일부 노드에서의 병목 현상으로 지연시간 증가와 패킷 손실이 발생하며, 네트워크 전체의 안정성이 감소할 수 있다는 것이다. 이를 해결하기 위해 RPL과 관련해 부하 분산 기술 연구가 활발히 진행되고 있다. 한편 네트워크 라우팅 분야에서도 강화학습을 적용하여 네트워크의 성능을 높이려는 연구가 진행되고 있다. 기존 라우팅 알고리즘은 노드가 패킷을 전달하기 위해 부모 노드를 선택한다. 이때 부모 노드는 노드와 이웃 노드와의 거리, 링크 상태에 기반한 성능을 바탕으로 선택한다. 강화학습인 Q-learning을 적용한 라우팅 알고리즘의 경우 경로를 선택할 때 이웃 노드의 성능과 패킷이 전해지는 경로에 있는 선조 노드들의 성능도 예측하여 부모 노드를 선택할 수 있다. 강화학습을 적용한 라우팅 알고리즘은 다양한 네트워크 상황에서 기존 라우팅 알고리즘보다 더 효율적으로 부하 분산을 할 수 있다.

Ⅱ. 관련 연구

강화학습의 한 종류인 Q-learning은 의사 결정자(agent)가 행동(action)을 선택했을 때 얻는 보상(reward)의 기댓값을 예측하는 Q 함수의 학습을 통해 최적의 정책(policy)을 학습한다[3]. 이때 정책은 주어진 상태(state)에서 의사결정자가 어떤 행동을 선택할지 나타내는 규칙이다.

$$Q(state_t,action_k) = R(state_t,action_k) + \gamma * MAX(Q(state_{t+1},action_1), Q(state_{t+1},action_2),...,Q(state_{t+1},action_n)) \quad (1)$$

(1)에서 $Q(state_t,action_k)$는 상태 $state_t$에서 행동 $action_k$을 선택했을 때 보상과 그다음 상태 $state_{t+1}$에서 선택할 수 있는 행동 중 가장 큰 $Q(state_{t+1},action_q)$의 합이다. $Q(state_t,action_k)$를 Q-value라 부르도록 한다.

Q-learning은 현재 상태부터 연속적인 상태들을 모두 거쳤을 때 얻게 되는 전체 보상의 예측값을 극대화한다. 이런 특성은 한 상태에서 다음 상태로 전이할 때의 보상이 확률적으로 주어지는 환경에서도 별다른 변형 없이 적용될 수 있다. Q 함수의 학습 후에는 각 상태에서 최고의 Q-value를 가지는 행동을 선택해 최적의 정책을 유도한다. Q-learning은 모델이 없어도 행동의 기댓값(Q-value)을 비교할 수 있다.

Q-routing은 기존 라우팅 알고리즘에 강화학습의

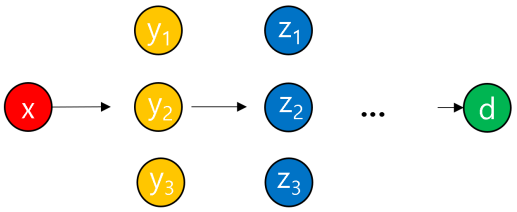

그림 1. 예시 네트워크 토폴로지
Fig. 1. Example network topology

일종인 Q-learning을 적용한 라우팅 알고리즘이다. 그림 1을 Q-routing의 예시 네트워크 토폴로지라 가정할 때, 노드 x는 패킷을 전달하기 위해 이웃 노드 y_i 중 하나를 부모 노드로 선택하여 패킷을 전달한다. 이 과정은 목적지 노드 d까지 패킷이 전달될 때까지 계속된다. 노드 x의 부모 노드를 선택할 때 노드 x가 y_i에게 패킷을 전달하는데 걸리는 총 시간과 이웃 노드 y_i가 목적지 노드까지 패킷을 전달하는 데 걸리는 총 예측 시간의 합이 가장 작은 이웃 노드를 선택한다[4].

$$t = \min_{z \in N(y)} Q_{y(d,z)} \quad (2)$$

$$new\,Q_{x(d,y)} = old\,Q_x(d,y) + \eta(q+s+t-old\,Q_x(d,y)) \quad (3)$$

(2)에서 t는 이웃 노드 y_i가 목적지가 d인 패킷의 전달을 위해 이웃 노드 z_k에 패킷을 전달할 때 걸리는 시간 중 최소 예측 시간이다. (3)을 바탕으로 노드 x는 Q-value인 $Q_{x(d,y)}$를 갱신한다. x가 이웃 노드 y_i를 통해 패킷의 목적 노드 d에게 패킷을 전달할 때의 Q-value를 갱신하는 수식이다. $Q_{x(d,y)}$는 t, 패킷이 x의 큐에 머무는 시간(q)과 노드 x에서 노드 y까지 패킷을 전달하는데 걸리는 시간(s)을 포함한다. 이때 Q-value는 패킷이 목적지 노드까지 전달되는데 걸리는 총 예측 시간을 의미한다. 학습률 η만큼 $Q_{x(d,y)}$에 측정한 q,s,t를 반영한다. 노드 x가 노드 y_1에게 패킷을 전달하면 노드 y_1은 그 즉시 y_1의 이웃 노드 z_i를 통해 패킷을 전달하는데 걸리는 최소 예측 시간(t)을 노드 x에게 전달한다. Q-table은 패킷의 목적지 별로 이웃 노드의 Q-value 값을 모아 놓은 테이블이다. 노드 x는 이웃 노드 y_i중 부모 노드를 선택할 때 Q-table에서 Q-value가 가장 작은 노드를 선택한다. 그림 2는 기존 최단 경로 알고리즘을 사용한 라우팅 알고리즘과 Q-routing의 성능을 비교한 표이다.

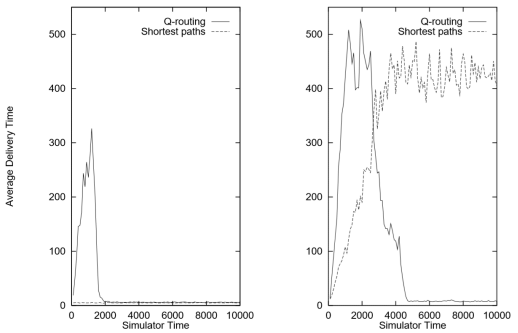

그림 2. Q-routing과 shortest path의 성능 비교. 왼쪽은 부하가 낮은 경우. 오른쪽은 부하가 큰 경우
Fig. 2. Q-routing and shortest path's performance based on low and high load.

Q-routing이 네트워크의 트래픽이 적은 상황에서 평균 전송 시간(Average Delivery Time)이 초반에 높지만, 네트워크의 부하가 큰 상황에서는 시뮬레이션이 수행될수록 평균 전송 시간이 감소하는 것을 확인할 수 있다.

Full Echo Q-routing은 Q-routing을 바탕으로 한다. Full Echo Q-routing은 노드가 패킷을 전달하기 전에 이웃 노드에게 t를 요청한다[4]. 그 후 이웃 노드들이 보낸 t를 바탕으로 Q-table의 Q-value를 갱신하고, Q-value가 가장 적은 부모 노드에게 패킷을 전달한다. Full Echo Q-routing을 Q-routing과 비교했을 때, exploration이 더 많이 발생하여 Q-value가 고정되는 문제를 개선할 수 있다. 그림 2를 보면 Full echo Q-routing은 초반 전송 시간이 수렴하지 않는 문제(oscillation)가 발생한다는 단점이 있다. Adaptive Q-routing Full Echo(AQFE)의 Q-value를 갱신하는 방법은 Full Echo Q-routing에 기반한다[5]. AQFE에서는 노드가 2가지의 학습률 η_1(original)과 η_2(additional)를 사용하며, 평균 전송 시간에 따라 추가 학습률 (η_2)을 변경한다. (4)는 η_2을 갱신하는 공식이다. T_{est}는 전송 시간(delivery time)의 평균 예측값이고, $T_{\max}$는 전송 시간의 최대 예측값이다. k는 이 수식에서 사용된 값이 정해진 상수이다. 즉, T_{est}는 $Q_{x(d,y)}$의 평균값이고, $T_{\max}$는 과거 T_{est}의 최대값이다.

$$\eta_2 = \eta_1 k \frac{T_{est}}{T_{\max}} \tag{4}$$

AQFE는 학습률 η_2의 피드백 시스템을 통해 exploitation과 exploration의 조화를 이룰 수 있게 된다. Exploitation은 부모 노드를 선택할 때 Q-table의 Q-value가 가장 작은 노드를 선택하는 것을, exploration은 Q-value와 상관없이 이웃 노드 중 랜덤 노드를 부모 노드로 선택하는 것을 의미한다. Exploitation은 선택된 이웃 노드의 Q-value만 갱신된다는 한계를 가진다. 그래서 부모 노드 선택 시 일정 확률로 exploration을 수행한다. 노드는 패킷을 받은 이웃 노드의 Q-value를 갱신할 때 학습률 η_1으로 갱신한다. 그 외 이웃 노드의 Q-value를 갱신할 때는 학습률 η_2을 사용한다. η_2는 (3)을 바탕으로 갱신된다. 평균 전송 시간이 증가하는 경우 네트워크의 효율이 떨어진 것이므로 더 많은 exploration의 발생을 위해 η_2가 증가한다. 반대로 평균 전송 시간이 감소하는 경우 네트워크의 효율이 올라간 것이므로 η_2 가 감소 된다. AQFE는 Full Echo Q-routing과 비교했을 때 전송 시간이 수렴하지 않는 현상을 줄여서 높은 효율을 낼 수 있다. AQFE는 네트워크의 부하가 큰 경우에 Full Echo Q-routing보다 높은 효율을 보이지만, 네트워크가 불안정하거나, k값이 커지는 경우 전송 시간이 수렴하지 않는 문제가 있다.

Adaptive Q-routing with Random Echo and Route Memory(AQRERM)는 AQFE에 Random Echo와 Route Memory를 추가로 제안하였다[6]. Random Echo는 임의의 한 이웃 노드 y_i에게만 t를 요구한다. Route memory는 지금까지 패킷이 방문한 일련의 노드 정보를 크기 L만큼 유지한다. 즉 Route memory에 노드의 정보가 있다면 패킷의 경로(route)에 있던 노드이다. 부모 노드를 선택할 때 Route memory에 없는 이웃 노드 중에서 Q-value가 가장 작은 이웃 노드를 선택하고, 이웃 노드가 모두 Route memory에 있다면 Route memory를 고려하지 않고 부모 노드를 선택한다.

Ⅲ. Routing protocol with periodic Q-table update (RPQU)

AQFE에서는 학습률 η_1이외에 추가 학습률 η_2를 사용한다. η_2를 구하기 위해서 패킷의 평균 전송 시간 (T_{est})을 구해야 한다. T_{est}는 주변에 이웃 노드가 많고 유입이 활발한 저전력 저손실 네트워크에서는 정확한 값을 구하기 어려워, η_2의 신뢰성이 떨어진다. AQRERM의 경우 패킷의 경로를 저장하는 Route

memory가 필요하다. 저전력 저손실 네트워크에서는 일반적으로 노드의 성능이 떨어지고 목적지 노드까지의 경로가 길어 Route memory 사용으로 노드에 추가적 부하가 발생할 수 있다. 따라서 AQFE와 AQRERM은 저전력 저손실 네트워크에 적용하기에 적절하지 않다.

Full echo Q-routing에서 패킷을 전송하기 전에 노드는 패킷 전송 전에 항상 이웃 노드 y_i에게 t를 받아 Q-table을 갱신한다. 이렇게 모든 패킷의 전송마다 t 값을 요청하고 받는 과정은 전체 네트워크의 혼잡 및 지연으로 이어질 수 있다. 더구나 패킷을 보내고자 하는 노드의 주변 네트워크 상황만 반영하며 전체 네트워크의 상황은 반영하지 못한다. 이는 네트워크가 트래픽이 많은 상황에서 더 심해질 수 있다. 최근 네트워크는 많은 노드가 하나의 네트워크에 연결되어, 노드 주변에는 많은 노드가 존재한다. 기존 Q-routing에 적용된 Q-table을 처리하기에는 과도한 전력 소모, 전송 지연이 발생할 수 있어 네트워크의 전체 성능 저하가 발생할 수 있다. RPL은 저전력 저손실 네트워크에 적합한 라우팅 알고리즘이므로 Q-learning을 적용하면 해당 문제를 해결할 수 있다[7].

본 논문에서 제안하는 알고리즘을 Wireless routing protocol with periodic Q-table update (RPQU)로 부르도록 한다. RPQU의 핵심 아이디어는 일정한 주기로 DODAG에 속한 전체 노드의 Q-table을 갱신하는 것이다. Full Echo Q-routing에서 모든 노드는 목적지 노드가 될 수 있다. 노드 x의 Q-table은 그림 3과 같이 행에는 목적지 노드가 열에는 노드 x의 모든 주위 노드 y_i가 요소로 들어간다. 이때 Q-value는 같은 이웃 노드라 하더라도 패킷의 목적지 노드에 따라 다를 수 있다.

RPL에서는 루트 노드가 하나 존재한다. 네트워크에 연결된 노드는 이웃 노드와 DODAG (Destination Oriented Directed Acyclic Graph)를 형성하고, 루트 노드까지 패킷을 전달한다. 그림 3의 왼쪽은 Full Echo Q-routing의 Q-table이고, 오른쪽은 RPL기반 RPQU의 Q-table이다. 노드 x의 Q-table은 행에는 루트 노드 1개와 열에는 노드 x의 이웃 노드 y_i가 요소로 들어가게 된다. 이때 Q-table은 같은 이웃 노드들은 모두 같은 Q-value 값을 가지게 된다.

destination / neighbor	y_1	z_1	y_2	...
y_1				
y_2				
...				
y_n				

destination / neighbor	d
y_1	
y_2	
...	
y_n	

그림 3. 노드 x의 Full Echo Q-routing(왼쪽)과 RPQU(오른쪽)의 Q-table 예시
Fig. 3. Full Echo Q-routing's (left) and RPQU's (right) Q-table example

$$Q_{new} = Q_{old} + \eta(R - Q_{old}) \tag{5}$$

$$R = \delta MQP + (1-\delta)QL \tag{6}$$

(5)는 Q-value를 갱신하는 수식이다. R은 보상(reward)이다. R은 보상(reward)이고, η는 학습률로 새로 측정한 R이 Q_{new}에 반영되는 정도를 결정한다. Q_{new}는 기존 Q_{old} 값과 R을 바탕으로 계산된다. (6)은 R을 정의하는 수식이다. QL(Queue length)는 Q-value를 측정하는 시점의 노드의 큐 길이이다. MQP(Min Q-value of parents)는 같은 시점에 노드의 이웃 노드의 Q-table에 있는 Q-value 중 가장 작은 값이다. δ는 반영률로 R에 QL와 MQP가 반영되는 정도를 결정한다.

표 1은 RPQU의 알고리즘을 의사 코드로 나타낸 것이다. 프레임은 n개의 슬롯으로 구성되는데, 슬롯이 Q-table을 갱신해야 하는 주기가 되면 하나의 DODAG에 속한 모든 노드의 Q-table을 갱신한다. 그 후 해당 슬롯에서 패킷을 전송할 노드(Tx node)는 자신의 Q-table에서 가장 작은 Q-value를 가지는 이웃 노드를 선호 부모 노드로 선택한다. 그 후 선호 부모 노드가 해당 슬롯에서 수신 상태이고, 큐에 패킷을 받을 공간이 있으면, 선호 부모 노드로 패킷을 전송한다.

Full Echo Q-routing에서 노드 x는 패킷 전송 전에 항상 패킷의 목적지 노드와 관련된 Q-table을 갱신한다. 그림 3의 왼쪽 테이블의 한 행을 갱신한다. Q-table을 갱신하기 위해 패킷 전송 전에 모든 이웃 노드 y_i에게 t를 요청한다. RPQU에서 노드 y는 기존 t 대신 가장 작은 Q-value를 노드 x로 전달한다. 노드 x는 이 값을 바탕으로 Q-table을 갱신한다. 노드 x는 Q-table 갱신 후에 가장 작은 Q-value를 가지는 부모 노드 y_i에게 패킷을 전달한다. 이 과정은 패킷이 최종적으로 목적지 노드에 도착할 때까지 계속된다.

그림 4의 왼쪽은 Full echo Q-routing, 오른쪽은 RPQU의 이웃 노드로부터 t와 MQP를 수신하는 방

표 1. RPQU 알고리즘의 의사 코드
Table 1. RPQU Algorithm's pseudo code.

```
Wireless routing protocol with periodic Q-table update

Period = Q-table update period
Tx node
        = Node to send a packet in current slot
Rx node
        = Node to receive a packet in current slot
Preferred Parent node(PP node)
        = Node which have min Q-value of
    Q-table(i)

If ( slot%Period == 0 )
   for i = 1 : total node
      node i's Q-table(i) update

PP node
        = node which have minimal value
          of Tx node's Q-table

If ( PP node == Rx node & PP node's queue
!= Full )
   Tx node sends a packet to PP node

slot++;
```

법을 나타낸다. Full echo Q-routing은 패킷을 전송 전에 모든 이웃 노드에게 t를 요청한다. 요청을 받은 이웃 노드들은 MQP를 노드 x에게 유니캐스트한다. 노드 x는 전송받은 MQP를 바탕으로 Q-table을 갱신한다. 반면 RPQU는 Q-table을 갱신하는 주기가 있어, 이웃 노드들은 특정 시점에 MQP를 브로드캐스트한다. 노드 x는 전송받은 MQP를 바탕으로 전체 Q-table을 갱신한다. 노드는 갱신한 Q-table을 바탕으로 선호 부모 노드를 선택한다.

Q-table을 주기적으로 갱신하면 네트워크의 트래픽과 상관없이 선호 부모 노드를 선택할 때 네트워크와 노드의 상황을 안정적으로 반영할 수 있다. 결과적으로 네트워크의 안정성을 높이고 전송 지연을 줄일 수 있다.

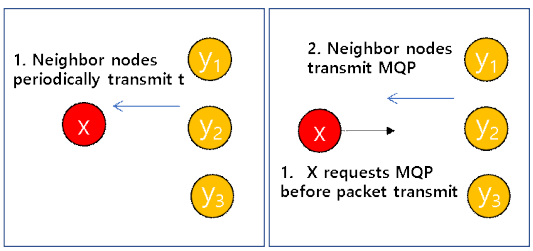

그림 4. Full Echo Q-routing의 t, RPQU의 MQP 갱신 방법
Fig. 4. Full Echo Q-routing's t and RPQU's MQP update method

Ⅳ. 성능 비교

Full Echo Q-routing과 RPQU를 각각 구현하여 두 알고리즘의 성능을 비교하였다. 시뮬레이션 환경은 Matlab을 통해 구축하였다. 시뮬레이션에서 사용하고 있는 변수는 표 2와 같으며, 그림 5는 시뮬레이션에서 사용된 토폴로지를 나타낸다.

그림 5에 점으로 표시된 곳이 노드의 위치이다. 시뮬레이션에 사용된 노드는 총 16개이며 그림 5의 (10, 10)에 위치한 노드가 루트 노드이다. 스케줄링은 TSCH을 바탕으로 구현하였다.

표 2는 시뮬레이션에서 사용한 파라미터의 값이다. 시뮬레이션은 128프레임 동안 수행된다. exploration은 30%, 패킷의 time_to_live는 2프레임이며, 1프레임은 16개의 슬롯으로 구성된다. Exploration은 30%로 선호 부모 노드를 결정할 때 30% 확률로 Q-value와

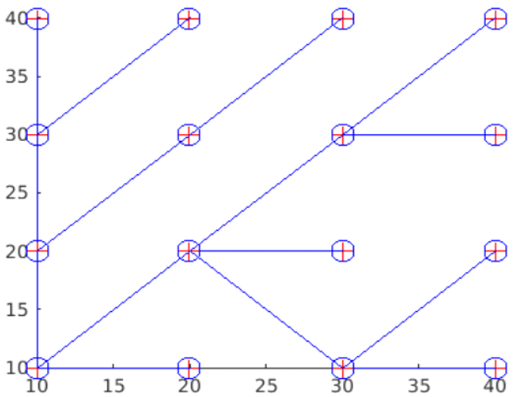

그림 5. RPQU simulation에 사용된 토폴로지
Fig. 5. RPQU topology used in the simulation

표 2. 시뮬레이션에 사용된 파라미터
Table 2. Parameters used in the simulation

Contents	Value
Area	40m X 40m
노드	16개
프레임	128 프레임
프레임당 슬롯 수	16 슬롯 / 프레임
전송 범위	15m
간섭 범위	30m
시뮬레이션 수행 횟수	10번
큐 길이	10

상관없이 무작위로 선호 부모 노드를 선택한다. 시뮬레이션에서 exploration이 30%보다 작은 경우 최소 Q-value인 노드만 선택되어 경로가 고정되었고, 30%보다 큰 경우 Q-value가 경로 설정에서 영향이 줄어들었다. 패킷의 time_to_live는 2프레임으로, 패킷은 생성되고 2프레임 안에 목적지(루트) 노드까지 도달하지 못하면 폐기된다. 2 프레임보다 작으면 패킷이 목적지 노드까지 도달하기 전에 폐기되는 비율이 높았고, 그 이상일 경우 time_to_live가 제대로 작동하지 못했다. 시뮬레이션은 1프레임은 16개의 슬롯으로 구성되며, 128프레임 동안 수행했다.

그림 6은 RPQU와 Full echo Q-routing의 PDR 비교표로 갱신 횟수(frequency)가 2인 경우를 제외하고 RPQU가 Full echo Q-routing보다 PDR이 높다. Packet Delivery Rate(PDR)의 경우 노드에서 생성된 패킷의 수를 마지막 목적지 노드인 루트 노드에 도착한 전체 패킷의 합으로 나눈 값이다. 그림 6의 갱신 주기(period)는 한 프레임 당 DODAG에 속한 전체 노드의 Q-table을 갱신하는 횟수이다. 이때 갱신 주기가 4인 경우 1프레임 동안 4번 Q-table이 갱신되며, 시뮬레이션을 수행하는 동안 총 512번 Q-table이 갱신된다. 그림 6에서 RPQU의 갱신 횟수가 증가할수록 PDR이 증가한다. Q-table의 갱신 횟수가 많을수록 Q-value에 노드와 네트워크의 상황 잘 반영된다. 즉 선호 부모 노드 선택을 효율적으로 할 수 있어 PDR이 증가한다. 다만 갱신 횟수가 8과 16의 경우 PDR의 차이가 거의 없다. 즉 Q-table의 갱신 횟수가 일정 이상이 되면 PDR 차이가 크지 않다.

그림 7은 전송 지연(delay)을 비교한 그림이다. RPQU는 갱신 주기 대부분에서 비슷한 수준의 전송 지연을 보인다. 전송 지연이 낮은 것은 패킷 하나가

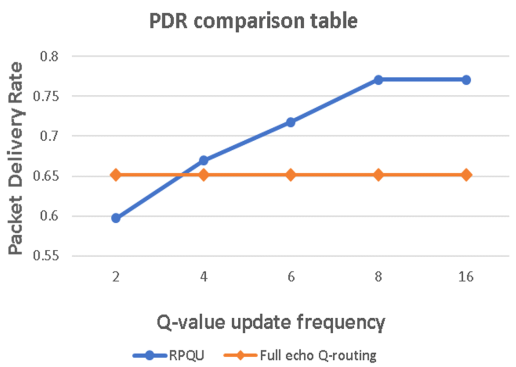

그림 6. RPQU와 Full echo Q-routing의 PDR
Fig. 6. RPQU and Full echo Q-routing's PDR

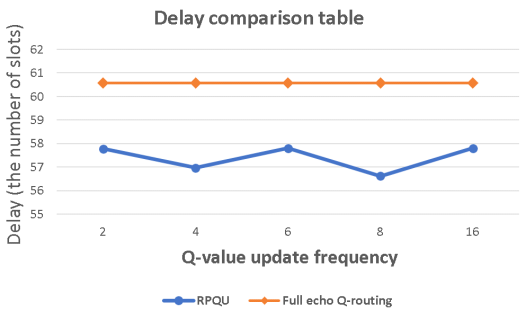

그림 7. RPQU와 Full echo Q-routing의 전송 지연(delay) 비교표
Fig. 7. RPQU and Full echo Q-routing's Delay comparative table.

전송되는데 필요한 슬롯 수가 더 적다는 것을 의미하며 전체 패킷의 총 전달 시간이 줄어든다. 전송 지연은 한 개의 패킷 전송에 필요한 슬롯 수이다. RPQU는 Full echo Q-routing보다 PDR이 높으면서 지연시간은 더 적다. 즉 RPQU가 패킷을 목적지 노드까지 안정적으로 빠르게 보내는 것을 알 수 있다.

V. 결 론

본 연구에서 제안하는 RPQU는 노드가 선호 부모 노드를 선택할 때 노드의 Q-table에 가장 작은 Q-value를 가지는 이웃 노드를 선호 부모 노드로 선택한다. 노드는 자기 이웃 노드들의 Q-value를 모아놓은 Q-table을 미리 정한 주기마다 갱신한다. 이때 갱신된 Q-value(Q_{new})는 갱신 전의 Q-value(Q_{old})와 이웃 노드의 Q-table의 가장 작은 Q-value인 MQP의 합으로 구성된다. MQP가 Q-value에 반영되므로 패킷의 경로상에 있는 선조 노드의 성능이 반영된다. 시뮬레이션을 통해 PDR과 전송 지연이 RPQU가 Full Echo Q-routing보다 좋은 효율을 보였다. 무선 네트워크는 노드의 이동, 추가, 사라짐이 빈번하게 발생한다. 본 연구는 노드를 고정하여 시뮬레이션을 진행하였다. 향후 노드가 이동, 추가되는 시뮬레이션을 진행할 예정이다. 이를 위해 노드의 이동 경로와 네트워크에 추가되거나 사라지는 노드를 인식하는 방법을 고려해 시뮬레이션을 진행할 예정이다. 본 연구는 Q-learning을 라우팅 알고리즘에 적용하였다. 향후 RPL에 다른 강화학습을 적용하여, Q-learning을 적용한 RPQU와 비교하는 연구를 수행할 계획이다.

References

[1] S. H. Chung, Y. S. Lee, and Y. Ha, "IEEE 802.15.4e TSCH based wireless network technology in smart factory," *J. KICS*, vol. 35, no. 4, pp. 3-10, 2018.

[2] R. T. Hermeto, A. Gallais, and F. Theoleyre, "Scheduling for IEEE802.15.4-TSCH and slow channel hopping MAC in low power industrial wireless networks: A survey," *Computer Commun.*, vol. 114, pp. 84-105, 2017.

[3] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach Learn 8*, pp. 279-292, 1992.

[4] J. Boyan and M. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," *Advances in Neural Inf. Process. Syst. 6*, 1994.

[5] Y. Shilova, M. Kavalerov, and I. Bezukladnikov, "Full echo Q-routing with adaptive learning rates: A reinforcement learning approach to network routing," 2016 *IEEE NW Russia Young Researchers in Electrical and Electronic Eng. Conf. (EIConRusNW)*, pp. 341-344, 2016.

[6] M. Kavalerov, Y. Likhacheva, and Y. Shilova, "A reinforcement learning approach to network routing based on adaptive learning rates and route memory," *SoutheastCon 2017*, pp. 1-6, 2017.

[7] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Strui, JP. Vasseur, and R. Alexander, "RPL: IPv6 routing protocol for low-power and lossy networks," *IETF*, Mar. 2012.

김 태 정 (Taejung Kim)

2019년 2월 : 상명대학교 컴퓨터과학과 졸업

2019년 3월~현재 : 상명대학교 일반대학원 컴퓨터과학과 석사과정

<관심분야> 인공지능, 무선네트워크

정 진 우 (Jinoo Joung)

1992년 2월 : KAIST 전자공학과 학사

1994년 8월 : NYU 전기전자공학과 Master

1997년 8월 : NYU 전기전자공학과 Ph.D.

1997년 10월~2005년 2월 : 삼성전자 종합기술원

2005년 3월~현재 : 상명대학교 휴먼지능정보공학과 교수

<관심분야> 유무선통신, 네트워크, 임베디드 시스템

[ORCID:0000-0003-3053-9691]