

RPL에서 이동성 향상을 위한 DIO 전송 간격 조절

신예진¹ · 설순욱^{2*}

The DIO Interval Adjustment to Enhance Mobility in RPL

Yejin Shin¹ · Soonuk Seol^{2*}

¹Ph.D. Student, School of Electrical, Electronics & Communication Engineering, Korea University of Technology & Education, Cheonan, 31253 Korea

^{2*}Associate Professor, School of Electrical, Electronics & Communication Engineering, Korea University of Technology & Education, Cheonan, 31253 Korea

요 약

본 논문에서는 RPL을 사용하는 사물인터넷 환경에서 노드들이 이동할 때에도 토폴로지 변경에 빠르게 적응하여 패킷 손실 문제를 해결하기 위한 방안을 제안한다. 이동성을 향상시키기 위해 모든 노드는 이웃 노드들의 이동성을 인식하고, 전체 수신 패킷과 제어메시지 수를 고려하여 이동 정도를 수치화한다. 이동 정도에 따라 DIO 타이머를 동적으로 설정하여 토폴로지 변경을 빠르게 인식하고 목적지까지의 경로를 업데이트할 수 있도록 한다. 제안 방식의 성능은 Contiki 기반 Cooja 시뮬레이터를 이용하여 다양한 이동 속도에 대해서 평가한다. 시뮬레이션 결과, 제안된 방식은 패킷 전달률이 31.03% 개선됨을 확인하여 표준 RPL보다 이동성 시나리오에 잘 대처함을 보여준다.

ABSTRACT

The main purpose of this research is to propose an approach for solving the packet loss problem by quickly adapting to topology change when nodes move in RPL-based IoT environment. In order to enhance mobility, every node is aware of the mobility of its neighbor nodes and quantifies the mobility level based on the number of control messages and all received packets. According to the mobility level, the DIO timer is changed. The proposed approach allows nodes to change their DIO timers according to their mobility levels to adapt topology changes and update paths to the sink. The performance of the proposed approach is evaluated using a Contiki-based Cooja simulator in various moving speeds. The simulation results show that the proposed approach copes with mobility scenarios better than the standard RPL by ascertaining that the packet delivery ratio is improved by 31.03%.

키워드 : RPL, 저전력 손실 네트워크, 사물인터넷, 이동성

Keywords : RPL, LLN, IoT, Mobility

Received 8 October 2019, Revised 18 October 2019, Accepted 29 October 2019

* Corresponding Author Soonuk Seol(E-mail:suseol@koreatech.ac.kr, Tel:+82-41-560-1426)

Associate Professor, School of Electrical, Electronics & Communication Engineering, Korea University of Technology & Education, Cheonan, 31253 Korea

Open Access <http://doi.org/10.6109/jkiice.2019.23.12.1679>

print ISSN: 2234-4772 online ISSN: 2288-4165

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

사물인터넷 (IoT: Internet of Things)은 수많은 장치들을 인터넷에 연결하는 기술로, 헬스케어용 디바이스와 같은 일상적인 기기뿐만 아니라 스마트 공장 등 산업용 응용에 적용하는데 큰 관심이 모아지고 있다. IoT 응용에 적용되는 장치들은 일반적으로 전력, 대역폭 등 엄격한 리소스 제약 조건 하에서 작동한다. 따라서 IoT 네트워크는 신뢰성, 안정성 등의 측면에서의 성능 저하가 불가피하기 때문에 제한된 리소스 환경에서도 안정적인 성능을 제공할 수 있는 네트워킹 기술이 요구되었다. 이러한 필요에 따라 IETF (Internet Engineering Task Force)에서는 저전력 손실 네트워크 (LLN: Low power and Lossy Network)를 위한 라우팅 프로토콜인 RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) [1]을 표준으로 제정하였으며, 대표적인 IoT 라우팅 프로토콜로서 활발한 연구가 진행되고 있다.

RPL은 IPv6 (Internet Protocol version 6)[2]를 적용하고 IEEE 802.15.4 표준[3]을 기반으로 하는 LLN용 라우팅 프로토콜이다. RPL은 원래 정적 네트워크용으로 설계되어 이동성이 요구되는 환경에는 최적화되지 않았다[4]. 이에 따라 노드가 이동할 경우 빈번한 연결 끊김으로 인해 패킷이 손실되고 에너지 소비가 증가하여 RPL 전체 성능을 저하시킨다. 이는 커넥티드 카, 스마트 공장, 헬스케어 서비스 등 이동성이 요구되는 응용에 적용하기에는 한계가 있다.

이러한 한계를 극복하기 위해 RPL에서의 이동성 지원에 관한 많은 연구들이 진행되어왔다. 그 중 Kevin C. Lee, et al.의 연구에서는 RPL 제어메시지의 전송 간격을 고정하여 주기적으로 제어메시지가 전송되도록 하는 RPL-VANET[4]을 제안하였다. 표준 RPL에서는 제어메시지의 전송을 통해 토폴로지를 구성하고 토폴로지 변경 시 적용하는데, 만약 토폴로지 변경이 없는 경우에는 제어메시지 전송 간격을 늘린다. RPL-VANET 처럼 제어메시지 전송주기를 짧게 설정한다면 노드 이동에 따른 토폴로지 변경에 빠르게 적응하여 목적지 (Sink)까지의 경로를 업데이트할 수 있다. 그러나 일시적으로 노드가 이동하지 않아 토폴로지 변경사항이 없는데도 계속 짧은 주기로 제어메시지를 보낸다면 불필요한 메시지로 인한 네트워크 트래픽 및 메시지 처리에 의한 부하 측면의 오버헤드가 발생한다.

본 논문에서는 RPL에서의 이동성 지원을 위해 이동 여부를 판단하여 RPL 제어메시지 전송 간격을 변경하는 방안을 제안한다. 제안 방식에서 모든 노드는 이웃 노드들의 이동성을 인식하며, 이동성 인식을 위해 이웃 노드들로부터 전송받은 패킷을 활용한다. RPL 제어메시지는 IP (Internet Protocol)계층 프로토콜인 ICMPv6 (Internet Control Message Protocol version 6)를 통해 전송된다. 만약 전체 수신 패킷 중 ICMPv6 패킷의 비중이 크다면 주변 노드들의 상태 정보가 자주 변경된다고 판단할 수 있다. 따라서 일정 시간 동안 수신된 전체 패킷 중 ICMPv6 패킷의 비중을 계산하여 이동 여부를 판단하고, 이동 정도에 따라 RPL 제어메시지 전송 간격을 동적으로 설정한다. 이를 통해 이동 정도가 클 경우 제어메시지 전송 간격을 짧게 설정함으로써 토폴로지 변화에 신속하게 적응하도록 한다.

제안 방식의 성능은 Contiki[5] 기반 Cooja 시뮬레이터에서 다양한 이동 속도에 대해 평가한다. 기존 연구 대비 성능 향상 여부를 확인하기 위해 패킷 전달률 및 제어메시지 오버헤드 측면에서 표준 RPL, RPL-VANET[4]과 비교한다.

본 논문의 구성은 다음과 같다. 2장에서는 RPL에서의 DODAG (Destination-Oriented Directed Acyclic Graph) 구성 절차와 제어메시지 전송 간격을 설정하는 트리클 알고리즘에 대해 소개한다. 그리고 RPL에서의 이동성 향상에 관한 기존 연구들을 3장에서 설명한다. 4장에서 본 논문에서 제안하는 제어메시지 전송 간격 조절 기법을 소개하고, 5장에서는 제안 기법의 성능을 평가한다. 마지막으로 6장에서는 결론 및 향후 연구를 제시한다.

II. RPL 개요

본 장에서는 RPL의 기본 동작에 대해 설명한다. RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks)은 IETF 워킹그룹에서 제안한 표준으로, IPv6 기반 저전력 무선 네트워크를 위한 라우팅 프로토콜이다. RPL에서는 세 종류의 제어메시지를 이용하여 트리 형태의 비순환 그래프 (DODAG: Destination-Oriented Directed Acyclic Graph)를 구성하고 목적지 (Sink)로의 경로를 형성한다.

2.1. RPL제어메시지

DODAG을 구성하기 위한 RPL 제어메시지에는 DIS (DODAG Information Solicitation), DIO (DODAG Information Object), DAO (Destination Advertisement Object)가 있다. RPL 제어메시지를 활용한 DODAG 구성 예시를 그림 1에 나타내었다.

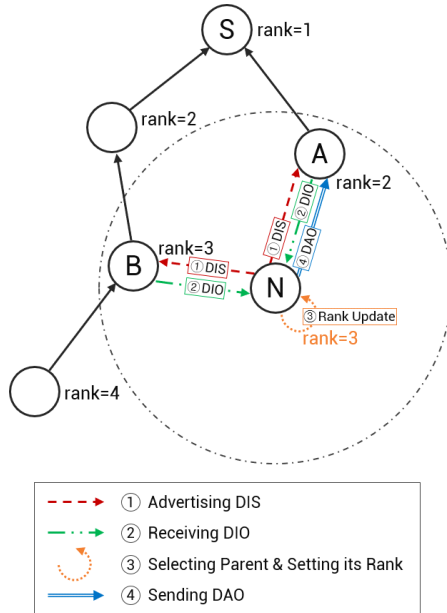


Fig. 1 An example of DODAG construction procedure using RPL control messages

그림 1에서 새로운 노드 N이 생성되면 노드 N은 DODAG에 가입하기 위해 주변 노드들에게 DODAG 정보를 요청하는 DIS를 브로드캐스팅하는데, 이는 그림 1의 과정 ①에 해당한다. DIS를 받은 주변 노드 A, B는 그에 대한 응답으로 자신이 속한 DODAG 및 자신의 정보를 담은 DIO 메시지를 전송하며, 이 과정은 그림 1의 ②에 해당한다. 이때 DIO 메시지에는 DODAG ID, Rank 정보 등이 포함된다. 이처럼 DIO는 DODAG 구성 단계에서 DIS에 대한 응답으로 유니캐스트 될 수 있으며, DODAG 구성이 완료된 후에는 DIO 브로드캐스팅을 통해 자신이 속한 DODAG 및 자신의 노드 정보를 주변에 알릴 수 있다.

주변 노드들 (노드 A, B)로부터 DIO를 받은 노드 N은 가장 적합한 부모를 선택하기 위해 목적 함수 (OF: Objective Function)라고 불리는 부모 선정 알고리즘을

활용한다. 목적 함수에 따라 Sink와 가까운 부모를 우선적으로 선택하거나 노드 간 링크 품질이 좋은 부모[6], 또는 배터리 잔여량이 많은 부모[7-9] 등을 선택할 수 있다. 그림 1에서의 방식은 Sink (노드 S)까지의 홉 (hop) 수를 기반으로 노드 S에 가까운 부모를 선택하는 OF를 적용한 예시이다. 노드 N은 각 노드들이 보낸 DIO 메시지의 Rank 필드를 비교한다. Rank가 높을수록 노드 S와 가깝다는 의미이므로 더 높은 rank인 A를 부모 노드로 선택한다. 그리고 부모 노드의 rank에 1을 더한 값 (= Sink까지의 홉 수)을 자신의 rank로 설정한다. OF를 활용한 부모 선택 과정은 그림 1의 ③에 해당하며, 이를 통해 노드 A를 부모로 선택한 노드 N은 부모 노드에게 자식 노드의 정보를 전달하는 DAO를 노드 A에게 전송함으로써 DODAG 가입을 완료한다.

2.2. 트리클 알고리즘

2.1절에서 설명한 것처럼 RPL에서의 DIO 브로드캐스팅은 DODAG 변경 사항에 대한 업데이트를 통해 DODAG의 유지보수를 가능하게 한다. 따라서 토폴로지가 변경되었을 경우 DODAG을 재구성함으로써 목적지 (Sink)까지의 경로를 유지할 수 있다. 그러나 만약 변경사항이 없는데도 계속 주기적으로 DIO를 브로드캐스팅 한다면 불필요한 메시지로 인한 네트워크 트래픽 및 메시지 처리에 의한 부하 측면의 오버헤드가 발생한다.

불필요한 DIO 전송에 따른 오버헤드를 줄이기 위해 RPL은 트리클 알고리즘 (Trickle Algorithm)[10]을 적용하여 DIO 메시지의 전송 간격을 제어한다. 전송 간격 (Interval) I 는 I_{MIN} 과 I_{MAX} 사이 범위에서 설정된다. 초깃값은 I_{MIN} 이며 ($I = I_{MIN}$), 실제 DIO 전송 간격은 $I/2$ 부터 I 사이의 값을 랜덤으로 선택한다. 그 후 토폴로지 변경이 확인되지 않는다면 I 를 두 배로 증가시키고 ($I = I \times 2$), 또 다시 $[I/2, I]$ 범위의 값을 랜덤으로 선택하여 DIO 전송 간격을 설정한다. 이 동작은 I 가 증가하여 I_{MAX} 에 도달할 때까지 계속되며, 그 과정에서 만약 토폴로지 변경이 발생한다면 다시 DIO 전송 간격을 MIN 값으로 초기화한다.

III. RPL 이동성 향상 관련 연구

본 장에서는 RPL에서의 이동성 향상을 위한 기존 연

구들을 소개한다. 주로 RPL을 VANET (Vehicular Ad-hoc Networks)이나 WSN (Wireless Sensor Networks)에 적용하기 위해 노드 이동 시에도 네트워크 연결성을 향상시킬 수 있는 RPL 수정 방안에 초점을 둔 연구들이다.

[4]에서는 RPL을 VANET에 적용한 RPL-VANET을 제안하였다. 차량 이동 시에도 빠르게 토폴로지를 변경할 수 있도록 표준 RPL에 적용된 트리클 알고리즘[10] 대신 고정된 타이머를 사용한다. 트리클 알고리즘은 정적 네트워크를 위해 설계되었기 때문에 VANET과 같이 차량들이 빠르게 이동하는 시나리오에는 적합하지 않다. 따라서 이 연구에서는 DIO 전송 간격을 고정하였다. 그러나 DIO 전송 간격을 짧은 시간으로 고정할 경우, 이동 속도가 느리거나 정지되어 있는 노드가 있다면 DIO 메시지에 대한 오버헤드가 증가한다는 문제가 있다.

ME-RPL[11]에서는 RPL을 WSN에 적용하고, 네트워크 내에 고정 노드와 이동 노드가 공존하는 환경을 위한 RPL 수정 방안을 제시하였다. DIO 메시지 포맷에 이동 상태를 나타내는 MO (Mobility Option) 필드를 추가하고, 이동 노드들은 MO 필드의 값을 1로 설정하여 DIO를 전송한다. 주변 노드는 부모를 선택할 때 이동 노드보다 고정 노드를 선호하는 방식으로 DODAG 유지 확률을 높인다. 또한 적응형 DIS 타이머를 적용하여 부모 변경 횟수가 많은 경우에는 네트워크의 변경이 많은 것으로 여겨 DIS 전송 간격을 반으로 줄이고, 부모 변경 횟수가 적을 경우에는 DIS 전송 간격을 두 배로 늘린다. 이 방식은 네트워크 변경 빈도가 일정한 경우에는 적용 가능하나, 일시적으로 네트워크가 안정된 상태였다가 갑자기 네트워크가 변경될 경우에는 긴 DIS 전송 간격으로 인해 DODAG 업데이트에 오랜 시간이 걸린다는 한계가 있다.

Reverse-RPL[12]에서는 이동 노드가 부모 노드를 선택한 직후에는 일정 시간동안 부모 노드의 통신범위를 벗어나지 않을 것이며, 부모 노드의 통신범위에 오래 머물렀을 경우에는 통신 범위를 벗어날 확률이 높다고 가정한다. 따라서 부모 노드는 이동 노드와 연결된 직후에는 DIO 전송 간격을 최대로 설정하고, 시간이 지날수록 계속 DIO 전송 간격을 절반으로 줄이는 Reverse Trickle Timer를 제안하였다. 하지만 노드의 이동 속도나 고정 노드의 통신 범위에 따라 빠른 시간 내에 부모의 통신 범위를 벗어날 가능성이 있다. 반대로 이동 노드가 부모 선택 후 이동을 멈추는 상황이 발생할 수도 있는데, 이

와 같은 시나리오에서 Reverse Trickle Timer는 좋은 성능을 내지 못한다.

MOD-RPL[13]에서는 ME-RPL과 Reverse-RPL의 한계를 극복하기 위해 노드의 현재 이동 속도에 따라 DIO 전송 간격을 조절하는 적응형 타이머를 제안하였다. 모든 이동 노드는 자신의 부모 노드의 범위 내 어느 위치에 있는지 인식한다. 그리고 TL (Time to Leave)을 계산하는데, TL은 부모 노드 통신 범위 가장자리로 부터의 거리를 노드 이동 속도로 나눈 값이다. 현재의 TL 값 (TL_2)보다 이전에 측정된 TL 값 (TL_1)이 크다면 부모 노드 범위의 가장자리와 가까워지고 있다는 의미이므로 DIS 전송을 통해 다른 부모를 찾는다. 이 방식은 모든 이동 노드가 자신의 이동 속도 및 위치를 인식하고 있어야만 적용할 수 있다는 한계점이 존재한다.

또한 Reverse-RPL과 MOD-RPL에서의 이동 노드는 항상 leaf node로 동작하여 다른 노드들로부터 부모 노드로 선택될 수 없다. 즉 이동 노드를 통한 DODAG의 구성이 불가능하다는 것을 뜻하는데, 이는 고정 노드만으로 DODAG 구성이 완료된 네트워크 환경에서만 제안 기법들을 적용할 수 있다는 한계가 있다.

대부분의 기존 연구에서의 네트워크 구성은 AP (Access Point) 또는 Sink, 고정 노드, 이동 노드로 이루어진다. 고정 노드와 이동 노드를 식별하기 위해 각 노드는 자신의 노드 유형 (고정 노드 또는 이동 노드)을 인지하고 있음을 가정하고, 제어메시지에 노드 타입을 포함하여 주변 노드와 공유한다. 이러한 가정은 몇 가지의 한계에 부딪힐 수 있는데, 본 논문에서는 이 한계를 두 가지로 정의한다.

- 1) 노드 유형 변경 불가: [11-15]에서의 모든 노드는 노드 자신의 이동 여부에 대해서 인지하고 있으며, 제어메시지에 노드 유형을 포함하여 주변 노드들과 공유한다. 또한 노드의 상태는 변경될 수 없기 때문에 이동 노드가 이동을 멈추고 고정되거나, 고정 노드가 이동을 시작하는 등의 동작은 수행될 수 없다.
- 2) 네트워크 내에 이동 노드가 많을수록 DODAG 생성 불가: [11-13, 16]에서는 DODAG 유지 확률을 높이기 위해 부모 선택 시 고정 노드를 우선적으로 선택하거나, 이동 노드를 leaf로 설정하여 무조건 부모 노드로 선택될 수 없도록 한다. 이러한 구성 방식은 이동 노드가 중간노드로서 DODAG에 연결될 수 없기 때문에 고정 노드 간의 연결만으로도 네트워크 전체

범위를 커버할 수 있는 환경에서만 적용할 수 있다. 따라서 이동 노드의 비율이 높을수록 DODAG에 연결되지 않고 고립되는 노드가 많아지므로 다수의 노드가 Sink까지의 경로를 확보할 수 없는 경우가 발생한다. 이 한계점은 전체 네트워크의 성능, 특히 패킷 전달률 (PDR: Packet Delivery Ratio)이 떨어지는 문제를 초래한다.

본 논문의 제안 방식에서는 상기 한계점을 극복하기 위해 모든 노드는 고정되거나 이동할 수 있도록 하여 노드 유형을 제한하지 않으며, 이동 노드도 부모 노드로 선택될 수 있도록 한다. 더불어 주변 노드들로부터 수신한 패킷을 통해 이동 정도를 분석하므로 노드의 이동 속도나 위치 등을 측정하기 위한 별도의 장치가 요구되지 않아 여러 응용에 쉽게 적용될 수 있다.

IV. DIO 전송 간격 조절 방안

본 장에서는 이동 정도에 따른 DIO 전송 간격을 조절하는 방법을 제안한다. 이동 정도를 인식하기 위해 주변 노드들로부터 수신한 패킷을 활용한다. RPL 제어메시지는 IP 계층 프로토콜인 ICMPv6의 Type 15로 전송된다. 만약 전체 수신 패킷 중 ICMPv6 패킷의 비중이 크다면 주변 노드들의 상태 정보가 자주 변경된다고 판단할 수 있다. 따라서 일정 시간 동안 수신된 전체 패킷 중 ICMPv6 패킷의 비중을 계산하여 이동 여부를 판단하고, 이동 정도에 따라 DIO 전송 간격을 동적으로 설정한다.

이동 정도에 따른 DIO 전송 간격 조절 방안을 그림 2에 나타내었다. 현재 설정된 DIO (D_t) 전송시점 t_d 와 이전 DIO (D_{t-1}) 전송시점 t_{d-1} 사이의 시간 T_1 동안 수신한 전체 패킷 수와 제어메시지 수신 횟수를 측정한다. 네트워크 내 노드들의 이동에 따라 새로운 노드가 발견되었다면 DIS, DIO, DAO의 전송을 통해 Sink까지의 경로를 업데이트할 것이다. 따라서 데이터를 포함한 전체 수신 패킷 중 제어메시지의 비중을 통해 네트워크 내 노드들의 이동 정도를 인식할 수 있다.

만약 이동 정도 (Mobility Level, ML)가 낮으면 토폴로지가 안정되어있다는 뜻이므로 DIO 전송 간격 I 를 두 배로 증가시킨다 ($I = I \times 2$). 반대로 ML이 크다면 토폴로지가 변경되었을 확률이 높으므로 I 를 절반으로

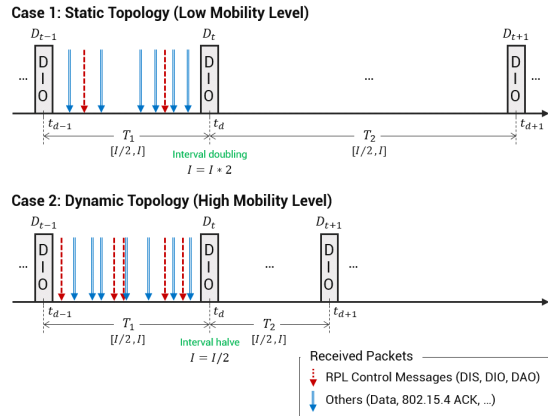


Fig. 2 DIO period adjustment depending on mobility level

줄인다 ($I = I/2$). 그 후 $[I/2, I]$ 범위의 값을 랜덤으로 선택한 T_2 후에 다음 DIO (D_{t+1})의 전송을 설정한다. ML 계산 방식은 식 (1)과 같이 표현될 수 있다.

$$ML = 1 - \frac{Packet_received}{(Control_message_received)^2} \quad (1)$$

이동 정도를 인식하기 위해 식 (1)을 활용하여 전체 수신 패킷 중 RPL 제어메시지 비중을 계산한다. 만약 ML이 0 이상일 경우, 식 (1)에서 분모인 제어 메시지 비중이 높다는 것이며, 이는 토폴로지 변경이 일어나고 있다는 의미이다. 따라서 DIO 전송 간격 (DIO_Period)을 절반으로 줄이고, ML이 0 미만일 경우에는 DIO 전송 간격을 두 배로 늘인다. 식 (2)에 DIO 전송 간격 변경 조건을 나타내었다.

$$DIO_Period = \begin{cases} DIO_Period \times 2 & (ML < 0) \\ DIO_Period / 2 & (ML \geq 0) \end{cases} \quad (2)$$

V. 성능 분석

본 장에서는 이동 정도에 따른 DIO 전송 간격 조절 방안을 트리클 알고리즘을 적용한 표준 RPL 및 고정 타이머를 적용한 RPL-VANET[4]과 비교한다. 제안 방식의 성능 평가를 위해 Contiki 3.0 기반 Cooja 시뮬레이터를 활용하였으며 다양한 이동 속도에 대해 실험한다.

5.1. 실험 환경

Table. 1 Simulation Parameters

Simulation Time [s]	300
Number of Nodes	31 (1 sink node, 30 mobile nodes)
X-size [m]	300
Y-size [m]	300
Reception Range [m]	75
Interference Range [m]	100
Data Period [s]	30
Speed [m/s]	0, 0.5, 1, 1.5, 2
Mobility Model	Random Waypoint

시뮬레이션 환경을 표 1에 나타내었다. 총 31개의 노드를 300m x 300m 공간에 설치하고, 각 노드의 전송 거리는 75m이다. 모든 노드는 CC2538[17] 인터페이스를 사용하며, 하나의 고정된 Sink 노드와 30개의 이동 노드로 구성된다. Sink 노드는 실험 공간의 정중앙에 위치하며, 이동 노드는 Random Waypoint 모델에 의해 랜덤으로 배치 및 이동한다. 실험은 0 m/s (정지상태)부터 2 m/s까지 0.5 m/s 간격으로 노드 이동 속도를 다양하게 변화시키며, 각 속도에 대해 총 5회씩 측정한다.

5.2. 실험 결과

본 실험에서는 PDR, 제어메시지 오버헤드 측면에서 제안 방식이 표준 RPL 및 RPL-VANET[4]보다 얼마나 개선되는지 확인한다. RPL-VANET에서의 DIO 전송주기는 가장 좋은 성능을 보였던 2초로 설정한다.

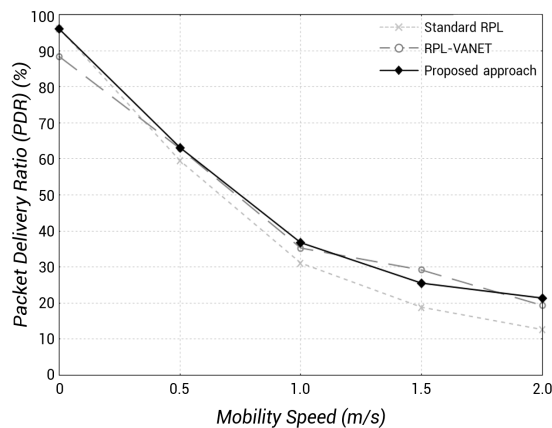


Fig. 3 Average packet delivery ratio in random network topology with different mobility speed

1) 패킷 전달률: 노드 이동에 따른 토폴로지 변경에 얼마나 빠른 속도로 적응하는지를 평가하기 위해 패킷 전달률 (PDR)을 측정한다. 각 노드가 보낸 데이터 패킷이 얼마나 Sink 노드에게 전달되었는지 측정한 후 평균한 결과를 그림 3에 나타내었다. 제안 방식은 표준 RPL 및 RPL-VANET에 비해 각각 31.03%, 3.67% 높은 PDR을 보인다. 특히 노드가 고정상태일 때 RPL-VANET의 경우 PDR이 약 6% 떨어지는데, 이를 통해 RPL-VANET이 이동 환경에서는 잘 적용될 수 있으나 고정 환경에 적용하기에는 적절하지 않은 것을 알 수 있다.

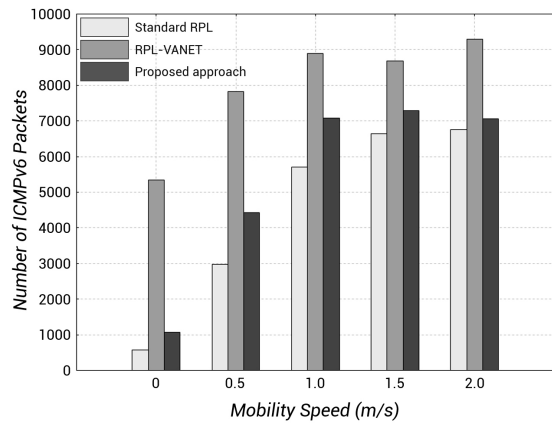


Fig. 4 Average total control traffic overhead in random network topology with different mobility speed

2) RPL 제어메시지 오버헤드: DIO 전송 간격을 조절함에 따른 제어메시지 오버헤드를 평가하기 위해 시뮬레이션 시간 동안 발생된 모든 패킷을 캡처하고 Wire-Shark[18]를 활용하여 분석한다. RPL 제어메시지는 ICMPv6 패킷을 통해 전송되므로, 데이터를 포함한 전체 수신 패킷 중 ICMPv6 패킷의 수를 계산한 결과를 그림 4에 나타내었다. 실험 결과, 제안 방식의 제어메시지 오버헤드는 표준 RPL보다 18.87% 높고, RPL-VANET에 비해 32.74% 낮다. 제안 방식은 이동 정도를 인식하여 이동 정도가 높은 경우에만 DIO 전송 간격을 조절하므로 이동 정도와 관계없이 고정된 전송 간격을 사용하는 RPL-VANET보다 오버헤드가 적은 것을 알 수 있다.

VI. 결론 및 향후연구

본 논문에서는 RPL에서의 노드들이 이동할 때에도 토폴로지 변경에 빠르게 적응하여 패킷 손실 문제를 해결하기 위한 방안을 제안하였다. 제안 방안에서는 이동성을 향상시키기 위해 모든 노드는 이웃 노드들의 이동성을 인식한다. 이동성 인식을 위해 전체 수신 패킷 중 제어메시지 비율을 측정하여 이동 알림을 정의한다. 이동 정도가 높을 경우에는 DIO 전송 간격을 절반으로 줄이고, 이동 정도가 낮을 경우에는 DIO 전송 간격을 두 배로 늘린다.

제안 방식은 3장에서 언급한 기존 연구들이 가지는 한계점들을 극복하기 위해 모든 센서 노드가 노드 유형에 구애받지 않고 자유롭게 고정 및 이동할 수 있도록 설계되었다는 점에서 기존 연구들과 차별성을 가진다. 더불어 이동성 인식을 위해 주변 노드들로부터 수신한 패킷을 통해 이동 정도를 분석하므로 노드의 이동 속도나 위치 등을 측정하기 위한 별도의 장치가 요구되지 않아 여러 응용에 쉽게 적용될 수 있을 것이다.

성능 측정 결과, 제안된 방식은 표준 RPL에 비해 패킷 전달률이 31.03%, RPL-VANET보다 제어메시지 오버헤드가 32.74% 개선됨을 확인하여 이동성 시나리오에 잘 대처함을 보여준다.

향후에는 본 논문에서 제안한 방식에 부모 선정 알고리즘 (OF)에 대한 연구를 추가하여 패킷 전달률을 더 향상시킬 수 있는 방안을 수립할 계획이다.

ACKNOWLEDGEMENT

This paper was supported by the Sabbatical Year Research Program of Koreatech in 2018.

REFERENCES

- [1] RFC 6550, *IETF Standard for RPL: IPv6 routing protocol for low-power and lossy networks*, IETF, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, 2012.
- [2] RFC 2460, *IETF Standard for Internet protocol, version 6 (IPv6) specification*, IETF, S. Deering, and Hinden R, 2017.
- [3] IEEE Std. 802.15.4, *IEEE Standard for Low-Rate Wireless Networks*, IEEE, R. Heile, 2015.
- [4] K. C. Lee, R. Sudhaakar, L. Dai, S. Addepalli, and M. Gerla, "RPL under mobility," in *2012 IEEE consumer communications and networking conference*, Las Vegas: NV, pp. 300-304, 2012.
- [5] Contiki: The Open Source Operating System for the Internet of Things [Internet]. Available: <http://contiki-os.org/>.
- [6] RFC 6719, *IETF Standard for The Minimum Rank with Hysteresis Objective Function*, IETF, O. Gnawali, and P. Levis, 2012.
- [7] M. Banh, N. Nguyen, K. H. Phung, L. Nguyen, N. H. Thanh, and K. Steenhaut, "Energy balancing RPL-based routing for Internet of Things," in *2016 IEEE Sixth International Conference on Communications and Electronics*, Quang Ninh: Vietnam, pp. 125-130, 2016.
- [8] H. Lamaazi, and N. Benamar, "OF-EC: A novel energy consumption aware objective function for RPL based on fuzzy logic," *Journal of Network and Computer Applications*, vol. 117, pp. 42-58, Sep. 2018.
- [9] C. Abreu, M. Ricardo, and P. M. Mendes, "Energy-aware routing for biomedical wireless sensor networks," *Journal of Network and Computer Applications*, vol. 40, pp. 270-278, Apr. 2014.
- [10] RFC 6206, *IETF Standard for the Trickle Algorithm*, IETF, P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, 2011.
- [11] I. E. Korbi, M. B. Brahim, C. Adjih, and L. A. Saidane, "Mobility enhanced RPL for wireless sensor networks," in *2012 Third international conference on the network of the future*, Gammarth: Tunisia, pp. 1-8, 2012.
- [12] C. Cobarzan, J. Montavont, and T. Noel, "Analysis and performance evaluation of RPL under mobility," in *2014 IEEE symposium on computers and communications*, Funchal: Portugal, pp. 1-6, 2014.
- [13] F. Gara, L. B. Saad, E. B. Hamida, B. Tourancheau, and R. B. Ayed, "An adaptive timer for RPL to handle mobility in wireless sensor networks," in *2016 International wireless communications and mobile computing conference*, Paphos: Cyprus, pp. 678-683, 2016.
- [14] S. Sanshi, and C. D. Jaidhar, "Enhanced mobility aware routing protocol for Low Power and Lossy Networks," *Wireless Networks*, vol. 25, no. 4, pp.1641-1655, May. 2019.
- [15] P. Satanasawapak, and C. Khunboa, "The improvement of node mobility in RPL to increase transmission efficiency," *International Journal of Electrical & Computer Engineering*, vol. 9, no. 5, pp. 4238-4249, Oct. 2019.
- [16] H. Kharrufa, H. Al-Kashoash, and A. H. Kemp, "A game

theoretic optimization of RPL for mobile Internet of Things applications,” *IEEE Sensors Journal*, vol. 18, no. 6, pp.2520-2530, Jan. 2018.

[17] CC2538: A Powerful System-On-Chip for 2.4-GHz IEEE 802.15.4-2006 and ZigBee Applications [Internet]. Available: <http://www.ti.com/product/CC2538/>.

[18] Wireshark: a free and open-source packet analyzer [Internet]. Available: <https://www.wireshark.org/>.



신예진(Yejin Shin)

2014년 2월 KOREATECH 전기전자통신공학부 (공학사)
2016년 2월 KOREATECH 전기전자통신공학과 (공학석사)
2016년 2월~현재 KOREATECH 전기전자통신공학과 박사과정
※관심분야: 무선통신, IoT, RPL



설순욱(Soonuk Seol)

1998년 2월 KOREATECH 정보통신공학과 (공학사)
2000년 2월 KAIST 정보통신공학과 (공학석사)
2004년 2월 KAIST 정보통신공학과 (공학박사)
1998년 3월~1999년 10월 한국전자통신연구원 위촉연구원
2004년 2월~2012년 2월 KT 종합기술원 차장
2012년 3월~현재 KOREATECH 전기전자통신공학부 부교수
※관심분야: 모바일 VoIP/IPTV, 유무선 네트워크 QoS, 프로토콜 공학