

자율적인 TSCH 네트워크에서 충돌 회피를 위한 슬롯 재할당

박혜빈*, 정진우^o

Slot Reallocation for Collision Avoidance in Autonomous TSCH Networks

Hye-bin Park*, Jinoo Joung^o

요약

IEEE 802.15.4e TSCH(Time Slotted Channel Hopping) 표준은 채널 호핑과 TDMA 방식을 사용하여 높은 안정성, 결정론적 지연과 효율적인 에너지 사용 등을 제공하나 표준 내 스케줄링 방법을 명시하지 않아, 이에 관한 연구가 활발하게 이루어졌다. Orchestra[6]는 최근 연구되고 있는 자율적인 스케줄링 방식 중 하나로 노드 간 협상 과정이 필요하지 않아서 제어 메시지 교환으로 인한 오버헤드가 적다. 하지만, Orchestra는 슬롯프레임에서 노드당 하나의 슬롯만을 할당하기 때문에, 트래픽 부하가 높은 경우 성능이 크게 저하된다. 이에 대한 해결책으로 제안된 e-TSCH-Orch[7]는 임시 슬롯 할당을 통해 일시적으로 패킷 전달 속도를 증가시키지만, 트래픽 부하가 지속적으로 높은 경우에 대한 근본적인 해결책은 되지 못한다. 따라서 본 논문에서는 자율적으로 스케줄을 업데이트하는 알고리즘을 제안하여, Orchestra의 문제점을 해결하고 e-TSCH-Orch를 개선하고자 한다. 세 가지 성능 지표를 사용하여 기존의 Orchestra와 e-TSCH-Orch와 성능 비교를 진행하였고, 그 결과 제안하는 알고리즘이 기존 기술에 비해 저지연과 고신뢰성을 제공한다는 것이 검증되었다.

Key Words : TSCH, Autonomous scheduling, Orchestra, low latency, high reliability

ABSTRACT

IEEE 802.15.4e TSCH(Time Slotted Channel Hopping) standard provides high reliability, deterministic delay, and efficient energy use by using channel hopping and TDMA. Research on scheduling method in TSCH has been proliferated since it is not explicitly specified in the standard. Orchestra^[6] is an autonomous scheduling scheme that was proposed recently. Because it doesn't require a negotiation process between nodes, there is less overhead due to the exchange of control messages. However, since a fixed slot is allocated for a node in a slotframe, its performance degrades severely with high traffic load. As a solution to this problem, e-TSCH-Orch temporarily assigns additional slots, but it doesn't solve the problem with persistent high traffic load. Therefore, in this paper, we propose an algorithm for schedule update in autonomous scheduling to solve the Orchestra's problem and improve e-TSCH-Orch's performance. Using three performance indicators, we conduct performance comparisons with Orchestra and e-TSCH-Orch, which prove that the proposed algorithm improves latency and reliability compared to existing solutions.

* 이 성과는 2018년 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2016R1D1A1B03932066).

• First Author : Sangmyung University, Department of Computer Science, reiwkei@gmail.com, 정희원

^o Corresponding Author : Sangmyung University, Department of Human Intelligence Information Engineering, jjoung@smu.ac.kr, 정희원
논문번호 : 201905-089-B-RN, Received May 25, 2019; Revised August 23, 2019; Accepted August 28, 2019

I. 서론

최근 사물인터넷 기술은 스마트 빌딩, 스마트 팩토리, 차량 네트워크 등 다양한 분야에 적용되고 있다. 특히 무선센서 네트워크 기술은 다양한 응용 서비스에 필요한 핵심 기술로 주목받고 있다¹⁾. 기존 무선센서 네트워크에서 사용되던 IEEE 802.15.4 프로토콜 표준의 경우 지속적인 관심을 받았지만, 채널 간섭 및 수용 가능한 노드 수의 한계 등의 문제점이 드러났다^{2,3)}. 또한 이 표준은 응용들에서 요구하는 지연과 신뢰성들의 성능을 제공하지 못한다^{2,3)}. 이러한 문제를 해결하기 위해 IEEE 802.15.4e에서는 WirelessHART나 ISA100.11a 같은 산업표준에서 사용되던 TSCH(Time Slotted Channel Hopping) 모드가 표준에 채택되었다^{1,4)}. TSCH는 TDMA와 채널 호핑을 통해 안전성이 높고 결정론적인 지연시간을 제공하며, 적은 에너지를 소비한다⁵⁾. TSCH 표준에서는 스케줄링 방법을 명시하고 있지 않기 때문에 링크 스케줄링에 대한 연구들이 활발히 진행되고 있다⁴⁾.

기존에 제시된 스케줄링 방법은 중앙집중식, 계층식, 분배식의 세 가지로 나눌 수 있다⁴⁾. 중앙집중식은 싱크 노드가 전체 네트워크에 대한 정보를 파악하여 각 무선 링크에 대한 스케줄링을 진행한다⁴⁾. 계층식은 싱크 노드로 향하는 최적의 경로를 라우팅 프로토콜을 이용해 설정한 후 각 노드에서 다음 홉까지의 링크에 재귀적으로 대역폭을 할당하는 방법으로 스케줄링이 이루어진다⁴⁾. 분배식은 노드가 최대 홉이나 거리 내에 존재하는 이웃 노드와만 통신하여 자신이 사용할 채널과 타임 슬롯을 결정한다⁴⁾. 이 세 방식은 노드 간 제어 메시지 교환을 통해 스케줄링이 이루어지므로 오버헤드가 크다는 문제를 안고 있다.

이에 비해 비교적 최근에 발표된 Orchestra와 같은 자율적인(autonomous) 스케줄링 방식은 노드가 자율적으로 타임 슬롯을 할당하기 때문에, 제어 메시지로 인한 오버헤드가 적고 별도의 핸드셰이크 과정이 필요 없다⁶⁾. 하지만 Orchestra는 노드 당 하나의 슬롯만을 할당하여 해당 노드의 모든 트래픽을 처리하기 때문에, 트래픽 부하가 높은 경우 성능이 저하된다. 이에 대한 해결책으로 나온 알고리즘들도 트래픽 부하가 지속적으로 높은 경우에 대한 근본적인 해결책은 되지 못한다.

본 논문에서는 자율적인 스케줄링을 기반으로 중앙 제어기 없이 지연, 고신뢰성을 제공하는 스케줄 업데이트 알고리즘을 제안한다. 본 논문은 다음과 같이 구성되었다. 2장에서는 자율적인 스케줄링 방식에 관한

기존 연구들을 살펴보고 문제점을 분석한다. 3장에서는 본 논문에서 제안하는 알고리즘을 설명하고, 4장에서는 알고리즘의 비교 성능 평가를 진행한다. 마지막으로 5장에서 본 논문의 결론과 추후 개선 방향에 대해 논의한다.

II. 관련 연구

TSCH에서 사용되는 채널 호핑은 저대역 간섭이나 지속적인 다중 경로 페이딩을 방지할 수 있어 산업현장에서 잘 동작한다는 것이 증명되었다^{10,12)}. 하지만 TDMA 방식처럼 채널 호핑은 주어지는 슬롯 스케줄에 의존하기 때문에, “좋은 스케줄”을 생성하는 것이 더 중요하다. 따라서 본 논문에서는 채널 호핑에 대한 자세한 내용은 다루지 않는다^{4,10)}.

Orchestra는 TSCH 네트워크를 위한 자율적인 스케줄링 방법이다⁶⁾. 기존 솔루션들은 중앙집중식이나 노드 간의 협상, 신호 전달, 멀티 홉 경로 예약 등으로 무선 노드의 전송 슬롯을 스케줄한다^{6,7)}. 반면 Orchestra는 각 노드들이 라우팅 계층 정보와 스케줄링 규칙을 이용해 자기 스스로 스케줄을 생성하고, 국지적으로 이를 유지한다^{6,7)}. Orchestra는 트래픽 종류에 따라 슬롯 프레임들 각각 생성하며, 각 슬롯프레임의 길이는 다를 수 있으나 소수(prime number)여야 한다⁶⁾. Orchestra에는 4가지 종류의 슬롯이 존재하는데, 본 논문에서는 RBS (Receiver-based Shared Orchestra Slots)를 사용하여 스케줄을 생성한다. RBS는 두 노드 간 통신을 위한 슬롯을 할당할 때, 수신 노드의 특성(예: MAC 주소, 식별자 등)을 이용하여 시간과 채널 오프셋을 계산한다⁶⁾. 또한 스케줄링 결과로 모든 노드들은 하나의 Rx(Reception) 슬롯과 이웃 노드마다 하나의 Tx(Transmission) 슬롯을 가지게 된다⁶⁾. 다음 그림 1은 예시 토폴로지에서 RBS를 사용해 계산한 스케줄을 보여준다.

Orchestra는 6TiSCH(IPv6 over the TSCH mode of IEEE 802.15.4e) 네트워크에서도 성능 검증이 이루어

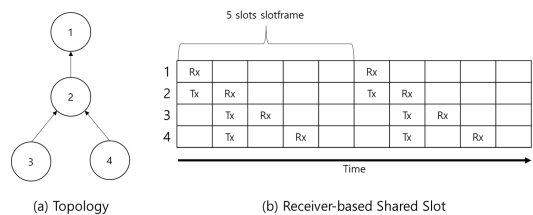


그림 1. RBS 슬롯으로 만들어진 스케줄
Fig. 1. The schedule made of RBS slots.

졌는데, 해당 WG에서 제안된 SF0나 SF1이 아니라 Orchestra를 사용해 TSCH 스케줄을 생성하였다⁸⁾. 시뮬레이션을 통해서 트래픽 밀도가 평균일 때, PDR(Packet Delivery Rate), 에너지 효율성, 지연의 관점에서 안정적인 성능을 보여주었다⁸⁾.

하지만 Orchestra는 트래픽의 양을 고려하지 않고 슬롯을 할당하기 때문에, 트래픽 부하가 증가하면 급격한 성능 저하가 나타난다^{7,9,10)}. [9]는 Orchestra가 전송할 패킷이 없더라도 노드가 슬롯을 계속 점유하고 있어 링크 이용률이 낮다는 문제를 제시했다. 이에 대한 해결책으로 Orchestra의 송신자 기반 방식을 확장하고 상태 머신을 사용해 노드의 필요에 따라 슬롯을 할당하는 적응적인 MAC 스케줄링 방법을 고안하였다⁹⁾. 또한 제안한 방법을 적용하여 자율적인 헬스케어 모니터링을 위한 프레임워크를 구축하였다⁹⁾.

PAAS(Parameterized Adaptive and Autonomous Scheduling)¹⁰⁾는 측정된 트래픽 강도를 적용하여 Orchestra를 개선한 알고리즘이다. 트래픽 강도 p 가 주어졌을 때, discrete time Markov chain 분석을 통해 계산된 정확한 충돌 확률에서 n 을 도출한다¹⁰⁾. 이때, n 은 한 슬롯에 할당될 수 있는 최대 송신기 수와 대응되며, n 값에 따라 Tx 슬롯과 Rx 슬롯이 할당된다¹⁰⁾. PAAS는 스케줄링에 트래픽을 반영하여 PRR(Packet Reception Rate)과 지연 등 성능을 개선시켰다¹⁰⁾. 하지만 슬롯 스케줄 정보가 담긴 DIO(DODAG Information Object) 교환을 통해 노드들의 스케줄링이 진행되기 때문에¹⁰⁾ 제어 메시지 교환으로 인한 오버헤드가 크게 발생한다.

한편, 트래픽 부하는 두 가지 방향으로 정의할 수 있다. 첫 번째는 각 노드와 그 노드가 속한 서브 노드가 생성하는 트래픽의 합이고, 두 번째는 각 노드의 큐(버퍼)에서 처리를 기다리는 패킷의 수이다⁷⁾. e-TSCH-Orch⁷⁾은 후자를 적용한다. 트래픽 부하를 대기 큐(버퍼)에서 전송되기 위해 남아 있는 패킷의 수(N_{rp})로 정의하고, 매 Tx 슬롯마다 값을 계산한다⁷⁾. 큐에 저장된 패킷의 수를 줄이기 위해서, 기존 Tx 슬롯 뒤에 연달아 $N_{rp} - 1$ 개 Tx 슬롯을 할당한다⁷⁾. 또한 전송하는 패킷에 이에 대한 정보를 피기배킹하여 수신 노드에게 통보한 후 추가한 Tx 슬롯을 이용해 큐(버퍼)에 있는 패킷을 처리한다⁷⁾. e-TSCH-Orch는 큐(버퍼) 오버 플로우를 회피하는 방법으로 PDR을 향상시켰다⁷⁾. 하지만 수신 노드의 스케줄에 상관없이 송신 노드가 Tx 슬롯을 추가하기 때문에, 패킷 충돌이 일어날 수 있다.

III. 제안하는 스케줄링 알고리즘

Orchestra RBS를 사용하는 경우, 모든 노드는 하나의 Rx 슬롯만을 가지게 된다. 자식 노드가 둘 이상이면, 해당 슬롯을 사용하기 위해서 자식 노드 간 경쟁이 발생하게 되고, 이는 패킷 손실로 이어지게 된다. 부모가 변경되지 않는 이상, 슬롯프레임 내 자식노드의 Rx 슬롯과 Tx 슬롯은 고정된다. 따라서 자식 노드는 패킷을 송신하기 위해서 부모 노드를 공유하는 다른 자식 노드와 매번 경쟁하게 된다. 필연적으로 패킷 충돌로 인한 성능 저하가 발생한다. 따라서 자식 노드가 둘 이상일 때 Orchestra를 사용하는 경우 랜덤 액세스의 경쟁 기반 매체 접근 방식이 된다. 경쟁 기반 방식은 충돌을 내재하고 있으므로 부하가 커지면 성능이 저하될 수밖에 없다. 이러한 Orchestra의 성능 저하를 감소시키기 위해, e-TSCH-Orch는 임시 슬롯을 할당하여 PDR을 증가시켰다. 하지만 Orchestra의 근본적인 문제를 해결한 것이 아니므로, 특정 노드의 트래픽이 증가하는 경우에 대한 일시적인 해결책은 될 수 있으나 네트워크 내 부하가 전체적으로 높은 경우에는 실효성이 적다.

Orchestra를 사용할 때 충돌이 발생하는 근본적인 이유는 스케줄이 고정되어 있기 때문이다. 따라서 노드의 스케줄을 수정할 수 있다면 패킷 충돌 확률을 어느 정도 낮출 수 있다. 제어 메시지 교환을 최소화하기 위해서, 처음 스케줄을 생성할 때는 수신 노드, 즉 부모 노드의 식별자로 Rx 슬롯을 자동 할당하고, 이후 ACK를 통해 부모 노드에게 새로운 Rx 슬롯을 할당받는다. 이렇게 자식 노드마다 다른 슬롯을 사용해 패킷을 보내게 되므로, 충돌이 발생할 확률이 낮아지고 이로 인해 지연시간 등의 성능도 개선된다.

본 논문이 제안하는 알고리즘을 SRCA(Slot Reallocation for Collision Avoidance)로 부르도록 한다. SRCA의 핵심 아이디어는 전송에 성공한 자식 노드를 선제적으로 다른 슬롯으로 이동시켜서 다른 자식 노드와의 충돌을 방지한다는 것이다. SRCA의 세부 동작은 다음과 같다. 먼저 모든 노드들은 자신의 ID와 부모 노드의 ID로 Tx 슬롯과 Rx 슬롯을 계산하여 자동으로 스케줄을 생성하고, REQUEST 모드를 시작한다. 자식 노드는 자신의 Tx 슬롯에서 부모 노드에게 패킷(TxPkt)을 보내는데, 이때 REQ 헤더 필드 값을 true로 설정한다. 자식 노드가 ACK를 수신하면, ACK의 TX 필드 값으로 Tx 슬롯을 갱신하고 NORMAL 모드로 전환한다. NORMAL 모드에서는 변경된 Tx 슬롯을 사용하여 패킷을 전송하게 된다. 만약 TIMEOUT 동

안 ACK가 도착하지 않을 경우, TSCH 재전송 절차^[5]를 진행한다. 한편, 자식 노드에게 패킷(RxPkt)을 받은 부모 노드는 REQ 헤더 필드를 확인하여, 값이 true일 경우 자식 노드에게 새로운 슬롯을 할당한다. 스케줄 내 빈 슬롯이 존재하지 않을 경우 가장 적은 수의 자식 노드가 할당된 슬롯을 택한다. 이를 TX 헤더 필드 값으로 설정하고, 자식 노드에게 ACK를 보낸다. 아래 표 1은 SRCA 알고리즘을 정리한 내용이다.

그림 1의 토폴로지에 SRCA를 적용하면 그림 2와 같이 동작한다. 이때 슬롯프레임의 길이는 11이다.

그림 1의 모든 노드는 자신의 ID와 부모 노드의 ID로 스케줄을 자동 생성하고 REQUEST 모드를 시작한다. 단 루트 노드인 노드 1의 경우 스케줄 업데이트가 불필요하므로 NORMAL 모드로 시작한다. 노드 2는 슬롯 1에 부모 노드인 노드 1에게 슬롯 재할당을 요청한다. 경쟁 노드가 없으므로 노드 1은 ACK를 통해 노드 2에게 새로 슬롯 3을 할당해준다. 슬롯 3에서 노드 3과 노드 4가 노드 2에게 슬롯 재할당을 요청하나 패킷 충돌이 발생하여 두 노드는 back-off 후 재전송 절차를 진행한다. 2번째 슬롯프레임에서 노드 3이 노드 2

Slotframe 1	2->1	3->2 4->2									
Slotframe 2			2->1								
Slotframe 3	3->2	4->2	2->1								
Slotframe 4	3->2		2->1	4->2							
	0	1	2	3	4	5	6	7	8	9	10

그림 2. 그림 1에서 SRCA의 동작 과정
Fig. 2. SRCA behavior on the topology of Fig. 1.

에게 슬롯 재할당을 요청하고 이때 노드 4가 back-off 중이므로 충돌이 발생하지 않는다. 노드 2는 ACK를 통해 노드 3에게 슬롯 1을 할당한다. 3번째 슬롯프레임에서 노드 4가 노드 2에게 슬롯 재할당을 요청하고 경쟁 노드가 없으므로 노드 2는 슬롯 5를 노드 4에게 할당해준다. 마지막 4번째 슬롯프레임에서는 모든 노드의 슬롯 재할당이 완료되어 충돌이 발생하지 않게 된다.

IV. 성능 비교

4.1 시뮬레이션 환경

본 논문에서 제안하는 스케줄링 알고리즘을 검증하기 위해서 Orchestra와 e-TSCH-Orch와 성능 비교를 진행했다. 시뮬레이션 환경은 e-TSCH-Orch에서 사용된 설정을 참조하였다^[7]. 시뮬레이션은 아래 그림 3과 같이 균일한 간격으로 노드가 배치된 그리드 토폴로지에서 이루어졌다.

루트 노드는 (0,0)에 위치하고 ID는 1로 고정되며 다른 노드는 사용하지 않는다. 루트 노드를 제외한 노드는 모두 데이터를 생성하며 루트 노드와 가까운 순으로 1씩 증가된 ID가 할당된다.

성능 평가는 2가지 시나리오에서 이루어진다. 첫 번째 시나리오에서는 노드 수와 성능 간의 상관관계를 살펴본다. 이를 위해 데이터 생성율은 초당 3 패킷으로 고정시킨 상태에서 노드의 수를 3x3~10x10으로 변화시킨다. 두 번째 시나리오에서는 Markov 트래픽 모델

표 1. SRCA 알고리즘
Table 1. SRCA algorithm.

Slot Reallocation for Collision Avoidance:
Tx slot ← parent ID % slotframe length Rx slot ← node ID % slotframe length Mode ← REQUEST
Transmission: packet TxPkt // a packet sending to parent node bool REQ // header field in TxPkt if Mode = REQUEST REQ ← true endif transmit TxPkt if ACK is arrived and Mode = REQUEST Tx slot ← TX value in ACK Mode ← NORMAL else if TIMEOUT TSCH retransmission procedure endif
ACK: packet RxPkt // a received packet integer TX // header field in ACK if REQ value in RxPkt = true Tx ← the slot with the fewest chil nodes assigned endif transmit ACK to sender

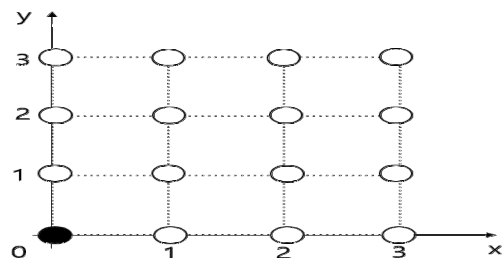


그림 3. 성능 평가에 사용된 토폴로지
Fig. 3. The topology used in performance evaluation.

[13]을 적용하여 버스트 트래픽이 발생하는 환경에서 성능을 관찰한다. 이번 시뮬레이션에서 사용된 Markov 모델은 초당 1 패킷을 생성하는 normal 상태와 초당 6 패킷을 생성하는 burst 상태가 존재한다. 상태 전이 확률은 아래 표 2와 같다.

성능 평가 결과는 일반성을 위해서 파라미터를 고정시킨 상태로 여러 번 반복 수행한 후 그 평균값을 구했다. 아래 표 3은 성능 평가에 사용된 파라미터 값을 정리한 내용이다.

본 성능 평가에서는 세 가지 성능 지표를 사용하여 비교가 이루어졌다. 첫째는 latency로, 송신 노드가 패킷을 생성한 후 수신 노드가 보낸 ACK를 정상적으로 수신할 때까지 걸린 슬롯 수로 정의한다. 둘째로는 링크 품질을 측정하기 위한 성능 지표인 ETX(Expected Transmission Count)를 사용한다^[11]. ETX는 한 홉 데이터를 전송할 때 오류 없이 목적지 노드가 패킷을 수신받기 위해 필요한 전송 시도 횟수로 정의된다^[11]. ETX는 링크의 신뢰도를 수치로 표현해주며, 신뢰도가 높을수록 1에 수렴하고, 낮을수록 값이 커진다^[11]. 본 성능평가에서는 $ETX=1/(1-PFR)$ 로 정의했으며, 이때 PFR(Packet Failure Rate)은 전체 전송 횟수 중 전송 실패 비율을 의미한다. 마지막 성능 지표는 PLR(Packet Loss Rate)인데, 버퍼에 유입된 패킷 중 최대 재전송 횟수 초과나 버퍼 오버 플로우로 인해 손실된 패킷의 비율로 정의된다.

표 2. 전이 확률 행렬
Table 2. Transition probability matrix.

	Normal	Burst
Normal	0.9	0.1
Burst	0.9	0.1

표 3. 시뮬레이션에서 사용한 파라미터 값
Table 3. Parameter values used in the simulation.

Parameter	Values
Length of slotframe ^[6,7]	11
MAC_MIN_BE/ MAC_MAX_BE ^[2,3,5]	3/5
MAC_MAX_FRAME_RETRIES[5]	7
Simulation time	50 min
Number of nodes	9 ~ 100

4.2 성능 평가 결과

모든 성능 평가 결과는 알고리즘 별로 Orchestra는

붉은색, e-Tsch-Orch는 초록색, SRCA는 파랑색으로 표현되었다. 또한 다른 지표와는 달리 latency의 경우 노드 수에 따른 값의 변동폭이 크므로 로그 스케일을 적용하여 반대수 그래프로 표현하였다.

첫 번째 시나리오는 노드 수가 스케줄링 알고리즘 성능에 어떤 영향을 주는 지 알아보기 위한 시뮬레이션이다. 먼저 latency로 세 알고리즘을 비교한 결과는 그림 4와 같다. Orchestra의 경우 전반적으로 다른 두 알고리즘에 비해 latency가 높아서 노드 수가 9일 때도 600 슬롯의 latency를 가진다. 또한 노드 수가 증가함에 따라 latency가 급격하게 증가하여 노드 수가 100인 경우 2336 슬롯의 latency를 가지게 된다. e-TSCH-Orch의 경우 Orchestra에 비해 노드 수에 따라 latency가 점진적으로 상승하며 노드 수가 9와 100일 때 각각 279와 872 슬롯의 latency를 가진다. 반면 SRCA는 모든 경우에 Orchestra와 e-TSCH-Orch 보다 낮은 latency를 가진다. 또한 노드 수에 큰 영향을 받지 않아 80~160 슬롯 사이의 균일한 latency를 가진다.

다음 그림 5는 ETX 측면에서 성능을 비교한 결과

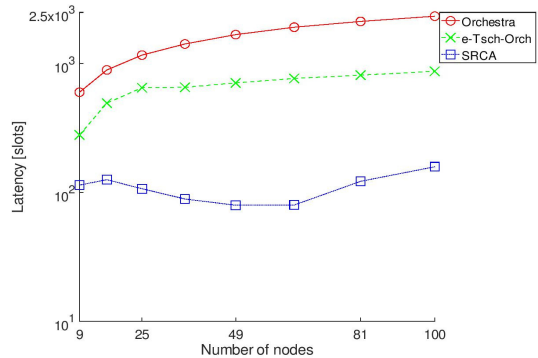


그림 4. 노드 수에 따른 latency
Fig. 4. Latency according to number of nodes.

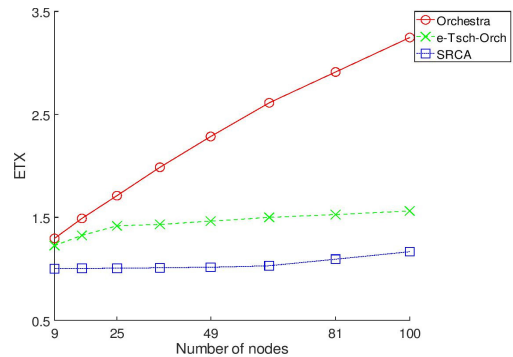


그림 5. 노드 수에 따른 ETX
Fig. 5. ETX according to number of nodes.

이다. Orchestra의 경우 노드 수에 민감하게 반응하여 노드 수 증가에 따른 ETX 증가폭이 커서 최소 1.3에서 최대 3.2의 값을 가진다. 반면 e-TSCH-Orch와 SRCA는 노드 수가 ETX에 큰 영향을 끼치지 않는다. 하지만 e-TSCH-Orch의 경우 노드 수가 증가함에 따라 ETX가 1.2에서 1.6으로 소폭 상승한다. 반면 SRCA는 노드 수가 81 미만일 경우 거의 1에 수렴하며 81 이상인 경우도 e-TSCH-Orch 보다 낮은 ETX 값을 가진다.

그림 6은 노드 수에 따른 PLR 변화를 나타낸 그림으로 세 알고리즘 모두 노드 수와 PLR이 비례하는 양상을 보인다. Orchestra와 e-TSCH-Orch의 경우 전반적으로 높은 PLR을 가지며 최소 절반 이상의 패킷이 손실된다. 반면 SRCA는 노드 수가 64이하 일 때 평균 0.3 정도의 PLR을 가진다. 노드 수가 그 이상으로 증가하는 경우 PLR이 급증하여 최대 53% 정도의 패킷이 손실된다. 하지만 모든 노드 수에 대하여 항상 다른 알고리즘에 비해 적은 PLR을 가진다.

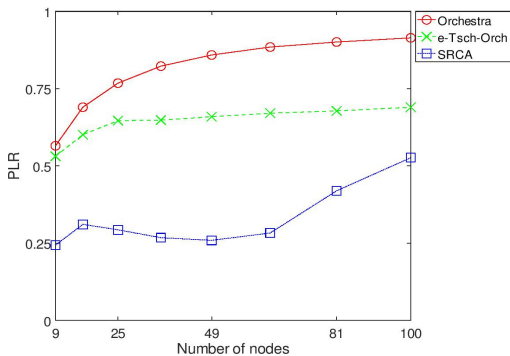


그림 6. 노드 수에 따른 PLR
Fig. 6. PLR according to number of nodes.

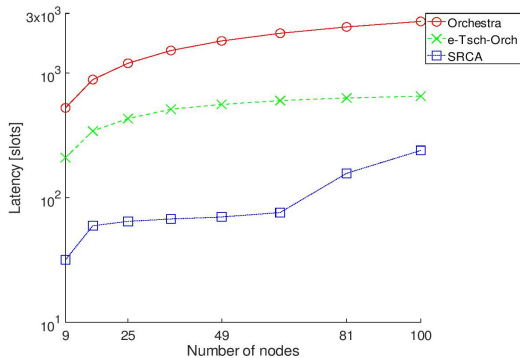


그림 7. 버스트 트래픽에서 노드 수에 따른 latency
Fig. 7. Latency according to number of nodes under burst traffic.

표 4. 첫 번째 시나리오에서 시뮬레이션 결과
Table 4. Simulation results in scenario 1.

Algorithm	Latency		ETX		PLR	
	min	max	min	max	min	max
Orchestra	600	2337	1.30	3.25	0.56	0.91
e-TSCH-Orch	280	873	1.23	1.56	0.53	0.69
SRCA	80	159	1.00	1.17	0.24	0.53

첫 번째 시나리오에 대한 성능 평가를 정리한 결과는 표 4와 같다.

두 번째 시나리오는 버스트 트래픽과 성능 간의 상관관계를 살펴보기 위한 환경으로 노드 수에 따라 성능을 측정하였다. 먼저 latency로 알고리즘의 성능을 평가한 결과는 그림 7과 같다.

Orchestra와 e-TSCH-Orch는 첫 번째와 시나리오와 비슷한 수준으로 latency가 증가한다. 하지만 Orchestra는 노드 수가 9일 때는 첫 번째 시나리오보다 낮은 524 슬롯으로, 노드 수가 100일 때는 2590 슬롯으로 latency가 높아졌다. 반면 e-TSCH-Orch는 둘 다 낮아져서 각 경우에 대하여 209 슬롯과 651 슬롯의 latency를 가진다. 한편 SRCA는 전반적으로 Orchestra와 e-TSCH-Orch 보다 latency가 낮다. 그러나 버스트 트래픽의 영향으로 노드 수가 64 이하인 경우 latency가 점진적으로 증가하지만, 노드 수가 그 보다 커지면 latency가 급증한다. SRCA는 노드 수가 9, 64와 100인 경우에 대하여 32 슬롯, 76 슬롯과 239 슬롯의 latency를 가진다.

아래 그림 8은 버스트 트래픽이 발생하는 경우에 ETX 측면에서 성능을 비교한 결과 그래프다. 첫 번째 시나리오의 결과와 매우 유사하여 ETX 결과가 소수점 자리 수준에서 차이를 보인다. 단 SRCA의 경우 노드

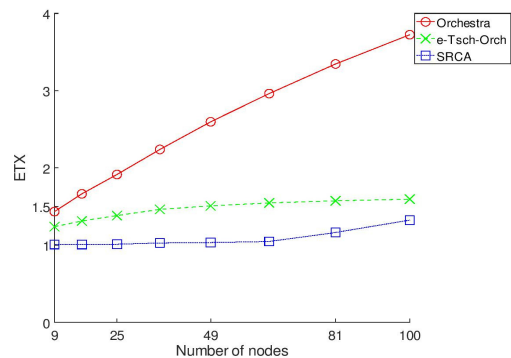


그림 8. 버스트 트래픽에서 노드 수에 따른 ETX
Fig. 8. ETX according to number of nodes under burst traffic.

V. 결 론

수가 81 이상인 경우 앞선 시나리오에서 보다 큰 폭으로 ETX가 증가한다. 노드 수가 81인 경우 1.16, 노드 수가 100인 경우 1.32의 ETX 값을 가진다.

마지막으로 아래 그림 9는 노드 수에 따른 PLR을 관찰한 결과로 버스트 트래픽의 영향으로 다른 지표에 비해 전 시나리오 결과와 큰 차이를 보인다.

Orchestra의 경우 노드 수에 더욱 민감하게 영향을 받게 되어 노드 수가 9와 100인 경우 PLR 값의 차이가 0.6 이상이 되었다. 반면 e-TSCH-Orch는 일시적으로 트래픽이 증가하는 환경을 상정한 솔루션이므로, 이번 시나리오에서 오히려 이점을 얻는다. 그 결과 노드 수가 100인 경우에도 31% 정도의 패킷만이 손실된다. 한편 SRCA는 버스트 트래픽의 영향으로 노드 수에 따른 PLR 증가폭이 커진다. 노드 수가 9인 경우와 노드 수가 64인 경우 PLR이 각각 0.03과 0.16으로 노드 수에 따라 PLR이 점진적으로 증가하지만 노드 수가 그 이상인 경우 PLR이 배로 증가하여 노드 수가 100인 경우 40% 패킷이 손실된다.

아래 표 4는 두 번째 시나리오에서 성능을 평가한 결과를 정리한 내용이다.

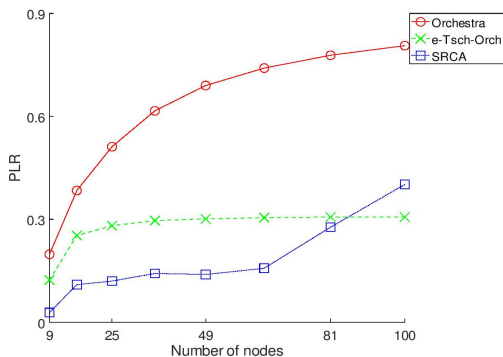


그림 9. 버스트 트래픽에서 노드 수에 따른 PLR
Fig. 9. PLR according to number of nodes under burst traffic.

표 5. 두 번째 시나리오에서 시뮬레이션 결과
Table 5. Simulation results in scenario 2.

Algorithm	Latency		ETX		PLR	
	min	max	min	max	min	max
Orchestra	524	2590	1.43	3.72	0.20	0.81
e-TSCH-Orch	209	651	1.24	1.60	0.13	0.31
SRCA	32	239	1.01	1.32	0.03	0.40

IEEE 802.15.4e TSCH는 높은 안전성, 결정론적인 지연과 효율적인 에너지 사용 등 여러 장점이 있지만, 표준에 스케줄링 방법을 명시하지 않아 이에 관한 연구가 활발히 이루어졌다. 기존 연구 방향은 노드 간 제어 메시지 교환을 요구하므로, 오버헤드가 크다는 문제를 내포하고 있다. 이에 반해 최근 연구되고 있는 자율적인 스케줄링 방식 중 하나인 Orchestra는 제어 메시지 없이 노드가 자율적으로 슬롯을 할당하기 때문에, 별도의 제어 메시지 교환 과정과 이로 인한 오버헤드가 적다. 하지만, Orchestra는 슬롯프레임에서 노드 당 하나의 슬롯만을 할당하기 때문에, 트래픽 부하가 높으면 성능이 크게 저하된다. 이에 대한 해결책으로 나온 e-TSCH-Orch는 버퍼 오버플로우 방지를 위해 임시 슬롯을 할당함으로써, 일시적으로 트래픽이 증가하는 경우에는 성능이 향상되었다. 하지만 Orchestra의 근본 원인을 해결한 것이 아니므로, e-TSCH-Orch는 트래픽 부하가 지속적으로 높은 경우에 대한 해결책은 될 수 없다.

따라서 본 논문에서는 Orchestra의 충돌 원인을 파악하고 e-TSCH-Orch를 개선한 스케줄 업데이트 알고리즘인 SRCA를 제안하였다. SRCA의 핵심 아이디어는 전송에 성공한 지식 노드를 선제적으로 다른 슬롯으로 이동시켜서 다른 지식 노드와의 충돌을 방지한다는 것이다. SRCA를 검증하기 위해서, Orchestra와 e-TSCH-Orch 스케줄링과 성능 비교하였다. 시뮬레이션은 초당 일정한 패킷이 발생하는 환경과 버스트 트래픽이 발생하는 환경에서 이루어졌다. Latency, ETX와 PLR을 성능 지표로 사용하여 알고리즘을 분석한 결과, 첫 번째 시나리오에서는 SRCA가 두 알고리즘에 비해 낮은 지연과 높은 신뢰성을 보장한다는 것이 검증되었다. SRCA는 다른 방식에 비해 최소 3배 이상 latency가 낮았으며, ETX의 경우 거의 1에 수렴함으로써 높은 링크 신뢰도를 보여주었다. PLR의 경우 노드가 많은 경우에도 최대 0.3을 넘지 않았다. 두 번째 시나리오에서는 버스트 트래픽의 영향으로 세 성능 지표 모두 변화폭이 커졌으나 전반적으로 SRCA가 더 좋은 성능을 보여주었다. e-TSCH-Orch가 일시적인 트래픽이 발생하는 환경을 가정한 알고리즘이기 때문에, 노드 수가 많은 경우 SRCA가 e-TSCH-Orch 보다 높은 패킷 손실율을 보여준다. 하지만 SRCA는 특수 환경을 가정하고 설계한 알고리즘이 아니므로 모든 환경에서 범용으로 사용할 수 있다.

이번 성능 평가는 규모가 작은 환경에서 이루어졌

다. 따라서 차후 연구를 통해 시뮬레이션 환경을 더욱 확장할 필요가 있다. 시뮬레이션 시나리오를 다양화하고, 대규모 환경에서 여러 성능 지표를 통해 프로토콜을 검증함으로써 연구의 완성도를 높인다. 기존 Orchestra는 한 슬롯프레임에서 하나의 슬롯만을 할당 받을 수 있으므로, 트래픽 변동에 따라 여러 슬롯을 할당받도록 SRCA를 개선한다.

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] M.-R. Park and D. Kim, "Scheduling scheme and performance analysis of IEEE802.15.4e TSCH," *The J. Inst. Internet, Broadcasting and Commun.*, vol. 17, no. 5, pp. 43-49, 2017.
- [3] D. Kim and M.-H. Youn, "Performance evaluation on the power consumption of IEEE802.15.4e TSCH," *The J. Inst. Internet, Broadcasting and Commun.*, vol. 18, no. 1, pp. 37-41, 2018.
- [4] 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.
- [5] IEEE Std 802.15.4™-2015: *IEEE Standard for Low-Rate Wireless Personal Area Networks (WPANs)*, Apr. 2016.
- [6] S. Duquenooy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust mesh networks through autonomously scheduled TSCH," in *Proc. SenSys '15*, pp. 337-350, Seoul, Korea, Nov. 2015.
- [7] S. Rekik, N. Baccour, M. Jmaiel, K. Drira, and L. A. Grieco, "Autonomous and traffic-aware scheduling for TSCH networks," *Computer Networks*, vol. 135, pp. 201-212, 2018.
- [8] C. M. G. Algora, V. A. Reguera, and K. Steenhaut, *Experimental evaluation of IEEE 802.15.4 TSCH on a 6TiSCH network(2017)*, Retrieved Apr. 30, 2019, from https://cris.vub.be/files/37446842/TE_36.pdf.
- [9] M. V. Ngo, Q. D. La, D. Leong, and T. Q. Quek, "User behavior driven MAC scheduling for body sensor networks," in *Proc. 2017 IEEE 19th Int. Conf. e-Health Netw., Appl. and Serv. (Healthcom)*, pp. 1-6, Dalian, China, Oct. 2017.
- [10] J. Jung, D. Kim, J. Hong, J. Kang, and Y. Yi, "Parameterized slot scheduling for adaptive and autonomous TSCH networks," in *Proc. IEEE INFOCOM 2018*, pp. 76-81, Honolulu, HI, USA, Apr. 2018.
- [11] J. Park, H. Kim, and S. Bahk, "RSSI-based link quality estimation in low-power and lossy networks," in *Proc. KICS Winter Conf.*, pp. 1-2, Pyeongchang, Korea, Jan. 2019.
- [12] T. Watteyne, A. Mehta, and K. Pister, "Reliability through frequency diversity: Why channel hopping makes sense," in *Proc. PE-WASUN'09*, pp. 116-123, Tenerife, Canary Islands, Spain, Nov. 2009.
- [13] V. S. Frost and B. Melamed, "Traffic modeling for telecommunications networks," *IEEE Commun. Mag.*, vol. 32, no. 3, pp. 70-81, 1994.

박 혜 빈 (Hye-bin Park)



2015년 8월 : 상명대학교 컴퓨터과학과 졸업
2017년 2월 : 상명대학교 컴퓨터과학과 석사
2017년 3월~현재 : 상명대학교 컴퓨터과학과 박사과정
<관심분야> 인공지능, 음성인식, 유무선 네트워크

[ORCID:0000-0002-5324-2906]

정 진 우 (Jinoo Joung)



1992년 2월 : KAIST 전기공학
과 졸업
1994년 8월 : NYU 전기전자공
학과 Master
1997년 8월 : NYU 전기전자공
학과 Ph.D.
1997년 10월~2005년 2월 : 삼성
전자 종합기술원 수석연구원

2005년 2월~현재 : 상명대학교 휴먼지능정보공학과
교수

<관심분야> 유무선통신, 네트워크, 임베디드 시스템

[ORCID:0000-0003-3053-9691]