

IoT 연결성 향상: 다중 프로토콜 네트워크에서 저속 저전력 프로토콜로서의 Thread 네트워크 구현

박성현, 주재한, 황다빈, 김석찬*

부산대학교, 부산대학교, 부산대학교, *부산대학교

shpark24@pusan.ac.kr, jhjoo2018@pusan.ac.kr, dabinsor@pusan.ac.kr, *sckim@pusan.ac.kr

Enhancing IoT connectivity: Implementation of Thread network as a Low-rate, Low-Power Protocol in Multiprotocol Networks

Seonghyeon Park, Jaehan Joo, Dabin Hwang, Sukchan Kim*

Pusan National Univ., Pusan National Univ., Pusan National Univ., *Pusan National Univ.

요약

본 스마트 홈 IoT 프로토콜 스택 Matter의 Low-rate, Low-power 프로토콜인 Thread의 기기 간 연결 구조와 이후 Matter 생태계에서 상호연결성을 높일 수 있도록 네트워크를 형성하는 방법을 소개한다. 보더 라우터라는 특수한 디바이스를 이용하여 Thread 네트워크와 IP 통신을 가능케 하고 차후 다른 IoT 프로토콜과도 통신할 수 있게 할 수 있는 Co-Processor Communication Daemon(CPCd)을 추가한 시뮬레이션 환경을 구성하여 데이터 및 신호를 주고받았다. 이후 CPCd를 활용하면 Matter 스택 이외의 프로토콜들을 Matter 생태계에 포함시킬 수 있을 것으로 예측된다.

I. 서론

현재 인터넷에 연결되는 객체의 수가 전례 없는 속도로 증가하면서 사물인터넷(IoT)의 개념이 실현되고 있다. 이러한 객체의 기본적인 예로는 스마트 홈을 가능하게 하는 온도 조절기 및 HVAC (Heating, Ventilation, and Air Conditioning) 모니터링 및 제어 시스템을 포함한다. [1]

홈 IoT 생태계를 구성하는 데 있어 제품 제작자와 소비자 공통의 여러 도전점 중 하나는 서로 다른 프로토콜을 사용하는 제품들의 상호 운용성 문제이다. 소비자는 새로운 기능 및 스펙의 제품들을 자신의 홈 IoT 생태계에 등록하여 사용하고 싶지만, 기존에 구성된 생태계와 호환되지 않아 사용할 수 없는 문제가 발생하는 것이다. 이는 홈 IoT 시장에서 해결하지 못하면 장기적으로 악화될 수 있는 사안으로 이를 해결하기 위해 Matter라는 프로토콜이 개발되고 있다. Matter는 2019년에 애플, 구글, 아마존 등 여러 주요 기술 회사들이 참여하는 Connectivity Standards Alliance(CSA)에 의해 처음 발표되었다. 스마트 홈 장치 간의 상호 호환성을 향상하기 위해 개발된 오픈 소스 프로토콜이며 다양한 스마트 홈 제품이 서로 연결되고 통신할 수 있도록 하여 사용자 경험을 개선하고, 장치 간의 통합을 촉진하는 것을 목표로 한다.

Matter의 스택 구조를 보면 WLAN의 Wi-Fi, LAN의 Ethernet, WPAN의 Thread, Bluetooth low Energy(BLE) 그리고 추가적인 네트워크 레이어로 구성 되어있다. BLE는 Commissioning만을 위해 사용하며 실질적인 WPAN은 Thread를 채택하고 있다. 본 논문에서는 Thread에 대하여 주요하게 설명하고 네트워크를 구성하여 시뮬레이션한다.

II. 본론

스마트 시티와 스마트 홈은 센싱 장치의 배터리 수명에 높은 요구치를 가지고 있으며, 센싱 장치의 안정적인 가용성을 보장하기 위해 배터리 수

명을 개선할 필요가 있다. 스마트 시나리오에서 기대되는 가장 긴 배터리 수명은 20년으로 추정된다. [2], [3], [4]

이러한 요구치를 충족하기 위해 Matter가 채택한 저속, 저전력 프로토콜이 Thread이다. Thread를 채택한 이유는 먼저 다른 프로토콜의 허브와 같은 역할을 하는 보더 라우터를 통해 IPv6와 같은 IP routing이 가능하기 때문이다. 허브와 다른 점은 보더 라우터 역할을 하던 디바이스가 네트워크에서 제외되었을 때 네트워크에 다른 보더 라우터 기능을 할 수 있는 디바이스가 존재한다면 자동으로 역할을 이어받아 자연스럽게 기능을 수행한다는 것이다. 따라서 보더 라우터를 여러 개 배치하기도 한다.

Thread 네트워크에서는 보더 라우터가 Thread 기기가 아닌 디바이스와 IPv6 기반 통신을 하기 위해서 Off Mesh Routable (OMR) 주소가 사용된다. 이는 OMR 접두어 prefix로 시작하는 IPv6 주소의 형태이며 WiFi/Ethernet 네트워크 또는 다른 메쉬의 Thread 네트워크와 연결을 가능하게 한다.

시뮬레이션에서는 위의 WiFi/Ethernet 이외의 네트워크와 연결을 하기 위해서 Linux Host에서 Co-Processor Communication Daemon (CPCd)

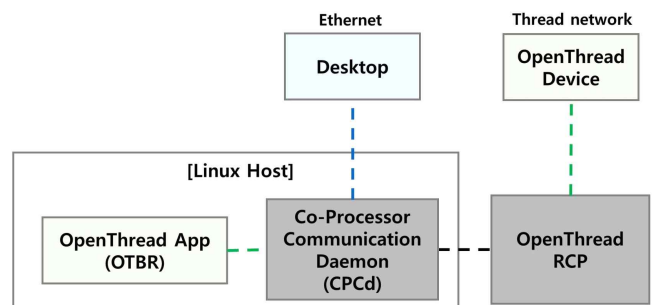


그림 1. 시뮬레이션 네트워크 구조

을 사용한다. CPCd는 여러 엔드포인트로 공유된 PHY 링크를 통해 보조 프로세서와 상호 작용하는 다수의 프로토콜을 사용할 수 있도록 해준다. 이는 차후 연구에서 새로운 프로토콜의 네트워크와의 연결을 위해 구성하였다.

시뮬레이션을 수행할 네트워크의 구조는 그림 2와 같다. Linux Host로는 raspberry pi 4를, OpenThread RCP와 Device로는 SiliconLabs사의 EFR32MG21 WSTK를 사용하였다. Linux host는 데스크탑과 같은 Ethernet 라우터에 연결하였고 CPCd를 통한 OpenThread command line interface(OT-CLI)를 사용하여 시뮬레이션에 필요한 기능을 수행하였다.

시뮬레이션은 네트워크 구성과 핑 송수신, UDP를 통한 payload 송수신 순서로 수행하였다. 보더 라우터(Linux Host)를 Leader로, OT 디바이스를 router로 설정하였고, 각각 UDP 포트를 열고 번호를 할당하였다.

```
Pinging fd61:b5d0:bf3:1:f102:e1e5:b7ff:1e3f with 32 bytes of data:
Reply from fd61:b5d0:bf3:1:f102:e1e5:b7ff:1e3f: time=138ms
Reply from fd61:b5d0:bf3:1:f102:e1e5:b7ff:1e3f: time=46ms
Reply from fd61:b5d0:bf3:1:f102:e1e5:b7ff:1e3f: time=51ms
Reply from fd61:b5d0:bf3:1:f102:e1e5:b7ff:1e3f: time=46ms
```

```
Ping statistics for fd61:b5d0:bf3:1:f102:e1e5:b7ff:1e3f:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 46ms, Maximum = 138ms, Average = 70ms
```

(a)

Age	Type	Len	Chksun	Sec	Prio	RSS	Dir	Neighb	Radio
00:08:53.162	ICMP6(EchoReply)	40	0x2081	yes	norm		NA	TX	0x7000 15.4
00:08:53.167	ICMP6(EchoReqst)	40	0x2181	yes	norm	-56	RX	0x7000	15.4
00:08:54.172	ICMP6(EchoReply)	40	0x2082	yes	norm		NA	TX	0x7000 15.4
00:08:54.179	ICMP6(EchoReqst)	40	0x2182	yes	norm	-55	RX	0x7000	15.4
00:08:55.190	ICMP6(EchoReply)	40	0x2083	yes	norm		NA	TX	0x7000 15.4
00:08:55.196	ICMP6(EchoReqst)	40	0x2183	yes	norm	-56	RX	0x7000	15.4
00:08:56.103	HopOpts	46	0x0000	yes	norm		NA	TX	bcast 15.4
00:08:56.111	ICMP6(EchoReply)	40	0x2084	yes	norm		NA	TX	0x7000 15.4
00:08:56.117	ICMP6(EchoReqst)	40	0x2184	yes	norm	-56	RX	0x7000	15.4

그림 2. PC to Thread 핑 신호 (a) 송신 결과 (b) 수신 로그

```
> udp send fd00:4562:1d9c:ffff:0:ff:fe00:f800 5678 CPCd_OT_TEST
Done
> 11 bytes from fd00:4562:1d9c:ffff:f189:6cec:5205:c0b8 5678 FTD_OT_TEST
history rxtx
```

Age	Type	Len	Chksun	Sec	Prio	RSS	Dir	Neighb	Radio
00:00:59.821	UDP	19	0xe5ec	yes	norm	-62	RX	0xf800	15.4
00:02:00.379	UDP	20	0xccd1	yes	norm		NA	TX	0xf800 15.4

그림 3. 보더 라우터 to Thread 디바이스 UDP Payload 송수신 로그

```
> udp send fd00:4562:1d9c:ffff:0:ff:fe00:fc00 1234 FTD_OT_TEST
Done
> history rxtx
```

Age	Type	Len	Chksun	Sec	Prio	RSS	Dir	Neighb	Radio
00:00:04.556	UDP	19	0xe5ec	yes	norm		NA	TX	0x8000 15.4
00:00:40.590	UDP	19	0xe5ec	yes	norm		NA	TX-F	0x8000 15.4
00:01:05.109	UDP	20	0xccd1	yes	norm	-52	RX	0x8000	15.4
00:01:55.107	ICMP6(EchoReply)	16	0xb8d	yes	norm	-55	RX	0x8000	15.4
00:01:55.139	ICMP6(EchoReqst)	16	0xb8d	yes	norm		NA	TX	0x8000 15.4
00:02:20.511	ICMP6(EchoReply)	16	0xbf59	yes	norm		NA	TX	0x8000 15.4
00:02:20.514	ICMP6(EchoReqst)	16	0xc059	yes	norm	-54	RX	0x8000	15.4

그림 4. Thread 디바이스 to 보더 라우터 UDP Payload 송수신 로그

먼저 OMR 주소를 사용하여 Desktop에서 OT 디바이스로 핑을 전송하였고 핑은 보더 라우터를 경유하여 도착하였다. 결과로 그림 2의 (b), 수신 결과의 “Neighb”가 “0x7000”인 것을 보아 보더 라우터를 경유했음을 확인할 수 있다.

다음으로 할당된 UDP 포트를 통하여 보더 라우터에서 OT 디바이스로 “CPCD_OT_TEST”를 그리고 OT 디바이스에서 보더 라우터로 “FTD_OT_TEST”를 전송하였고 그림 3, 4와 같이 메시지 Payload를 송수신한 것을 확인하였다. RSSI는 대략 -55dBm ~ -60 dBm으로 약 2m의 실험 환경에서 보이는 양호한 수치이다.

끝으로 실질적인 기능 수행은 불가능하여 자료를 첨부하지 않았으나 Thread 서비스를 생성하고 Linux Host를 통해 Wi-Fi로 서비스를 스마트폰으로 검색하여 서비스의 이름과 수행하는 디바이스의 주소를 확인할 수 있었다.

III. 결론

본 논문에서는 다중 프로토콜 네트워크에서 지속 저전력 프로토콜로서 주로 사용될 Thread 네트워크를 구현하였다. 이를 통해 CPCd 및 보더 라우터를 네트워크 구현에 포함하여 LAN 네트워크와 WPAN Mesh 네트워크 사이의 통신을 수행하였다. 차후 CPCd를 통해 다른 프로토콜들을 추가하여 IoT 생태계의 연결성을 더욱 향상할 수 있을 것으로 생각된다. 하지만 연구 결과의 간단한 데이터 교환 이상으로 홈 IoT에 적용하기 위해서는 복잡한 기능 수행에 있어 데이터 구조의 호환성도 고려해야 할 부분으로 보인다.

ACKNOWLEDGMENT

- (국)이 연구는 2023년도 산업통상자원부 및 산업기술평가관리원(KEIT) 연구비 지원에 의한 연구임 (00144500)
- 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2022R1A2C1092737)

참고 문헌

- [1] A. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, 2015, pp. 2347–2376.
- [2] “Key drivers and research challenges for 6G ubiquitous wireless intelligence,” 6G Flagship, Oulu, Finland, White Paper, Sep. 2019. [Online]. Available: https://www.mobilewirelesstesting.com/wp-content/uploads/2019/10/5G-evolution-on-the-path-to-6G-_wp_en_3608-3326-52_v0100.pdf
- [3] W. Tong and P. Zhu, 6G, *The Next Horizon: From Connected People and Things to Connected Intelligence*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 2021.
- [4] J. R. Bhat and S. A. Alqahtani, “6G ecosystem: Current status and future perspective,” *IEEE Access*, vol. 9, pp. 43134–43167, 2021.
- [5] Silicon Labs, “AN1333: Running Zigbee, OpenThread, and Bluetooth Concurrently on a Linux Host with a Multiprotocol Co-Processor,” 1.2, 2023.