

RSSI 기반 다중 노드 거리 측정 시스템 설계 및 구현

박기완, 허준*, 무하마드 하리스*, 남해운*

한양대학교 에리카, 한양대학교*

{rldhks6899, inas2056, haris737, hnam}@hanyang.ac.kr

Design and Implementation of a Multi-Node Distance Measuring System Based on RSSI

Kiwan Park, Jun Hur, Muhammad Haris, Haewoon Nam

Hanyang Univ.

요약

본 논문은 RSSI 기반의 다중 노드 거리측정 시스템을 설계하고 구현했다. 무선 신호의 강도를 측정하는데 사용되는 방법의 하나로 RSSI를 사용하면 간단하면서도 낮은 비용으로 넓은 범위를 측정할 수 있고, 실시간 거리 정보를 제공할 수 있다. 본 연구에서는 RSSI 기반의 거리 측정 시스템을 구현하기 위해 WiFi 기능을 갖춘 저비용, 저전력 보드인 ESP 32 모듈을 채택하였으며, 이를 이용해 1개의 서버 노드와 5개의 클라이언트 노드를 만들어 다중 노드 거리 측정 시스템을 설계했다. 각 클라이언트 노드는 RSSI 값이 기준 이하로 떨어지면 경고 신호가 발생하도록 설계되었는데, 이에 대한 신뢰성을 평가하기 위해 1m부터 3m까지 RSSI 값을 측정하는 실험을 진행했다. 그 결과 실제 거리가 기준 거리(1m)에 가까울수록 거리 추정 오차가 줄어드는 것을 확인할 수 있었다.

I. 서론

로봇 공학, 내비게이션, 무선 센서 네트워크 및 사물 인터넷(IoT)과 같은 분야에서 거리측정은 네트워크 연결 유지, 센서 노드 위치 파악 등에 중요한 역할을 한다. 특히 무선 노드 간의 거리를 정확하게 결정하는 기능은 위치 추적과 물체 회피 및 매핑과 같은 응용 프로그램에 필수적이다. 무선 센서 네트워크 분야에서 거리측정은 네트워크 연결을 유지하고 센서 노드의 위치 파악에 중요한 역할을 하며, 마찬가지로 IoT 분야에서는 산업 자동화, 스마트 시티와 같은 다양한 응용 분야에 적용된다.

RSSI (Received Signal Strength Indicator)를 이용하여 거리 추정에 사용할 수 있다. RSSI는 무선 신호의 강도를 측정하는데 사용되는 방법의 하나로 간단하면서도 낮은 비용으로 넓은 범위를 측정할 수 있고 실시간 거리 정보를 제공할 수 있다는 장점이 존재한다. 특히 RSSI와 거리 사이 강한 상관관계가 있는데, 수신기와 송신기 사이가 가까울수록 신호가 강해져 RSSI 값도 함께 커진다. Bluetooth[1, 2], Zigbee, WiFi[3]와 같은 다양한 통신 프로토콜에서 RSSI를 기반으로 거리를 추정하는 기존의 연구들이 있었다.

본 논문에서는 WiFi 통신을 이용하여, RSSI 기반의 다중 노드 간 거리 측정 시스템을 설계 및 구현한 것을 소개한다. 이를 구현하고자 WiFi 기능을 갖춘 저비용, 저전력 보드로 IoT 애플리케이션에 많이 사용되는 ESP 32를 채택했다. 해당 모듈을 통해 거리 측정 시스템을 설계하였으며, 노드 간의 거리가 일정 거리보다 가까워지면 사용자와 관리자에게 안내 메시지를 보내는 기능을 추가하였다. 코로나와 같은 질병의 확산을 방지 또는 안전성 확보를 위해 안전거리 유지가 중요한 현 시점에서 제안하는 RSSI 기반의 다중 노드 간 거리 측정 시스템은 연구 가치가 높은 것으로 판단한다.

II. 제안하는 다중 노드 거리측정 시스템

제안하는 시스템은 ESP 32 보드를 이용해 1개의 서버 노드와 5개의 클라이언트 노드를 설계하였다. 그림 1은 각 노드 간 관계를 보여준다.

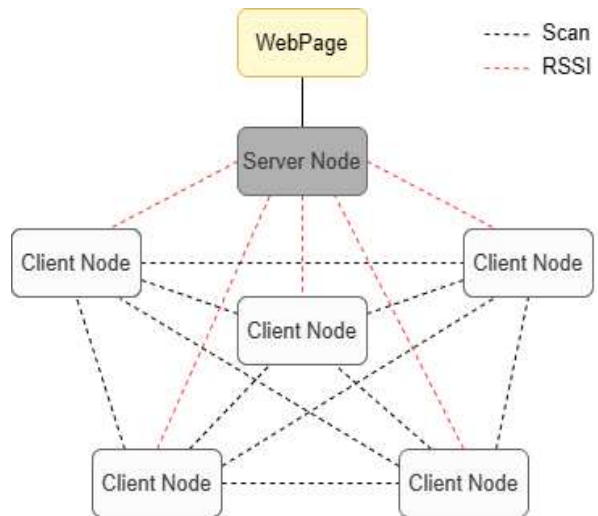


그림 1. 시스템 구조

클라이언트 노드는 주변의 다른 클라이언트 노드를 스캔하고 RSSI 값을 수집한다. 서버 노드는 WiFi 네트워크를 만들어 Access Point(AP)로서 동작하며, 클라이언트 노드는 해당 네트워크를 식별하면 자동으로 연결되어 수집한 데이터를 서버 노드로 보낸다. 서버 노드는 수집된 데이터를 처리하고 웹 페이지에 3초에 한 번씩 업데이트를 진행한다. 웹 페이지는 노드와 연동을 위해 Node.js 및 Express.js를 사용하였으며, 이를 통해 노드 간 연결 상태 및 RSSI 값을 실시간으로 확인할 수 있다.

PainLessMesh 라이브러리를 사용해 모듈의 BSSID 정보를 이용해 노드마다 고유한 Node ID를 지정하고 서버 노드와 클라이언트 노드를 구별할 수 있도록 한다. 따라서 클라이언트 노드 간의 RSSI 값만을 얻어낼 수 있으며, 이 값은 다시 Node ID가 일치하는 서버 노드로 보낼 수 있다.

제안하는 거리 기반 경고 시스템을 테스트하기 위하여 제작된 모듈은 그림 2에서 볼 수 있다. 각 모듈에는 노드 간 통신 상태 및 RSSI 값에



그림 2. 제작된 노드 모듈

```
07:27:23.823 -> ScanNetworks : 26 , delayCnt : 0, delayFlag : 0, delayTime : 44730, Wifi status : 3 , BSSID : 24:D7:EB:0C:8D:51
07:27:10.392 -> NODE : 1 | BSSID : E8:9F:6D:E8:CD:29 | RSSI : -3 dBm
07:27:10.437 -> NODE : 4 | BSSID : B8:D6:1A:82:2D:BD | RSSI : -7 dBm
07:27:10.437 -> NODE : 3 | BSSID : 10:97:BD:34:60:71 | RSSI : -16 dBm
07:27:10.484 -> NODE : 5 | BSSID : 24:D7:EB:0C:8D:51 | RSSI : -22 dBm
```

```
07:27:23.823 -> NODE : 4 | BSSID : B8:D6:1A:82:2D:BD | RSSI : -9 dBm
07:27:23.869 -> NODE : 3 | BSSID : 10:97:BD:34:60:71 | RSSI : -16 dBm
07:27:23.869 -> NODE : 5 | BSSID : 24:D7:EB:0C:8D:51 | RSSI : -21 dBm
```

그림 3. 노드 2번에서 수집된 RSSI 데이터

대한 실시간 정보를 제공하기 위해 소형 OLED 디스플레이를 장착했다. 또한, 진동 모터를 보드 뒷면에 장착하여 노드 간 일정 거리에 대한 경고 신호를 사용자에게 보낼 수 있게 하였다. 제작한 모듈을 통해 서버 노드와 클라이언트 노드 간 안정적인 통신이 가능했다. 그림 3은 클라이언트 노드 2번에서 주변을 스캔하고 RSSI 값을 수집하는 과정을 나타낸다. 서버 노드의 네트워크 환경에 연결된 클라이언트 노드는 그림 2와 같이 주변 노드를 식별하여 RSSI 값을 측정하고 수집할 수 있다. 다른 노드로부터 얻은 RSSI 값을 이용하면 식 (1) 를 통해 거리를 판단할 수 있다[4].

$$RSSI = -10\log_{10}(d) + \alpha \quad (1)$$

n 은 경로 손실 지수(Path loss exponent)로 거리에 따라 신호 강도가 얼마나 빨리 감소하는지를 나타낸다. 장애물의 여부 등에 따라 보통 2에서 4 사이의 값을 가진다. d 는 실제 거리, α 는 기준 거리 (1m)에서 수신 기로부터 측정된 기준 RSSI 값이다. 1번 식은 다음과 같이 거리 d 에 대한 수식으로 변형할 수 있다.

$$d = 10^{\frac{\alpha - RSSI}{10n}} \quad (2)$$

그림 4에서 서버 노드의 클라이언트 노드 관리 웹 페이지를 볼 수 있다. 서버 노드에 클라이언트 노드가 연결되면 웹 페이지에서 해당하는 노드 블록이 녹색으로 바뀌어 연결이 성공했음을 나타내고, 그 아래에서 주변 노드로부터 수집한 RSSI 값을 기록한다. 반면, 노드를 찾지 못하는 경우 노드 블록이 빨간색으로 유지되며, 연결될 때까지 계속 스캔을 한다. 일련의 과정은 3초마다 반복되어 노드 상태 및 RSSI 값을 업데이트한다.

III. 결과

RSSI 값으로부터 식 (2)를 이용해 추정 거리를 계산했을 때, 기준 거리 이하로 나오면 사용자에게 경고 신호를 보낸다. 경고 신호에 대한 신뢰성을 평가하기 위해 1m부터 3m까지 0.5m 간격으로 RSSI 값을 측정하는 실험을 30회 반복했다. 표1은 실험을 통해 얻은 RSSI 값의 평균과 이를 통해 구한 추정 거리 및 실제 거리와의 오차를 보여준다. 실험을 통해 구한 RSSI 값으로 각 거리 마다의 평균 경로 손실 지수를 구했으며, 그 평균값인 1.8362를 경로 손실 지수 n 값으로 설정하여, 거리를 추정하였다. 실험

BSSID : E8:9F:6D:E8:CD:29 RSSI : 444742549 NODE 1	BSSID : B8:D6:1A:82:2E:8D RSSI : 0000000 NODE 2	BSSID : 10:97:BD:34:60:71 RSSI : 0000000 NODE 3	BSSID : B8:D6:1A:82:2D:BD RSSI : 0000000 NODE 4	BSSID : 24:D7:EB:0C:8D:51 RSSI : 0000000 NODE 5
N1 0 dBm	N1 0 dBm	N1 0 dBm	N1 0 dBm	N1 0 dBm
N2 0 dBm	N2 0 dBm	N2 -10 dBm	N2 0 dBm	N2 -42 dBm
N3 0 dBm	N3 -10 dBm	N3 0 dBm	N3 0 dBm	N3 -23 dBm
N4 0 dBm	N4 0 dBm	N4 0 dBm	N4 0 dBm	N4 0 dBm
N5 0 dBm	N5 -41 dBm	N5 -20 dBm	N5 0 dBm	N5 0 dBm

그림4. 서버 노드 웹 페이지

표 1. 추정 거리와 실제 거리 오차 (경로 손실 지수 $n = 1.8362$)

실제 거리	측정된 RSSI (Avg.)	추정 거리	오차
1m	-35.13 dBm	1m	0
1.5m	-38.53 dBm	1.532m	2%
2m	-41.03 dBm	2.096m	5%
2.5m	-42.07 dBm	2.386m	5%
3m	-43.30 dBm	2.785m	7%

결과 실제 거리가 기준 거리(1m)에 가까울수록 거리 추정 오차가 줄어드는 것을 확인할 수 있었다.

IV. 결론

본 연구에서는 RSSI를 이용한 거리측정 시스템에서 거리측정 오차를 분석하기 위해 실제로 ESP 32 모듈로 6개의 노드를 구현하고, RSSI 값을 수집하여 추정 거리를 계산했다. 실험 결과, 기준 거리(1m)에 가까울수록 추정 거리의 오차가 낮았으며, 보건복지부에서 정한 사회적 거리 두기 기준 2m에서의 추정 거리에 대한 오차는 5%를 보여주었다. 이번 실험에서는 3m에 대한 신뢰도를 측정하였지만, 거리가 멀어짐에 따라 오차가 커질 수 있다는 것을 예측할 수 있다. 따라서 RSSI를 기반한 거리측정 시스템은 장애물이 없고, 단거리에서 활용되거나 높은 정밀도를 요구하지 않는 시스템에 적합할 것으로 판단된다.

ACKNOWLEDGMENT

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No.2022R1A2C1011862)

참 고 문 헌

- [1] Zhao, Qingchuan, et al. "On the accuracy of measured proximity of bluetooth-based contact tracing apps." *International Conference on Security and Privacy in Communication Systems*. Springer, Cham, pp. 49-60, 2020.
- [2] Barsocchi, P., Girolami, M., and La Rosa, D. "Detecting proximity with bluetooth low energy beacons for cultural heritage," *Sensors* vol. 21, no..21, p. 7089, 2021.
- [3] Whitney, Ann, et al, "RSSI informed phase method for distance calculations," in *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 1138-1142, 2015.
- [4] Y. Sung, "RSSI-based distance estimation framework using a Kalman filter for sustainable indoor computing environments," *Sustainability*, vol. 8, no. 11, p. 1136, 2016.