

위치기반 화재대피 서비스를 위한 혼잡도 기반 서버시스템 구현

송동혁*, 민진기*, 서효승*, 이준범*, 김현정*, 손봉기**, 이재호***

*서원대학교 정보통신공학과

**서원대학교 컴퓨터공학과

***e-mail:izeoho@seowon.ac.kr

An Implementation of Congestion-based Server System for LBS Fire Evacuation Service

Dong-Hyuk Song*, Hyo-Seung Seo*, Jin-Ki Min*, Joon-beom Lee*,
Hyeon-Jung Kim*, Bong-Ki Son**, Jaeho Lee*

*Dept. of Information and Communication Engineering, Seowon University

**Dept. of Computer Engineering, Seowon University

요 약

화재와 지진 등 재난 발생 시 인원이 많이 밀집되어 있는 건물 내에서는 혼란스러운 상황이 발생하는 것이 일반적이고 각 건물 내 사람들이 통솔하기도 어렵다. 이러한 문제를 해소하기 위해 본 논문에서는 상황에 따른 장치 간 전송하는 데이터를 나누어 데이터 전송에 효율성을 높이며 데이터베이스를 제안하여 데이터 간 관련성과 관리에 효율성을 높이고 사용자에게 직접 건물 내 상황을 알리고 안정적으로 대피를 할 수 있도록 도움을 주는 기술과 그에 필요한 Protocol을 제안한다.

1. 서론

최근 백화점, 마트, 기업 등 실내에 많은 인원을 수용하는 건물 실내에서는 화재 및 재난 발생 시 혼잡스런 상황이 발생하며, 이를 해결하기 위해 사용자에게 건물 내의 혼잡도, 상황 발생 등을 쉽게 알려 줄 수 있는 프로그램 개발이 필요하다. GPS, Wi-Fi 기술을 이용한 위치인식 프로그램이 사용가능하지만 사람이 많이 밀집된 실내에서 GPS는 큰 효과를 보기가 어렵고, Wi-Fi는 사용자의 이동에 따른 Traffic이 실시간으로 변경되어 Resurveying 문제가 발생할 수 있으며, 경제성 또한 낮으면서 유지 및 보수에 어려움이 있다.[1]

최근 주목을 받는 BLE(Bluetooth Low Energy)기술은 저전력으로 경제성면에서 좋은 인식을 가지고 있으며, 유지 및 보수에서도 어려움이 없으며 10m이내에서 Wi-Fi 견주었을 때 정확도가 뛰어나다는 평을 받고 있다. 실내측위 기술에서 BLE기반 Beacon의 RSSI(Received Signal Strength Indication)의 신뢰구간 신호 값을 이용한 핑거프린팅 기반의 위치인식 시스템 및 실내 혼잡도 파악 기술을 연구가 진행되며 그에 필요한 BLE 기반의 실내 위치인식 및 혼잡도 파악에 필요한 Protocol이 필요하다.[2] 본 논문에서는 BLE를 이용한 위치인식 및 혼잡도 기술에 사용되는 Protocol을 제시하며 상황 발생여부에 따른 데이터를 나누고 정의한다.

BLE(Bluetooth Low Energy)를 이용한 위치인식 과 혼잡도를 위해 Beacon의 Type을 A, B 두 가지로 나누었다.

Beacon Type A는 건물 내 각 비상구에 설치되어 비상구들의 혼잡 상태를 측정하고, 혼잡도측정 정보를 서버로 전송하는 역할을 하며, 서버에서 받은 혼잡도를 토대로 사용자들의 App으로 전송해 비상사태시 각 비상구의 혼잡상태를 알기 쉬워 큰 피해를 막을 수가 있다. Beacon Type B는 10m간격으로 설치되며 각 Beacon들의 RSSI값들을 사용자의 App으로 보내 App 자체 내에서 삼각측량 및 핑거프린팅 기술을 이용하여 사용자가 자신의 위치를 쉽게 파악 할 수가 있다.

2. 제안하는 Protocol 기술

본 논문에서는 혼잡도 파악의 정확한 기준과 사용자의 편리성을 고려해 센서를 이용한 시간 단위 카운트 기술을 사용하였는데, 시간단위 카운트 기술은 센서에 감지될 때 카운트가 점차 증가하며 비상구를 통해 사람이 지나갈 때마다 카운트를 올리며, 일정시간 단위로 카운트를 산출해 기준을 넘어가면 혼잡, 기준이하일시 여유로운 상태를 Beacon Type A가 사용자에게 알기 쉽게 알릴 수가 있다.[3] 이 과정에서 요구되는 데이터베이스 개발이 요구되며, 데이터의 타입 또한 나누어야 한다.

본 장에서는 Protocol에 이용되는 데이터베이스와 Beacon Type A와 서버, 사용자의 App간의 데이터베이스 및 데이터 타입을 나누는 것을 설명한다.

2.1 데이터 타입

Beacon Type A에서 서버를 거쳐 사용자의 App(단말

기)으로 데이터가 전송되는 동안 화재 및 재난 등 상황이 발생하였을 때와 평상시에 사용자가 받는 데이터가 같다면 많은 혼란을 야기 할 수가 있다. 그렇기 때문에 평상시와 상황발생시 그에 맞는 데이터를 나누어야 할 필요가 있다.

<표 1> 전송형태에 따른 데이터

데이터타입	Periodic Data	Event-Driven Data
Beacon Type A → Server	BeaconID, 전원형태, 배터리레벨, 연결상태, Protocol Response, 혼잡도	Beacon(비상구)의 상태
Server → Beacon Type A	BeaconID, Protocol Request	N/A
Beacon Type A → Server	Beacon(비상구)의 상태, BeaconID	N/A
Server → App	사용자 단말기 정보	화재발생시 push 메시지, 비상구 폐쇄유무

표1과 같이 평상시 주기적으로 전송하는 Periodic Data가 있고, 화재 발생 등 상황이 발생하였을 때 동작하는 Event-Driven Data가 있다. Periodic Data는 주기적으로(Periodic) 보내는 데이터를 말하는 것이다. Beacon Type A와 서버, 서버와 사용자의 App사이의 데이터를 Periodic Data로 분류하면 다음과 같다.

위 그림1의 Periodic Data를 자세히 살펴보면 Beacon Type A에서 서버로 가는 Periodic Data를 보면 Beacon ID로 어느 Beacon에서 오는 데이터인지 알 수가 있다. 배터리를 쓸 경우에는 배터리레벨을 이용하여 잔여 배터리를 표시 할 수가 있고, 비상구의 전원과 연결되어 사용된다면 전원이 교류, 직류를 어떠한 전원을 사용하는지 알 수가 있다. Protocol Request를 받은 후 Request를 보내 Beacon의 상태(손실되었는지)파악과 Beacon Type A와 서버 간 연결 상태가 Wi-Fi, Ethernet 등 어떠한 방식으로 연결되었는지 파악 할 수가 있고, 평소의 혼잡도를 주기적으로 전송하여야 한다. 서버에서 Beacon Type A의 경우에는 어느 Beacon으로 Protocol Request를 전송해야하는지 알아야 하기 때문에 BeaconID가 필요하다.

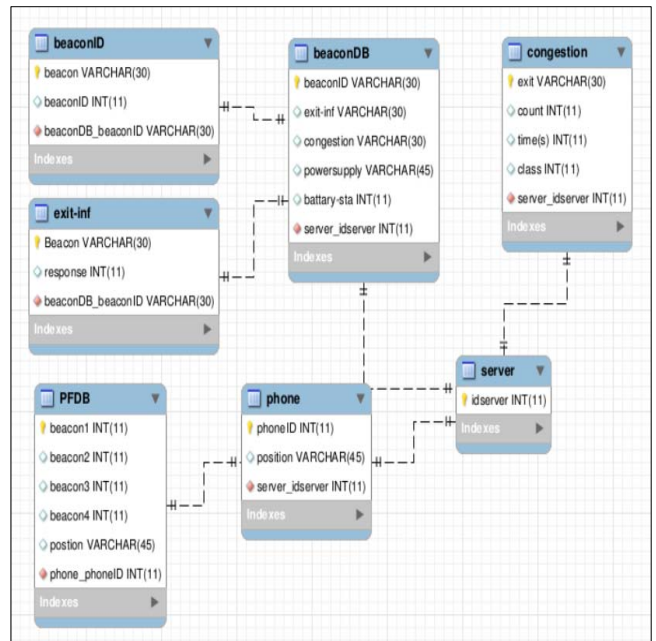
다음으로 서버와 App간의 Periodic Data를 보면 Beacon Type A에서 전송된 Beacon의 상태가 App으로 전송되어 App상의 맵에 비상구의 상태를 알 수가 있으며, BeaconID를 전송받아 어느 비상구인지 또한 알 수가 있다. App에서 서버로 사용자의 단말기 정보를 보냄으로 정보를 받아 누가 재난이 발생했음에도 탈출을 못했는지 알 수가 있다.

Event-Driven Data는 상황(화재 및 재난)발생 시 동작을 하는 데이터를 말하는데 Beacon Type A→ 서버,

서버→App에 대한 데이터가 있다. 위의 Event-Driven Data를 살펴보면 Beacon의 상태가 다시 한번 보이는데 이것은 혼잡한 상황이 발생 시 비상구 폐쇄, 재난 push 메시지를 보내는 동작과 연관되기 때문에 동작이 상황에 따라 다를 수가 있다.

2.2 데이터베이스 구조

서버와 Beacon 사이에 데이터 교환을 위해 데이터베이스를 만들 필요가 있다. 처음으로 데이터 테이블로 Beacon-Sta, 사용자의 단말기 정보, 각 Beacon의 Rssi의 값에 따른 위치로 구성되어 있다.



(그림1) 개발된 데이터베이스 구조

그림1의 데이터베이스 구조에서 BeaconDB에는 Beacon ID, 비상구의 상태(Exit-Inf), 배터리의 상태, 전원형태가 나타나 있고, 혼잡도(Congestion)에는 비상구를 지나간 인원을 나타내는 Count, 시간, 혼잡상태의 등급이 나타나 있다. PFDB에는 각 Beacon의 RSSI의 값과 그에 해당하는 위치정보가 있고, Phone(사용자 단말기)에는 PFDB에서 받은 위치정보와 PhoneID가 있다.

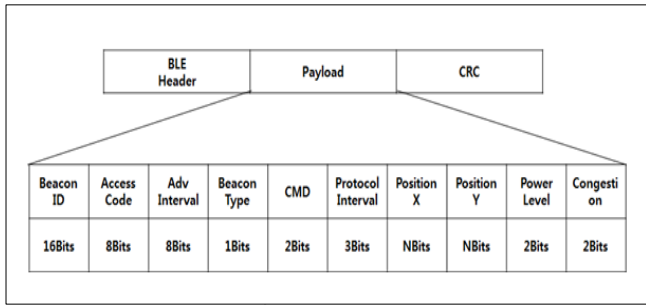
BeaconID와 Congestion 경우는 Beacon Type A와 서버 사이의 DB, PFDB와 PhoneDB의 경우는 단말기와 서버 사이의 DB로 나누어 볼 수가 있다. 각 데이터의 역할은 위 데이터 타입 항목에 기술된 것과 같으며 데이터베이스를 생성함으로 데이터의 관리의 전송에 중추적인 역할을 할 수가 있다.

2.3 Protocol 구조

혼잡도의 파악을 위해 위의 시간단위 카운트의 구동을 위해 데이터베이스와 가장 중추적 역할을 하는 프로토콜 기술이 필요하다. 본 장에서는 혼잡도 파악의 역할과 혼잡도의 기준을 표준하기 위한 프로토콜을 제시한다.

참고문헌

- [1] 하일규, 장철호, 박희주, 김종근, “실내 위치 측정을 위한 Wi-Fi 신호 특성 분석”, 한국정보통신학회논문지 제 16 권 제10호 : 2177~2184 October.2012
- [2] 윤창표, 황치곤, “실내 위치 기반 서비스제공을 위한 효율적인 실내 위치 측위 시스템”, 한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) 제19권 제6호 : 1368~1373 Jun. 2015
- [3] 차진만, 김선일, 양성룡, 이상욱, 박연식, “센서를 이용한 인원 관리 시스템에 관한 연구“, 한국정보통신학회논문지 제16권 제2호 : 373~177 February.2012



(그림2) 프로토콜 구조

그림 2의 프로토콜을 통해 BeaconID를 통해 Beacon을 파악하고 해당 Beacon의 위치를 파악 할 수가 있다. Access Code를 사용하며 해당 Protocol임을 식별하여 Beacon Type을 사용하여 혼잡도를 측정하기 위한 Type A와 단순 위치를 측정하기 위한 Type B를 식별한다. CMD를 이용함으로 서버에서 Beacon으로 Heart Bit 응답 요청을 위해 Heart Bit Request를 할당하고, 반대의 역할로 Beacon에서 서버로 Heart Bit에 대한 응답 메시지를 보내는 Heart Bit Response를 할당하며, Response의 응답과 Position X, Y를 사용하여 각 Beacon의 위치를 알 수가 있어 어떠한 Beacon의 상태에 문제가 생겼는지 알 수가 있어 Beacon관리에도 편리하고 빠르게 대처 해결할 수가 있다. Congestion 항목을 추가함으로 혼잡도의 기준을 잡고 혼잡하지 않은 상태(00)와 혼잡한 상태(01)를 파악 할 수가 있어 낮은 Bit로 혼잡도를 파악할 수가 있어 효율성이 높아진다.

3. 결론

본 논문은 위와 같이 BLE(Bluetooth Low Energy)를 이용한 위치인식과 혼잡도 파악 기술을 위해 데이터베이스 및 Heart-Bit Protocol을 이용하여 기술에 필요한 데이터와 혼잡도 파악 기술을 증대시킴으로 기술의 효율성을 증대시켰다. 하지만 외부요인으로 인한 서버 불안과 기기와 App의 호환이 맞지 않을 경우 상황 발생 시 큰 도움을 주지 못한다는 문제점을 가지고 있다.

App의 호환은 계속되는 개발로 해소가 되어 질 수 있지만 급작스런 서버 불안은 IT기술에도 같이 피해를 입히는 테러나 해킹 등에 불안정성을 염두해 두어야 한다. 또한 Beacon의 정확한 관리 시스템과 해당 범위에서만 사용자를 관리하는 사용자 관리 시스템 또한 필요하다.

본 논문에서 제안한 시스템 기반으로 서버관련 데이터 전송과 보안 문제와 사용자, Beacon의 관리 시스템을 연구할 계획이다.

Acknowledgement

본 연구는 중소기업청에서 지원하는 2016년도 산학연협력 기술개발사업(No. C0396518)의 연구수행으로 인한 결과물임을 밝힙니다.