

Mobile Edge Computing을 활용한 건물 재난 알림 시스템 구축 방안^{*}

Mobile Edge Computing based Building Disaster Alert System Implementation

하 태 영¹ 김 준 성¹ 정 중 문^{1*}
Taeyoung Ha Jungsung Kim Jong-Moon Chung

요 약

본 논문은 MEC (Mobile Edge Computing) 기술을 이용하여 건물에 재난이 발생 하였을 때 건물 내 사람들에게 재난에 대해 알리는 건물재난 알림 시스템 구현 방안을 대하여 제안한다. MEC의 개요를 설명하고, MEC를 활용한 네트워크의 구조와 특성을 파악한다. 추가적으로 기업 통합 패턴기반의 Apache Camel의 특성을 파악하고, 이를 활용한 MEC 구현 방안에 대해서 설명한다. 마지막으로 Apache Camel 기반의 MEC를 활용하여 재난 발생시, 센서들을 통해 재난상황을 빠르게 인식하고, 건물 내 사람들을 신속하게 대피할 수 있도록 돕는 건물재난 알림 시스템 구현 방안을 제시한다.

☞ 주제어 : 모바일 에지 컴퓨팅, 아파치 카멜, 건물 재난 알림 시스템

ABSTRACT

In this paper, a building disaster notification system with MEC (Mobile Edge Computing) technology is proposed, which informs people in a building about the disaster. The overview of MEC is presented, and the structure and characteristics of network using MEC are described. In addition, the characteristics of a enterprise integration pattern based Apache Camel is described, and how to implement MEC with Apache Camel is presented. Finally, an implementation method of building disaster notification system with Apache Camel based MEC is proposed to quickly recognize disasters through sensors and to rapidly evacuate people from buildings.

☞ keyword : Mobile Edge Computing, Apache Camel, Building Disaster Alert System

1. 서 론

우리나라는 1994년 10월 성수대교 붕괴사고를 시작으로 1995년 6월 삼풍백화점 붕괴사고, 2003년 대구 지하철 화재사고 등 대형 사고들이 발생하면서 재난에 대한 체계적인 대처가 필요하다는 인식이 생기게 되었다 [1]. 최근에는 건물이 고층화되고 거대화됨에 따라서, 건물에서 발생할 수 있는 재난상황이 일상화, 다양화, 대형화되고 있으며 재난 발생 시에 피해 규모 또한 증가함에 따라 재난에 대비할 수 있는 시스템의 필요성이 증가하고 있다.

재난상황에 대비할 수 있도록 건물들은 다양한 센서들을 건물에 배치하고 이들을 통해서 재난 상황을 감지하거나 안전한 피난경로를 알려주는 시스템들을 구축하고 있다 [2,3]. 하지만 재난상황이 다양화됨에 따라 이를 감지할 수 있는 다양한 종류의 센서들을 배치해야할 뿐만 아니라 건물의 크기가 커짐에 따라 관리해야하는 센서의 개수가 기하급수적으로 증가하게 된다. 뿐만 아니라 많은 센서들을 제대로 관리하지 못 하면 제때 재난상황을 감지하지 못 하거나, 시스템이 잘못 작동한 경우에는 사람들에게 혼란을 줄 수 있기 때문에 센서들을 효율적으로 관리하고 활용할 수 있는 시스템을 구축해야 한다.

MEC (Mobile Edge Computing)는 분산 클라우드 컴퓨팅 기술을 활용하여, 다양한 서비스를 사용자의 인근에 위치시킴으로써 백본 네트워크의 혼잡을 줄이고 낮은 지연시간을 유지할 수 있는 로컬 서비스를 제공할 수 있는 기술이다. 그리고 MEC는 다양한 IoT (Internet of Things) 센서들을 관리하고, 센서에서 발생된 데이터들을 분배,

¹ School of Electrical and Electronic Engineering, Yonsei University, Seoul, 03722, Korea.

* Corresponding author (jmc@yonsei.ac.kr)

[Received 07 July 2017, Reviewed 18 July 2017, Accepted 18 July 2017]

☆ 본 연구는 정부(국민안전처)의 재원으로 재난안전기술개발사업단의 지원을 받아 수행된 연구임 [MPSS-재난-2015-10]

처리 할 수 있는 IoT 게이트웨이 (gateway)로서의 역할을 수행할 수 있다 [4].

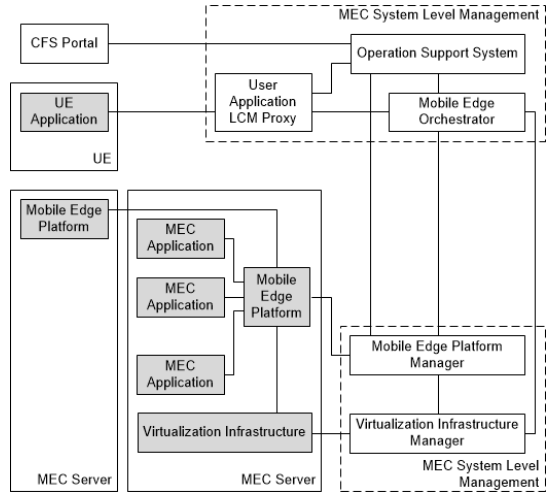
Apache Camel은 Java를 사용하는 오픈 소스 기반의 프레임워크로, 라우팅 (Routing)과 중재 (Mediation)기능을 제공 한다 [5]. 그리고 Apache Camel은 다양한 IoT 응용층 프로토콜 (Application protocol)을 기본 라이브러리로 제공하기 때문에 각기 다른 응용층 프로토콜을 사용하는 IoT 센서에서 발생하는 데이터들을 쉽게 처리가능하고, 다양한 외부 시스템과의 연동시킬 수 있는 다양한 라이브러리도 기본적으로 제공한다. 이러한 Apache Camel 특징을 활용하면 IoT 게이트웨이역할을 수행하는 MEC를 보다 쉽게 구축할 수 있다.

본 논문에서는 MEC의 대한 개요에 대해서 설명하고, Apache Camel을 활용한 MEC를 구축하는 방안을 제시한다. 그리고 Apache Camel 기반의 MEC를 활용한 건물 재난 알림 시스템을 구축하는 방안을 제시한다.

2. Mobile Edge Computing

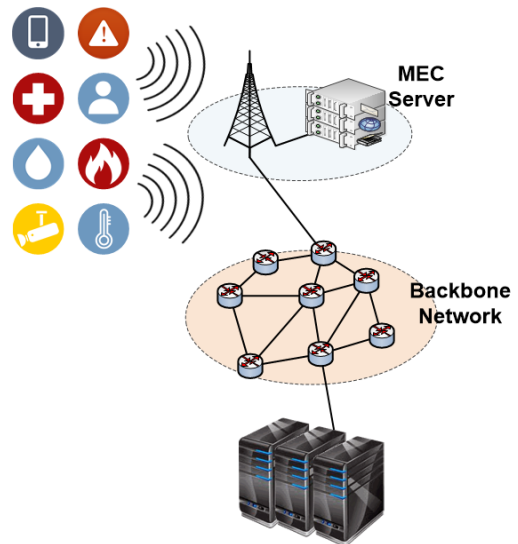
정보통신기술이 발전함에 따라서 IoT가 미래 인터넷의 하나의 형태로 각광받고 있으며, IoT 서비스를 제공하는 개체들이 2020년에는 전 세계적으로 2120억 개에 달할 것으로 예측 된다 [6]. 뿐만 아니라 사물인터넷의 큰 부분인 사물 간 통신 (Machine to Machine, M2M)이 2020년에는 전체 인터넷 트래픽의 45%를 차지할 것으로 예측 된다 [7]. 그리고 2021년에는 전 세계 IP기반 통신의 연간 트래픽이 2016년 대비 3.2배 증가하여 약 3.3 ZB (1 ZB = 10²¹ byte)에 다다를 것으로 예측되며, 트래픽이 집중되는 시간 (하루 중 가장 트래픽이 많은 1시간)에 발생하는 트래픽의 증가량은, 평균적인 트래픽의 증가량보다 더 커서 약 4.6배 증가할 것으로 예측 된다 [8]. 이러한 트래픽의 증가는 네트워크 과부하를 초래하고, 모든 트래픽이 백본 네트워크에 집중될 경우에는 지연시간이 기하급수적으로 커질 수 있다. 그렇기 때문에 지연시간을 일정 이내로 유지하려면 트래픽을 분산하여 처리할 수 있는 네트워크의 구조적 개선이 필요하다.

MEC는 ETSI (European Telecommunications Standards Institute)의 주도로 표준화 되고 있으며, 그림 1과 같은 MEC의 기준 구조를 제시하고 있다 [9,10]. MEC는 네트워크의 에지에서 클라우드 컴퓨팅 능력과 서비스를 바로 할 수 있는 환경을 제공하여 사용자의 요청을 MEC에서 처리할 수 있다. 그림 2는 MEC를 활용한 네트워크 구조



(그림 1) MEC의 기준 구조 [10]

(Figure 1) MEC Reference Architecture [10]



(그림 2) MEC가 포함된 네트워크 구조

(Figure 2) Network Architecture with MEC

를 보여준다. MEC를 활용하면 백본네트워크를 거치지 않고 서비스를 즉각적으로 제공할 수 있기 때문에 저 지연 서비스가 가능하며, 상대적으로 사용자의 근처에 위치한 MEC를 통해 실시간 네트워크 정보 접근이 가능하다. 뿐만 아니라 MEC는 기기에서 처리해야 할 태스크의 일부 또는 전체를 오프로드 받아 처리한 후 결과를 사용자에게 결과물을 제공할 수 있다. MEC의 오프로딩 기능

을 사용할 경우, 기기에서 태스크를 처리하는데 사용되는 에너지를 줄일 수 있어 기기의 동작 시간을 늘릴 수 있다.

3. Apache Camel 기반 건물 재난 알림 시스템

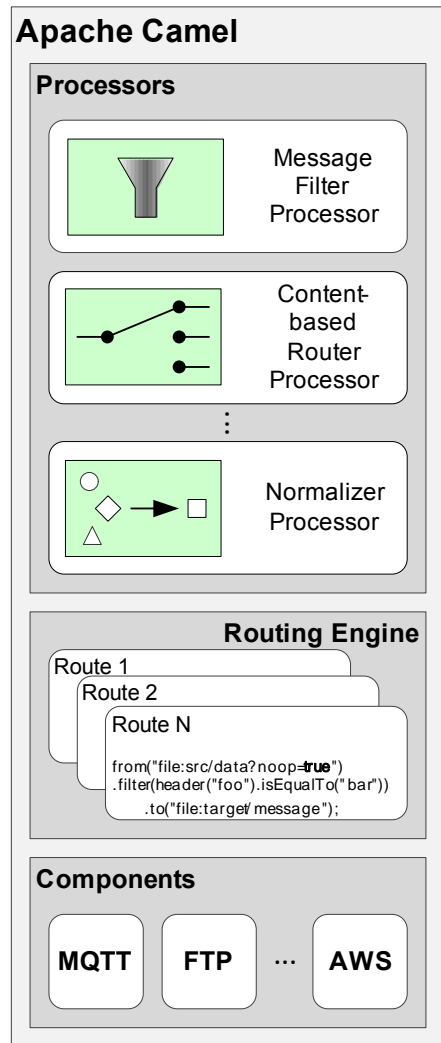
3.1 Apache Camel

Apache Camel은 2007년부터 개발이 시작된 프로젝트지만 이미 안정된 서비스를 제공할 수 있는 단계의 프로젝트다 [11]. Apache Camel은 시스템 통합을 위한 Java 프레임워크로 라우팅과 중재 기능에 특화되어 있으며, 기업 통합 패턴 (Enterprise Integration Patterns, EIPs)에 기반을 둔다. Apache Camel에서 사용되는 EIPs는 내용기반 라우터 (Content Based Router), 분할기 (Splitter), 필터 (Filter)와 같은 고전적인 패턴을 포함하며, 이외에도 15가지의 다양한 메시지 라우팅 패턴을 제공 한다 [12]. 뿐만 아니라 기존의 EIP의 메시지 시스템 (Messaging systems), 메시지 채널 (Messaging Channels), 메시지 구축 (Message Construction), 메시지 종단 (Message Endpoints), 그리고 시스템 관리 (System Management) 기능도 제공한다.

그림 3은 Apache Camel의 대략적인 구조도를 보여주며, Apache Camel은 크게 프로세서 (Processors), 컴포넌트 (Components), 그리고 라우팅 엔진 (Routing Engine)으로 구성된다. 프로세서는 종단 (Endpoint)간의 작업들을 다룬다. 예를 들면, 메시지 필터 프로세서 (Message Filter Processor)는 사전에 정의된 필터의 조건에 따라서 메시지들을 필터링하고, 콘텐츠 기반 라우터 (Content Based Router Processor)는 사전에 정의된 조건에 맞는 메시지들을 해당 종단으로 라우팅 해주는 역할을 한다. 컴포넌트는 종단들을 구성하며, 이미 구성된 외부 시스템과의 연결점이 되기도 한다. 라우팅 엔진은 종단과 프로세서들을 DSL을 통하여 루트를 생성하여 연결하여 준다.

3.2 Apache Camel의 장점

Apache Camel은 기존의 통합 프레임워크 대비 다음의 장점을 갖는다. 우선 첫 번째로 다른 통합 프레임워크들은 라우팅을 위해 주로 독자적인 DSL (Domain-Specific Language)를 사용한다. 반면에 Apache Camel은 Java는 주로 작성되지만, XML (eXtensible Markup Language)뿐만 아니라 Java, Scala, Rest, Groovy, Annotation, Kotlin과 같은 다양한 프로그래밍 언어로 DSL을 사용할 수 있다.



(그림 3) Apache Camel의 구조도
(Figure 3) Architecture of Apache Camel

다음으로 Apache Camel은 서로 다른 플랫폼을 통합하기 위해 80개 이상의 다양한 기본 라이브러리를 제공한다. 뿐만 아니라 플랫폼 사이에 Apache Camel을 통해 전송되는 페이로드는 특정 페이로드의 형식을 따르지 않아도 되기 때문에, 개발자들이 페이로드의 형식을 통일시키는 작업을 줄일 수 있다. 게다가 Apache Camel은 150개 이상의 형 변환기 (type-converter)를 내장하고 있어서 개발자들이 따로 데이터의 형을 변환하는 규칙을 만들지 않아도 된다.

마지막으로 Apache Camel은 Core 프로그램이 상대적

으로 작은 1.6 MB이며, 1.6 MB의 이 Core 프로그램은 Apache Commons Logging과 Fuse-Source Commons Management에 대해서만 의존적으로 동작한다. Apache Camel Core 프로그램의 용량은 상대적으로 작기 때문에, Raspberry Pi와 같은 소형 PC에도 독립적으로 동작하는 Apache Camel 환경을 구축할 수 있다.

3.3 Apache Camel 기반 MEC

네트워크 에지에 위치한 MEC는 백본 네트워크 보다 근접한 환경에서 클라우드 컴퓨팅 능력과 서비스를 제공한다. 이 때, MEC는 백본네트워크와 사용자 중간에 위치함으로써 중재기 (Mediator)나 게이트웨이 역할을 수행하게 된다. 따라서 MEC는 IoT 센서에서 발생할 수 있는 다양한 형태의 IoT, 빅데이터 (Big Data), 모바일 (Mobile) 서비스 요청을 처리할 수 있어야 한다. Apache Camel은 폭넓은 IoT 응용층 프로토콜을 라이브러리로 제공하여, IoT 센서에서 발생하는 데이터를 쉽게 처리할 수 있으며 다양한 서비스의 컴포넌트를 제공함으로써 다양한 외부 시스템 간 연결이 용이하다. 그렇기 때문에 Apache Camel를 활용하면 MEC의 중재 기능, 게이트웨이 기능 및 다양한 태스크의 처리기능을 구현 할 수 있다.

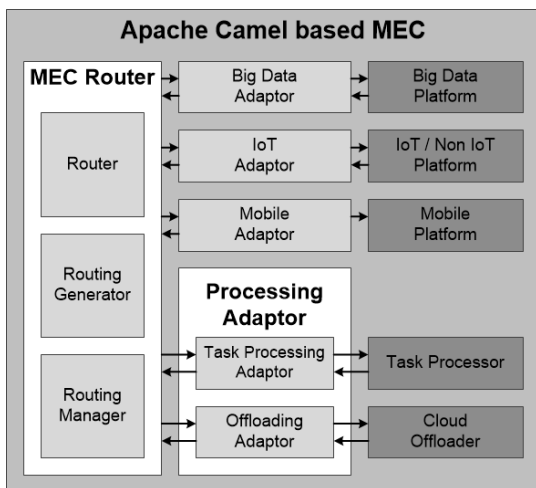
그림 4는 Apache Camel을 기반으로 구성한 MEC의 구조를 나타낸다. Apache Camel기반 MEC는 MEC 라우터 (Router), 프로세싱 어댑터 (Processing Adaptor) 그리고

IoT, 빅데이터, 모바일과 같은 여러 응용 서비스 플랫폼 및 프로토콜과의 통신을 위한 어댑터 (Adaptor)로 구성되어 있다. MEC 라우터는 태스크의 처리 과정 중 요구되는 서비스에 따라 종단과 종단 또는 종단과 컴포넌트를 연결하는 라우팅 (Routing) 경로를 생성하는 라우팅 제네레이터 (Routing Generator)와 생성된 라우팅 경로를 관리하는 라우팅 매니저 (Routing Manager), 그리고 라우팅을 수행하는 라우터 (Router)로 구성된다. 프로세싱 어댑터는 MEC로 오프로딩된 태스크가 처리될 수 있도록 태스크 프로세서 (Task Processor)와 연결해주는 태스크 프로세싱 어댑터 (Task Processing Adaptor), MEC에서 처리할 수 없는 태스크를 외부 클라우드 서비스로 다시 오프로딩하여 처리될 수 있도록 하는 오프로딩 어댑터 (Offloading Adaptor)로 구성된다.

IoT 어댑터는 다양한 IoT 플랫폼뿐만 아니라 non-IoT 플랫폼을 수용하기 위한 어댑터로써, 내부 API (Application Programming Interface)를 활용해 플랫폼에 IoT 센서 등록, 정보조회, 센서 값 조회, 데이터 포맷 정형화와 같은 태스크 요청이 있을 때 이용된다. 또한 IoT 어댑터는 Mobius, IoTivity와 같은 one2M2M 국제 산업계 표준으로 인정되는 플랫폼에 접속 할 수 있도록 OCF (Open Connectivity Foundation) 표준을 적용하여 구현된다. IoT 어댑터는 제공되는 플랫폼에 연결된 IoT 센서들과 관련된 태스크 요청 (센서의 정보 변경, 정보 삭제, 센서로부터 데이터 전송)이 있을 때 동작한다.

빅데이터 어댑터는 Apache Storm, Apache Spark와 같은 빅데이터 플랫폼과 연관 있는 태스크의 처리를 위해 필요한 어댑터이다. 센서로부터 발생하는 대량의 데이터에 대하여 저장, 연산 및 분석하는 태스크 요청이 있을 때, 빅데이터 어댑터를 통해 요청이 처리된다. 이를 위해 데이터의 타입 (CSV, JSON)의 정형화를 수행한다. 또한 빅데이터 어댑터를 통해 SQL (Structured Query Language) 기반 질의를 생성하고 전송할 수 있으며, 이를 통해 빅데이터 데이터베이스 (Database)를 활용할 수 있다.

모바일 어댑터는 모바일 애플리케이션 (Application), 사회 관계망 서비스 (Social Network Service, SNS), 메신저 (Messenger) 플랫폼에 대한 태스크 요청이 있을 때 사용된다. 모바일 플랫폼 전용 API 및 프로토콜을 MEC에서 사용 가능하게 전환하여 모바일 서비스 요청 시 MEC에서의 즉각적인 정보 전달을 담당한다. 따라서 모바일 서비스 사용자 통합 인증 절차, 데이터 캐싱 및 동기화, 데이터 필터링, 다중 외부 API의 분석 및 통합의 역할을 수행한다.



(그림 4) Apache Camel기반 MEC의 구조
(Figure 4) Architecture of Apache Camel based MEC

3.4 건물 재난 알림 시스템

건물 재난 알림 시스템의 경우 재난을 가능한 빨리 감지하고, 재난이 발생한 경우 사용자에게 가능한 빨리 재난 발생 사실을 알리고 대피할 수 있도록 안내해야 한다. Apache Camel 기반의 MEC를 활용할 경우 건물 재난 알림 시스템을 구현할 수 있으며, 기존의 네트워크 구조를 이용할 때보다 빠르게 사용자에게 알림을 제공할 수 있다.

건물 재난 알림 시스템에서 Apache Camel 기반 MEC의 구성요소들은 다음과 같은 역할을 한다. IoT 어댑터는 건물 내에 위치한 IoT 센서와 non-IoT 센서들을 관리하는 IoT 및 non-IoT 플랫폼에 접속하는 데 사용된다. 사용자가 IoT 센서와 non-IoT 센서에 대한 측정값을 요청할 경우 IoT 어댑터를 이용하여 해당 센서의 값을 제공하며, 주기적으로 센서의 측정값을 가져와 빅데이터 어댑터를 이용하여 빅데이터 플랫폼에 접근하여, 빅데이터 데이터베이스에 측정값을 저장한다.

빅데이터 플랫폼은 빅데이터 데이터베이스에 저장된 센서 데이터를 분석하여 재난의 발생 여부를 파악한다. 빅데이터 플랫폼을 사용할 경우, 재난의 발생 여부를 단순히 해당 재난을 감지하는 센서에 의존하지 않고 다양한 센서에서 얻어진 데이터를 분석하여 보다 더 정교하고 빠르게 재난 상황을 감지할 수 있다. 예를 들면 건물에 발생한 화재를 감지하는데 단순히 화재감지 센서만 이용하는 것이 아니라, 건물의 환기 시스템에 이용되는 온도 센서와 이산화탄소 센서를 활용하여 빅데이터 분석을 통해 추가적으로 화재 발생 여부를 판단할 수 있다. 화재의 발생 여부는 주기적으로 센서에 의해 측정되는 온도와 이산화탄소의 값의 시간에 따른 변화와 주변의 다른 센서에서 측정된 값과의 차이를 분석하고, 변화량 또는 차이값이 특정 한계점 (threshold) 값을 넘을 경우 화재가 발생했다고 인식하는 알고리즘을 바탕으로 판단할 수 있다.

모바일 어댑터는 건물에서 재난이 발생하였을 때 건물 내에 있는 사람들과 건물 인근에 있는 사람들에게 알림 메시지를 전송하는데 사용된다. IoT 및 non-IoT 플랫폼 또는 빅데이터 플랫폼에서 건물 내의 재난이 발생했음을 인지할 경우, 모바일 어댑터를 통해 휴대폰으로 재난 발생 알림을 제공하여 사람들이 재난이 발생한 장소로부터 빠르게 대피할 수 있도록 한다.

BLE (Bluetooth Low Energy) 비콘들을 이용하여 실내 위치 확인 시스템을 구축할 경우, 사람들에게 재난이 발생한 지역을 피해 안전하게 대피할 수 있는 경로 안내가

가능하다. 추가적으로 건물이 붕괴된 경우, 실내 위치 확인 시스템이 구축되어 있으면 매몰된 사람의 위치를 확인하여 보다 빠르게 인명 구조가 가능하다. 실내 위치 확인 시스템의 경우, 실내에서는 기존에 위치 확인 시스템에 널리 사용되는 위성 기반의 GPS (Global Positioning System)을 사용할 수 없기 때문에 추가적인 시스템 구축이 필요하다. 실내 위치 확인 시스템은 다양한 비콘을 활용하여 구축할 수 있다 [13]. 스마트폰과 통신이 가능한 비콘을 실내에 균일하게 배치하고, 배치된 비콘은 주기적으로 자신의 좌표를 포함한 메시지를 브로드캐스트 한다. 스마트폰에서는 비콘으로부터 수신된 메시지의 수신 전계 강도 (Received Signal Strength Indication, RSSI) 값을 바탕으로 비콘과 스마트폰 사이의 거리를 추정한다. 그 후에 비콘 메시지의 포함된 비콘의 좌표값과 추정된 비콘과의 거리를 바탕으로 삼각측량법을 활용하여 자신의 위치를 확인할 수 있다.

또한, MEC 기반의 건물 재난 알림 시스템을 활용하면, 재난 상황으로 건물 외부와의 네트워크가 단절되더라도 건물을 관리하는 MEC가 따로 존재하기 때문에 해당 시스템은 지속적으로 동작할 수 있다.

3.5 MEC 기반 시뮬레이션

제안하는 건물 재난 알림 시스템은 MEC를 기반으로 동작하기 때문에, 기존의 클라우드 서버를 기반으로 동작할 때보다 대부분의 상황에서 줄어든 지연시간을 제공한다는 것을 보여주기 위해 시뮬레이션을 수행하였다. 그림 5는 MEC가 사용된 네트워크의 구조와 지연시간의 구성 요소를 나타내며, 그림 5를 바탕으로 MEC를 기반으로 동작할 때의 지연시간 (D^M)과 기존의 클라우드 서버를 기반으로 동작할 때의 지연시간 (D^C)은 다음과 같이 나타낼 수 있다 [14].

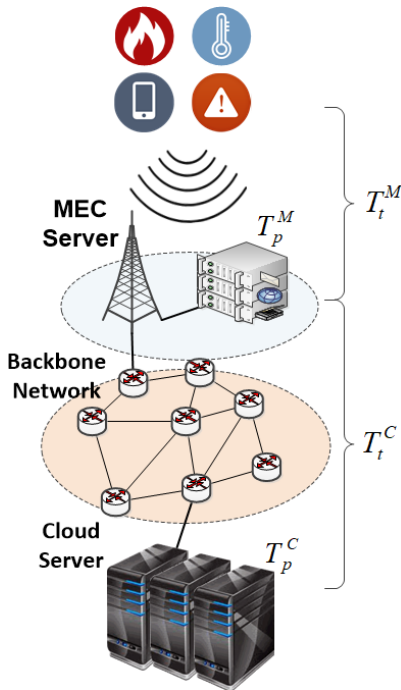
$$D^M = T_t^M + T_p^M = d/r + c/f^M$$

$$D^C = T_t^M + T_t^C + T_p^C = d/r + \varphi + c/f^C$$

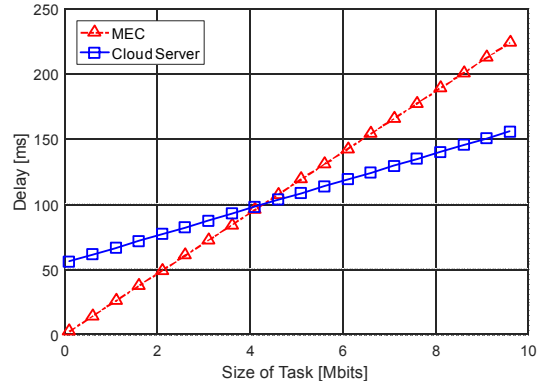
위 식에서 T_t^M 은 단말에서 MEC (기지국)까지 무선 환경으로 전송되는데 걸리는 시간이며, T_p^M 은 MEC에서 태스크를 처리하는데 걸리는 시간을 나타낸다. d 는 무선으로 전송되는 패킷의 크기를 나타내며, r 은 무선 환경에서의 데이터 전송 속도를 나타낸다. c 는 해당 태스크가 처리되는데 필요한 CPU (Central Processing Unit)의 사이클의 수를 나타내고, f^M 은 MEC에서 태스크를 처리하는

데 사용되는 CPU 동작 주파수이다. T_r^C 는 기지국에서 클라우드 서버까지 유선 환경으로 전송되는데 걸리는 시간이며, T_p^C 는 클라우드 서버에서 태스크를 처리하는데 걸리는 시간을 나타낸다. φ 는 기지국과 클라우드 서버 사이의 거리에 따라 결정되는 값이고, f^C 는 클라우드 서버에서 태스크를 처리하는데 사용되는 CPU 동작 주파수이다. 일반적으로 MEC보다 클라우드 서버의 태스크 처리용량이 더 크기 때문에 $f^M < f^C$ 라고 할 수 있다. 그리고 데이터 패킷의 크기가 처리해야하는 비트수와 동일할 경우 c 는 처리 밀도 (processing density) Γ (cycles/bit) 값을 이용하여 다음과 같이 $c = d\Gamma$ 로 나타낼 수 있으며, Γ 값은 CPU의 처리 능력에 따라 달라지고 $\Gamma^M > \Gamma^C$ 라고 할 수 있다 [15]. 이때 Γ^M 은 MEC의 처리 밀도이고 Γ^C 는 클라우드 서버의 처리 밀도이다.

그림 6은 처리해야하는 태스크의 크기에 따른 지연시간을 나타내는 그래프이다. 태스크의 크기가 작을수록 MEC에서 처리할 때 지연시간이 짧고, 태스크의 크기가 클수록 클라우드 서버에서 처리할 때 지연시간이 짧은



(그림 5) MEC가 포함된 네트워크 구조와 지연시간
(Figure 5) Network Architecture and Delay of MEC based Network



(그림 6) 시뮬레이션 결과
(Figure 6) Simulation Result

것을 확인 할 수 있다. 이는 클라우드 서버의 처리 용량이 MEC에 비해 상대적으로 크기 때문이다.

대부분 IoT기기에서 발생하는 태스크의 크기는 작기 때문에, 건물 재난 알림 시스템에서 낮은 지연시간을 유지하기 위해서는 MEC기반의 네트워크를 구성하는 것이 유리하다는 것을 시뮬레이션 결과를 통해 확인할 수 있다.

4. 결 론

본 논문에서는 Apache Camel기반의 MEC를 구축하고 이를 활용하여 건물 재난 알림 및 대피 시스템 구축방안을 제안하였다. 다양한 프로토콜을 사용하는 IoT 센서들에서 제공되는 정보를 MEC의 IoT 어댑터를 통하여 수신하고, 빅데이터 어댑터를 활용하여 수신된 정보를 빅데이터 데이터베이스에 저장한다. 저장과 동시에 데이터베이스에 저장된 정보들을 분석하여 재난 상황을 인지할 경우, 모바일 어댑터를 통하여 건물 내의 사람들과 인근지역의 사람들에게 재난 상황에 대해 알려준다.

Apache Camel기반의 MEC를 활용할 경우 백본네트워크를 통하지 않아도 되기 때문에 좀 더 신속하게 사용자들에게 재난상황에 대하여 알릴 수 있으며, 재난으로 인하여 외부와 네트워크가 단절되어도 재난상황에 대한 알림을 사람들에게 전송할 수 있다. 더 나아가 실내위치를 파악할 수 있는 시스템을 구성할 경우, 사람들이 안전하게 건물을 빠져나갈 수 있는 대피경로를 안내해줄 수 있으며, 건물 붕괴 시 매몰된 사람의 위치를 보다 빠르게 파악하여 인명 구조에 도움이 될 수 있을 것으로 예상된다.

참고문헌(Reference)

- [1] J. Jeong, "Disaster Management ICT Trends and Implications in Overseas Major Countries," KISTEP Issue Paper, vol. 9, no. 11, pp. 1-41, Nov. 2014. <http://www.kistep.re.kr/c3/sub3.jsp?brdType=R&bbIdx=8181>
- [2] M. Mardiyono, R. Suryanita, and A. Adnan, "Intelligent Monitoring System on Prediction of Building Damage Index using Neural-Network," TELKOMNIKA Telecommunication, Computing, Electronics and Control, vol. 10, no. 1, pp. 155-164, Mar. 2012. <http://www.iaesjournal.com/online/index.php/IJEECS/article/view/665>
- [3] K. Lorincz, D. J. Malan, T. R.F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, and M. Welsh, "Sensor networks for emergency response: challenges and opportunities," IEEE Pervasive Computing, vol. 3, no. 4, pp. 16-23, Oct.-Dec 2004. <https://doi.org/10.1109/MPRV.2004.18>
- [4] S. Kim, J. Park, "MEC Technology Trends for 5G," ETRI Electronics and Telecommunications Trends, vol. 31, no. 1, pp. 25-35, Feb. 2016. <https://ettrends.etri.re.kr/ettrends/pubreader.do?volume=31&issue=1&page=25&paperno=0905002099>
- [5] <http://camel.apache.org/>
- [6] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," IDC iView: IDC Analyze the Future, vol. 2007, pp. 1-16, Dec. 2012. <https://www.emc.com/leadership/digital-universe/2012iview/index.htm>
- [7] S. Taylor, "The next generation of the Internet revolutionizing the way we work, live, play, and learn," CISCO Point of View, Apr. 2013. <https://communities.cisco.com/docs/DOC-33604>
- [8] The Zettabyte Era: Trends and Analysis; Cisco White Paper, Jun. 2017. <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html>
- [9] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," IEEE Communications Surveys & Tutorials, Early Access. <https://doi.org/10.1109/COMST.2017.2682318>
- [10] ETSI GS MEC 003: Mobile Edge Computing (MEC); Framework and Reference Architecture V1.1.1, Mar. 2016. https://portal.etsi.org/webapp/workprogram/Report_WorkItem.asp?WKI_ID=50984
- [11] C. Ibsen and J. Anstey, *Camel in Action, Second Edition*, Manning Publications, Jul. 2015.
- [12] <http://www.enterpriseintegrationpatterns.com/>
- [13] H. Liu, H. Darabi, and P. Banerjee, "Survey of Wireless Indoor Positioning Techniques and Systems," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 37, no. 6, pp. 1067-1080, Nov. 2017. <https://doi.org/10.1109/TSMCC.2007.905750>
- [14] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, Li Pan, S. Maharjan, and Y. Zhang, "Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks," IEEE Access, vol. 4, pp. 5896-5907, 2016. <https://doi.org/10.1109/ACCESS.2016.2597169>
- [15] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic Resource and Task Allocation for Energy Minimization in Mobile Cloud Systems," IEEE Journal on Selected Areas in Communications, vol. 33, no. 12, pp. 2510-2523, Dec. 2015. <https://doi.org/10.1109/JSAC.2015.2478718>

◎ 저 자 소 개 ◎



하 태 영(Taeyoung Ha)

2012년 연세대학교 전기전자공학과 (공학사)

2012년~현재 연세대학교 대학원 전기전자공학과 석박사통합과정

관심분야 : IoT & Smartphones, Energy Harvesting, Mobile Edge Computing.

E-mail : taeyoungha@yonsei.ac.kr



김 준 성(Junsung Kim)

2015년 연세대학교 전기전자공학과 (공학사)

2015년~현재 연세대학교 대학원 전기전자공학과 석박사통합과정

관심분야 : IoT & Smartphones, Mobile Edge Computing, Fog Computing.

E-mail : june.kim27@yonsei.ac.kr



정 종 문(Jong-Moon Chung)

1992년 연세대학교 전자공학과 (공학사)

1994년 연세대학교 대학원 전자공학과 (공학석사)

1999년 Pennsylvania State University, Dept. of Electrical Engineering (공학박사)

1997년~1999년 Assistant Professor & Instructor, Pennsylvania State University, Dept. of Electrical Engineering

2000년~2005년 Director, Advanced Communication Systems Engineering Laboratory (ACSEL)

2000년~2005년 Director, Oklahoma Communication Laboratory for Networking & Bioengineering (OCLNB)

2000년~2005년 Associate Professor (Tenured), Oklahoma State University, School of Electrical & Computer Engineering

2005년~현재 연세대학교 전기전자공학과 교수

관심분야 : IoT & Smartphones, Smart Cars & ITS, Military Communications, NFV / ICN / SDN, LTE-A / 5G Networks, Cloud Computing & Big Data, Public Safety AR Simulators and Networking.

E-mail : jmc@yonsei.ac.kr