ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 21-04-03

j.inst.Korean.electr.electron.eng.Vol.25,No.4,598~606,December 2021
https://doi.org/10.7471/ikeee.2021.25.4.598
15

상황인지 기반 IoT-MEC 서비스를 위한 계층적 서비스 바인딩 및 자원관리 구조 설계

Hierarchical Service Binding and Resource Allocation Design for Context-based IoT Service in MEC Networks

노 원 종*★

Wonjong Noh*★

Abstract

In this paper, we presents a new service binding and resource management model for context based services in mobile edge computing (MEC) networks. The proposed control is composed of two layers: MEC service bindng control layer (MCL) and user context control layer (UCL). The MCL manages service binding construction, resource allocation, and service policy construction from a system point of view; and the UCL manages real-time service adaptation using meta-objects. Through simulations, we confirmed that the proposed control offers enhanced throughput and content transfer time when it is compared to the legacy computing and control models. The proposed control model can be employed as a key component for the context based various internet-of-things (IoT) services in MEC environments.

요 약

본 논문에서는 mobile edge computing (MEC) 네트워크에서 컨텍스트 기반 IoT 서비스를 위한 새로운 서비스 바인딩 및 리소스 관리 모델을 제시한다. 제안하는 제어는 MEC 서비스 바인딩 제어 계층 (MCL)과 사용자 컨텍스트 제어 계층 (UCL)의 두 가지 계층으로 구성되며, MCL은 시스템 관점에서 서비스 바인딩 구성, 리소스 할당 및 서비스 정책 구성을 관리하고, UCL은 사용자 관점에서 메타 객체를 사용하여 실시간 서비스 적응을 관리한다. 본 논문에서 제안하는 제어 모델은, 실험을 통해 기존 컴퓨팅 모델과 비교할 때 향상된 정보 처리량과 콘텐츠 전송 시간을 제공함을 확인했다. 제안하는 제어 모델은 차세대 MEC 환경에서 컨텍스트 기반의 다양한 사물인터넷 서비스의 핵심 구성 요소로 적용될 수 있다.

Key words : Adaptive Computing, Binding Control, Context IoT, Meta-Objects, Mobile Edge Computing, Resource Control

Ⅰ. 서론

최근, IT 기술의 빠른 발전은, 서비스, 네트워크, 및 컴퓨팅 패턴에 대해 다양한 변화를 가져오고 있다. 첫째, 네트워크 및 단말의 급속한 발전으로 인해, 단순한 애플리케이션에서 다양한 사용자 지향

* School of Software, Hallym University
★ Corresponding author
E-mail : wonjong.noh@hallym.ac.kr, Tel : +82-33-248-2337
※Acknowledgment
This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (Ministry of Science and ICT) under Grants 2020R1F1A1069119.
Manuscript received Oct. 12, 2021; revised Dec. 1, 2021; accepted Dec. 7, 2021.
This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

지능형 애플리케이션으로 서비스가 변경되고 있다[1]. 둘째, 진화된 상호 운용성으로 인해, 셀룰러 및 Wi-Fi 기반 네트워크에서 all-IP, 장치 대 장치(D2D), 애드혹, 및 센서 네트워크와 같은 이기종 네트워크로 네트워크가 다양화되고 있다[2]. 셋째, 임베디드 시스템 및 컴버전스 기술의 발전으로 인해, IoT(Internet-of-Things) 시스템이 일반화되고 있다[3]. 마지막으로 새로운 리소스 가상화 및 네트워크 보안 기술로 인해, 사용자 중심의 지능적인 컴퓨팅이 가능해지고 있다[4]-[6]. 이와 같은, 다양한 변화에 따라, 최근 클라우드 컴퓨팅 및 mobile edge computing(MEC) 환경에서의 상황인식 기반 서비스가 매우 중요해지고 있으며, 이를 효율적으로 지원하기 위한 다양한 상황인식 기반 컴퓨팅 제어 플랫폼이 제안되었다[7]-[12]. Xing[7]은 프로그램 가능한 네트워킹 기술을 통해 가상 네트워크 시스템을 구축 할 수 있도록 함으로써, 인터넷 기반 자원관리를 개선 한 분산 모바일 컴퓨팅 아키텍처를 제시했다. Haw[8]는 최적의 컨텐츠 전달을 지원하는 아키텍처를 제안했다. 제안 시스템은 콘텐츠 제공 서비스 품질(QoS)을 향상시키기 위해 SDN (Software Defined Networking) 및 CCN(Content Centric Networking)과 함께 보강 학습 기반 콘텐츠 전달 프레임 워크를 사용했다. Mitra[9]는 모바일 클라우드 컴퓨팅을 위한 이동성 관리 시스템을 제안했다. 제안 시스템은, 사용자의 애플리케이션 요구 사항과 네트워크 및 클라우드 부하를 기반으로 멀티 홈, 클라우드 네트워크 프로빙, 그리고 클라우드 네트워크 선택 메커니즘 등을 제공했다. Srirama[10]는 모바일 코드의 효율적인 오프로딩를 위해, 모바일 장치 및 클라우드 리소스의 동적인 컨텍스트를 활용하는 프레임 워크를 제안하였다. Zhou[11]는 모바일 클라우드 컴퓨팅에서 사용자 및 서비스 컨텍스트를 효율적으로 인식하는 아키텍처를 제시했다. Gabriel[12]는 모바일 클라우드에서 서비스의 가용성을 증가시키기 위한, 상황인식 기반 에너지 효율적인 소프트웨어 아키텍처를 제안했다. 이는 서비스 복제 기술과 함께 모바일 시스템의 컨텍스트 정보에 따라, 서비스의 복제본을 활성화하고 최대 절전 모드로 전환하는 방식을 기반으로 한다. 본 연구에서는, 보다 효율적인 상황인식 기반 서비스를 위한 컴퓨팅 플랫폼의 계층적 제어 모델을 제안한다. 본 논문의 주요 특징은 다음과 같이 요약 할 수 있다. 첫째, [7]-[12]와 다르게, 본 논문에서 제안된 시스템은 MEC 플랫폼의 제어 계층(CCL)에서, 모든 컨텍스트 인식 IoT 서비스를 고려하여 자원 할당을 거시적으로 제어한다. 또한, 사용자 제어 계층(UCL)에서는 각 상황 인식 서비스를 미시적으로 제어한다. 둘째, [7]-[12]와 다르게, 본 논문에서는, 비균일 서비스 바인딩 모델을 제공한다. 기존 [7]-[12]의 클라우드 컴퓨팅 플랫폼에서는, OMG(Object Management Group)의 공통 객체 브로커 아키텍처(CORBA) [13], Java의 원격 메소드 호출(RMI) [14], Open Group의 DCE (Distributed Computing Environment) [15], ISO의 RM-ODP [16] 등의 균일 바인딩 모델이 적용되어 왔다. 기존 모델들은 주로 상호 운용성과 이식성에 중점을 두고 있으며, 애플리케이션 작성자는 균일한 계산 모델을 볼 수 있고, 다른 시스템, 즉 다른 네트워크, 최종 시스템, 통신 프로토콜, 운영 체제 및 언어 환경의 이질성으로부터 격리된다는 장점이 있다. 이러한 접근 방식은 분산 응용 프로그램의 광범위한 배포를 향상 시켰으나, 플랫폼 수준에서 응용 프로그램에게 최적화된 서비스별 컴퓨팅 시맨틱을 지원하지 않고, 모든 애플리케이션에서 공통적으로 사용되는 최소한의 시맨틱만을 지원하고 있다. 즉, 대부분의 응용프로그램 별 요구되는 컴퓨팅 시맨틱은 주로 최종 사용자가 직접 처리하였다. 이로 인해 효율성이 낮고 보안 수준이 낮은 서비스가 발생하는 단점이 발생하였다. 반면, 제안하는 비균일 서비스 바인딩 모델은 사용자 수준이 아닌 제어 플랫폼 수준에서 응용 프로그램에 적합한 컴퓨팅 시맨틱을 제공함으로서, 시스템에서 보다 향상된 성능을 응용프로그램에게 제공하게 되었다. 셋째, [7]-[12]와 다르게, 본 논문에서 제안된 시스템은 메타-객체 기반의 실시간 적응형 서비스 바인딩 모델을 사용한다. 즉, 제공되는 컴퓨팅 제어 환경이 서비스 컨텍스트의 변경에 따라 적응적으로 조정될 수 있게 된다.

본 논문에서는, [7]-[12]에서의 기존 균일 컴퓨팅 모델 방식과 비교할 때, 제안하는 계층적 제어 모델이 향상된 전송률을 제공함을 실험을 통해 보였다. Ⅱ장에서는 제안하는 시스템 모델을 설명하며, Ⅲ장에서는 제안하는 모델에 적용된 주요 제어 알고리즘과 절차들을 제시한다. 끝으로, 성능 평가 및 결론을 Ⅳ 및 Ⅴ에 각각 제시하였다.

II. 시스템 모델

그림 1은, 상황인지 기반 IoT 서비스를 위해 제안하는 MEC 서비스 바인딩 및 자원할당 제어 구조를 보여준다. 사용자는, 네트워크 제공자에 의해 제공되는 상황 인식 서비스에 액세스 할 수 있으며, 사용자가 서비스에 액세스하면 사용자와 서비스 간에 연결이 생성된다. 이를 서비스 바인딩 이라고 한다. 이 서비스 바인딩은 컨텍스트 스위칭, 자원 할당, 작업 스케줄링 및 서비스 적응을 담당하는 제어 엔진에 의해 유지된다.

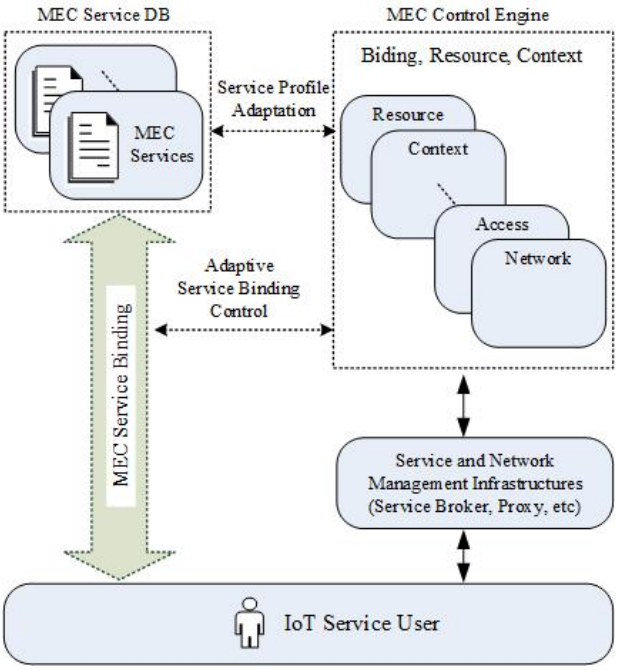

Fig. 1. Proposed MEC service control architecture.
그림 1. 제안하는 MEC 서비스 제어 구조

1. 상황인식 서비스 모델

시스템에서, 각 상황 인식 서비스 S_i는 여러 가지 서비스 컨텍스트를 갖고 있으며, 그 중에서 자신의 서비스 컨텍스트를 다음을 통해 결정한다.

$$C(S_i) = f_i\left(c_{i1}, c_{i2}, c_{i3}, \ldots, c_{in}\right)$$

여기서, c_{ij}는 서비스 S_i에 대한 컨텍스트를 결정하는 j번째 컨텍스트-인자요소(factor)를 의미한다. 또한 컨텍스트-인자요소 정보들을 하나의 서비스 컨텍스트로 매핑하는 서비스 컨텍스트 매핑 함수를 $f(\bullet)$로 표현한다. 서비스 S_i는 결정된 서비스 $C(S_i)$에 따라 맞춤형 서비스를 적응적으로 수행하게 된다.

2. 서비스 바인딩 모델

그림 2는 어플리케이션-level 바인딩과 전송-level 바인딩으로 구성된 계층적 서비스 바인딩 모델을 보여준다.

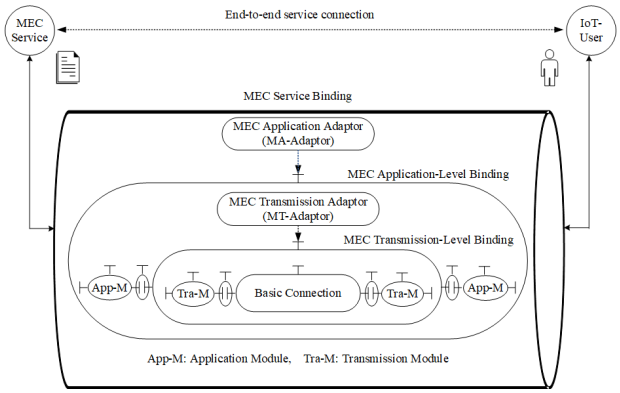

Fig. 2. Proposed hierarchical service binding model.
그림 2. 제안하는 계층적 서비스 바인딩 모델

애플리케이션-level 바인딩은 애플리케이션 관련 객체, 즉 사용자 인터페이스, 언어 및 애플리케이션 프로토콜 간의 연관을 나타낸다. 전송-level 바인딩은 데이터 전송 관련 객체, 즉 압축기, 필터, 변조기, 큐, 캐시 및 전송 프로토콜 간의 연관성을 나타낸다. 여기서 서비스 바인딩에는 고유한 바인딩 식별자(BID)가 부여되며, 바인딩 내부의 각 객체에도 고유한 시스템 객체 식별자(OID)가 부여된다. 이러한 식별자는 MEC 컴퓨팅 플랫폼에 의한 바인딩의 동적 제어에 사용된다. 본 논문에서, 서비스 바인딩의 실시간 동적 적응을 가능하게 하기 위해 제안된 제어 구조는 시스템의 내부 동작을 검사하고 변경할 수 있는 메타-클래스 또는 메타-오브젝트 기반 reflective 시스템을 이용한다. 여기서, reflective 시스템은 소프트웨어 엔지니어링에 대한 블랙 박스 접근 방식의 한계를 극복하고, 개방형 엔지니어링을 위한 수단을 제공하게 된다[17]. 그리고, 메타-클래스는 바인딩을 제어하는 일련의 함수들을 지칭하며, 객체 지향 클래스에서 다음과 같은 메소드의 형태로 정의된다.

- REPLACE(BID, 객체/프로토콜, OID, OID*) : 서비스 바인딩 BID에서, OID에 해당되는 기존 시스템 객체 또는 프로토콜을 OID*를 갖는 새

로운 것으로 대체.

- INSERT(BID, 객체/프로토콜, OID*, pos) : 서비스 바인딩 BID의 pos 위치에 OID*를 갖는 새로운 시스템 객체 또는 프로토콜을 삽입.
- REMOVE(BID, 객체/프로토콜, OID) : 서비스 바인딩 BID에서 OID를 갖는 기존 객체 또는 프로토콜을 제거.
- MONITOR(BID, 객체/프로토콜, OID) : 서비스 바인딩 BID에서 OID를 갖는 객체 또는 프로토콜의 성능을 모니터링.

이러한 메소드가 정의된 오브젝트 지향 클래스의 인스턴스를 메타-오브젝트라고 한다. 제안된 메타-오브젝트 공간은 configuration 및 functional 메타-오브젝트들로 구성된다. Configuration 메타-오브젝트는 바인딩 내부의 메소드, 시스템 오브젝트 및 연결 토폴로지의 구성을 조사하고 제어한다. 예를 들어, configuration 메타-오브젝트는 REPLACE (BID, object, FIFO, CB-LLQ)를 통해 바인딩 내부의 큐잉구조를 변화시킨다. 한편, functional 메타-오브젝트는 바인딩의 프로토콜, QoS 및 보안 레벨을 조사하고 제어한다. 예를 들어, functional 메타-오브젝트는 REPLACE(BID, 프로토콜, TCP, UDP)를 통해 응용 프로그램 상태에 따라 TCP에서 UDP로 바인딩의 전송 프로토콜을 수정한다.

3. 제어 엔진 모델

그림 3은 MCL과 UCL로 구성된 계층적 MEC 제어 엔진 모델을 보여준다.

첫째, MCL은 MEC 최적화 제어 모듈(MOM), MEC 서비스 제어 모듈(MSM) 및 MEC 네트워크 제어 모듈(MNM)의 세 가지 기본 모듈로 구성되며, 시스템 전체의 리소스 및 서비스 관리를 담당합니다. MOM은 다음의 하위 구성 요소들을 사용하여 MEC 리소스 및 서비스 일정을 제어한다.

- MEC 시스템 자원 할당 컴포넌트 : 서비스 유틸리티 극대화 측면에서 네트워크 프로세서, 스토리지, 대역폭 및 전송 전력과 같은 리소스를 서비스에 최적으로 할당하는 기능.
- MEC 스케줄링 컴포넌트 : 서비스의 활성화 및 비활성화 시간을 관리하는 기능.

MSM은 다음의 하위 구성 요소들을 사용하여 사용자 서비스 구성을 제어한다.

- MEC 서비스 프로파일 컴포넌트 : 서비스 컨텍스트에 따라 컨텍스트 인식 기반 서비스가 어떻게 작동해야 하는지를 관리하는 기능. 또한 새 서비스 프로파일을 생성하고 기존 서비스 프로파일을 업데이트하거나 삭제하며, 서비스 프로파일간의 서비스 무결성 및 서비스 충돌을 제어하는 기능.
- MEC 서비스 적응 컴포넌트 : 서비스 적응 정책을 관리하는 기능. 즉 컨텍스트 전환에 따라 서비스가 어떻게 변해야 하는지를 제어함.

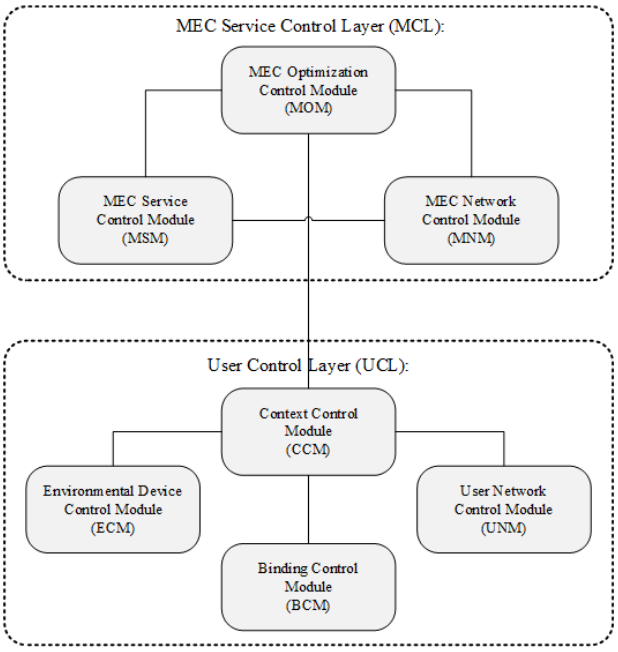

Fig. 3. Proposed hierarchical MEC control engin.
그림 3. 제안하는 계층적 MEC 제어 엔진 모델

MNM은 다음의 하위 구성 요소들을 사용하여 MEC 관련 네트워크 정보를 제어한다.

- MEC 액세스 컴포넌트 : 로그인, 연결 설정 및 연결 종료와 같은 사용자 및 서비스 인증을 관리하는 기능을 하며, 또한 이 구성 요소는 로그인 절차를 통과 한 사용자에게 사용 가능한 모든 상황 인식 서비스의 리스트를 보여줌.
- MEC 보안 컴포넌트 : 예기치 않은 사용자 또는 행동으로부터 클라우드 시스템을 보호하는 기능.
- MEC 네트워크 인터페이스 컴포넌트: 클라우드 무선 액세스 네트워크(Cloud-RAN) 라고 하는 통일 된 네트워크 액세스 인터페이스를 관리한다. 표준화 된 네트워크 인터페이스를 통해 LTE, 3G, Wi-Fi 및 Bluetooth와 같은 네트워크를 지원하는 기능.
- MEC/클라우드 코어 네트워크 모니터링 컴포넌

트: 시스템 전반의 네트워크 정보, 즉 MEC/클라우드 네트워크 상태 및 리소스 사용 상태를 모니터링하는 기능.

둘째, UCL은 컨텍스트 관리 및 실시간 서비스 바인딩 적응을 담당하며. 컨텍스트 제어 모듈(CCM), 바인딩 제어 모듈(BCM), 사용자 네트워크 제어 모듈(UNM) 및 환경 장치 제어 모듈(ECM)로 구성되어 있다. 우선, CCM은 다음의 하위 구성 요소들을 사용하여 서비스 컨텍스트를 제어한다.

- 컨텍스트 수집 컴포넌트 : 환경, 네트워크 및 사용자로부터 하위 컨텍스트 정보를 수집하는 기능.
- 컨텍스트 해석 컴포넌트 : 하위 컨텍스트 정보를 측정 가능한 실수값으로 변환하는 컨텍스트 해석 기능.
- 컨텍스트 추론 컴포넌트 : 하위 컨텍스트 인자요소들로부터, 서비스 컨텍스트를 추론 기능.

BCM은 다음의 하위 구성 요소들을 사용하여, 컨텍스트의 변경에 따라 서비스 바인딩 적응을 제어한다.

- QoS 컴포넌트 : A-어댑터와 T-어댑터를 통해 애플리케이션과 전송 바인딩에 목표 QoS를 설정하고, 달성된 QoS를 모니터링 기능.
- 실시간 코드 제어 컴포넌트 : A-어댑터 및 T-어댑터를 통한 DLL(dynamic link library) 및 실시간 작업을 관리 기능.

UNM은 다음의 하위 구성 요소들을 사용하여 사용자 관련 네트워크 정보를 제어한다.

- 이동성 컴포넌트 : 홈 위치 등록기(HLR) 또는 방문자 위치 등록기(VLR)를 통해 사용자 이동성을 관리하는 기능.
- 액세스 네트워크 모니터링 컴포넌트 : 사용자에게 할당 된 네트워크 리소스를 모니터링하는 액세스 네트워크 모니터링 기능.

ECM은 다음의 하위 구성 요소들을 사용하여 센서 및 액추에이터와 같은 환경 장치를 제어한다.

- 센싱 컴포넌트 : 환경 센서로부터 하위 컨텍스트 인자요소 정보를 수집하고 감지하는 기능.
- 실행 컴포넌트 : 실시간 센서 작동을 관리하는 기능.

III. 주요 제어 알고리즘

본 논문에서 제안하는 컴퓨팅 제어의 주요 알고리즘과 절차는 다음과 같다.

1. 컨텍스트 결정

CCM은 다음과 같이 절차로 서비스 컨텍스트를 결정한다.

① ECM은 센서로부터 컨텍스트 인자요소(factor) 데이터를 수집하여 CCM에 보고한다.

② UNM은 네트워크 데이터를 수집하여 CCM에 보고한다.

③ CCM은 그림 4의 기계 학습 모듈을 이용하여 서비스 문맥을 결정한다.

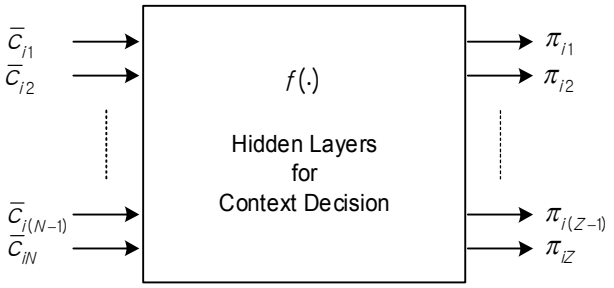

Fig. 4. Machine-learning based context decision.
그림 4. 기계 학습 기반 컨텍스트 결정 모델

여기서, $\bar{C}_{ik}$는 서비스 S_i의 k번째 컨텍스트 인자요소 c_{ik}에 대한 측정값이며, π_{ik}는 서비스 S_i의 k번째 서비스 컨텍스트를 의미한다. 이를 바탕으로, 최종적으로 적용할 서비스 컨텍스트를 다음과 같이 결정한다.

$$C(S_i) = \arg max_{j \in 1,\dots Z} \pi_{ij}$$

이를 위해, CCM은 사전에 컨텍스트 트레이닝 데이터를 사용하여 숨겨진 계층함수 $f(\bullet)$을 찾아 놓는다.

2. 자원 할당

MOM은 주기적으로 최적의 클라우드 리소스 할당을 수행한다.

① 각각의 서비스들은 QoS 요소로서, rate(r), packet loss(l), delay(d)를 갖고 있다고 가정한다.

② 각 QoS 변수는 할당된 자원 θ의 함수로서, 값을 갖는다: $r(\theta), l(\theta), d(\theta)$

③ 만약, i번째 자원이 유저 n에게 $\theta_{n,i}$ 만큼 할당된다면, 유저 n이 얻는 서비스 유틸리티는 다

음과 같이 나타낼 수 있다.

$$u_n^i(r(\theta_{n,i}),l(\theta_{n,i}),d(\theta_{n,i})).$$

④ 상황인식 기반 서비스 사용자 1....N 들에게 최적으로 클라우드 리소스를 할당하기 위해 COM은 다음과 같은 최대화 문제를 해결한다.

$$\max\sum_{n=1}^{N}\sum_{i=1}^{M}c_n u_n^i(r(\theta_{n,i}),l(\theta_{n,i}),d(\theta_{n,i}))$$

여기서 c_n은 각 사용자의 서비스 가중치를 나타낸다. 한편, 최적화 문제를 해결하는데 있어서, 다음과 같은 자원 제약이 존재한다.

$$\sum_{n=1}^{N}\theta_{n,j}\le\Theta_i,\ \text{for } i=1....M,$$

여기서 Θ_i는 자원 i의 최대 사용 가능량을 나타낸다. 제약 조건은 convex 영역을 형성하며, 유틸리티 함수가 사용자들 사이의 자원 공정성을 반영할 수 있는 로그함수라면, 목적 함수 역시 convex 함수가 된다. 이것은, Karush-Kuhn-Tucker(KKT) 조건과 Lagrangian relaxation[18] 방법을 사용하여 최적의 자원을 찾을 수 있다는 것을 의미한다. 즉, COM은 아래의 문제를 해결함으로서, 최적의 자원 할당을 결정한다

$$\max\sum_{n=1}^{N}\sum_{i=1}^{M}c_n u_n^i(r(\theta_{n,i}),l(\theta_{n,i}),d(\theta_{n,i}))+\sum_{i=1}^{M}\lambda_i(\Theta_i-\sum_{n=1}^{N}\theta_{n,j})$$

3. 동적 서비스 바인딩 적응 관리

제안된 제어 모델은 서비스 컨텍스트의 변동에 따라 서비스 바인딩의 실시간 적응을 수행하며, 이는 그림 5에 나와 있다.

① ECM과 UNM은 서비스 환경을 모니터링하며, 환경이 변경 될 때마다 변경된 정보를 CCM에게 알린다. BCM은 바인딩 상태를 모니터링하며, 마찬가지로, 환경의 변화가 있을 때마다 BCM은 CCM에게 알린다.

② CCM은 새로운 서비스 컨텍스트를 업데이트한다.

③ CCM은 서비스 컨텍스트 정보를 MOM으로 보낸다.

④ MOM은 MCM과 MNM으로부터 각각 서비스와 네트워크 정보를 받는다.

⑤ MOM은 새로운 리소스 할당을 결정한다.

⑥ MOM은 새로운 자원 할당 및 적응 스크립트를 CCM으로 전달한다.

⑦ CCM은 BCM에 adaptation-rule을 전달한다.

⑧ BCM은 MA-어댑터와 MT-어댑터를 통해 서비스 바인딩을 제어한다.

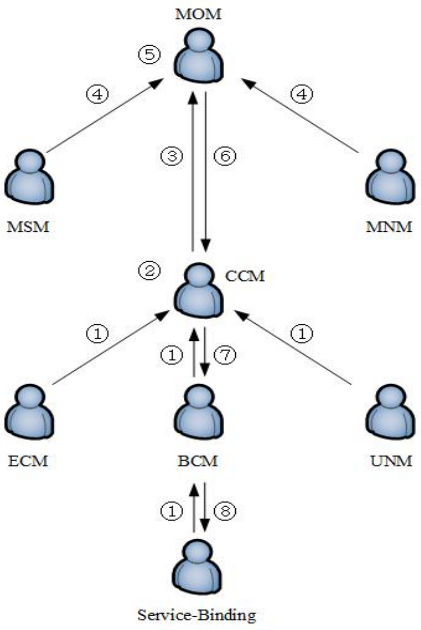

Fig. 5. Proposed dynamic binding adaptation.
그림 5. 제안하는 동적 바인딩 적응 절차

Ⅳ. 구현 및 평가

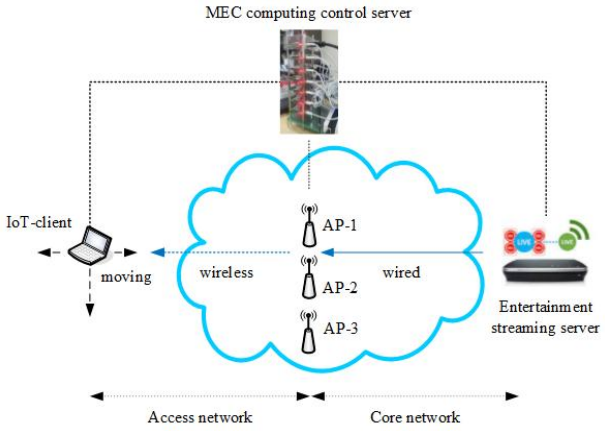

Fig. 6. Multimedia transmission testbed.
그림 6. 멀티미디어 전송 테스트베드 네트워크

본 논문은, 제안하는 제어 플랫폼의 성능을 확인하기 위해, 그림 6과 같은 테스트베드를 구성하였다.

위 테스트베드에서, 상황인지 기반 서비스로서, 멀티미디어 스트리밍 서비스를 시험하였으며, 이 때, 서비스의 컨텍스트는 신호 대 간섭 및 잡음 비율(SINR)을 기반으로 하는 네트워크 상태에 따라서 아래와 같이 정의하였다.

Table 1. SINR to context mapping rule.
표 1. SINR에서 컨텍스트 매핑 규칙

Context	SINR
Best	19.1 dB ~ 100 dB
Good	13.2 dB ~ 19.1 dB
Normal	6.2 dB ~ 13.2 dB
Bad	1.7 dB ~ 6.2 dB
Worst	-100 dB ~ 1.7 dB

테스트베드에서, 멀티미디어 클라이언트는 최초에는 액세스 포인트(AP)-1에 연결되어 있으면서, AP-1을 통해 미디어 서버에서 스트리밍 데이터를 수신받는다. AP의 송신 전력 및 대역폭은 각각 30 (dBm) 및 5(MHz)로 설정되어 있다. 사용자는 이동 속도 1(m/sec)로 무작위로 이동하며, 이동하는 사용자는 SINR의 관점에서 서비스 컨텍스트의 변동을 겪게 된다. 그림 7은 제안 시스템과 기존 시스템의 처리율을 비교하여 보여준다. 네트워크 상태의 변화에 따라, 기존 균일 컴퓨팅 플랫폼은 적응적 바인딩을 적용 할 수 없으며, 다만, 아래의 표에 따른 변조 및 코딩 속도만을 제어한다.

Table 2. Modulation and coding mapping rule.
표 2. 변조 및 코딩 매핑 규칙

Context	SINR	Modulation	Code rate
Best	19.1 ~ 100	64-QAM	5/6
Good	17.6 ~ 19.1	64-QAM	3/4
Good	15.6 ~ 17.6	64-QAM	2/3
Good	13.2 ~ 15.6	16-QAM	5/6
Normal	12.2 ~ 13.2	64-QAM	1/2
Normal	12.0 ~ 12.2	16-QAM	3/4
Normal	10.3 ~ 12.0	16-QAM	2/3
Normal	7.73 ~ 10.3	16-QAM	1/2
Normal	6.18 ~ 7.73	QPSK	5/6
Bad	5.17 ~ 6.18	QPSK	3/4
Bad	3.71 ~ 5.17	QPSK	2/3
Bad	1.72 ~ 3.71	QPSK	1/2
Worst	-100 ~ 1.72	QPSK	1/4

반면에, 제안하는 플랫폼은, 표 2의 변조 및 코딩 외에도, 아래와 같은 적응적 서비스 바인딩을 수행한다.

- 네트워크 상태가 'Best'에서 'Good'으로 변경되면 제안 된 플랫폼은 압축률 모듈을 사용하여 서비스 속도를 유지한다. 제안 된 플랫폼은 균일 한 시스템보다 대략 10% 높은 처리량을 제공한다.
- 네트워크 상태가 'Good'에서 'Normal'로 변경되면 제안된 플랫폼은 보다 강력한 압축 모듈을 사용하여 서비스 속도를 유지하게 된다. 제안된 플랫폼은 균일 한 시스템보다 대략 20% 높은 처리량을 제공한다.
- 네트워크 상태가 'Normal'에서 'Bad'로 바뀌면 제안된 플랫폼은 더 나은 경로를 찾으려고 시도한다. 다른 AP, 즉 AP-2나 AP-3를 발견하면 그 중 더 나은 신호를 갖는 AP로 동적으로 전환된다. 제안된 플랫폼은 균일한 시스템보다 대략 30% 높은 처리량을 제공한다.
- 네트워크 상태가 'Bad'에서 'Worst'로 변경되면 제안된 플랫폼은 글로벌 바인딩 적응을 요청한다. 이를 통해, MOM 및 CCM은 서비스 사용자에게 더 많은 자원을 할당함으로서, 향상된 성능을 제공하게 된다.

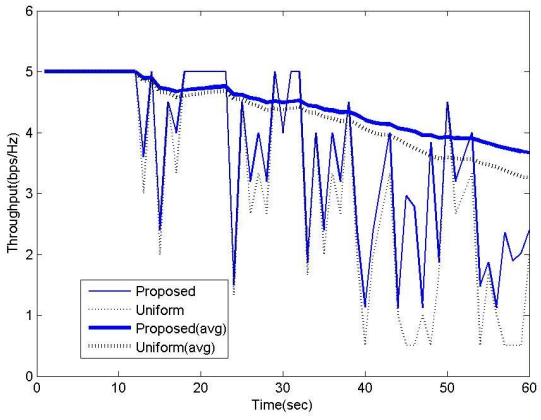

Fig. 7. Throughput.
그림 7. 전송량

한편, 그림 8은 사용자 수를 10 명으로 늘리면서, 파일 전송 시간을 비교하였다.

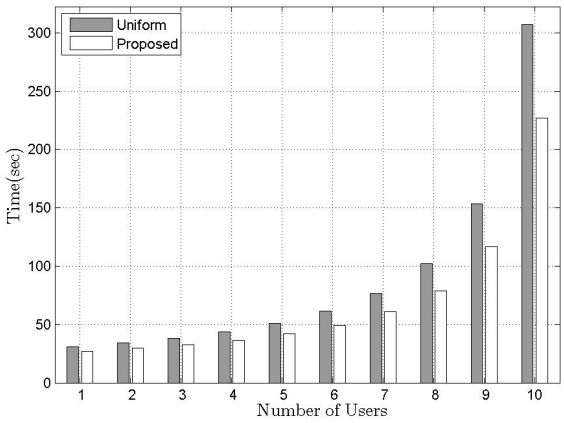

Fig. 8. File transfer time.
그림 8. 파일 전송 시간

기존 시스템과 제안하는 시스템에서의 파일전송 시간의 차이는 사용자 수가 증가함에 따라 증가한다는 것을 알 수 있다. 사용자 수가 10명에 근접 할 때, 제안된 컴퓨팅 플랫폼은 약 25% 더 짧은 전송 시간을 제공한다. 예를 들어, 1(GB) 동영상 파일을 다운로드하기 위해서, 기존의 균일 플랫폼에서는 307(sec)가 소요되지만, 제안된 컴퓨팅 플랫폼은 단지 227(sec)이 소요된다.

Ⅴ. 결론

본 논문은 상황인식 기반 IoT 서비스를 위한 계층적 클라우드 컴퓨팅 모델을 제시했다. 제안된 모델은 MEC 서비스 제어 계층(MCL)과 사용자 제어 계층(UCL)으로 구성된다. 먼저, MCL은 상황인식 기반 서비스를 거시적으로 제어하며, 모든 상황인식 기반 서비스에 대해 최적의 자원 할당 및 스케줄링을 결정하고 서비스 프로파일 적응 정책을 제어한다. 한편, UCL은 상황인식 기반 서비스를 미시적으로 제어하며, 이는 서비스 컨텍스트의 변환에 따라 실시간 서비스 바인딩의 적응을 가능하게 하는 configuration 및 functional 메타-오브젝트를 지원한다. 본 논문은, 제안하는 계층적 제어 모델이 기존의 균일한 컴퓨팅 모델보다 향상된 성능을 제공함을 실험을 통해 확인하였으며, 본 논문에서 제안된 모델은 상황인지 기반 서비스를 위한 핵심 기반 구조로 사용될 수 있다. 향후 연구로서, 제안된 컴퓨팅 제어 모델의 활용도를 극대화할 수 있는 진화된 적응 기법 및 이동성 관리 프레임워크 구조에 대한 설계를 추가로 진행할 예정이다.

References

[1] R. Gupta, V. Sekhri, and A. Somani, “CompuP2P: An architecture for internet computing using peer-to-peer networks,” *IEEE Trans. Parallel and Distributed Systems,* vol.17, no.11, pp.1306-1320, 2006. DOI: 10.1109/TPDS.2006.149

[2] T. A. Pham, X. Li, G. Cong, Z. Zhang, “A General Recommendation Model for Heterogeneous Networks,” *IEEE Trans. on Knowledge and Data Engineering, PrePrints,* 2016.
DOI: 10.1109/TKDE.2016.2601091.

[3] S. M. Riazul Islam, M. N. Uddin, and K. S. Kwak, “The IoT: Exciting Possibilities for Bettering Lives: Special application scenarios,” *IEEE Consumer Electronics Magazine*, vol.5, no.2, pp.49-57, 2016. DOI: 10.1109/MCE.2016.2516079

[4] F. Hao, T. Lakshman, S. Mukherjee, and H. Song, “Enhancing dynamic cloud-based services using network virtualization,” *ACM Sigcomm. Computer Communication Review*, vol.40, no.1, pp.67-74, 2010. DOI: 10.1145/1592648.1592655

[5] D. Sanchez, A. Martin, D. Proserpio, and P. Cabarcos, “Media cloud: an open cloud computing middleware for content management,” *IEEE Trans. Consumer Electronics*, vol.57, no.2, pp.970-978, 2011.

[6] W. Zhu, C. Luo, J. Wang, and S. Li, “Multimedia cloud computing: an emerging technology for providing multimedia services and applications,” *IEEE Signal Processing Magazine,* vol.28, no.3, pp.59-69, 2011.
DOI: 10.1007/978-3-030-62124-7_18

[7] T. Xing, and D. Huang, “MobiCloud: A geo-distributed mobile cloud computing platform,” *IEEE International Conference on Systems Virtualization Management*, pp.164-168, 2012.

[8] R. Haw, M. Alarm, and C. Hong, “A context-aware content delivery framework for QoS in mobile cloud,” *IEEE Network Operations and Management Symposium,* pp.1-6, 2014.
DOI: 10.1109/APNOMS.2014.6996607

[9] K. Mitra, S. Saguna, C. Ahlund, and D. Lulea,

"A mobility management system for mobile cloud computing," *IEEE Wireless Communications and Networking Conference*, pp.1608-1613, 2015.
DOI: 10.1109/WCNC.2015.7127708
[10] B. Zhou, A. Dastjerdi, R. Calheiros, S. Srirama, and R. Buyya, "mCloud: A context-aware offloading framework for heterogeneous mobile cloud," *IEEE Trans. on Services Computing,* issue.99, pp.1-1, 2015.
DOI: 10.1109/TSC.2015.2511002
[11] J. Zhou, and Z. Zhang, "The context awareness architecture in mobile cloud computing," *IEEE International Symposium on Computational Intelligence and Design,* pp.302-305, 2012.
DOI: 10.1109/ISCID.2012.83
[12] G. Contreras, J. Garrido, S. Diaz, C. Dominguez, "A context-aware architecture supporting service availability in mobile cloud computing," *IEEE Trans. on Services Computing*, no.99, pp.1-1, 2016. DOI: 10.1109/TSC.2016.2540629
[13] Object Management Group, "Object request broker," CORBA Specification 3.2, 2011.
[14] A. Wollrath, R. Riggs, and J. Waldo, "A distributed object model for the Java system," *Proceedings of the USENIX Conference on Object-Oriented Technologies,* pp.219-232, 1996.
[15] Open Group, "Telecommunications information networking architecture," *Distributed Computing Environment,* 2005.
[16] P. Linington, Z. Milosevic, A. Tanaka, and A. Vallecillo, "Building enterprise systems with ODP: An Introduction to Open Distributed Processing," *Chapman & Hall/CRC Press,* 2011.
[17] P. Maes, "Concepts and experiments in computational reflection," Conference on Object-Oriented Programming Systems, Languages and Applications, pp.147-155, 1987.
[18] S. Boyd, and L. Vandenberghe, "Convex Optimization," Cambridge University Press, 2004.

BIOGRAPHY

Wonjong Noh (Member)

1998 : BS degree in Electronics Engineering, Korea University.
2000 : MS degree in Electronics Engineering, Korea University.
2005 : PhD degree in Electronics Engineering, Korea University.
2005~2007 : Post-Doc. Researcher, Purdue University, UC-Irvine, USA
2007~2014 : Research Engineer, Samsung Electronics.
2014~2018 : Assistant Professor, Gyeonggi University of Science and Technology.
2019~present : Associate Professor, Hallym University.