Journal of the Korea Academia-Industrial cooperation Society
Vol. 19, No. 6 pp. 46-54, 2018

https://doi.org/10.5762/KAIS.2018.19.6.46
ISSN 1975-4701 / eISSN 2288-4688

IoT에서 효율적인 서비스 제공을 위한 이름 기반 서비스 탐색 메커니즘

조국현[1], 김정재[1], 류민우[2], 차시호[3*]
[1]광운대학교 소프트웨어학부, [2]KT 융합기술원, [3]청운대학교 멀티미디어학과

A Name-based Service Discovering Mechanism for Efficient Service Delivery in IoT

Kuk-Hyun Cho[1], Jung-Jae Kim[1], Minwoo Ryu[2], Si-Ho Cha[3*]
[1]**Department of Computer Software, Kwangwoon University**
[2]**Institute of Convergence Technology, Korea Telecom (KT)**
[3]**Department of Multimedia Science, Chungwoon University**

요 약 IoT는 다양한 디바이스들이 통신을 통해 사용자에게 서비스를 제공하는 환경이다. IoT의 특성으로 인해 데이터들은 이종간의 정보시스템에 분산되어 저장된다. 이러한 상황에서 IoT 엔드 애플리케이션은 데이터가 어디에 있는지 또는 스토리지의 형태가 어떠한지 알 수 없어도 데이터를 액세스할 수 있어야 한다. 이러한 메커니즘을 SD(Service Discovery)라고 한다. 그러나 현재까지의 SD 구조는 물리적 디바이스를 중심으로 탐색하기 때문에 몇 가지 문제점이 발생한다. 첫째, 물리적 위치에 따른 서비스 탐색으로 인해 반환시간이 증대된다. 둘째, 디바이스와 서비스를 따로 관리하는 데이터 구조가 요구된다. 이는 관리자의 서비스 구성복잡도를 증가시킨다. 이로 인해 디바이스 중심의 SD 구조는 실제 IoT에 적용하기에는 적합하지 않은 구조로 되어 있다. 이러한 문제점을 해결하기 위하여 본 논문에서는 NSSD(Name-based Service Centric Service Discovery)라는 SD 구조를 제안한다. NSSD는 이름 기반의 중앙집중형 SD를 제공하며 IoT 에지 게이트웨이를 캐싱 서버로 사용해 서비스 탐색속도를 향상시킨다. 기존의 DNS와 DHT 기반 DS 구조와의 시뮬레이션을 통해 NSSD가 평균 반환시간에 있어 약 2배 정도 향상된 성능을 제공함을 입증하였다.

Abstract The Internet of Things (IoT) is an environment in which various devices provide services to users through communications. Because of the nature of the IoT, data are stored and distributed in heterogeneous information systems. In this situation, IoT end applications should be able to access data without having information on where the data are or what the type of storage is. This mechanism is called Service Discovery (SD). However, some problems arise, since the current SD architectures search for data in physical devices. First, turnaround time increases from searching for services based on physical location. Second, there is a need for a data structure to manage devices and services separately. These increase the administrator's service configuration complexity. As a result, the device-oriented SD structure is not suitable to the IoT. Therefore, we propose an SD structure called Name-based Service-centric Service Discovery (NSSD). NSSD provides name-based centralized SD and uses the IoT edge gateway as a cache server to speed up service discovery. Simulation results show that NSSD provides about twice the improvement in average turnaround time, compared to existing domain name system and distributed hash table SD architectures.

Keywords : Internet of Things (IoT), Service Discovery (SD), Name-based Service, IoT edge-Gateway, SNT

이 논문은 2016년도 광운대학교 연구년에 의하여 연구되었음.
*Corresponding Author : Si-Ho Cha(Chungwoon Univ.)
Tel: +82-32-770-8205 email: shcha@chungwoon.ac.kr
Received April 30, 2018 Revised May 14, 2018
Accepted June 1, 2018 Published June 30, 2018

1. 서론

IoT는 다양한 디바이스들이 서로 간의 통신을 통해 사용자에게 서비스를 제공하는 환경으로, IoT 내의 다양한 디바이스들은 인터넷으로 연결되어 사용자나 센싱을 통해 주변 환경 정보 (실제 사물의 상태나 이벤트에 대한 글로벌 데이터)를 얻게 된다[1,**19**]. 현재 IoT의 디바이스와 서비스는 대부분 개별적이고 독립적인 도메인 플랫폼과 데이터베이스를 사용하고 있으며, IoT 디바이스 데이터는 공급업체별로 파편화되어 있다[2]. 즉 IoT에서의 데이터는 이기종 정보시스템에 저장되어 분산되어 존재한다[3]. 이러한 상황에서 사용자의 엔드 애플리케이션은 서비스를 위해 특정 데이터를 다른 엔드 애플리케이션에 요구한다. 이는 결과적으로 각각의 디바이스에 있는 애플리케이션에서 필요로 하는 데이터에 액세스할 수 있는 메커니즘이 요구된다**[15-18]**. 또한, 요청 애플리케이션은 데이터가 어디에 있는지 혹은 스토리지의 형태가 어떠한지 알 수 없어도 가능해야 하며, 이러한 메커니즘을 SD(Service Discovery)라 한다[4].

이러한 SD를 제공하는 일반적인 방법은 DNS나 프록시 역할을 하는 구조를 만드는 것이다. 이러한 구조의 종류는 크게 DNS 기반 방식과 DHT(Distributed Hash Table) 기반 방식의 두 종류로 나뉘며, 현재는 DHT를 활용해 SD 구조를 구성하는 연구가 활발히 이루어지고 있다. DHT를 이용한 SD 구조는 빠른 응답시간과 디바이스 측면의 큰 확장성 등 매우 많은 장점이 있다. DNS 기반 구조 또한 현재 시스템과의 확장성 면에서 매우 큰 장점이 있다[5]. 그러나 현재 연구되고 있는 SD 구조는 주로 디바이스에 집중되고 있어서 다음의 두 가지 문제로 인해 실제 IoT 환경에는 적합하지 않다.

첫째, 디바이스 중심의 구조는 디바이스에 대한 검색만 제공한다. 그러나 실제 IoT 환경은 멀티 도메인 상에서 모바일 서비스를 중심으로 이루어져 있다. 예를 들어, 사용자가 위치 기반의 서비스를 받는 디바이스를 가지고 이동할 경우 기존의 SD는 디바이스 중심이기 때문에, 자신의 위치를 기반으로 서버에 재요청을 해야지만 서비스를 검색할 수 있다. 이는 데이터에 대한 응답시간을 저해하는 요인이 된다.

둘째, 기존의 SD 구조는 디바이스 중심의 확장성을 제공한다. DHT는 키와 값의 1:1 매칭 구조로 이루어져 있고 피어(peer)를 기반으로 복잡한 구조를 갖는다. 또한, 서비스 제공자가 DHT에 디바이스를 등록하거나 삭제하는 과정은 자동으로 이루어지지만, 서비스를 추가하거나 삭제하는 등의 서비스 확장은 제공하지 않는다. 이는 관리자의 서비스 구성복잡도를 증가시키는 요인이 된다.

이러한 문제점들로 인해 디바이스 중심의 SD 구조를 서비스 중심으로 이루어진 실제 IoT에 적용하기에는 적합하지 않다. 따라서 IoT 환경에서는 IoT의 이동성, 이기종성, 높은 분산성이라는 고유의 특성을 고려한 서비스 중심의 SD 구조를 적용하여야 한다. 이를 위해 본 논문에서는 IoT에서 효율적인 서비스 제공을 위한 서비스 중심의 SD 구조(NSSD, Name-based Service centric Service Discovery)를 제안한다. 본 논문에서 제안한 서비스 중심의 SD 구조는 다음과 같은 3가지 특성을 갖는다: 1) 서비스 이름 기반 SD 구조, 2) 서비스 중심 SD 구조, 3) IoT 에지 게이트웨이 사용. 본 논문에서 제안된 NSSD는 서비스 이름 기반의 중앙집중형 SD 구조를 제공하며, 속도의 향상을 위해 IoT 에지 게이트웨이를 캐싱 서버로 이용한다. 이를 통해 지역 안에 있는 사물 간의 통신에서는 서버를 거치지 않기 때문에 빠른 속도를 제공한다. 또한, 서비스 캐싱을 통해 이동성 있는 디바이스에 대한 즉각적인 응답이 가능하다. NSSD는 또한 서비스 측면의 통일성을 위해 서비스 제공자에 대한 URI를 사용하고 가장 긴 접두어 매치를 통해 정확하지 않은 키워드에 대한 검색도 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존에 제안된 관련 기술들을 설명하고, 3장에서는 본 논문에서 제안하는 NSSD의 기본 구조에 관해 기술한다. 4장에서는 본 논문에서 제안하는 서비스 발견 과정에 관해 기술한다. 5장에서는 시뮬레이션을 통해 기존의 DNS, DHT 방법과 성능을 비교한다. 6장에서는 본 논문에서 제안하는 방법에 대한 전체적인 평가와 고찰을 기술한다.

2. 관련연구

과거에는 SD를 위해 SLP(Service Location Protocol), UPnP(Universal Plug and Play), INS(International Naming System)와 같은 서비스 발견 프로토콜이 퍼베이시브 컴퓨팅 환경에서 제안되었다[6-8]. 그러나 이러한 프로토콜들은 서비스 발견에서는 효율적이고 빠른 탐색 방법을 제공하지만, 서비스 요청과 디바이스 광고 간의 동적 매치 메이킹 문제는 해결하지 못한다.

EPCglobal[9]은 동적 매치 메이킹 문제를 해결하기 위하여 RFID 환경에서 전역 추적과 가시성을 제공해 주는 구조를 제안하였다. EPCglobal은 RFID에서 제공되는 데이터에 대해 번호를 붙여 URI 형태로 저장하고, 이를 단순 조회 서비스를 통해 SD를 제공한다. EPCglobal은 같은 RFID 공급체인 안에서는 잘 동작하지만, 공급간 체인에 대한 SD는 고려하지 않는다. BRIDGE 프로젝트[10]는 RFID 공급체인 상에서 서비스 중심으로 동작하기 위하여 SD를 디렉토리 서비스로써 구현하였다. 그리고 SD를 위한 서버를 배치하여 EPCnumber를 기반으로 데이터에 대한 시리얼 레벨 조회를 제공한다. 그러나 이 역시 공급내 체인에서의 SD 구조라는 한계점을 가지고 있다.

Ion Stoica[11]는 DNS 기반의 SD 구조의 모든 도메인을 평등하게 간주하는 P2P 시스템을 도입하고, 다른 도메인 간의 SD를 위하여 데이터 조회 방법을 처음으로 제안하였다. 최근의 P2P 시스템의 연구 동향은 확장성 및 견고한 분산 검색 서비스이다. Nina Schoenemann[12]은 공급내 체인에서의 서비스를 제공하기 위해 P2P 기반 구조를 제안하였다. 이와 비슷하게 Shrestha 등은 각각의 서비스 제공자가 공급체인을 가지는 노드를 가지고 있고, 각 노드가 다른 노드의 부분적인 뷰를 가지는 구조화된 P2P 오버레이 네트워크를 제안하였다. 이러한 P2P 방식은 일반적으로 DHT(Distributed Hash Table) 기술을 사용한다. DHT는 분산된 데이터 구조에서 정보 객체의 유일한 키와 디스턴스 메트릭에 대응하는 피어로 이루어져 있다. DHT에서 각각의 노드는 일반적으로 MAC 또는 IP 주소인 노드 키에 의해 식별된다. 노드 내부의 정보 아이템도 마찬가지로 정보 키에 의해 식별된다. 이러한 두 가지의 키 유형은 해시 함수(예, SHA-1나 MD5)에 의해 맵핑된다. 결과적으로 DHT 기반의 SD는 노드(디바이스)나 데이터에만 집중하고 있다[13].

그러나 IoT는 여러 가지의 서비스가 통합되고 재구성되어 최종적으로 사용자에게 서비스를 제공하는 것을 목적으로 하고 있기 때문에[14], IoT를 위한 SD의 설계 원칙은 서비스 지향 설계 패러다임을 내포해야 한다.

3. 기본 구조

3.1 Overview of SD architecture

본 논문에서는 실제 IoT 환경에서 발생할 수 있는 문제점들을 해결하기 위해 서비스 중심의 SD 구조를 제안한다. 제안하는 구조는 서비스 이름을 기반으로 메타 서비스를 수집하고, 중앙집중형 SD를 제공한다. 따라서, NSSD는 서비스 이름을 기반으로 IoT 환경 내의 모든 서비스와 디바이스를 구분할 수 있다. 서비스 이름은 전형적인 계층구조로 되어 있으며 접두어 매치를 통해 검색된다. 그림 1은 NSSD의 이름 구조를 보여준다. 서비스 제공자를 식별할 수 있는 'company name', 서비스를 식별하는 'service name', 디바이스의 위치를 식별하는 'location', 디바이스 자체를 식별하는 'device name', 물리적인 디바이스의 주소인 'address'로 구성된다. Location의 기준은 3.3절에서 설명한다.

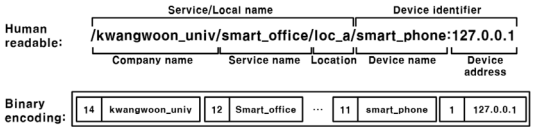

Fig. 1. Example of Service name

그림 2는 본 논문에서 제안하는 NSSD의 기본 구조이다. NSSD는 크게 서비스 사이드와 디바이스 사이드로 구성된다. 서비스 사이드는 서비스 제공자에 의해 서비스를 구성하고 해지하는 역할을 한다. 디바이스 사이드는 IoT 에지 게이트웨이를 통해 디바이스를 관리하고 서비스를 연결해주는 역할을 한다.

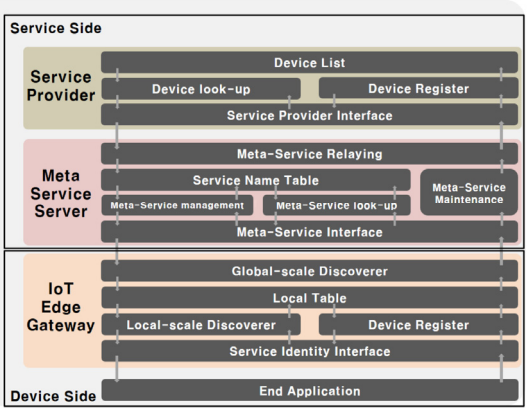

Fig. 2. Architecture of NSSD

디바이스 사이드는 IoT의 물리적 구성요소인 다양한 사물과 IoT 에지 게이트웨이(IeG, IoT-edge Gateway)로 이루어진다. IeG는 MSS(Meta-Service Server)와 사물

사이에서 질의를 중계해 주는 역할을 한다. 또한, 기본적으로 캐싱 기능이 있어 이미 접근했던 서비스에 대한 질의는 재질의를 하지 않기 때문에 속도를 향상시키는 역할을 한다.

서비스 사이드는 서비스 제공자와 MSS로 이루어진다. 서비스 제공자는 기존의 서비스를 통합하거나 새로 추가할 수 있으며, 서비스를 제공할 수 있는 디바이스의 리스트를 가지고 있다. 이 리스트는 차후 서비스를 검색할 때의 기준이 된다. MSS는 디바이스의 서비스 쿼리가 왔을 때 디바이스 리스트를 가지고 있는 서비스 제공자의 주소를 가지고 있다. 따라서 MSS는 서비스 이름과 서비스 제공자의 주소를 쌍으로 가지고 있다. 이를 SNT(Service Name Table)라 한다. 이렇게 서비스 사이드의 두 구성요소는 서비스 이름 기반의 SD를 제공하는 역할을 한다.

SD를 제공하는 순서는 다음과 같다. 먼저 서비스 제공자에 의해 구성된 서비스 이름을 MSS에 등록한다. 이때 서비스에 포함된 디바이스 리스트는 서비스 제공자가 가지고 있으며, MSS는 서비스 이름과 디바이스 리스트에 접근할 수 있는 주소만을 가지고 있다. 이후 데이터를 원하는 디바이스들이 IeG를 통해서 메타 서비스에 자신의 서비스를 질의하고, MSS는 디바이스 리스트를 가지고 있는 서비스 제공자에게 연결해준다. 그리고 결과로 받은 디바이스에 접속하여 원하는 데이터를 서비스받게 된다.

3.2 IoT edge-Gateway

본 논문에서 제안하는 IeG는 네트워크의 최말단에 위치한다. 이는 디바이스와 가장 맞닿아있는 네트워크 장비를 의미한다(예, AP, 브릿지, 스위치). 이렇게 게이트웨이의 역할을 네트워크의 말단에 맡김으로써 임의로 부여한 게이트웨이 ID가 지역적 위치를 나타낼 수 있는 수단이 된다. 그림 3은 IeG의 역할을 간략히 나타낸 것이다.

IeG는 SD의 속도 향상을 위해 로컬 네트워크 안에서 제공하는 서비스에 대한 캐싱 기능을 하고 있다. 또한, 기기가 이동했을 때 기기의 이동을 서비스 제공자에게 알리는 역할도 한다. 캐싱 기능은 매칭 검색 과정으로 인해 저하되는 데이터 응답 시간을 보완하기 위한 기능이다. IeG는 캐싱 기능을 위해 그림 1의 구조를 가지는 LT(Local Table)를 가진다.

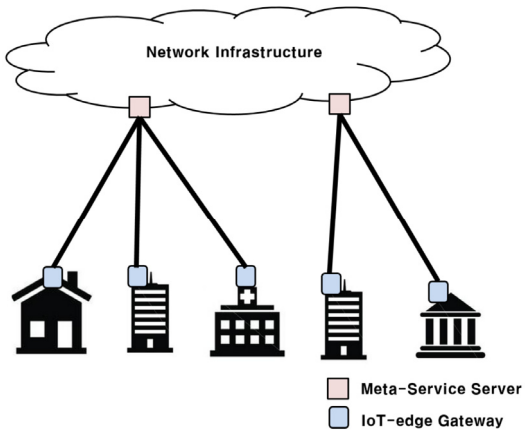

Fig. 3. Overview of IoT-edge Gateway

LT는 이전에 접속했던 서비스에 대한 캐시를 통해 디바이스에 대한 재질의를 막는 기능을 한다. 디바이스 등록 기능은 이동하는 디바이스의 위치를 식별하기 위한 기능이다. 디바이스가 이동했을 경우, 디바이스는 IeG와의 통신을 통해 자신이 이동했음을 인식한다. 이후 IeG에게 디바이스 등록을 요청한다. 요청받은 IeG는 자신의 LT에 해당 디바이스를 추가하고 MSS에 IeG ID와 해당 디바이스의 이름을 전송해 디바이스가 로컬 영역에 들어왔음을 알리고 서비스 제공자의 리스트를 업데이트한다.

3.3 Meta-Service Server

MSS는 메타 서비스 등록/해제, 메타 서비스 조회, 메타 서비스 유지/전파 기능을 제공한다.

메타 서비스 등록/해제 기능은 서비스 제공자가 새로운 서비스를 등록하거나 등록한 서비스를 삭제하는 기능이다. 서비스 제공자는 등록 툴을 통해 MSS에 서비스를 등록한다. 이때 서비스 제공자는 등록할 서비스 이름과 디바이스 리스트가 있는 서버의 주소를 등록한다. 그 후 MSS는 서비스 제공자에게 서비스 등록자와 서비스 이름으로 이루어진 이름을 부여한다. 또한, 저장된 서비스에 대한 조회를 통해 서비스 네임을 삭제하는 기능도 제공한다.

메타 서비스 조회 기능은 IeG의 요청을 서비스 제공자에게 중계하는 기능이다. IeG에서 찾는 서비스가 목록에 없으면 서비스에 대한 조회를 요청하고, MSS가 서비스 제공자의 서버에 대한 전달을 통해 실제 데이터를 제공하는 디바이스의 주소를 받아와 데이터에 접근할 수 있다.

메타 서비스 유지/전파 기능은 지역적으로 존재하는 MSS끼리 SNT의 교환을 통해 서비스 목록을 유지보수 하는 기능이다. MSS는 주변의 다른 MSS 서버와 SNT의 교환을 통해 메타 서비스를 유지한다. 서비스 제공자가 서비스를 등록하면 MSS는 SNT를 주변의 MSS와 교환한다. 이 과정은 링크 상태 프로토콜인 OSPF의 테이블 교환 과정과 유사하다.

4. Service Discovery Process

본 논문에서 제안하는 SD는 IeG를 기준으로 로컬 네트워크 안에서 서비스를 찾는 지역 조회와 로컬 네트워크 외부 서비스에 접근하는 전역 조회의 두 가지로 이루어져 있다. 모든 디바이스는 지역 조회를 한 후에 전역 조회를 통해 SD를 한다. 표1은 NSSD의 SD과정을 나타낸 의사코드이다.

Table 1. Service Discovery Procedure

```
    //when IeG n receives a message
 1. n.receive_message(message)
 2.     if(message.attr == query)
 3.         ip_addr = n.find_service(message)
 4.         if(ip_addr == -1)
 5.             device_state = n.check_device (message)
 6.             if(device_state == new)
 7.                 message = n.reg_update(message)
 8.             end if
 9.             send_to_MSS(message)
 10.        else
 11.            return ip_addr
 12.        end if
 13.    else if(message.attr == return_msg)
 14.        n.add_to_LT(message)
 15.        return message.ip_addr
 16.    Else if(message.attr == reg_msg)
 17.        n.add_to_checklist(message)
 18.        message = n.reg_update(message)
 19.        send_to_MSS(message)
 20.    end if
 21.end

    //when MSS n receives a message
 22. MSS.receive_message(message)
 23.     find_service_provider(message)
 24.     Send_to_service_provider(message)
 25. End

    //when service provider n receives a message
 26. N.receive_message(message)
 27.     if(message.attr == reg_msg)
 28.         n.update_device_list(message)
 29.     else if(message.attr == query)
 30.         ip_addr = n.find_addr(message)
 31.         return ip_addr
 32.     end if
 33. end
```

NSSD의 SD과정은 IeG를 중심으로 이루어진다.

1행부터 21행은 IeG에서 하는 역할을 나타낸다. 2행에서 12행까지는 지역 및 전역 조회의 과정을 나타내고 있다. 메시지를 받은 IeG는 LT에서 메시지에서 질의하는 서비스가 있는지 확인한다. LT에 질의한 서비스가 존재할 경우 바로 반환한다. 4행부터 9행은 전역 조회를 위해 메시지를 생성하는 과정이다. 이동한 후에 등록과정에 앞서 질의하는 디바이스를 처리하기 위하여 LT를 이용해 확인한다. 확인을 통해 새로 발견된 디바이스의 경우 메시지의 상태를 등록으로 바꾼 후 MSS에 메시지를 전송한다. 13행부터 15행은 IeG에 도착한 메시지가 전역 조회에 대한 응답 메시지일 경우이다. 도착한 메시지를 LT에 추가하고 질의한 디바이스에 목적지에 대한 주소값을 전달한다. 이후부터 캐싱을 통해 같은 서비스에 대해서는 빠른 접근이 가능하다. 16행부터 20행은 이동한 디바이스가 자신의 위치를 서비스 제공자에 등록하는 부분이다. IeG에서 현재 자신의 영역에 있는 디바이스를 구별하기 위한 체크리스트에 디바이스 이름을 삽입한 후 MSS로 메시지를 전송한다.

MSS는 서비스에 대한 메타정보를 가지고 직접적으로 서비스 제공자에 접근할 수 있게 해주는 역할을 한다. 22행부터 25행은 MSS가 하는 역할이다. 전송받은 메시지로 제공자의 주소를 찾아 서비스 제공자에게 메시지를 전달하는 역할을 한다. 서비스 제공자는 SD 과정에서 서비스 이름과 위치를 통해 물리 디바이스에 접근할 수 있는 주소를 반환해주는 역할을 한다.

서비스 제공자가 메시지를 받았을 때 26행부터 28행은 이동한 디바이스에 대한 위치수정을 나타낸다. 이동한 디바이스는 현재 위치의 IeG이름을 통해 위치를 식별할 수 있다. 29행부터 31행은 디바이스 리스트에서 위치를 찾아 주소를 반환한다.

5. 실험 및 시뮬레이션

5.1 시뮬레이션 환경

본 장에서는 NSSD의 우수성을 검증하기 위하여 NS3의 시뮬레이션 파라미터를 사용하여 기존의 DNS와 DHT 기반 DS 구조와의 성능평가를 진행하였다. 본 논문에서 사용하는 모델은 데이터를 요청하는 디바이스와 라우터로 이루어져 있다. 서비스 구성은 2백만개의 디바

이스 중 랜덤으로 클라이언트가 요청하게 되며, 요청하는 클라이언트에서부터 디바이스가 떨어져 있는 위치는 랜덤으로 초기화 시기에 결정된다. 또한, 구조의 규칙성과 다수의 다중 경로 설정을 위한 메쉬 토폴로지 구조로 되어 있다. 본 논문에서 제안하는 기법의 성능을 평가하기 위한 시뮬레이션 환경은 표 1과 같다.

Table 2. Simulation Parameters

Parameters	Values
Link bandwidth	100 Mbps
Link delay	2 ~ 5 ms
Data Packet Size	100 Kbytes
Number of Services	10000
Number of Devices of each Service	1000
IeG caching policy	LRU
IeG cache size	100

5.2 비교 및 분석

그림 4는 요청 횟수에 따른 데이터의 평균 반환시간을 나타낸 것이다.

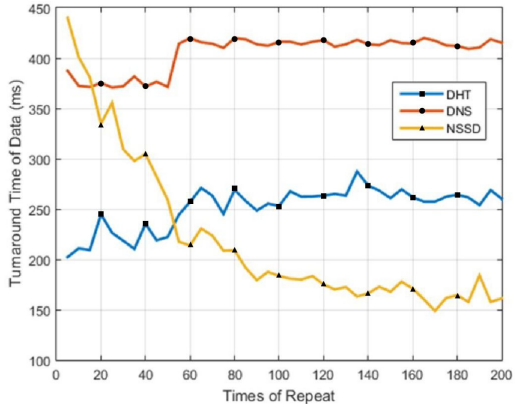

Fig. 4. Average Turnaround Time of Data

DHT와 DNS 방식은 요청 횟수가 증가하더라도 변화가 크지 않다. 그러나 본 논문에서 제안하는 NSSD의 경우 요청 횟수를 반복할수록 평균반환시간이 급격하게 줄어들었다. 이러한 이유는 NSSD의 경우 IeG의 캐싱 기능으로 인해 중복 요청 디바이스에 대한 접근속도가 현저하게 빨라졌기 때문이다. 이는 NSSD가 기존 방식에 비교해 성능이 향상되었다는 것을 의미한다.

그림 5는 그림 4의 시뮬레이션 상황에 따른 최선의 경우와 최악의 경우에 대한 변화를 나타낸 것이다.

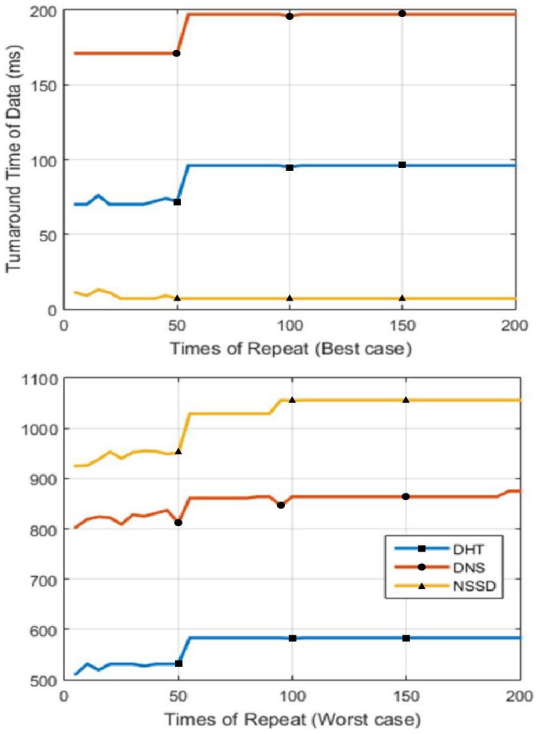

Fig. 5. Best and Worst Case of Turnaround Time According to Number of Repeat

최선의 경우에서 DHT는 평균 70ms, DNS의 경우 170ms의 평균 반환시간을 보였다. NSSD의 경우 지역 조회를 통한 접근으로 인해 평균 8ms의 빠른 반환시간을 보였다. 그러나 최악의 경우는 최선의 경우와 반대의 상황을 보였다. DHT의 경우 최악의 경우임에도 평균 550ms 정도의 비교적 빠른 속도를 보였고, DNS의 경우 평균 850ms 정도의 속도를 나타냈다. NSSD의 경우에는 IeG에서의 초기 조회와 MSS의 조회, 서비스 제공자에서의 조회로 인해 평균 1초가 넘는 시간이 소비된 것으로 나타났다.

그림 6은 네트워크의 크기에 따른 데이터의 평균 반환시간을 나타낸 것이다.

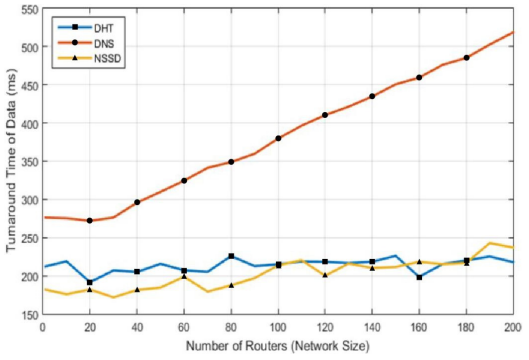

Fig. 6. Average Turnaround Time of Data According to Network Size

DNS는 네트워크의 크기가 커질수록 선형적으로 증가하는 그래프를 나타내지만, DHT의 경우에는 네트워크의 크기와 관계없이 일정한 속도를 유지하는 것을 볼 수 있다. NSSD 또한 DNS 기반의 조회를 수행함에도 네트워크의 크기와 관계없이 일정한 속도를 유지하였다.

그림 7은 사용자가 이동시 목적지의 IeG에서 서비스가 존재하는 확률당 평균 반환시간을 나타낸 것이다.

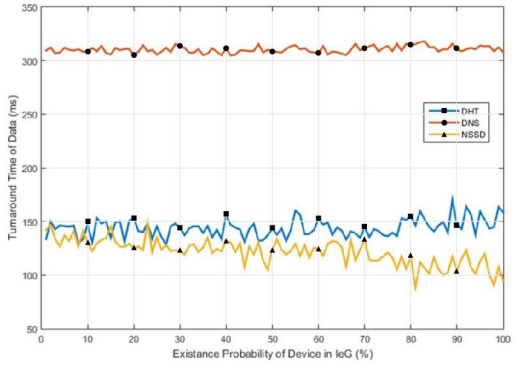

Fig. 7. Average Turnaround Time of Data According to Existence Probability

DNS와 DHT는 IeG가 없기 때문에 지속적으로 일정한 반환시간의 성능을 발휘하였다. 그러나 NSSD의 경우에는 IeG의 LT에 서비스 목록이 존재하는 확률이 높아질수록 반환시간이 빨라 지는 것을 볼 수 있다. 이러한 이유는 NSSD의 캐싱 기능으로 인해 IeG에 서비스 목록이 존재할수록 조회 시간이 줄어들었기 때문이다. 이는 서비스 이름 중심의 SD가 이동성이 빈번한 위치 기반 서비스 환경에서 더 좋은 성능을 발휘할 수 있음을 의미한다.

6. 결론

본 논문에서는 기존의 디바이스 중심의 SD가 가지는 문제점들을 해결하기 위하여 IoT의 물리적인 환경을 고려한 서비스 중심의 SD 구조인 NSSD를 제안하였다. 제안된 NSSD 구조는 서비스 중심의 SD를 위하여 SD 기준을 서비스 이름으로 변경하였고, DNS 기반 조회 방법의 속도 저하를 해결하기 위하여 IeG의 캐싱 기능으로 속도를 향상시켰다. 또한, 성능평가를 통하여 기존의 SD 구조와 제안된 NSSD를 비교하였으며, 그 결과 기존의 DHT에 비해 성능이 향상되었음을 확인하였다. 또한, IoT에서 자주 발생하는 위치 기반 서비스로 인한 재요청을 IeG를 통해 해소할 수 있음을 확인하였다.

그러나 최악의 경우에서는 NSSD가 DHT에 비해 2배 이상의 반환시간이 소요되는 문제점을 보였다. 이러한 문제를 해결하기 위해서는 현재 최악의 경우에서 소요시간의 대부분을 차지하는 3단계 조회 알고리즘을 개선하고, 개선된 연구결과를 적용하는 연구가 반드시 이루어져야 한다.

References

[1] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, Moussa Ayyash, “Internet of Things: A Survey on Enabling techonologies, Protocols, and Applications”, *IEEE Communications Surveys & Tutorials*, pp. 2347-2376, June, 2015.
DOI: https://doi.org/10.1109/COMST.2015.2444095

[2] Dimiter V. Dimitrov, “Medical Internet of Things and Big Data in Healthcare”, *Healthcare Informatics Research*, pp. 156-163, July, 2016.
DOI: https://doi.org/10.4258/hir.2016.22.3.156

[3] Vaclav Jirkovsky, Marek Obitko, Vladimir Marik, “Understanding Data Heterogeneity in the Context of Cyber-Physical Systems Integration”, *IEEE Transactions of Industrial Informatics*, pp. 660-667, July, 2016.
DOI: https://doi.org/10.1109/TII.2016.2596101

[4] M. Nidd, "Service discovery in DEAPspace", *IEEE Personal Communication*, pp. 39-45, August, 2001.
DOI: https://doi.org/10.1109/98.944002

[5] Anne H. Ngu, Mario Gutierrez, Vangelis Metsis, Surya Nepal, Quan Z. Sheng, “IoT Middleware: A Survey on Issues and Enabling Technologies”, *IEEE Internet of Things Journal,* pp. 1-20, October, 2016.
DOI: https://doi.org/10.1109/JIOT.2016.2615180

[6] J. Velzades, E. Guttman, C. Perkins, S. Kaplan, "Service Location Protocol", Network Working Group, ieft, 1997.

[7] B.A. Miller, T. Nixon, C. Tai, M.D. Wood, "Home Networking with Universal Plug and Play", *IEEE*

Communication Magazine, pp. 104-109, December, 2001.
DOI: https://doi.org/10.1109/35.968819

[8] Magdalena Balazinska, Hari Balakrishnan, David Karger, "INS/Twine: A Scalable Peer-to-Peer Architecture for Intentional Resource Discovery", *Lecture Notes in Computer Science*, pp. 195-210, 2002.
DOI: https://doi.org/10.1007/3-540-45866-2_16

[9] Felice Armenio et al., "The EPCglobal Architecture Framework", EPCglobal, pp. 1-74. March, 2009

[10] The BRIDGE project, www.bridge-project.eu, accessed 11/01/2010.

[11] Ion Stoica, Robert Morris, David Karger, M. Fans Kaashoek, Hari Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications", *ACM SIGCOMM Computer Communication Review*, pp. 149-160, October, 2001.
DOI: https://doi.org/10.1145/964723.383071

[12] Nina Schoenemann, Kai Fischbach, Detlef Schoder, "P2P architecture for ubiquitous supply chain systems", ECIS 2009 Proceedings, pp. 357, 2009.

[13] Gluseppe Pirro, Domenico Talla, Paolo Trufio, "A DHT-based semantic overlay network for service discovery", *Future Generation Computer Systems,* pp. 689-707, April, 2012.
DOI: https://doi.org/10.1016/j.future.2011.11.007

[14] Christian Cabrera, Andrei Palade, Slobhan Clarke, "An evaluation of service discovery protocol in the internet of things', *ACM, SAC '17*, pp. 469-476, April, 2017.
DOI: https://doi.org/10.1145/3019612.3019698

[15] Min-soo Kang, Chunhwa Ihm, Jaeyeon Lee, Eun-Hye Choi, Sang Kwang LeeA Study on Object Recognition for Safe Operation of Hospital Logistics Robot Based on IoT, *The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)*, vol. 17, no. 2, pp. 141-146, Apr. 2017.
DOI: https://doi.org/10.7236/JIIBC.2017.17.2.141

[16] Hoon-Shik Woo, Application of Analytic Hierarchy Process for Relative Importance Determination of Internet of Things Standardization, *J. Soc. Korea Ind. Syst. Eng*, vol. 39, no. 1, pp. 47-55, Mar. 2016.
DOI: http://dx.doi.org/10.11627/jkise.2016.39.1.047

[17] Minzheong Song, A Study on Business Types of IoT-based Smarthome: Based on the Theory of Platform Typology, *The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)*, vol. 16, no. 2, pp. 27-40, Apr. 2016.
DOI: http://dx.doi.org/10.7236/JIIBC.2016.16.2.27

[18] Min-soo Kang, Chunhwa Ihm, Jaeyeon Lee, Eun-Hye Choi, Sang Kwang Lee, A Study on Object Recognition for Safe Operation of Hospital Logistics Robot Based on IoT, *The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)*, vol. 17, no. 2, pp. 141-146, Apr. 2017.
DOI: https://doi.org/10.7236/JIIBC.2017.17.2.141

[19] Kwang Seob Jeong, Sukjoo Bae, Hyoungtae Kim, Evaluation Criteria for Suitable Authentication Method for IoT Service Provider in Industry 4.0 Environment, *J. Soc. Korea Ind. Syst. Eng*, Vol. 40, No. 3, pp. 116-122, Sep. 2017.
DOI: https://doi.org/10.11627/jkise.2017.40.3.116

조 국 현(Kuk-Hyun Cho) **[정회원]**

- 1981년 2월 : 일본 Tohoku University (공학석사)
- 1984년 2월 : 일본 Tohoku University (공학박사)
- 1984년 3월 ~ 현재 : 광운대학교 소프트웨어학부 교수

<관심분야>
네트워크 관리, 차량 통신 네트워크, IoT

김 정 재(Jung-Jae Kim) **[준회원]**

- 2013년 2월 : 광운대학교 컴퓨터소프트웨어학과(공학사)
- 2015년 2월 : 광운대학교 컴퓨터과학과 (공학석사)
- 2015년 3월 ~ 현재 : 광운대학교 컴퓨터과학과 박사과정

<관심분야>
인공지능, 지능시스템, 딥러닝, 강화학습, Internet of Things

류 민 우(Minwoo Ryu) **[정회원]**

- 2009년 8월 : 광운대학교 컴퓨터과학과 (공학석사)
- 2012년 8월 : 광운대학교 컴퓨터과학과 (공학박사)
- 2011년 2월 ~ 2016년 12월 : 전자부품연구원 선임연구원
- 2017년 1월 ~ 현재 : KT 융합기술원 선임연구원

<관심분야>
시맨틱, 사물인터넷, 차량통신, 인공지능, 기계학습

차 시 호(Si-Ho Cha) **[종신회원]**

- 1997년 8월 : 광운대학교 전자계산학과 (이학석사)
- 2004년 2월 : 광운대학교 컴퓨터과학과 (공학박사)
- 1997년 7월 ~ 2000년 2월 : 대우통신 종합연구소 선임연구원
- 2009년 3월 ~ 현재 : 청운대학교 멀티미디어학과 교수

<관심분야>
네트워크 관리, 차량 통신 네트워크, 지능형 IoT