www.osia.or.kr

OSIA S&TR Journal

ISSN 1738-9887 Vol. 31, No. 1, March 2018
• 발행인 : 정성호 회장/OSIA • 편집위원 : 김평수 교수/한국산업기술대학교

02 Editorial
인터넷 환경에서 식별자, 네이밍, 그리고 어드레싱
김평수 교수/한국산업기술대학교

Article
04 사물인터넷 기기 관제를 위한 DNS 기반 네이밍 시스템
정재훈 교수/성균관대학교, 김석화 연구원/성균관대학교
12 정보 중심 네트워킹에서의 네임 레졸루션 기술
홍정하 선임/ETRI
17 정보중심 네트워킹 이후의 정보처리 네트워킹
정희영 박사/ETRI
23 LTE Network의 IPv6전환 기술들
고득녕 매니저/SK브로드밴드
31 사물인터넷 환경에서의 DNS와 DOA 기술 비교 및 인터넷 거버넌스 이슈 분석
김평수 교수/한국산업기술대학교

(사)한국컴퓨터통신연구회
Open Standards and ICT Association

사물인터넷 기기 관제를 위한 DNS 기반 네이밍 시스템

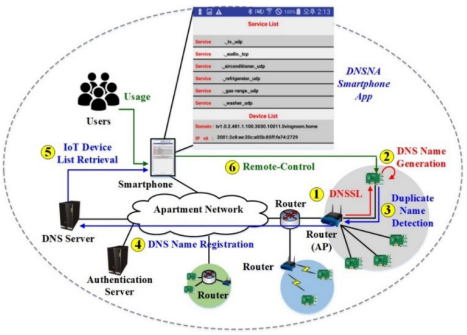
김석화[1] 정재훈[2]
[1]성균관대학교 전자전기컴퓨터학과, [2]성균관대학교 인터랙션사이언스학과

Abstract

스마트 홈, 스마트 팩토리 등 다양한 사물인터넷(Internet of Things, IoT) 기기를 식별하기 위한 DNS Name Autoconfiguration(DNSNA) 기법은 IPv4와 IPv6 인터넷에서의 IoT 기기 관제를 위해 연구되어 왔다. 본 기법은 DNS 네임을 사물인터넷 기기의 IP 주소로 변환하여 사용자가 손 쉽게 자신의 사물인터넷 기기를 모니터링 및 원격에서 제어할 수 있도록 자동으로 설정하는 방법이다. 본고에서는 DNSNA의 개념, DNS 네임 형식, DNSNA 프레임워크를 소개하고, DNSNA의 성능을 애플 사(Apple Inc.)가 개발한Multicast DNS(mDNS)와 비교한다.

I. 서론

사물인터넷(Internet of Things, IoT)기기가 증가함에 따라 기기를 식별 및 관리하는 기법에 관한 연구는 필수불가결한 요소이며, 이를 위해 DNS Name Autoconfiguration(DNSNA)은 사물인터넷 기기를 획기적으로 모니터링 및 관리를 위한 프레임워크를 Domain Name System(DNS) 을 기반으로 정의하고 구현하는 것을 목표로 한다. 사물인터넷 기기를 원격에서 제어하며, 서비스를 제공하기 위해서는 Internet Protocol(IP) 주소를 기반으로 해야 한다. IP 주소는 4바이트의 정수로 구성되며 일련의 숫자(1 ~ 255)와 점으로 구성이 된다. 하지만 기존의 IP Version 4(IPv4) 는 약 30억개의 IP 주소를 할당 할 수 있으나 2011년 IPv4 주소가 고갈되었다. 따라서 IPv4 주소 고갈의 문제를 해결하기 위해 IP Version 6(IPv6)가 제정되었다. 128 비트의 주소를 가진 IPv6는 최대 2^128개의 IP 주소를 할당할 수 있지만 아직까지 한정적인 환경(예, 이동통신 네트워크) 에서 사용되고 있으나 사물인터넷 기기가 증가함에 따라 점점 IPv6의 사용량은 높아질 것으로 예상된다. DNSNA는 사물인터넷 기기를 식별, 모니터링 및 원격에서 제어하기 위한 설정과정을 사용자의 개입을 최소화 하고 다양한 네트워크 환경(예, IPv6, IPv4, Security, Constrained Application Protocol(CoAP)) 에서 DNS를 활용하여 원격에서 관리 및 제어를 위한 자동설정을 제공하도록 하는 기술이며, 관련연구가 활발히 진행되었다[1-7]. 또한 본 논문은 본 저자의 이전 저널논문과 학술대회 논문을 기반으로 정리하였다[3, 4, 5, 6]. 각 연구 기법은 6절 DNSNA 프레임워크(Framework)에서 소개한다. 본 고를 통해서 이러한 DNSNA의 기술을 소개하며, 다음과 같이 구성되어 있다. 2장에서는 사물인터넷 기기를 관리하기 위한 관련된 연구들을 소개한다. 3장에서는 DNS 네임 포맷에 대해 소개한다. 4장에서는 DNSNA의 프레임워크를 소개한다. 5장에서는 DNSNA의 성능을 Multicast DNS(mDNS)와 비교하여 소개한다. 본 성능평가는 저자의 이전 저널 논문[6]에서 소개한 자료를 사용한다. 그리고 6장에서는 결론 및 향후 연구를 제시한다.



〈그림 1. Concept of DNSNA〉

II. 관련연구

많은 학교 및 기업에서 사물인터넷 기기를 위한 관리기법 혹은 네이밍 기법들이 사물인터넷 기기가 증가함에 따라 활발히 연구되고 있다. 본 절에서는 기기 제어 및 관리를 위한 관련 연구들을 소개한다.

미국 애플 사(Apple Inc.)의 봉주르(Bonjour)는 애플 디바이스를 사용하는 사용자에 한하여 자동으로 자신이 소유하고 있는 애플 기기들을 관리할 수 있도록 해주는 mDNS 기반의 Zero-Configuration 플랫폼이다[8]. 모든 애플 디바이스는 DNS 네임을 저장하고 있는 DNS 서버이며 동시에 자신의 DNS 네임을 다른 디바이스에 요청하는 DNS 클라이언트

가 된다. 또한 봉주르는 로컬네트워크에 위치한 다른 애플 디바이스를 탐색하는 디바이스 탐색(Device Discovery)과 각각 디바이스가 지원할 수 있는 서비스를 탐색하는 DNS 기반의 서비스 탐색(DNS Service Discovery, DNS-SD)를 지원한다[9]. 하지만 봉주르는 mDNS를 기반으로 하여, 많은 기기가 속한 네트워크에서는 멀티캐스트를 기반으로 하기 때문에 많은 패킷을 발생시킨다.

oneM2M은 Machine-to-Machine(M2M) 형식으로 통신하는 디바이스들을 위한 표준화 기구로서 통신프로토콜, 스마트 홈, 스마트 산업 등 사물인터넷 기기들을 위한 소프트웨어 플랫폼을 제공한다. oneM2M의 플랫폼은 개발자들에게 손쉽게 사물인터넷 기기들을 위한 서비스 개발을 할 수 있다. 또한 oneM2M 에서는 사물인터넷 기기를 위한 객체 식별자(Object Identifier, OID)를 표준화 했다[10]. oneM2M은 다양한 종류의 사물인터넷 기기들을 위한 통일된 서비스 플랫폼을 제공하는데 중점을 두어 Message Queue Telemetry Transport(MQTT), Constrained Application Protocol(CoAP) 등과 같은 사물인터넷 기기 제어 및 관리를 위한 다양한 프로토콜을 이용하여 서비스 제공을 지원한다[10]. CoAP은 UDP 기반의 제어 프로토콜로 바이너리로 구성되어 기존의 PC와 같은 디바이스와 달리 사물인터넷 기기, 센서 등과 같이 컴퓨팅 파워가 제한적인 기기에 에너지 효율성을 고려하여 디자인된 프로토콜이다. CoAP는 UDP/IP 상에서 동작한다. 메시지의 신뢰성을 위해 재전송을 제공하며, 제어를 위해 Hyper Text Transfer Protocol(HTTP) 에서 제공하는 메소드(즉, GET, PUT, POST, DELETE)를 이용해 제어메시지를 구성한다[11]. 또한 CoAP은 프록시를 통해 HTTP와 연동되어 인터넷의 단말이 사물인터넷의 네트워크 기기들을 웹 기반 인터페이스를 통해 쉽게 제어할 수 있다[12].

III. DNS 네임 포맷

본 절에서는 사물인터넷 기기 관리를 위한 DNS 네이밍 포맷에 대해서 설명한다. 본 고에서 제안하는 DNS 네임 포맷은 2장의 관련연구에서 소개한 것처럼 oneM2M에서 객체를 식별하기 위한 OID를 기반으로 하여 DNS 네임 포맷을 정의하였다[10].

```
<unique_id>.<object_identifier>.<location>.<domain_name>
```

〈그림 2. OID 기반의 DNS 네임 포맷〉

- **unique_id:** ASCII 문자로 생성된 DNS 네임의 유일성을 보장하기 위한 식별자로, 일련의 숫자의 조합 혹은 제품의 이름처럼 사용자가 읽을 때 가독성있는 알파벳과 숫자로 구성된다.
- **object_identifier:** 디바이스의 객체 식별자로서 oneM2M에서 정의한 지시 ID(관리기관, 관청, 데이터 국가 코드, M2M 노드의 결합 문자열)[10]이다. 이 OID는 일련의 4개로 이루어진 식별자(제조사 ID, 모델 ID, 시리얼 ID, 확장 ID)로 구성되며, 각 필드는 언더스코어('_')를 이용해 구분한다.
- **location:** 사물인터넷 기기의 물리적인 위치를 나타내며 미시적인 위치(예, 중앙, 코너, 오른쪽 벽 등)와 거시적 위치(예, 거실, 주방, 서재 등)로 구성된다.
- **domain_name:** RFC 3315의 도메인 네임의 표현에 따라 인코딩된 DNS 도메인 네임을 나타낸다. 본 필드는 사용자가 자신의 네트워크를 식별하기 위해 사용된다. 예를 들어, .home 및 .lab과 같은 도메인 네임이 있다.

IV. DNSNA 프레임워크

본 절에서는 다양한 네트워크 환경에서 어떻게 DNSNA 프레임워크를 적용하는지 소개하며, 3장의 DNS 네임 포맷을 활용하여 네트워크에 위치한 기기의 탐색 및 기기가 제공할 수 있는 서비스의 목록을 탐색하는 서비스 탐색을 제공한다.

4.1 DNSNA의 구성요소

본 DNSNA는 로컬에 위치한 DNS 서버, 라우터, 인증서버, 사물인터넷 기기, 스마트 기기로 구성된다. 본 장비들은 다양한 네트워크 환경(예, 집, 학교, 공장)에서 DNSNA 프레임워크를 적용하기 위한 장비로서 동작한다.

- **DNS 서버:** DNS 서버는 로컬에 위치한 사물인터넷 기기에서 생성된 DNS 네임을 등록하고 관리하는 서버이다. DNS 서버는 사용자의 DNS 리스트 요청이나 사물인터넷 기기의 DNS 업데이트 요청을 처리한다. DNS 서버는 사용자의 원활한 접속을 위해 글로벌 IP 주소를 사용해야 한다. DNS 서버는 사용자의 DNS 리스트 요청을 수신하면 사용자에게 사물인터넷 기기들의 DNS 네임 리스트를 전송한다. DNS 업데이트 요청이 들어오면 사물인터넷 기기의 DNS 네임을 DNS 서버 내의 DNS 네임 리스트에 등록한다.
- **라우터:** 네트워크 라우터로 동작하며, 유선 혹은 무선으로 반드시 네트워크에 연결되어, 사물인터넷 기기들과 유기적으로 통신할 수 있어야 한다. 또한 자동설정 기술인

Dynamic Host Configuration Protocol(DHCP)[13] 기능을 탑재하여 네트워크에 존재하는 사물인터넷 기기들의 IP를 관리하거나 혹은 Recursive DNS Server(RDNSS)서버의 주소를 지정하고 IPv6의 Router Advertisement(RA) 옵션을 활용하여 네트워크를 식별하는 DNS 탐색 리스트(DNS Search List, DNSSL)를 전송한다[14]. IPv4 네트워크에서는 DHCP의 옵션을 활용하여 DNSSL을 사물인터넷 기기들에게 전송한다[13].

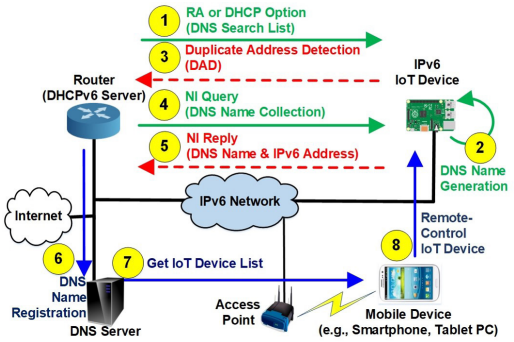
- **인증서버 (AS):** 인증서버는 보안이 고려된 DNSNA(SDNSNA)에서 RSA 기반의 키 생성 및 인증을 과정을 처리하는 서버이다. AS 서버는 한 쌍의 키를 생성하여 각각 스마트 기기와 라우터에게 전송한다. 이때 스마트 기기로 전송된 키는 Near Field Communication(NFC) 태그를 이용하여 사물인터넷 기기에게 전송된다.
- **사물인터넷 기기:** 사물인터넷 기기로써 사용자가 관리 및 제어하고자 하는 대상이 된다. 이는 많은 센서(예, 온도 센서, 습도 센서)를 포함하고 있으며 센서들을 관리하기 위한 컴퓨팅 기능을 제공하는 동시에 사용자의 스마트 기기 및 라우터와 유기적으로 통신할 수 있어야 한다. 또한 NFC를 이용하여 스마트 기기와 상호작용한다.
- **스마트 기기:** 사용자가 사용하는 스마트 폰 혹은 테블릿 PC와 같은 이동단말이다. 사물인터넷 기기와 통신을 위한 NFC 기능을 탑재하고 있으며 Wi-Fi 및 인터넷에 접속이 가능해야 한다. 사용자는 스마트 기기를 활용하여 사물인터넷 기기를 관리 및 제어할 수 있다.

V. 사물인터넷 기기 관제를 위한 DNS 기반 네이밍 시스템(DNSNA)

본 장에서는 다양한 네트워크환경에서 DNSNA의 개념을 적용하고 실제 환경에서의 적용방법을 소개한다. DNSNA는 IPv6 네트워크 환경에서의 적용 및 실험이 이루어졌으나, 현재 네트워크 환경인 IPv4와 보안성(Security)을 지원하도록 확장되었다.

5.1 IPv6 네트워크를 위한 DNSNA

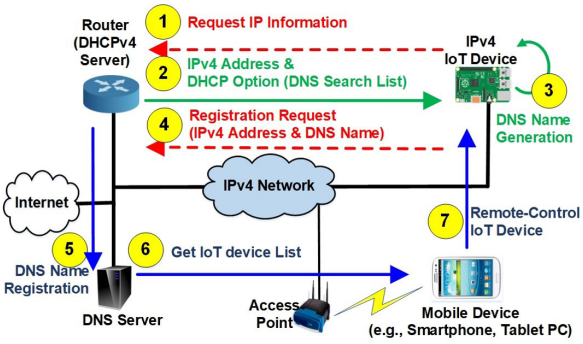
본 절에서는 IPv6 네트워크 에서의 DNSNA를 소개한다. 본 장에서 소개하는 DNSNA는 앞으로 소개할 IPv4, Secure 등의 환경에서의 연구에 근간이 되는 기술이다[1]. DNSNA의 시스템 설정은 그림 3과 같다.



〈그림 3. DNSNA 시스템 설정〉

DNSNA는 다음과 같이 동작한다[14]. 첫째, 사물인터넷 기기가 네트워크에 등록되면 라우터는 사물인터넷 기기에게 라우터 광고(RA)혹은 DHCPv6의 옵션을 활용하여 DNS 탐색 리스트(DNSSL)을 전송한다. 둘째, 사물인터넷 기기는 수신한 DNSSL을 이용하여 DNS 네임 포맷에 일치하게 자신의 DNS 네임을 생성한다. 셋째, 생성된 DNS 네임과 자신의 IPv6 주소를 포함하여 라우터에게 DNS 네임이 중복되었는지 중복 네임 검사(Duplicate Name Detection, DND)를 수행하여 자신의 DNS 네임의 유일성을 검증한다. 넷째, 라우터는 Node Information(NI) 쿼리(Query) 메시지를 이용하여 DNS 네임 수집 메시지를 전송한다. 사물인터넷 기기는 NI 응답(Reply) 메시지를 활용하여 자신의 DNS 네임을 DNS 서버에 등록해달라는 요청을 전송한다. 다섯째, 사물인터넷 기기의 DNS 네임과 IPv6 주소를 수신한 라우터는 DNS 동적 업데이트 프로토콜을 활용하여 DNS 서버로 전송할 쿼리를 생성한다. 여섯째, 사물인터넷 기기의 DNS 네임이 정상적으로 DNS 서버에 등록되어있다면 사용자는 스마트 기기를 이용하여 DNS 서버로부터 등록된 사물인터넷 기기들의 DNS 네임 리스트를 이용해 원격에서 사물인터넷 기기를 제어할 수 있다.

5.2 IPv4 네트워크를 위한 DNSNA(DNSNAv4)



〈그림 4. DNSNAv4 시스템 설정〉

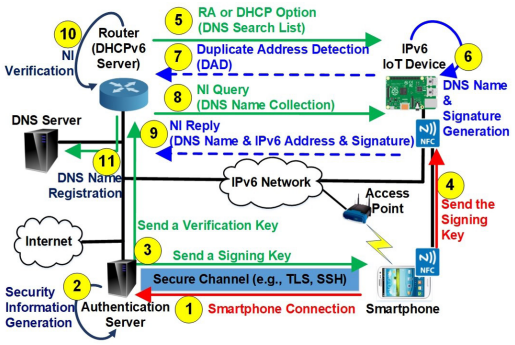
본 절에서는 현재 네트워크 환경인 IPv4 환경에서의

DNSNA(DNSNAv4)를 소개한다. DNSNAv4는 DHCPv4 서버를 라우터로 활용하여 구성하며, 시스템 설정은 그림 4와 같다.

DHCP 서버는 DHCP 옵션을 활용하여 DNSSL을 사물인터넷 기기에게 전송하며 사물인터넷 기기는 수신한 DNSSL 옵션을 활용하여 IPv6에서의 DNSNA와 동일하게 사물인터넷 기기의 DNS 네임을 생성하게 된다. 사물인터넷 기기는 생성된 DNS 네임과 IPv4 주소를 이용하여 메시지를 구성한 후 DHCP 서버에게 자신의 DNS 네임이 고유한지 검증하는 중복 네임 검사(DND)를 수행한 후 DNS 네임이 검증되었다면 DNS 동적 업데이트 메시지를 요청한다. 외의 과정은 IPv6와 동일하다.

5.3 보안을 고려한 DNSNA(SDNSNA)

기존의 DNSNA와 DNSNAv4는 악의적인 해커에 의해서 사물인터넷 기기를 등록하는 일련의 과정이 해킹 당할 수 있는 위험성을 보완하기 위해 연구되었다. 본 절에서 소개할 Secure DNSNA(SDNSNA)는 그림 5와 같이 시스템을 설정한다.



〈그림 5. SDNSNA 시스템 설정〉

기존의 DNSNA에서는 사물인터넷 기기 등록을 위해 별다른 인증과정을 진행하지 않지만, SDNSNA에서는 NFC를 활용해 AS 서버에서 발생시킨 키 쌍을 활용하여 디지털 서명을 생성하며 이를 검증하는 과정을 추가하여 사용자가 등록하고자 하는 사물인터넷 기기만 등록하도록 할 수 있다. 사용자 스마트 기기는 AS 서버(인증서버)에 인증된 채널(예, TLS, SSH)을 이용하여 접속하게 된다. AS 서버는 인증을 위한 키 쌍(Signing Key, Verification Key)을 생성하여 Signing Key를 스마트 기기로, Verification Key를 라우터로 전송한다. Signing Key를 스마트 기기에서 NFC를 이용해 사물인터넷 기기로 전송되며 Verification Key를 사물인터넷 기기로부터 생성된 디지털 서명을 검증하는데 사용된다. 이후 사물인터넷 기기는 라우터로부터 DNSSL을 수신하여, 자신의 DNS 네임을 제안한 DNS 네임 포맷에 일치하게 생성하며, 스마트 기기를 통해 수신한 Signing Key와 사물인터넷 기기의 고유한 정보(예, MAC 주소)를 이용하여 자신의 디지털 서명을 생성한다. 디지털 서명은 중복된 DNS 네임이 존재하는지 확인 한 후, NI 쿼리를 이용해 라우터로 전송된다. 라우터는 이전에 수신한 Verification Key를 이용하여 사물인터넷 기기의 디지털 서명을 검증하며 만일 수신한 디지털 서명이 검증된다면 라우터는 DNS 동적 업데이트 쿼리를 생성하여 DNS 서버로 업데이트를 요청한다. 사물인터넷 기기를 원격에서 관리 및 제어하기 위한 나머지과정은 일반적인 DNSNA와 동일하게 동작한다[7].

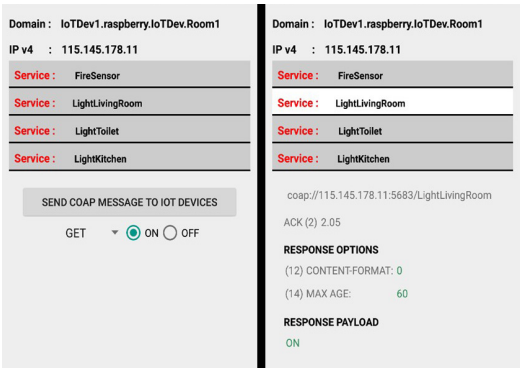
5.4 CoAP기반 원격제어

본 절에서는 CoAP 기반의 통신과 사물인터넷 기기의 서비스를 탐색하는 서비스탐색(DNS Service Discovery, DNS-SD)을 고려한 DNSNA를 소개한다 [4], [9], [11]. 기존의 DNSNA에서는 사물인터넷 기기 관리 및 제어를 위해서 단순한 소켓 프로그래밍을 이용하였다. 하지만 소켓 프로그래밍은 기존의 기기와 달리 컴퓨팅 파워가 제한적인 기기에는 적합하지 않은 관리 및 제어 방식이다. 따라서 적합한 관리 및 제어 메시지를 위하여 CoAP 기반의 관리 및 제어 메시지를 이용해야 한다[11]. CoAP은 2장 관련연구에서 소개한 것처럼 바이너리 기반의 헤더를 이용하여 기존의 PC처럼 컴퓨팅파워가 충분하지 않은 기기에 대해 적합한 관리 및 제어 메시지를 제공할 수 있다. 또한 DNS 기반의 서비스 탐색을 제공하기 위해 DNS 서비스 레코드 타입[15]을 이용하여 서비스 탐색을 제공한다. 서비스 탐색을 위한 SRV 리소스 레코드 포맷은 그림 6과 같다.

```
robotCleaner._RobotCleaner._tcp.local.  120  IN  SRV  0  0  515  robotcleaner.local.
```

〈그림 6. DNS Service Discovery를 위한 SRV 리소스 레코드 포맷〉

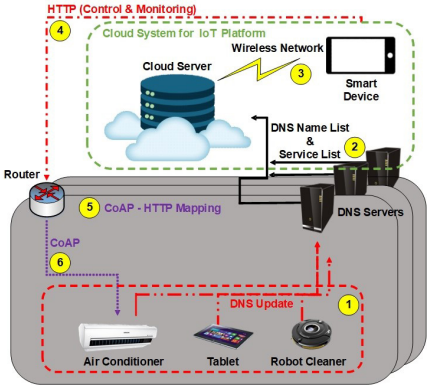
그림 6의 SRV 레코드의 첫 번째 필드는 〈Instance Name〉.〈Service Type〉.〈Domain〉로 구성된다. 자세한 설명은 아래와 같다. Instance Name은 사용자가 쉽게 이해할 수 있는 알파벳 및 숫자의 조합으로 서비스의 이름이 들어간다. Service Type에서는 해당 서비스가 사용하는 타입이 들어간다. Domain 해당 서비스가 제공되는 도메인의 정보를 나타낸다. 나머지 120, 0, 0, 515는 순서대로 Time-to-Live, Weight, Priority, Port Number를 나타낸다. 따라서 그림 6의 SRV 리소스 레코드는 로컬에서 사용되는 로봇 청소기의 대한 정보를 담고 있는 것이다. 따라서 CoAP 기반의 DNSNA를 이용하게 될 경우 그림 7과 같이 원격에서 사물인터넷 기기를 탐색하고 관리 및 제어할 수 있다[4].



〈그림 7. CoAP 기반의 DNSNA의 원격 관리 및 제어〉

5.5 Cloud 기반 DNSNA

본 절에서는 클라우드 컴퓨팅(Cloud Computing) 기반의 DNSNA를 소개한다. 클라우드 컴퓨팅은 최소한의 관리노력으로 사용자나 기업에게 데이터 센터의 데이터를 저장, 가공하는 다양한 기능을 제공한다. 클라우드 서버를 이용하게 되면 기존의 기업 및 연구기관에서 선행 투자비용 (예, 서버구현)을 획기적으로 감소시켜 핵심사업에 집중 할 수 있도록 하는 역할을 한다[16]. 본 클라우드 기반의 DNSNA 구조에서는 그림 8과 같이 시스템을 구성할 수 있으며 크게 6개의 동작으로 이루어진다[17].



〈그림 8. 클라우드 서비스 기반의 DNSNA 시스템 설정〉

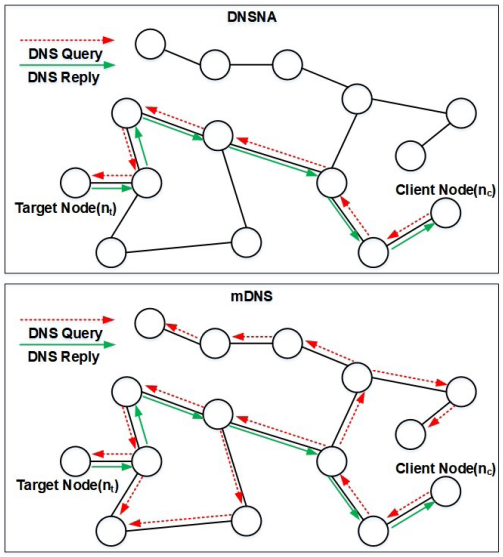
첫째, DNSNA, DNSNAv4 등의 과정을 통해 DNS 서버에 네임을 등록한다. 둘째, 로컬에 위치한 DNS 서버는 글로벌 서버인 Cloud 서버에 자신이 가지고 있는 DNS 네임 리스트와 서비스 리스트를 업로드 한다. 셋째, 사용자는 스마트 기기를 이용하여 Cloud 서버에 접속하여 자신의 Domain 정보를 활용해 사용자의 사물인터넷 기기를 식별한다. 넷째, 사용자는 HTTP 메시지를 이용해 원격에서 관리 및 제어 메시지를 생성한다. 다섯째, 수신한 HTTP 메시지를 라우터(또는 프락시)를 활용해 CoAP 메시지로 변환한다. 이때 변환방식은 [18]에서 제안한 표준방식을 준수한다. 여섯째, 변환된 CoAP 메시지를 활용하여 로컬에 위치한 사물인터넷 기기를 원격에서 관리 및 제어가 가능하다.

VI. 성능분석

본 절에서는 우리의 실험 환경과 실험에 사용된 장비들을 소개하고 가장 기본이 되는 DNSNA와 mDNS의 성능을 비교, 분석한 결과를 설명한다. 1장 서론에서 소개한 것처럼 본 성능평가의 내용은 [6]에서 사용한 자료를 이용해 설명한다.

- **실험환경:** DNSNA의 성능을 검증하기 위하여 두 가지의 케이스를 이용해 성능평가를 진행하였다. 먼저 첫 번째 실험 케이스는 IPv6 환경에서의 실험이며, 두 번째 실험 케이스는 IPv4 환경에서의 실험을 진행할 예정이다.
- **하드웨어설정:** 각 환경에서 DNSNA를 실험하기 위해 사물인터넷 기기는 Raspberry Pi, DNS 서버로는 Desktop PC, 인증서버로 사용되는 Desktop PC 및 라우터의 역할을 하는 Desktop PC를 활용하였다. 또한 삼성 Galaxy Note 3(SM-N900S)를 사용자 디바이스로 사용하였다. 라우터로 사용되는 Desktop PC는 'radvd'[19] 및 'isc-dhcp-server'[20]를 사용하였으며, DNS 서버로 활용된 Desktop은 'bind9'[21]을 이용해 DNS 서버로 동작하였다. 각각의 Desktop PC는 3.4GHz CPU(Intel i7-3770) 및 16GB Memory RAM을 장착하였으며, Raspberry Pi는 900Mhz 32비트 CPU(ARM Cortex-A7) 및 1GB Memory RAM을 장착하였고 동시에 IPv6 및 IPv4 환경에서 사물인터넷 기기로 활용되었다.

6.1 DNS 네임 레졸루션에 대한 메시지 사용량 비교



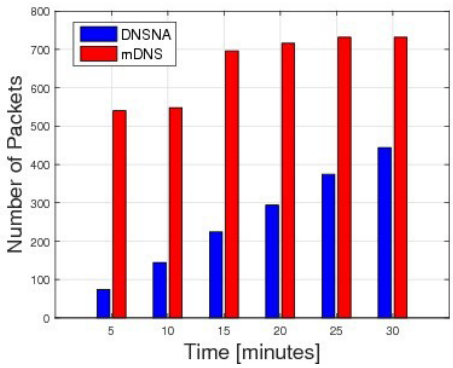
〈그림 9. 멀티 홉 네트워크 환경에서 DNSNA 및 mDNS의 메시지 사용량 비교〉

그림 9는 멀티 홉으로 구성된 네트워크에서 DNSNA와 mDNS의 메시지 사용량을 비교한다. 먼저 DNSNA의 경우

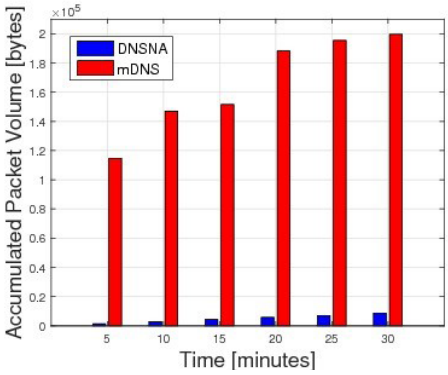
유니캐스트를 기반으로 DNS 네임 레졸루션을 수행하기 때문에 DNS 쿼리 메시지의 수는 클라이언트 노드(n_c)에서 대상 노드(n_t)까지의 최대 홉(Hop)수로 결정할 수 있다. 하지만 mDNS의 경우 네트워크 내에서 멀티캐스트를 이용해 레졸루션을 수행하기 때문에 DNS 레졸루션의 메시지 수는 네트워크의 총 링크 수로 결정된다. 따라서 DNSNA의 경우 클라이언트 노드부터 타겟 노드까지 m 홉이라고 가정하면 최악의 경우 2 × m 개의 메시지가 사용될 것이며 O(m)으로 표현할 수 있다. 하지만 mDNS의 경우는 네트워크 링크의 갯수에 해당하는 O(|E|)가 DNS 네임 레졸루션을 위한 메시지 개수가 된다.

6.2 DNS 네임 레졸루션에 대한 패킷 개수 비교

DNSNA와 mDNS의 패킷 사용량 비교를 위해 mDNS 구현체인 'Avahi' [22]을 사용하여 mDNS에 대한 패킷 사용량을 실험하였다. 3개의 사물인터넷 기기를 활용하여 mDNS 및 DNSNA의 대한 패킷 사용량을 'Wireshark' [23] 툴을 이용해 네트워크에 생성되는 패킷의 수를 계산하였다. 또한 각 테스트베드의 네트워크는 성균관대학교 자연과학캠퍼스의 하위 네트워크에 속해있으며, 각 장치의 패킷 사용량을 계산하기 위해 각 장치의 패킷을 5분 간격으로 30분 동안 측정하고 총 사용한 패킷의 양을 합산하였다. 패킷의 합산 기준은 mDNS의 경우 mDNS 프로토콜과 관련이 있으며 사전에 구성된 DNS 네임 레코드만 합산을 하였으며, 각각 사물인터넷 기기구분을 위해 사물인터넷 기기들의 IP 주소를 구분자로 사용하였다. DNSNA 패킷은 mDNS와 비슷하게 사전에 구성된 DNS 네임을 사용하여 패킷 개수를 합산하였으며, 각각 사물인터넷 기기 구분을 위해 IP 주소를 구분자로 사용했으며, DNSNA의 ICMPv6 기반 프로토콜(예, RA 및 ND)은 합산하지 않았다.



〈그림 10. 시간에 따른 DNSNA와 mDNS의 패킷 개수 비교 [6]〉



〈그림 11. 시간에 따른 DNSNA와 mDNS의 누적 패킷 개수 비교 [6]〉

그림 10과 그림 11은 시간에 따른 DNSNA와 mDNS의 패킷 개수 및 누적된 패킷의 데이터량(볼륨, Volume)의 비교를 보여준다. DNSNA는 모든 측정 시간에 대해 mDNS 보다 적은 수의 패킷 개수를 나타내고 있으며 시간에 따라 mDNS 보다 평균 60.8%까지 감소시킨다. 또한 그림 11은 모든 측정 시간에 대해 mDNS보다 작은 누적 패킷 볼륨(Byte)을 생성하며 최대 97% 감소된 누적 패킷 양을 보여준다. mDNS를 실행하는 사물인터넷 기기는 다른 디바이스들의 서비스정보를 포함한 자신의 DNS 쿼리를 멀티캐스트를 이용하여 네트워크로 전송하기 때문에 mDNS가 DNSNA보다 많은 트래픽을 생성한다. 반면에 DNSNA는 모든 사물인터넷 기기가 서비스를 포함한 DNS 쿼리를 유니캐스트를 이용하여 지정된 DNS 서버로 전송하기 때문에 적은 양의 트래픽을 발생시킨다. 따라서 네트워크가 현재 실험환경보다 거대해지는 경우 성능차이는 더욱 커질 것으로 예상된다. 또한 사물인터넷 기기는 기존의 기기와 달리 저전력 및 적은 용량의 메모리를 가지고 있어, 발생시키는 패킷 및 트래픽 량이 많아짐에 따라 사물인터넷 기기에게 무시할 수 없는 부담과 에너지 소비를 초래한다.

〈표 1. DNSNA 및 mDNS의 비교〉

Approaches	DNSNA	mDNS
Packet Forwarding	Unicasting	Multicasting
Packet Size	About 100 bytes	About 1500 bytes
Naming Scope	Global, Local	Local
Code Size	Hundreds	Thousands
Authoritative DNS Server	Used	Not used
Working Layer	Network Later	Application Layer
Energy Consumption for Resolution	O(m)	O(|E|)

DNSNA와 mDNS의 비교를 표 1에 나타낸다. 표 1에서 패킷의 크기는 경우에 따라 달라질 수 있지만 mDNS는 네트워

크에 위치한 다른 디바이스의 정보를 포함하는 패킷을 생성하기 때문에 패킷 사이즈가 증가한다. 또한 DNSNA는 mDNS보다 적은 코드라인으로 구현이 되며, 로컬 IoT 네트워크뿐만 아니라 글로벌 IoT 네트워크에서도 사용될 수 있다.

VII. 결론

본 고에서는 네트워크에 위치한 사물인터넷 기기를 위한 DNS 네임 자동설정 기법(DNSNA)에 대해서 소개하였다. 사물인터넷 기기가 증가함에 따라 기기 관리 및 제어를 위한 설정이나 관리를 사용자가 수동으로 하는 것은 매우 비효율적이다. 따라서 효율적으로 사물인터넷 기기 관제를 위해 DNSNA를 활용할 수 있다. 또한 DNSNA는 NFC를 활용하여 보안이 취약한 사물인터넷 기기에 대해 보안성을 제공할 수 있다. 추후에는 광범위하게 사물인터넷 기기 인증을 위하여 Wi-Fi, ZigBee 및 Bluetooth를 활용할 예정이며, 현재에는 단순텍스트를 활용하여 사물인터넷 기기의 정보를 전달하였지만 추후에는 기기의 위치를 추정하는 기법(예, SALA [24])을 활용하여, 직관적으로 사물인터넷 기기를 식별할 수 있도록 하는 기법을 개발할 예정이다.

Acknowledgement

본 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단 – 차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(No. 2017M3C4A7065980).

References

[1] Sejun Lee, Jaehoon (Paul) Jeong, and Jung-Soo Park, "DNSNA: DNS Name Autoconfiguration for Internet of Things Devices", The 18th International Conference on Advanced Communications Technology (ICACT 2016), Outstanding Paper, Phoenix Park, PyeongChang, Korea, Jan. 31-Feb. 3, 2016.

[2] Keuntae Lee, Seokhwa Kim, and Jaehoon (Paul) Jeong, "DNSNAv4: DNS Name Autoconfiguration for Internet-of-Things Devices in IPv4 Networks", The 31th International Conference on Advanced Information Networking and Applications Workshops – Device Centric Cloud (DC2), Best Paper Award, Taipei, Taiwan, Mar. 27-29, 2017.

[3] Seokhwa Kim and Jaehoon (Paul) Jeong, "Service Discovery and Remote Control for Internet-of-Things Devices", KICS 2017-Summer, Jun. 2017.

[4] Seokhwa Kim, Keuntae Lee, and Jaehoon (Paul) Jeong, "DNS Naming Services for Service Discovery and Remote Control for Internet-of-Things Devices", International Conference on ICT Convergence (ICTC 2017), Jeju, Korea, Oct. 18-20, 2017.

[5] Seokhwa Kim, and Jaehoon (Paul) Jeong, "DNS Naming Service for Internet-of-Things Devices Based on Cloud System", KICS 2017-Fall, Dec. 2017.

[6] Keuntae Lee, Seokhwa Kim, Jaehoon (Paul) Jeong, Sejun Lee, Hyoungshick Kim, and Jung-Soo Park, "A Framework for DNS Naming Services for Internet-of-Things Devices", Elsevier Future Generation Computer Systems, Jan. 2018.

[7] Keuntae Lee, Hyungsuk Kang, Jaehoon (Paul) Jeong, Hyoungshick Kim, and Jung-Soo Park, "Secure DNS Name Autoconfiguration for IPv6 Internet-of-Things Devices", International Conference on ICT Convergence (ICTC 2016), Jeju, Korea, Oct. 19-21, 2016.

[8] S. Cheshire and M. Krochmal, "Multicast DNS", IETF RFC 6762, Feb. 2013.

[9] S. Cheshire and M. Krochmal, "DNS-based Service Discovery", IETF RFC 6763, Feb. 2013.

[10] oneM2M, "Object Identifier based M2M Device Identification Scheme", http://www.onem2m.org/images/files/deliverables/TS-0001-Functional_Architecture-V1_13_1.pdf. Accessed Mar. 26, 2018.

[11] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)", IETF RFC 7252, Jun. 2014.

[12] A. Castellani, S. Loreto, A. Rahman, T. Fossati, and E. Dijk, "Guidelines for Mapping Implementations: HTTP to the Constrained Application Protocol (CoAP)", IETF RFC 8075, Feb. 2017.

[13] M. Stapp, B. Volz, and Y. Rekhter, "The Dynamic Host Configuration Protocol (DHCP) Client Fully Qualified Domain Name (FQDN) Otion", IETF RFC 4702, Oct. 2006.

[14] J. Jeong, S. Park, L. Beloeil, and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", IETF RFC 8106, Mar. 2017.

[15] A. Gulbrandsen, P. Vixie, and L. Esibov “A DNS RR for specifying the location of services (DNS SRV)”, IETF RFC 2782, Feb. 2000.
[16] AWS Cloud Computing, https://aws.amazon.com, Accessed Mar. 2013.
[17] Seokhwa Kim, and Jaehoon (Paul) Jeong, "DNS Naming Service for Internet-of-Things Devices Based on Cloud System", KICS 2017-Fall, Dec. 2017.
[18] Angelo P. Castellani, and Thomas Fossati, and Salvatore Loreto, “HTTP-CoAP Cross Protocol Proxy: An Implementation Viewpoint”, IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012), Las Vegas, Nov. 2012.
[19] RADVD, http://www.litech.org/radvd/. Accessed Dec. 13, 2017.
[20] DHCP, https://www.isc.org/downloads/dhcp/. Accessed Dec. 13, 2017.
[21] BIND, https://www.isc.org/downloads/bind/. Accessed Dec. 13, 2017.
[22] Avahi, https://www.avahi.org/doxygen/html/index.html. Accessed Dec. 13, 2017.
[23] Wireshark, https://www.wireshark.org/. Accessed Dec. 13, 2017.
[24] J. Jeong, S. Yeon, T. Kim, H. Lee, S. Kim, and S. Kim, “SALA: Smartphone-Assisted Localization Algorithm for Positioning Indoor IoT Devices”, Springer Wireless Networks, Vol. 24, No. 1, Jan. 2018.

Biographies

김석화

2017년: 백석대학교 정보통신학부 학사
2017년 ~ 현재: 성균관대학교 전자전기컴퓨터학과 석사과정
주요 관심분야: 사이버물리시스템, 사물인터넷, 차량네트워크
e-mail: seokhwakim@skku.edu

정재훈

1999년: 성균관대학교 정보공학과 학사
2001년: 서울대학교 컴퓨터공학과 석사
2009년: 미네소타대학교 컴퓨터공학과 박사
2001년 ~ 2004년: 한국전자통신연구원 표준연구센터 연구원
2010년 ~ 2012년: Brocade(현재 Broadcom) 소프트웨어 엔지니어
2012년 ~ 현재: 성균관대학교 소프트웨어학과 교수
주요 관심분야: 사이버물리시스템, 사물인터넷, 차량네트워크, 소프트웨어 정의 네트워크, 네트워크 기능 가상화, 네트워크 보안
e-mail: pauljeong@skku.edu