ISSN 2383-630X(Print) / ISSN 2383-6296(Online)
Journal of KIISE, Vol. 42, No. 11, pp. 1441-1451, 2015. 11
http://dx.doi.org/10.5626/JOK.2015.42.11.1441

사물인터넷 디바이스를 위한 DNS 네임 자동설정의 설계 및 구현

(Design and Implementation of DNS Name Autoconfiguration for Internet of Things Devices)

이 세 준 [†]　　정 재 훈 [††]
(Sejun Lee)　　(Jaehoon Jeong)

요 약 최근 가장 주목받고 있는 연구 분야 중 하나인 사물인터넷(Internet of Things, IoT)은 네트워크에 연결된 매우 많은 디바이스를 통해 사용자에게 다양한 서비스를 제공하는 것을 목표로 한다. IoT 환경에서 IoT 디바이스는 매우 많은 개수가 사용되는데 각 IoT 디바이스에 대한 DNS(Domain Name System) 네임을 일일이 수동으로 설정하는 것은 비효율적이다. 따라서 본 논문에서는 IPv6 기반의 IoT 환경에서 IoT 디바이스의 DNS 네임을 자동으로 생성하고 관리하는 DNS Name Autoconfiguration (DNSNA)이라는 기법을 제안한다. DNS 네임을 생성 및 등록하는 과정에서 Internet Engineering Task Force(IETF)에서 재정된 표준 프로토콜을 이용한다. 본 기법은 유니캐스트로 DNS 서버를 통해 IoT 디바이스의 DNS 네임을 IPv6 주소로 레졸루션(Resolution)하기 때문에 멀티링크 네트워크 환경에서는 기존의 멀티캐스트 기반의 mDNS(Multicast DNS) 기법보다 트래픽을 적게 발생시킨다. 따라서 본 기법은 멀티홉으로 구성된 IoT 네트워크에서 mDNS 보다 더 적합하다. 본 논문은 제안한 기법의 디자인과 스마트 홈과 스마트 로드에서의 서비스 시나리오를 설명한다. 또한 본 논문은 스마트 그리드 환경에서 구현 및 테스트에 대하여 설명한다.

키워드: IPv6, 사물인터넷, 디바이스, DNS, 자동설정, 이웃탐색

Abstract As one of the most spotlighted research areas, these days, the Internet of Things (IoT) aims to provide users with various services through many devices. Since there exist so many devices in IoT environments, it is inefficient to manually configure the domain name system (DNS) names of such devices. Thus, for IPv6-based IoT environments, this paper proposes a scheme called the DNS Name Autoconfiguration (DNSNA) that autoconfigures an IoT device's DNS name and manages it. In the procedure for generating and registering an IoT device's DNS name, the standard protocols of the Internet Engineering Task Force (IETF) are used. Since the proposed scheme resolves an IoT device's DNS name into an IPv6 address in unicast through a DNS server, it generates less traffic than multicast-based mDNS (Multicast DNS) which is a legacy DNS application for the DNS name service in the smart home. Thus, the proposed scheme is more appropriate in multi-hop IoT networks than

· 본 연구는 교육부의 재원으로 지원을 받아 수행된 산학협력 선도대학(LINC) 육성사업[2015-A-0002-010113]과 2014년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업[2014R1A1A1006438]과 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업[R0166-15-1041, 중앙 집중제어 기반 네트워크 보안기술 표준개발]의 일환으로 수행하였음

논문접수 : 2015년 8월 18일
(Received 18 August 2015)
논문수정 : 2015년 9월 9일
(Revised 9 September 2015)
심사완료 : 2015년 9월 9일
(Accepted 9 September 2015)

† 학생회원 : 성균관대학교 전자전기컴퓨터공학
prosejun14@gmail.com
†† 정 회 원 : 성균관대학교 인터랙션사이언스학과 교수
(Sungkyunkwan Univ.)
pauljeong@skku.edu
(Corresponding author임)

Copyright©2015 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

mDNS. This paper explains the design of the proposed scheme and its service scenarios, such as smart home and smart road. It also explains the implementation and testing of the proposed scheme in the smart grid.

Keywords: IPv6, internet of things, device, DNS, autoconfiguration, neighbor discovery

1. 서 론

Internet of Things(IoT)는 최근 가장 주목받고 있는 인터넷 관련 기술로 스마트 디바이스와의 연동을 통해 주변 센서로부터 수집된 정보를 가공 및 처리하여 다양한 서비스를 제공한다[1]. 미국의 시장조사기관인 가트너는 2014년도에 발표한 Hype Cycle에서 향후 5년에서 10년 정도 가장 유망한 미래의 기술로 IoT를 선정했고, 2018년도까지 80억 IoT 디바이스가 증가해서 총 160억 IoT 디바이스가 존재할 것이라고 발표했다[2].

IPv4는 인터넷 주소 고갈이라는 한계점이 있어서 이를 극복하기 위한 기술로 IPv6 인터넷 프로토콜이 등장하였다[3]. IoT 기술은 많은 분야에 적용되며 네트워킹 기능을 포함한 IoT 디바이스가 증가함에 따라 차세대 인터넷 프로토콜인 IPv6를 통해 IoT 디바이스 관리를 효율적으로 할 수 있다. 한편, Domain Name System (DNS)은 사용자가 IP 주소를 쉽게 기억할 수 있는 문자열로 대응시키는 서비스로써 인터넷에 연결된 개체에 대한 IP 주소와 DNS 네임을 관리한다[4]. 하지만 IoT 환경에서 IoT 디바이스에 대한 DNS 네임을 일일이 수동으로 설정하는 것은 매우 번거로운 일이다. 이를 해결하는 방안으로 애플(Apple Inc.)에서 개발한 봉주르(Bonjour)는 제로콘프(Zeroconf)를 구현한 소프트웨어인 multicast Domain Name System(mDNS)를 통해 DNS 네임을 자동으로 부여하고 도메인 네이밍 서비스를 지원한다[5]. mDNS에서 각 노드는 자신의 DNS 서버를 운영하며 자신의 DNS 네임에 대한 쿼리를 요청하는 노드에게 IP 주소를 전달해준다. 봉주르에서 제공해주는 네이밍은 같은 로컬 안에 사용 중인 기기들의 DNS 네임과 그에 대응하는 IPv4 또는 IPv6 주소를 특별한 보안키 없이 보여주기 때문에 프라이버시에 문제가 있다. 기본적으로 멀티캐스트 기반으로 DNS 네임 레졸루션(Resolution)을 수행하므로 네트워크에 많은 트래픽을 발생시킨다. 또한 센서와 같이 Duty Cycling을 통해 슬립모드(Sleep Mode)에 있는 디바이스는 DNS 네임쿼리에 응답하지 않을 수도 있다. 이와 같이 mDNS는 멀티홉으로 구성된 IP 네트워크와 Duty Cycling을 하는 센서와 같은 IoT 디바이스를 위한 네이밍 서비스로는 적합하지 않다.

이러한 IoT DNS 네이밍 서비스를 제공하기 위해 본 논문은 IPv6의 이웃탐색 프로토콜(Neighbor Discovery, ND)[6] 기반의 단순하고 효과적인 IoT 디바이스를 위한 DNS Name Autoconfiguration(DNSNA)이라는 기법을 제안한다. ND의 라우터광고(Router Advertisement, RA) 메시지는 DNS Search List(DNSSL)[7] 옵션을 통해 멀티캐스트로 IoT 디바이스에게 링크가 해당되는 도메인 정보인 DNS Suffix를 제공한다. IoT 디바이스는 이렇게 전달받은 DNS Suffix에 디바이스 정보를(예, Device Category, Device Model 및 Unique ID) 결합하여 DNS 네임을 생성한다. IoT 디바이스는 생성된 DNS 네임이 현재 링크에서 유일한지 검사한다. 차후에 현재 링크의 라우터는 Node Information(NI) Query[8]를 통해 링크에 있는 IoT 디바이스의 DNS 네임과 대응하는 IPv6 주소를 수집한다. 라우터는 수집하는 IoT 디바이스의 DNS 정보는 DNS 동적갱신(Dynamic Update)[9]을 통해 도메인을 관리하는 DNS 서버에게 등록한다. 이러한 등록절차를 통해 특정 네트워크 내의 IoT 디바이스의 DNS 정보는 DNS 서버에 등록되어 있기 때문에 어떤 노드가 다른 노드의 DNS 네임의 IPv6 주소를 알고 싶을 때는 기존의 DNS 레졸루션과 동일하게 유니캐스트로 DNS 서버에게 DNS 쿼리를 전송한다. 이와 같이 유니캐스트 기반의 네임 레졸루션이 가능하므로 mDNS보다 적은 트래픽으로 네임 레졸루션을 수행한다. 아울러 노드가 DNS 서버를 통해 IoT 디바이스에 대한 DNS 네임 정보를 참조하므로 IoT 디바이스가 슬립모드에 있을 때도 해당 IoT 디바이스에 대한 DNS 네임을 확인 및 제어를 할 수 있다. 따라서 제안하는 기법은 규모가 큰 네트워크뿐만 아니라 규모가 작은 네트워크에서도 IoT 디바이스 도메인 관리를 효과적으로 할 수 있다.

본 논문은 다음과 같이 구성된다. 2장에서는 본 제안하는 기법에 관련된 선행연구에 대한 내용을 다룬다. 3장에서는 제안하는 기법에 대한 프로토콜을 설명하고, 4장에서는 실제 네트워크에서 테스트결과를 분석한 후 기존의 다른 기법과 비교 분석을 한다. 마지막으로 5장에서는 결론 및 향후연구에 대해 논의한다.

2. 관련연구

ND는 IPv6 호스트가 자신의 디폴트 게이트웨이를 찾거나, IPv6 주소를 링크 계층의 주소로 변환하는 일을 할 수 있게 하는 IPv6 기본프로토콜이다. ND는 Internet Control Message Protocol(ICMPv6) 메시지[10]를 통해 동작하고 ND는 IPv6 호스트의 IPv6 주소 자동설정

(Autoconfiguration)을 가능하게 한다. Autoconfiguration은 Router Solicitation(RS) 및 RA를 통해 수행한다. RS는 새로운 네트워크 영역에 노드가 연결될 때 해당 서브넷을 관리하는 라우터에 신속하게 RA 메시지 전송을 요청하는 메시지로 노드의 인터페이스가 이용 가능 상태가 되면 RS를 통해 라우터에 RA 메시지를 즉시 전송하도록 요청한다. RA는 라우터가 RS 메시지에 대한 응답으로 RA 메시지를 전송하거나 멀티캐스팅이 가능할 경우 주기적으로 해당 서브넷에 위치한 노드에 라우터 자신의 존재를 알리기 위해 라우터의 링크로컬(Link Local) 주소와 네트워크 프리픽스(Prefix) 정보를 담은 RA 메시지를 전송한다. RA 메시지는 DNS 레졸루션을 수행하는 Recursive DNS Server(RDNSS) 주소를 포함하는 RDNSS 옵션과 링크를 포함하는 네트워크의 DNS 도메인을 포함하는 DNSSL를 포함하는 DNSSL 옵션을 전달할 수 있다[7].

애플에서 개발한 봉주르는 DNS 네이밍 및 서비스 디스커버리를 제공하는 제로콘프 소프트웨어이다[11]. 봉주르를 이용하면 로컬 네트워크상에서 애플기기 등을 검색 및 등록할 수 있으며, 기기들이 제공하는 서비스를 알 수 있다. 봉주르는 mDNS[5]을 이용해서 로컬 네트워크에 참여한 기기들의 DNS 네임을 기기 또는 사용자 ID와 DNS Suffix를 기반으로 자동생성하여 DNS 네이밍 서비스를 제공한다. mDNS는 각 노드는 자신의 DNS 서버를 데몬형태로 운영하며 자신의 DNS 네임에 대한 쿼리를 요청하는 노드에게 자신의 DNS 네임에 대한 IP 주소를 전달해준다. 하지만 mDNS는 기본적으로 멀티캐스트 기반으로 네임 레졸루션을 수행하므로 네트워크에 많은 트래픽을 발생시키므로 멀티홉으로 구성된 네트워크에서는 부적합하다. 또한 인터넷에 연결된 디바이스가 항상 동작하는 경우에만 DNS 네임 요청 메시지에 응답할 수 있고 에너지 절약을 위해 슬립모드를 지원하는 IoT 디바이스에 사용되기에는 부적합하다. 또한 제공하는 DNS 네임이 평문(Plaintext)으로 공개되므로 디바이스의 정보가 쉽게 공유되므로 의료기기와 같은 IoT 디바이스를 위한 네이밍 서비스를 위해서는 적합하지 않다. 봉주르는 DNS Service Discovery(DNS-SD)[12]을 이용해서 서비스 타입을 설정하여 특정 기기의 서비스 검색을 가능하도록 했다. DNS-SD는 등록된 기기의 서비스목록과 포트를 제공한다. DNS의 Pointer Record(PTR)을 이용해서 사용자가 요청할 서비스 타입을 설정한다. 형식은 <Service>.<Domain>이며 Service는 언더바(_)로 시작하여 서비스 이름을 나타내며, Domain은 특정 서비스 안에서 사용되는 도메인을 나타낸다. 위와 같이 서비스 타입을 설정하고 멀티캐스트로 각 기기를 검색하면, 해당하는 기기는 자신의 서비스 타입을 <Instance>.<Service>.<Domain> 형식으로 반환한다. Instance는 사용자가 알아보기 쉽게 고유한 이름을 나타낸다.

[13]은 이동 애드혹 네트워크(Mobile Ad-hoc Network, MANET)환경에서 DNS 네이밍 서비스를 제공한다. 노드의 DNS 네임을 자동으로 설정하기 위해 네트워크 인터페이스의 MAC 주소를 EUI-64(64-bit Extended Unique Identifier) 변환하여, 사용자 또는 디바이스 ID와 MANET을 위한 DNS Suffix(예, adhoc)를 이용해 DNS 네임을 자동생성 및 유일성을 검증한다. 하지만 EUI-64 기반의 DNS 네임은 일반 사용자들이 이해하기 어렵고, 다른 디바이스들이 이러한 DNS 네임을 통해 유용한 정보를 얻을 수 없다. 이와 반면에 본 논문이 제안하는 DNSNA는 디바이스 정보(예, Device Type과 Device Model)을 기반으로 DNS 네임을 생성하므로 DNS 네임만으로 디바이스의 기본적인 정보를 파악할 수 있다.

3. DNSNA 네이밍 시스템

본 논문은 향후 증가하는 IoT 디바이스의 DNS 네임 관리를 위해 IPv4 주소 고갈의 문제점이 있어 IPv6 주소 기반을 두어 IoT 디바이스 관리를 위한 도메인 네이밍 시스템을 제안한다. 그림 1은 응용분야인 한 예로 스마트 그리드 환경에서의 도메인 네이밍 시스템의 아키텍처를 보여준다. 과거에는 전력 사업자가 고객에게 전력을 공급한 전력량을 기록한 미터기를 읽기위해 방문 검침을 한다. 수많은 미터기 있는 환경에서는 방문 검침은 비효율적이다. 현재는 전력망에 정보통신기술을 접목해 무선으로 미터기를 읽는 기술이 개발되었다. 그림 1과 같이 IPv6/6lo[14]/Wi-SUN[15]을 통해 미터기의 전력량을 수집하여 미터기를 효율적으로 관리할 수 있다. 스마트 그리드 환경에서 댁내에 설치된 IoT 디바이스인 수많은 미터기의 IP주소를 이해 및 기억하기 쉽게 하기 위해서 DNS 네임이 필요하다. 본 논문에서는 IoT 디바이스의 DNS 네임을 자동으로 생성 및 등록하고 관리하는 방법을 제안한다.

3.1 DNS 네임의 형태

본 논문에서 제안하는 DNS 네임은 그림 2와 같은 형태를 지닌다. DNS 네임은 호스트 네임과 도메인 네임이 결합한 FQDN(Fully Qualified Domain Name)형태로 점(.)으로 구분되며, 그림 2와 같이 총 4개의 문자열로 구성되어 있다[16].

unique_id는 서브넷에 존재하는 동일한 모델(예, 냉장고 모델인 samsung_RH269LP)에 대해 시리얼번호를 가지는 고유 식별자(예, refrigerator1)를 의미한다. device_model은 IoT 디바이스를 제조한 제조사 또는 판매자가 제공하는 디바이스 모델(예, samsung_RH269LP)을 말한다. 디바이스 모델은 제품을 생산한 제조사의 이름이

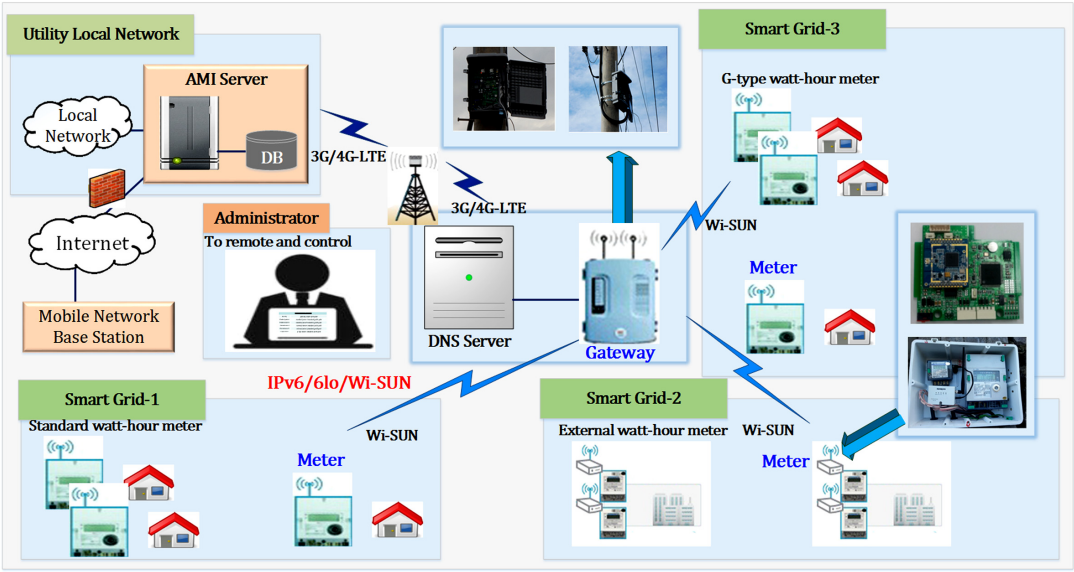


그림 1 스마트 그리드 환경에서의 도메인 네이밍 시스템 아키텍처
Fig. 1 The Architecture of Domain Naming in the Smart Grid

unique_id.device_model.device_category.domain_name

그림 2 DNS 네임 형태
Fig. 2 DNS Name Format

unique_id.device_model.device_category.location_of_device.domain_name

그림 3 위치정보를 포함한 DNS 네임 형태
Fig. 3 DNS Name Format with Location Information

포함될 수 있다. device_category는 IoT 디바이스의 카테고리(예, refrigerator)를 나타낸다. domain_name은 IoT 디바이스가 갖는 네트워크 도메인의 DNS Suffix (예, home)를 의미한다. 응용분야에 따라 DNS 네임을 구성하는 요소의 순서는 그림 2와 다를 수 있다. 디바이스 모델 및 디바이스 카테고리는 사전에 IoT 디바이스에 저장된 정보에 해당한다. 고유 식별자는 DNS 네임 생성 과정에서 IoT 디바이스가 생성하는 정보이다.

IoT 디바이스의 개체 수가 많을 경우 그림 2와 같은 DNS 네임 형태 정보만으로는 IoT 디바이스를 구분하기 어려울 수 있다. 이 경우 DNS 네임은 IoT 디바이스의 위치 정보라는 추가적인 정보를 포함할 수 있다. 장치의 구별 가능성과는 관계없이 DNS 네임에 포함되는 위치 정보는 새로운 정보를 사용자에게 제공할 수 있다. 위치정보가 포함된 DNS 네임의 예는 그림 3과 같다. 그림 3의 DNS 네임은 그림 2의 DNS 네임에 location_of_device가 추가되었다. location_of_device는 IoT 디바이스의 위치 정보를 의미한다. 위치 정보는 IoT 디바이스가 사전에 미리 획득한 것을 전제로 한다. 위치 정보는 다음과 같이 다양한 방법으로 수집할 수 있다. IoT 디바이스가 GPS 수신기와 같이 위치 추적 장치가 있다면 GPS 좌표와 같은 절대적인 좌표 정보를 사용할 수 있다. IoT 디바이스가 가속도 센서(Accelerometer), 자이로스코드 센서(Gyroscope), 지자기 센서(Magnetometer) 등을 내장하고 있다면 특정 기준 지점을 기준으로 IoT 디바이스가 이동하여 위치한 지점에 대한 상대적인 위치 정보를 생성할 수 있다. 특정 영역에 이미 일정한 위치 정보를 보유하고 있는 다른 IoT 디바이스나 장치가 있다고 가정하면, IoT 디바이스는 무선 통신을 통해 통신 가능한 거리에 특정 IoT 디바이스가 있다는 것을 알 수 있다. 따라서 IoT 디바이스는 주변에 있는 다른 장치의 위치 정보를 참조하여 자신의 위치를 대략 추정할 수 있다.

3.2 도메인 네이밍 시스템의 동작

본 절에서는 도메인 네이밍 시스템의 동작과정을 설명한다. 첫 번째는 IoT 디바이스가 네트워크에 최초로 접속하고 DNS 네임을 자동으로 생성하여 유일성을 테스트하는 단계를 설명한다. 두 번째는 IoT 디바이스가 DNS 네임을 생성한 후에 라우터가 NI Query 프로토콜[8]을 이용해서 DNS 네임을 수집하여 DNS 동적 갱신[9]을 통해 DNS 서버에 등록하는 단계를 설명한다. 세 번째는 사용자가 스마트폰같은 클라이언트 장치를

이용해 DNS 서버로부터 IoT 디바이스의 DNS 네임 목록을 가져와 IoT 디바이스를 제어하는 단계를 설명한다.

3.2.1 IoT 디바이스의 DNS 네임 생성

그림 4는 IoT 디바이스(냉장고)의 DNS 네임 생성 과정을 기술한다. IoT 디바이스는 서브넷에 접속되었을 때 서브넷을 관장하는 라우터를 통해 DNSSL 옵션을 IPv6 표준에 따른 RA 또는 Dynamic Host Configuration Protocol(DHCP)[17]을 통해 수신한다. DNSSL은 IoT 디바이스가 DNS 네임을 등록하는 과정에서 사용하는 DNS 도메인 리스트를 포함한다. IPv6 호스트인 IoT 디바이스는 수신한 DNSSL 옵션이 유효하다면 그림 2와 같이 자신의 DNS 네임(refrigerator1.samsung_RH269LP.refrigerator.kitchen.home)을 생성한다. 냉장고의 DNS 네임 중 “refrigerator1”은 같은 모델의 디바이스를 구별하기 위한 고유 식별자이고, “samsung_RH269LP”는 제조사 이름을 포함하는 디바이스 모델이고, “refrigerator”는 디바이스 카테고리, “home”은 DNS Suffix에 해당한다.

IoT 디바이스는 생성한 DNS 네임이 서브넷에서 유일한 DNS 네임인지 확인하는 것이 바람직하다. 이러한 DNS 네임의 유일성 보장을 위해서 IoT 디바이스는 생성한 DNS 네임을 링크로컬 범위의 멀티캐스트 주소를 생성하고 이 주소에 해당되는 멀티캐스트 그룹에 조인해서 다른 IoT 디바이스가 동일한 DNS 네임을 사용하고자 할 때 중복 검사과정에 응답하여 동일한 DNS 네임이 사용되는 것을 방지할 수 있다. DNS 네임에 해당되는 멀티캐스트 주소는 IPv6 멀티캐스트 프리픽스인 ff02::/16을 이용하고, 네트워크 인터페이스 ID에 해당하는 하위 64 비트를 위해 DNS 네임에 대한 MD5 해싱 값 128 비트 중에서 Most Significant Bit(MSB)를 포함하는 상위 64 비트를 취한다. 새로운 DNS 네임을 생성한 IoT 디바이스는 다음과 같이 ND 프로토콜에 따라 DNS 네임에 대한 중복 검사를 수행할 수 있다. IoT 디바이스는 IPv6 멀티캐스트 주소(Multicast Address)로 주소 중복 검사(Duplicate Address Detection, DAD)을 수행할 수 있다[18]. DNS 네임이 중복된 경우 IoT 디바이스는 종전의 DNS 네임에서 디바이스의 고유 식별자를 새로운 식별자로 변경한 새로운 DNS 네임을 생성하고, 새로운 DNS 네임에 대한 중복 검사를 수행한다. IoT 디바이스가 서브넷에서 중복되지 않은 DNS 네임을 생성했다고 확인해야 DNS 네임 생성과정이 종료된다.

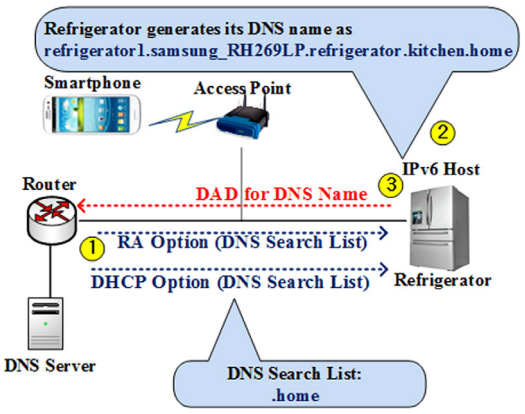


그림 4 IoT 디바이스의 DNS 네임 생성

Fig. 4 IoT Device DNS Name Generation

3.2.2 IoT 디바이스의 DNS 네임 등록

3.2.1절에서 IoT 디바이스는 DNS 네임 생성 및 서브넷에서의 DNS 네임 중복성 검사에 대해 설명했다. 본 절에서는 IoT 디바이스의 DNS 네임을 DNS 서버에 등록하는 과정을 설명한다.

사용자는 클라이언트 장치를 이용해서 DNS 서버에 있는 IoT 디바이스의 DNS 네임을 보고 제어하기 위해 DNS 서버는 서브넷에 있는 IoT 디바이스의 DNS 네임을 관리해야 한다. 그림 5는 IoT 디바이스의 DNS 네임 등록과정을 나타내고 있다. 우선, 서브넷을 관장하는 라우터는 해당 서브넷의 네트워크에 연결된 IoT 디바이스의 DNS 네임 및 대응하는 IPv6 주소를 수집한다. 라우터는 ICMPv6 기반의 NI 프로토콜의 NI Query를 이용하여 IoT 디바이스에 DNS 네임을 묻는다[8]. 이 과정은 종래 NI 프로토콜에서 새로운 NI Query 타입이 필요하다. 즉, 새로운 NI 타입에 대한 새로운 코드가 정의되어야 한다. IoT 디바이스는 NI Query에 대한 응답으로 자신의 DNS 네임 및 IPv6 주소를 포함하는 NI Reply를 랜덤한 지연시간 이후에 라우터에게 전달한다.

라우터는 서브넷에 있는 IoT 디바이스들의 DNS 네임 및 IPv6 주소를 수집한다. 수집이 끝나면 DNS 동적 갱신(Dynamic Update)[9]을 통해 DNS 서버에 전송하고, DNS 서버는 모든 IoT 디바이스의 DNS 네임 및 IPv6 주소를 로컬 저장소에 저장한다.

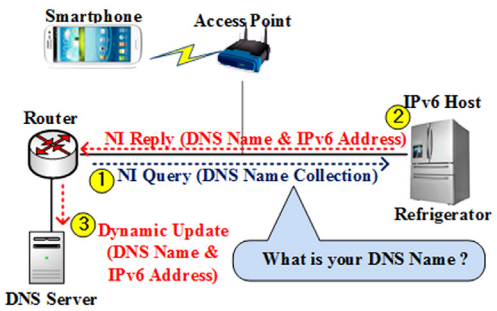


그림 5 IoT 디바이스의 DNS 네임 등록

Fig. 5 IoT Device DNS Name Registration

3.2.3 클라이언트 장치를 이용한 IoT 디바이스의 관리

3.2.2절에서는 DNS 서버에 IoT 디바이스의 DNS 네임 및 IPv6 주소를 저장하는 방법을 설명했다. 본 절에서는 사용자가 DNS 서버로부터 해당 서브넷에 있는 IoT 디바이스의 DNS 네임 목록을 불러와서 제어 및 모니터링을 하는 방법을 설명한다.

사용자는 스마트폰 및 태블릿 PC와 같은 클라이언트 장치를 통해 IoT 디바이스에 대한 정보를 확인하거나, IoT 디바이스를 제어할 수 있다. 클라이언트 장치는 DNS 서버에 접속하여 네트워크에 존재하는 IoT 디바이스에 대한 DNS 네임을 획득한다. 그림 6에서는 클라이언트 장치가 IoT 디바이스를 제어하는 과정이다. 클라이언트 장치는 DNS 서버로부터 네트워크에 존재하는 다수의 IoT 디바이스들에 대한 DNS 네임을 획득한다. 그림 6은 IPv6 호스트인 스마트 냉장고를 예로 보여주고 있으며 클라이언트 장치를 이용해 냉장고의 정보를 확인하거나, 냉장고의 동작(예, 식품의 유통기한, 부족한 식재료 확인)을 제어 및 모니터링을 할 수 있다.

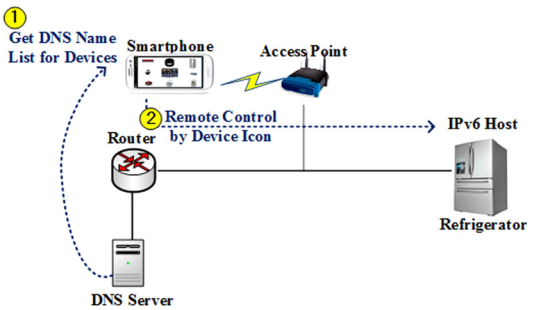


그림 6 클라이언트 장치를 이용한 IoT 디바이스 관리

Fig. 6 Using client machine for management

3.3 DNS 네임의 보안성

애플에서 개발한 봉주르[11]의 네이밍 방법은 보안키 없이 사용자에게 네이밍 정보를 그대로 전달해주기 때문에 개인 정보 유출에 문제가 발생될 수 있다. 따라서 본 논문에서는 DNS 네임의 메시지 전달은 보안성을 유지하기 위한 다양한 사용방법을 제안한다. 개인 정보 유출을 막기 위해 위치와 관련된 DNS 네임은 공유키 또는 공개키를 이용하여 암호화할 수 있다. 예를 들어 smart_tv1.living_room.home이라는 DNS 네임은 암호화 되어 smart_tv1.xxx.home과 같이 암호화될 수 있다. xxx는 living_room에 대해 암호화된 문자열이며, home에 대한 root 권한을 갖고 있는 사용자는 공유키 또는 개인키로 암호화된 위치정보인 living_room을 복호화할 수 있다.

4. DNSNA 서비스 디스커버리

본 논문은 DNS 서버를 관리함으로써 DNS 네임을 효율적으로 관리하며 규모에 제약을 받지 않고 작거나 큰 규모의 IoT 환경에서 DNS 네이밍 및 서비스 디스커버리를 제공한다. 본 절에서는 제안기법을 통해 서비스 디스커버리를 설명한다.

4.1 스마트 로드에서의 네이밍 서비스

IoT 디바이스가 이동하는 스마트 로드 환경에서의 DNS 네이밍 서비스를 설명한다. 그림 7은 스마트로드에서 IoT 디바이스의 DNS 네임 생성 및 등록은 3.2절과

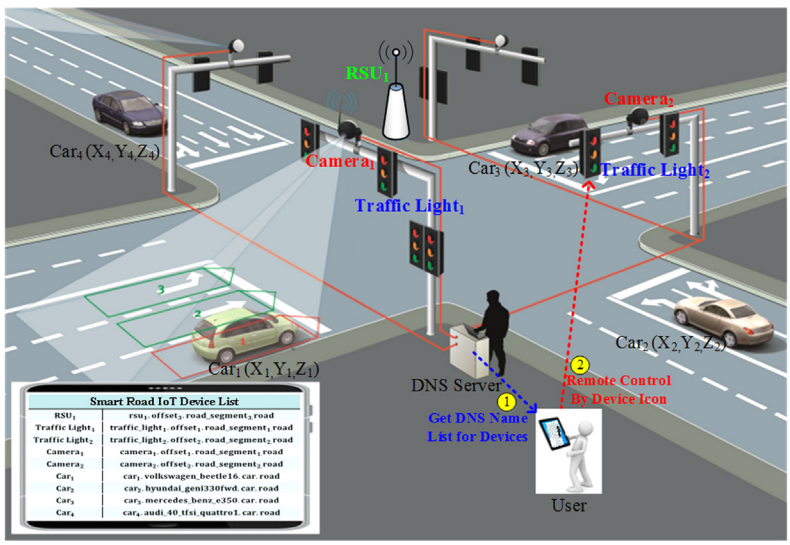
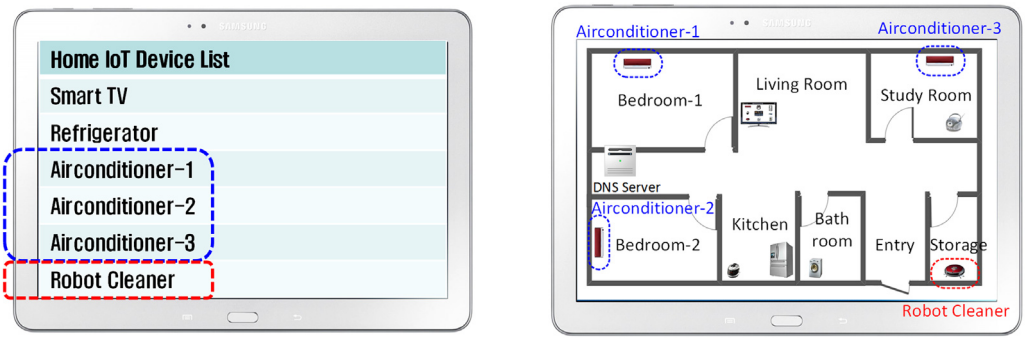


그림 7 스마트 로드에서의 IoT 디바이스들을 위한 DNS 네이밍 서비스

Fig. 7 DNS Naming Service for IoT Devices in the Smart Road



(a) Text-based IoT Device List (b) Graphic-based IoT Device Display

그림 8 DNS 네이밍 서비스 기반의 IoT 디바이스 관리

Fig. 8 Management of IoT Devices based on the DNS Naming Service

같이 동일한 과정으로 수행할 수 있다. 스마트로드에서 Car1 차량은 루프 무선 통신 기능을 제공하는 카메라(Camera1)로부터 DNSSL 옵션을 통해 “road”라는 도메인 정보를 수신한다. Car1 차량은 고유 식별자를 결합해서 car1.Volkswnagen_beetle16.car.road라는 유일한 DNS 네임을 생성하고 각 IoT 디바이스(루프 디텍터, 신호제어기, 노변기지국(Road-Side Unit, RSU)[19], 차량)에 주소 중복 검사를 수행한다. 만약 중복된 DNS 네임이 생성되면 다른 고유 식별자로 DNS 네임을 생성한다. 다음으로 카메라(Camera1)는 NI Query 프로토콜을 이용하여 차량의 DNS과 IPv6 주소를 수집하여 DNS 서버에 등록한다.

그림 7에서 스마트 로드의 관리자는 클라이언트 장치를 통해서 Traffic Light2를 관리하는 것을 보여주고 있다. 이러한 시스템을 통해 기본 교통자료 수집 및 분석(예, 교통량, 차량속도, 도로 점유율), 교통 정보의 실시간 전송, 교통사고 경보 신호, 교통사고인식을 수행할 수 있으며 스마트로드 환경에서 각 IoT 디바이스들을 쉽게 관리할 수 있다.

4.2 DNS 네이밍 서비스 기반의 IoT 디바이스 관리

사용자는 IoT 디바이스의 상태를 확인할 수 있고, 각 IoT 디바이스를 제어 및 모니터링 할 수 있다. 그림 8(a)은 사용자는 AllSeen Alliance 프로토콜[20]을 이용하여 태블릿 PC를 이용해 댁내에 배치된 IoT 디바이스의 DNS 네임을 볼 수 있다. 보다 진보된 형태로 그림 8(b)는 태블릿 PC를 통해 댁내에 배치된 IoT 디바이스들의 종류 및 위치를 화면에 보여주고 있다. 이러한 서비스를 위해서는 IoT 디바이스의 위치정보를 파악하는 위치측정(Localization) 서비스를 통해 가능하다[21]. 본 논문의 DNS 네이밍 서비스는 위치측정 서비스와 결합하여 그림 8(b)와 같이 그래픽 기반의 IoT 디바이스 디스플레이를 통해 동일한 IoT 디바이스 모델(예, 에어콘)이라도 위치정보를 통해 사용자가 쉽게 구분하여 관리할 수 있게 할 수 있다. 또한 로봇청소기(Robot Cleaner)의 위치도 실시간적으로 파악하여 관리할 수 있다.

5. 구현 및 실험

본 논문에서는 제안한 기법의 검증을 위해 실제 게이트웨이 및 저 스펙의 사양을 가진 임베디드 IoT 디바이스 2개를 사용해서 구현 및 실험을 했다. 게이트웨이(Gateway)와 미터기(Meter)에 대한 상세 사양은 표 1과 같다.

그림 9는 게이트웨이의 환경설정을 나타낸 것으로 게이트웨이에서는 미터기의 주소 자동 설정 기능을 지원하기 위해 라우터 광고 데몬(Router Advertisement Daemon, radvd)을 실행하며, 광고주기 설정 값은 MinRtrAdvInterval은 600 sec, MaxRtrAdvInterval은 1800 sec으로 설정하였고 RDNSS 옵션은 2001:2c8::1로 설정, DNSSL 옵션은 secter1.grid로 설정했다.

그림 10은 게이트웨이가 주기적으로 RA 메시지를 미터기에게 전송한다. Meter1, Meter2는 DNSSL 옵션인 secter1.grid를 받고 유일한 DNS 네임을 생성한다.

그림 11은 두 개의 미터기들이 DNS 네임을 생성해서 주소중복검사를 통해 DNS 네임의 중복여부를 검사한다. 주소중복검사는 MD5 알고리즘을 이용했으며 각 미터기들이 멀티캐스트로 로컬 링크 안에 있는 주변 장

표 1 실험에 사용된 게이트웨이와 미터기의 사양

Table 1 Specification of Gateway and Meter used in the Experiment

	(a) Gateway	(b) Meter
OS	Linux	FreeRTOS
CPU	ARM Cortex-A5	ARM Cortex-M4
RAM	DDR2 256Mbyte	64k-byte embedded SRAM
Flash Memory	256Mbyte	256/512Kbytes

```
Interface wisun0
{
        AdvSendAdvert on;
        MinRtrAdvInterval 600;
        MaxRtrAdvInterval 1800;

        Prefix 2001:2c8:ee:100::/64
        {
                AdvOnLink on;
                AdvAutonomous on;
                AdvRouterAddr off;
        };
                                        RDNSS 2001:2c8::1
                                        {
                                                AdvRDNSSLifetime 3600;
                                        };
                                        DNSSL secter1.grid
                                        {
                                                AdvDNSSLifetime 3600;
                                        };
                                };
```

그림 9 라우터 광고 데몬 설정

Fig. 9 Configuration of Router Advertisement Daemon

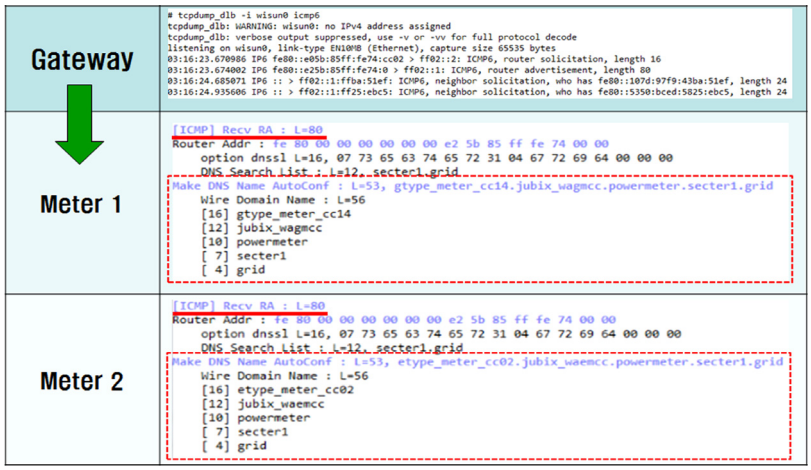


그림 10 라우터 광고

Fig. 10 Router Advertisement

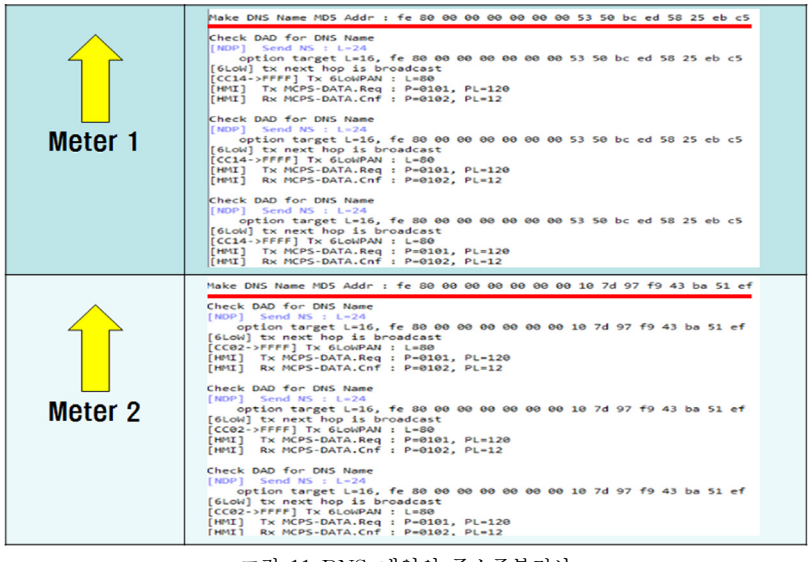


그림 11 DNS 네임의 주소중복검사

Fig. 11 Duplicate Address Detection for DNS Names

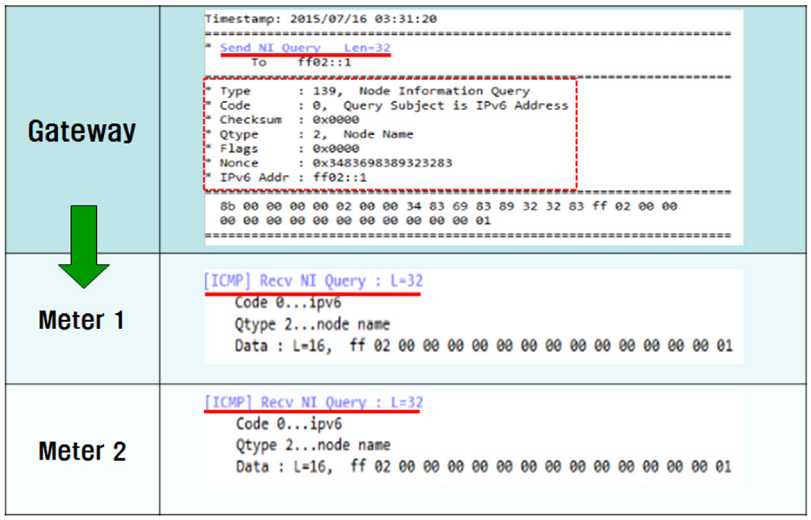


그림 12 DNS 네임 수집을 위한 Node Information 쿼리 메시지
Fig. 12 Node Information Query for DNS Name Collection

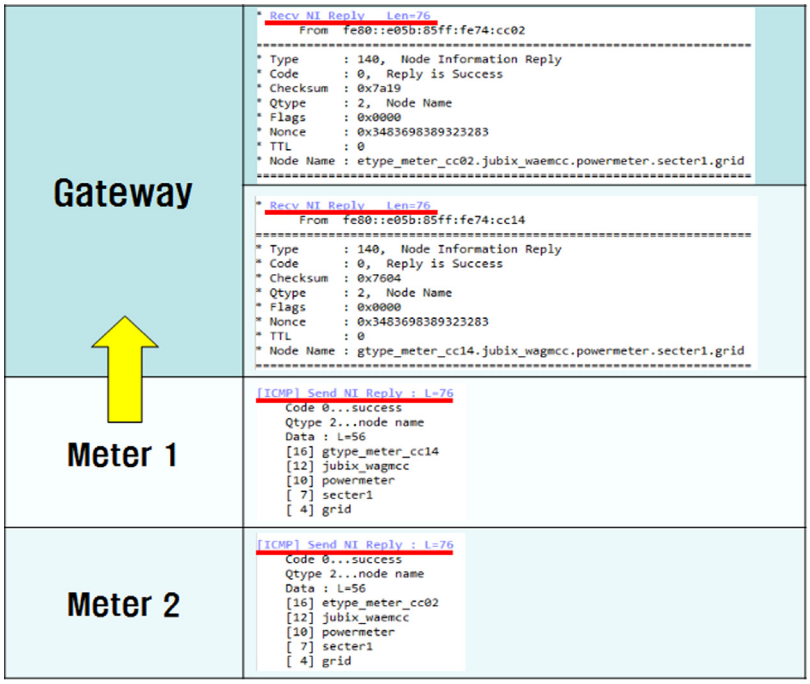


그림 13 DNS 네임 수집을 위한 Node Information Reply 메시지
Fig. 13 Node Information Reply for DNS Name Collection

치들에게 그 주소의 사용가능 여부를 알기 위해 Neighbor Solicitation(NS) 메시지를 보내서 중복된 주소가 있는 검사한다. 만약 미터기가 Neighbor Advertisement(NA) 메시지를 받으면 같은 DNS 네임을 사용하고 있다고 인식하고 다른 DNS 네임을 사용한다.

그림 12는 게이트웨이가 DNS 네임을 수집하기 위해서 NI Query를 미터기에게 메시지를 보낸다. ICMPv6의 NI Query는 타입 139로 정의되어 있다[8].

그림 13은 NI Query를 받은 미터기가 이에 대한 응답하기 위해 게이트웨이에게 NI Reply를 보낸다. ICMPv6

표 2 DNSNA와 mDNS의 비교
Table 2 Comparison of DNSNA and mDNS

Approaches	DNSNA	mDNS
Packet Forwarding	Unicasting	Multicasting
Authoritative DNS Server	Yes	No (host itself is server)
Naming Scope	Global, Local	Local
Target Networks	Small, Large	Small
Target Devices	IoT Devices	Apple Products
Code Size	Hundreds lines	Thousands lines
The Number of Messages for DNS Name Resolution	$O(2m)$	$O(\|E\|+m)$
Resource Usage (CPU, Memory, Energy)	Small	Big

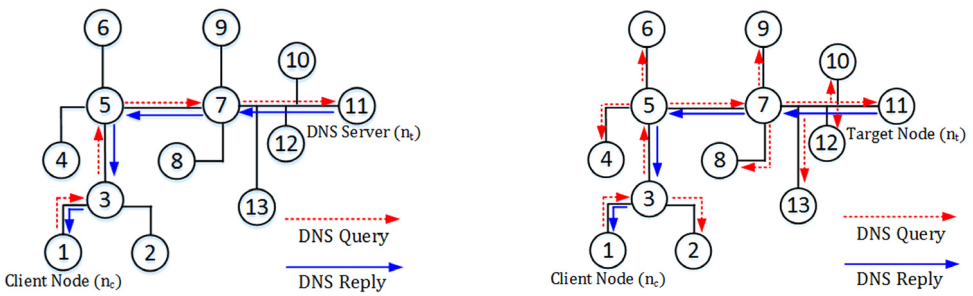
의 NI Reply는 타입 140로 정의 되어있다[8]. 게이트웨이는 각 미터기들의 DNS 네임을 수집한다.

끝으로 제안하는 기법인 DNSNA의 구현물과 mDNS의 비교는 표 2와 같다. DNSNA는 IoT 디바이스가 자동으로 DNS 네임을 생성 및 등록하는 과정에서 유니캐스트로 메시지를 전달한다. 따라서 DNS 관련 트래픽이 적게 발생하고, DNS 서버를 통해 IoT DNS 네임을 효율적으로 관리한다. 그러므로 DNSNA는 네트워크 규모에 제약을 받지 않고 작거나 큰 규모의 네트워크에 적용이 가능하다. 또한 실제 코드라인수는 mDNS 기법은 수천라인이지만 제안하는 기법 DNSNA는 비교적 간단하여 수백 라인이다. 따라서 DNSNA는 IoT 디바이스 DNS 네이밍을 위한 개발 오버헤드가 mDNS보다 훨씬 작다.

그림 14는 다중링크 네트워크 환경에서 제안하는 기법과 mDNS 기법에 대해 DNS 네임 레졸루션에 대해 발생하는 네트워크 트래픽을 비교 분석한다. 제안하는 기법의 경우 유니캐스트 기반으로 클라이언트 노드가 해당 노드의 DNS 네임 및 IPv6 주소를 얻으려면 해당 망을 관리하는 DNS 서버로부터 정보를 얻을 수 있다.

$$d(n_c,n_t)=m \quad (1)$$

(1)번 수식은 클라이언트 노드 n_c에서 DNS Server n_t까지의 홉거리(Hop Distance), 즉 패스 링크 개수이다. 그림 14에서 최대 홉거리는 m이라고 가정한다. 본 시나리오에서는 노드 n_c가 DNS 네임에 대해 레졸루션을 하고자 할 때, 노드 n_c는 DNS Server n_t까지 DNS Query를 전송하면, 그림 14(a)와 같이 m개의 DNS Query에 대한 패킷을 발생된다. 또한 DNS Server는 이에 상응하는 DNS Reply를 노드 n_c에게 전송하면, 그림 14(a)와 같이 m개의 DNS Reply에 대한 패킷을 발생된다. 따라서 Worst Case 발생 패킷 수는 $O(2m)$이다. 이와 반면에 mDNS가 DNS 네임 레졸루션에 이용될 때는 그림 14(b)와 같이 DNS Query는 네트워크 모두 링크로 멀티캐스트 되므로 발생하는 패킷 수는 네트워크 링크 개수인 $|E|$ 이다. 또한 DNS Query를 받은 Target Node n_t는 DNS Reply를 노드 n_c로 유니캐스트로 전송한다. 따라서 발생하는 패킷 수는 두 노드사이의 홉거리인 m이다. 따라서, Worst Case 발생 패킷 수는 $O(|E|+m)$이다. 사용되는 CPU, Memory, Energy는 DNS 패킷 수에 비례한다. 따라서 제안하는 DNSNA는 CPU, Memory 및 에너지소모 측면에서 mDNS보다 더 효과적임을 예측할 수 있다.



(a) DNS Name Resolution at DNSNA (b) DNS Name Resolution at mDNS

그림 14 다중링크 네트워크 환경에서 DNSNA기법과 mDNS기법의 트래픽 비교
Fig. 14 Comparison between DNSNA & mDNS for Traffic in a Multi-link Network

6. 결 론

본 논문에서는 IoT 디바이스의 DNS 네임을 자동으로 생성하고 관리하는 기법을 제안하였다. 제안 하는 기법은 IoT 디바이스의 증가하는 추세로 IPv4의 주소 고갈의 문제점에 대응하여 IPv6 주소 체계 기반의 DNS 네이밍 서비스에 적합하다. 본 논문은 DNS 네임을 생성 및 등록하는 과정에서는 IETF에서 재정된 프로토콜을 이용했다. 이때 유니캐스트로 메시지를 전달하여 트래픽이 적으며, DNS 서버를 관리함으로써 사용자는 DNS 서버로부터 IoT 디바이스의 DNS 네임 목록을 보고 제어 및 모니터링을 할 수 있다. 또한, 저 사양 IoT 디바이스에 적합하며 댁내, 스마트 그리드, 사무실, 쇼핑몰, 물류 창고와 같은 거대한 공간에서 사용될 수 있다.

향후연구로 제안하는 기법인 DNSNA와 mDNS를 시뮬레이션에서 구현해서 대규모 IoT 네트워크 환경에서의 네임 레졸루션을 위해 발생하는 트래픽을 정량적으로 비교 분석할 예정이다. 또한 DNSNA와 mDNS를 스마트 그리드 노드에서 운영할 때 소요되는 에너지양도 분석할 예정이다. 아울러 스마트폰, 태블릿 PC, 로봇청소기와 같이 이동성이 있는 IoT 디바이스의 DNS 네이밍 서비스를 위한 구현을 할 예정이다.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, Vol. 29, No. 7, pp. 1645-1660, Sep. 2013.

[2] Gartner's 2014 Hype Cycle [Online]. Available: http://www.gartner.com/newsroom/id/2819918

[3] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," *IETF RFC 2460*, Dec. 1998.

[4] P. Mockapetris, "Domain Names - Implementation and Specification," *IETF RFC 1035*, Nov. 1987.

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

[6] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," *IETF RFC 4861*, Sep. 2007.

[7] J. Jeong, S. Park, L. Beloeil, and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration," *IETF RFC 6106*, Nov. 2010.

[8] M. Crawford and B. Haberman, "IPv6 Node Information Queries," *IETF RFC 4620*, Aug. 2006.

[9] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)," *IETF RFC 2136*, Apr. 1997.

[10] A. Conta, S. Deering, and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," *IETF RFC 4443*, Mar. 2006.

[11] Apple Bonjour [Online]. Available: https://developer.apple.com/bonjour

[12] S. Cheshire and M. Krochmal, "DNS-Based Service Discovery," *IETF RFC 6763*, Feb. 2013.

[13] Jaehoon Jeong, Jungsoo Park, Hyounglun Kim and Kishik Park, "Name Service in IPv6 Mobile Ad-hoc Network," *ICOIN 2003*, Feb. 2003.

[14] C. Bormann, "6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)," *IETF RFC 7400*, Nov. 2014.

[15] Wi-SUN Allicance for Wireless and Smart Utility Networks (IEEE 802.15.4g) [Online]. Available: http://www.wi-sun.org

[16] P. Mockapetris, "Domain Names - Concepts and Facilities," *IETF RFC 1034*, Nov. 1987.

[17] R. Droms, "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," *IETF RFC 3646*, Dec. 2003.

[18] S. Thomson, T. Narten, and T. Jinmei, "IPv6 Stateless Address Autoconfiguration," *IETF RFC 4862*, Sep. 2007.

[19] A. Abdrabou and W. Zhuang, "Probabilistic Delay Control and Road Side Unit Placement for Vehicular Ad Hoc Networks with Disrupted Connectivity," *IEEE Journal on Selected Areas in Communications*, Vol. 29, No. 1, pp. 129-139, Jan. 2011.

[20] AllSeen Alliance. A Leading Internet of Things Initiative [Online]. Available: https://allseenalliance.org

[21] J. Jun, Y. Gu, L. Cheng, J. Sun, T. Zhu, and J. Niu, "Social-Loc: Improving Indoor Localization with Social Sensing," *ACM SenSys*, Nov. 2013.

이 세 준

2014년 목포해양대학교 정보통신공학(학사). 2014년~현재 성균관대학교 전자전기컴퓨터공학(석사과정). 관심분야는 사물인터넷, 차량 네트워크, 사이버물리시스템

정 재 훈

1999년 성균관대학교 정보공학(학사). 2001년 서울대학교 컴퓨터공학과(석사). 2009년 University of Minnesota 컴퓨터공학과(박사). 2001년~2004년 한국전자통신연구원 연구원. 2010년~2012년 Brocade Communications 소프트웨어 엔지니어. 2012년~현재 성균관대학교 인터랙션사이언스학과 조교수. 관심분야는 사이버물리시스템, 사물인터넷, 차량 네트워크