(주) 이디 리서치

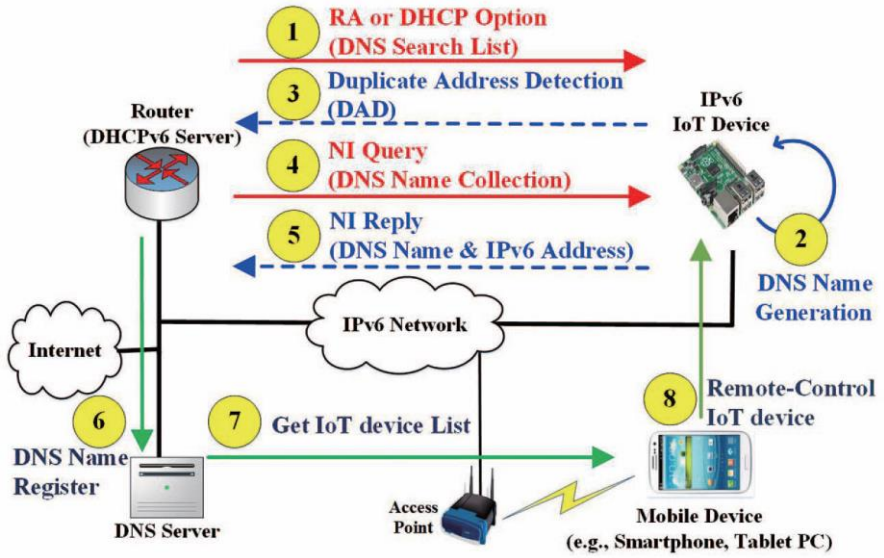
성균관대학교

# DNS 네임 자동설정 및 네이밍 서비스 기술

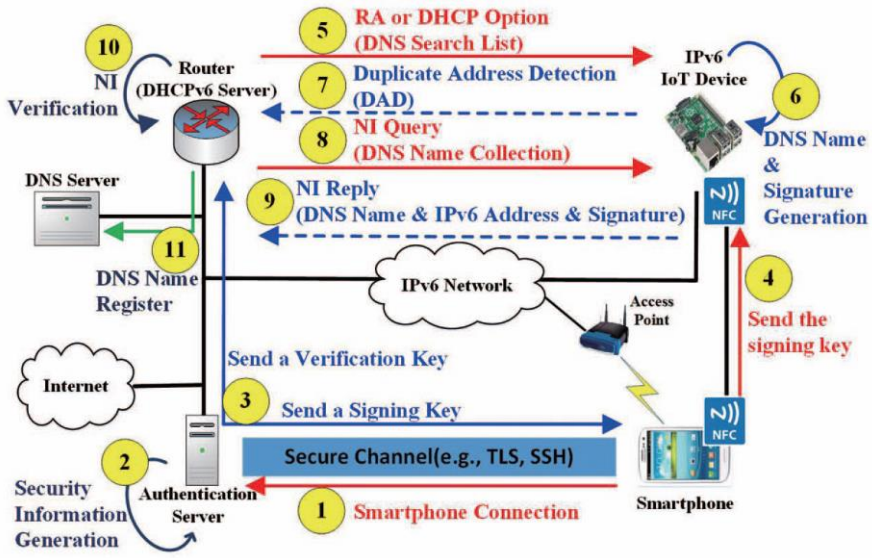
# DNS 네임 자동설정 및 네이밍 서비스 기술

## 기술 개요

- IoT 환경에서 수많은 IoT 디바이스의 관리를 위한 IoT 디바이스의 DNS 네임 자동설정 및 네이밍 서비스 기술
- 사용자는 IoT 디바이스를 DNS 네임으로 스마트폰과 태블릿과 같은 모바일 스마트 디바이스로 IoT 디바이스를 원격에서 모니터링 및 원격 제어할 수 있음




<DNSNA>




<SDNSNA>

# DNS 네임 자동설정 및 네이밍 서비스 기술

## 세부 기술 내용

- **DNSNA**
  - **IoT 디바이스를 위한 DNS 네이밍 서비스 표준 제정**
  - IoT 디바이스 관리의 효율증대를 위해 IoT 디바이스의 DNS 네임 자동 생성 및 자동 등록 서비스 제공
  - **IoT 디바이스를 위한 DNS 네임포맷 표준 제정**
  - IoT 디바이스 관리의 효율증대를 위해 IoT 디바이스에서 사용할 DNS 네임 포멧 제정
  - **IETF 표준을 이용한 서비스 구현 및 표준 제정**
  - 기존에 정의되어 있는 IPv6 라우터 광고 메시지(Router Advertisement, RA)와 노드 정보 쿼리(Node Information Query) 등의 표준 프로토콜을 이용한 최소화의 소프트웨어 확장을 통한 IoT DNS 네이밍 서비스로 프로토콜 구현
- **SDNSNA**
  - **IoT 디바이스를 위한 DNS 네이밍 서비스에 대한 보안 기술 제공**
  - DNSNA에 IoT 디바이스 인증 절차를 추가한 안전한 DNS 네이밍 서비스를 제공하는 기술
  - **NFC를 이용한 보안**
  - IoT 디바이스를 인증 서버에 접속된 관리자 스마트폰의 NFC(Near-Field Communication) 통신을 기반으로 한 물리적 IoT 디바이스 인증방식을 통한 불법 IoT 디바이스의 네트워크 접근 제어를 제공

# DNS 네임 자동설정 및 네이밍 서비스 기술

## 기술의 특징

- 기존 IoT 디바이스 간의 DNS 네이밍 서비스에 대한 표준이 없어 IoT 디바이스 관리에 어려움이 존재함
  - IoT 디바이스 간의 DNS 네이밍 서비스에 대한 표준을 제공하여 IoT 관리의 효율성 증가
- 기존 IoT 디바이스 간의 안전한 DNS 네이밍 서비스에 대한 표준이 없고, 이에 대한 보안 기능들이 없음
  - IoT 디바이스 간의 안전한 DNS 네이밍 서비스에 대한 표준을 제공하여 네트워크 보안 관리의 효율성 증가

## 관련 특허

| No. | 특허번호 | 발명의 명칭 |
|---|---|---|
| 1 | 10-1589413 | IoT 디바이스에 대한 DNS 네이밍 방법 |
| 2 | 10-1776882 | IoT 디바이스에 대한 보안성 있는 DNS 네이밍 방법 및 보안성 있는 DNS 네이밍 등록을 수행하는 라우터 장치 |

## 활용분야

- 사물인터넷(IoT)을 만드는 제조사
- 사물인터넷(IoT)을 활용하는 회사 및 서비스 제공자
- 사물인터넷(IoT)을 제공하는 IoT 환경

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년01월28일
(11) 등록번호 10-1589413
(24) 등록일자 2016년01월21일

(51) 국제특허분류(Int. Cl.)
*H04L 29/12* (2006.01)
(52) CPC특허분류
*H04L 61/3025* (2013.01)
*H04L 61/1511* (2013.01)
(21) 출원번호 10-2015-0101662
(22) 출원일자 2015년07월17일
심사청구일자 2015년07월17일
(56) 선행기술조사문헌
KR1020040097849 A*
KR1020130117436 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
**성균관대학교산학협력단**
경기도 수원시 장안구 서부로 2066, 성균관대학교 내 (천천동)
(72) 발명자
**정재훈**
경기도 수원시 장안구 화산로187번길 19, 1603호 (천천동, 삼성래미안 109동)
**이세준**
경기도 수원시 권선구 여기산로 42, 101동 401호 (서둔동, 성일아파트
(74) 대리인
**남정길**

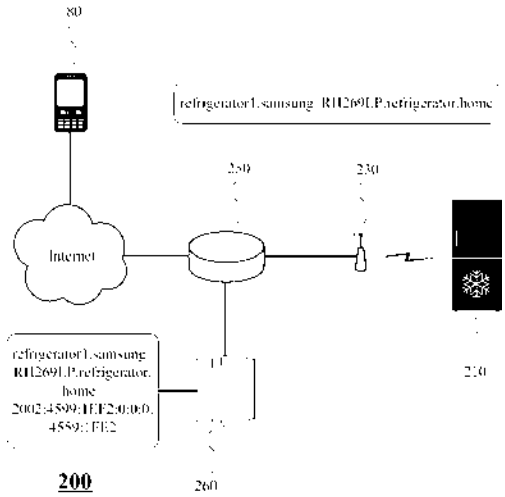
전체 청구항 수 : 총 18 항 심사관 : 박보미

(54) 발명의 명칭 **IoT 디바이스에 대한 DNS 네이밍 방법**

## (57) 요 약

IoT 디바이스에 대한 DNS 네이밍 방법은 IPv6 프로토콜에 따라 네트워크에 연결된 디바이스가 DNS 검색 리스트가 포함된 제1 메시지를 수신하는 단계, 상기 디바이스가 상기 디바이스의 모델 정보 및 식별자가 포함된 도메인 이름을 생성하는 단계, 상기 디바이스가 ND(neighbor discovery) 프로토콜에 따라 상기 도메인 이름에 대한 중복 검사를 수행하는 단계 및 상기 도메인 이름이 중복되지 않는 경우 상기 네트워크에 연결된 도메인 이름 수집 장치가 NI(Node Information) 프로토콜에 따라 상기 도메인 이름과 상기 디바이스에 대한 IPv6 주소를 DNS 서버에 등록하는 단계를 포함한다.

## 대 표 도



(52) CPC특허분류
*H04L 61/3005* (2013.01)
*H04L 61/6059* (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 R0101-14-241
부처명 정부)미래창조과학부
연구관리전문기관 정보통신기술진흥센터-정보통신·방송 연구개발사업
연구사업명 SW컴퓨팅산업원천기술개발사업(SW)
연구과제명 스마트TV 2.0 소프트웨어 플랫폼
기 여 율 1/2
주관기관 성균관대학교 산학협력단
연구기간 2014.12.01 ~ 2015.08.31
이 발명을 지원한 국가연구개발사업
과제고유번호 S-2015-0012-016
부처명 정부)교육부
연구관리전문기관 한국연구재단-LINC사업_기술개발과제
연구사업명 산학협력선도대학(LINC)육성사업 4/5
연구과제명 IoT 디바이스 관리를 위한 DNS 네이밍 시스템
기 여 율 1/2
주관기관 성균관대학교 산학협력단
연구기간 2015.05.22 ~ 2016.01.31
공지예외적용 : 있음

# 명 세 서

## 청구범위

### 청구항 1

IPv6 프로토콜에 따라 네트워크에 연결된 디바이스가 DNS 검색 리스트가 포함된 제1 메시지를 수신하는 단계;

상기 디바이스가 상기 디바이스의 모델 정보, 동일한 도메인 서픽스에서 동일한 모델의 디바이스를 구별하는 식별자 및 상기 디바이스의 위치 정보를 포함하는 도메인 이름을 생성하는 단계;

상기 디바이스가 ND(neighbor discovery) 프로토콜에 따라 상기 도메인 이름에 대한 중복 검사를 수행하는 단계; 및

상기 도메인 이름이 중복되지 않는 경우 상기 네트워크에 연결된 도메인 이름 수집장치가 NI(Node Information) 프로토콜에 따라 상기 도메인 이름과 상기 디바이스에 대한 IPv6 주소를 DNS 서버에 등록하는 단계를 포함하는 IoT 디바이스에 대한 DNS 네이밍 방법.

### 청구항 2

제1항에 있어서,

상기 수신하는 단계에서 상기 디바이스는 IPv6 표준에 따른 RA(Router Advertisement) 옵션 또는 DHCP(Dynamic Host Configuration Protocol) 옵션을 통해 상기 제1 메시지를 수신하는 IoT 디바이스에 대한 DNS 네이밍 방법.

### 청구항 3

제2항에 있어서,

상기 도메인 이름 수집장치 또는 상기 네트워크에 연결된 라우터가 상기 RA 옵션 또는 DHCP 옵션에 따라 상기 제1 메시지를 상기 디바이스에 전송하는 IoT 디바이스에 대한 DNS 네이밍 방법.

### 청구항 4

제1항에 있어서,

상기 도메인 이름은

모델 이름, 상기 디바이스의 종류를 나타내는 모델 카테고리, 동일한 모델 이름을 갖는 디바이스 마다 서로 다른 디바이스 식별자 및 상기 디바이스의 네트워크 도메인 중 적어도 하나를 포함하는 IoT 디바이스에 대한 DNS 네이밍 방법.

### 청구항 5

제1항에 있어서,

상기 모델 정보는 상기 디바이스의 모델 이름, 상기 디바이스의 생산자 또는 판매자 및 상기 디바이스의 종류를 나타내는 모델 카테고리 중 적어도 하나를 포함하는 IoT 디바이스에 대한 DNS 네이밍 방법

### 청구항 6

제1항에 있어서,

상기 DNS 검색 리스트는 DNS 도메인 서픽스(suffix)를 포함하고,

상기 도메인 이름은 모델이름, 상기 디바이스의 종류를 나타내는 모델 카테고리, 동일한 모델 이름을 갖는 디바이스 마다 서로 다른 디바이스 식별자 및 상기 DNS 도메인 서픽스 중 적어도 하나를 포함하는 IoT 디바이스에 대한 DNS 네이밍 방법.

**청구항 7**

제1항에 있어서,

상기 도메인 이름이 중복된 경우 상기 디바이스는 상기 식별자를 변경한 새로운 식별자를 포함하는 새로운 도메인 이름을 생성하고, 상기 새로운 도메인 이름에 대한 중복 검사를 수행하는 단계를 더 포함하는 IoT 디바이스에 대한 DNS 네이밍 방법.

**청구항 8**

삭제

**청구항 9**

제1항에 있어서,

상기 위치 정보는 상기 디바이스가 위치하는 지점에 대한 지리적 좌표 정보인 IoT 디바이스에 대한 DNS 네이밍 방법.

**청구항 10**

제1항에 있어서,

상기 위치 정보는 상기 디바이스의 주변에 위치한 다른 디바이스와의 상대적인 위치 정보인 IoT 디바이스에 대한 DNS 네이밍 방법.

**청구항 11**

제1항에 있어서,

상기 위치 정보는 기준 지점에서 이동하는 위치를 측정하는 센서 장치를 이용하여 측정되는 상대적인 위치 정보인 IoT 디바이스에 대한 DNS 네이밍 방법.

**청구항 12**

제11항에 있어서,

상기 디바이스는 상기 센서 장치로부터 측정된 상대적인 위치 정보를 수신하는 IoT 디바이스에 대한 DNS 네이밍 방법.

**청구항 13**

제1항에 있어서,

상기 위치 정보는 상기 네트워크에 연결된 사용자 단말을 통해 상기 디바이스에 입력되는 IoT 디바이스에 대한 DNS 네이밍 방법.

**청구항 14**

제1항에 있어서,

상기 등록하는 단계는

상기 도메인 이름 수집장치가 NI(Node Information) 쿼리를 상기 디바이스에 전송하는 단계;

상기 디바이스가 상기 도메인 이름을 상기 도메인 이름 수집장치에 전송하는 단계; 및

상기 도메인 이름 수집 장치가 수신한 상기 도메인 이름을 상기 DNS 서버에 전송하는 단계를 포함하는 IoT 디바이스에 대한 DNS 네이밍 방법.

**청구항 15**

제1항에 있어서,

상기 네트워크에 연결된 클라이언트 장치가 상기 DNS 서버에서 상기 디바이스에 대한 도메인 이름을 획득하고, 상기 도메인 이름을 이용하여 상기 디바이스를 모니터링하거나, 제어하는 단계를 더 포함하는 IoT 디바이스에 대한 DNS 네이밍 방법.

**청구항 16**

IPv6 프로토콜에 따라 네트워크에 연결된 라우터가 RA(Router Advertisement) 옵션 또는 DHCP(Dynamic Host Configuration Protocol) 옵션을 통해 DNS 검색 리스트를 IoT 디바이스에 전송하는 단계;

상기 IoT 디바이스가 자신의 모델 이름, 동일한 도메인 서픽스에서 동일한 모델의 IoT 디바이스를 구별하는 식별자, 상기 IoT 디바이스의 위치 정보 및 상기 DNS 검색 리스트에 포함된 도메인 서픽스(suffix)를 포함하는 도메인 이름을 생성하는 단계; 및

상기 IoT 디바이스를 관리하는 도메인 이름 수집장치가 NI(Node Information) 프로토콜에 따라 상기 도메인 이름과 상기 디바이스에 대한 IPv6 주소를 DNS 서버에 등록하는 단계를 포함하는 IoT 디바이스에 대한 DNS 네이밍 방법.

**청구항 17**

제16항에 있어서,

상기 도메인 이름은 상기 IoT 디바이스의 생산자, 상기 IoT 디바이스의 판매자 및 상기 IoT 디바이스의 종류를 나타내는 모델 카테고리 중 적어도 하나를 포함하는 IoT 디바이스에 대한 DNS 네이밍 방법.

**청구항 18**

삭제

**청구항 19**

제1항에 있어서,

상기 디바이스는 새로운 도메인 서픽스를 갖는 영역으로 이동하는 경우 상기 새로운 도메인 서픽스를 갖는 도메인 이름을 생성하는 IoT 디바이스에 대한 DNS 네이밍 방법.

**청구항 20**

제16항에 있어서,

상기 IoT 디바이스가 새로운 도메인 서픽스를 갖는 영역으로 이동하는 경우 상기 새로운 도메인 서픽스를 갖는 도메인 이름을 생성하는 IoT 디바이스에 대한 DNS 네이밍 방법.

## *발명의 설명*

### *기 술 분 야*

[0001] 이하 설명하는 기술은 IPv6에 기반한 자동 DNS 네이밍 기법에 관한 것이다.

### *배 경 기 술*

[0002] IoT(Internet of Things)는 최근 가장 주목받고 있는 인터넷 관련 기술이다. IoT를 이용한 서비스는 기본적으로 인터넷에 연결된 매우 많은 디바이스를 통해 일정한 서비스를 제공할 수 있다.

[0003] 인터넷에 연결되는 장치는 개별적인 인터넷 주소(IP address)를 갖고 있어야 한다. IoT 환경에서 종래 IPv4는 인터넷 주소 고갈이라는 한계를 안고 있었다. 이를 극복하기 위한 기술로 IPv6가 등장하였다. IETF(Internet Engineering Task Force)는 IPv6에 관련된 표준을 제정하고 있다.

[0004] 한편 DNS(Domain Name System)는 인터넷에 연결된 개체에 대한 IP 주소와 도메인 이름을 관리한다. 종래 인터넷 환경에서는 도메인 이름을 사용자가 수동으로 설정하였다.

### *선행기술문헌*

### 특허문헌

[0005] (특허문헌 0001) 한국공개특허 제10-2014-0099194호

(특허문헌 0002) 한국공개특허 제10-2011-0055392호

### 비특허문헌

[0006] (비특허문헌 0001) S. Thomson, T. Narten and T. Jinmei, "Pv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

(비특허문헌 0002) J. Jeong, S. Park, L. Beloeil and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.

## 발명의 내용

### 해결하려는 과제

[0007] IoT 환경에서 IoT 디바이스는 매우 많은 개수가 사용된다. 따라서 각 IoT 디바이스에 대한 도메인 이름을 일일이 설정하는 것은 매우 번거로운 일이 된다. mDNS(multicast Domain Name System)과 같이 도메인 이름을 자동으로 부여하기 위한 기술이 있다. 대표적으로 애플(apple)이 개발한 Bonjour와 같은 프로토콜이 있다. 하지만 mDNS는 기본적으로 멀티캐스트 기반으로 네트워크에 많은 트래픽을 발생시키고, 인터넷에 연결된 디바이스가 항상 동작하는(깨어 있는) 경우에만 네이밍 요청 메시지에 응답할 수 있다.

[0008] 이하 설명하는 기술은 IPv6의 이웃탐색 프로토콜(Neighbor Discovery, ND)을 기반하여 도메인 정보를 담은 DNS Search List(DNSSL) 옵션을 멀티캐스트(multicast)로 전송하면서, IoT 디바이스가 자동으로 DNS 이름을 생성하는 기법을 제공하고자 한다.

### 과제의 해결 수단

[0009] IoT 디바이스에 대한 DNS 네이밍 방법은 IPv6 프로토콜에 따라 네트워크에 연결된 디바이스가 DNS 검색 리스트가 포함된 제1 메시지를 수신하는 단계, 상기 디바이스가 상기 디바이스의 모델 정보 및 식별자가 포함된 도메인 이름을 생성하는 단계, 상기 디바이스가 ND(neighbor discovery) 프로토콜에 따라 상기 도메인 이름에 대한 중복 검사를 수행하는 단계 및 상기 도메인 이름이 중복되지 않는 경우 상기 네트워크에 연결된 도메인 이름 수집장치가 NI(Node Information) 프로토콜에 따라 상기 도메인 이름과 상기 디바이스에 대한 IPv6 주소를 DNS 서버에 등록하는 단계를 포함한다.

[0010] 다른 측면에서 IoT 디바이스에 대한 DNS 네이밍 방법은 IPv6 프로토콜에 따라 네트워크에 연결된 라우터가 RA(Router Advertisement) 옵션 또는 DHCP(Dynamic Host Configuration Protocol) 옵션을 통해 DNS 검색 리스트를 IoT 디바이스에 전송하는 단계, 상기 IoT 디바이스가 자신의 모델 정보, 식별자 및 상기 DNS 검색 리스트에 포함된 도메인 서픽스(suffix)를 포함하는 도메인 이름을 생성하는 단계 및

[0011] 상기 IoT 디바이스를 관리하는 도메인 이름 수집장치가 NI(Node Information) 프로토콜에 따라 상기 도메인 이름과 상기 디바이스에 대한 IPv6 주소를 DNS 서버에 등록하는 단계를 포함한다.

### 발명의 효과

[0012] 이하 설명하는 기술은 DNS 이름을 생성하는 과정에서 유니캐스트로 메시지를 전달하여 트래픽이 적다. 또한 이하 설명하는 기술은 클라이언트 장치가 DNS 서버를 통해 IoT 디바이스에 대한 이름 정보를 참조하므로 IoT 디바이스가 슬립모드(sleep mode)에 있을 때도 해당 IoT 디바이스에 대한 도메인 이름을 확인할 수 있다. 나아가 이하 설명하는 기술은 도메인 이름이 디바이스의 모델 정보 및/또는 위치 정보를 포함하여 도메인 이름만으로도 특정 디바이스에 대한 정보를 제공할 수 있다.

### 도면의 간단한 설명

[0013] 도 1은 종래 DNS의 구성을 도시한 블록도의 예이다.

도 2는 DNSNA의 구성을 도시한 블록도의 예이다.

도 3은 DNSNA에서 IoT 디바이스가 도메인 이름을 생성하는 과정에 대한 예이다.

도 4는 DNSNA에서 IoT 디바이스에 대한 도메인 이름을 등록하는 과정에 대한 예이다.

도 5는 DNSNA에서 클라이언트 장치가 IoT 디바이스를 제어하는 과정에 대한 예이다.

도 6은 댁내에서 사용하는 IoT 디바이스에 대한 DNS 이름에 대한 예이다.

도 7은 댁내에서 사용하는 IoT 디바이스에 대한 DNS 이름의 다른 예이다.

도 8은 스마트폰이 댁내에 위치한 IoT 디바이스에 대한 정보를 표시하는 예이다.

도 9는 도로에서 이동하는 차량에 대한 DNS 이름에 대한 예이다.

도 10은 교차로에서의 차량 및 고정 장치에 대한 DNS 이름에 대한 예이다.

## *발명을 실시하기 위한 구체적인 내용*

[0014] 이하 설명하는 기술은 다양한 변경을 가할 수 있고 여러 가지 실시 예를 가질 수 있는 바, 특정 실시 예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 이하 설명하는 기술을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 이하 설명하는 기술의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0015] 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 해당 구성요소들은 상기 용어들에 의해 한정되지는 않으며, 단지 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 이하 설명하는 기술의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

[0016] 본 명세서에서 사용되는 용어에서 단수의 표현은 문맥상 명백하게 다르게 해석되지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함한다" 등의 용어는 설시된 특징, 개수, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 의미하는 것이지, 하나 또는 그 이상의 다른 특징들이나 개수, 단계 동작 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 배제하지 않는 것으로 이해되어야 한다.

[0017] 도면에 대한 상세한 설명을 하기에 앞서, 본 명세서에서의 구성부들에 대한 구분은 각 구성부가 담당하는 주기능 별로 구분한 것에 불과함을 명확히 하고자 한다. 즉, 이하에서 설명할 2개 이상의 구성부가 하나의 구성부로 합쳐지거나 또는 하나의 구성부가 보다 세분화된 기능별로 2개 이상으로 분화되어 구비될 수도 있다. 그리고 이하에서 설명할 구성부 각각은 자신이 담당하는 주기능 이외에도 다른 구성부가 담당하는 기능 중 일부 또는 전부의 기능을 추가적으로 수행할 수도 있으며, 구성부 각각이 담당하는 주기능 중 일부 기능이 다른 구성부에 의해 전담되어 수행될 수도 있음은 물론이다.

[0018] 또, 방법 또는 동작 방법을 수행함에 있어서, 상기 방법을 이루는 각 과정들은 문맥상 명백하게 특정 순서를 기재하지 않은 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 과정들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.

[0020] 이하 설명하는 기술은 DNS 이름을 자동으로 생성하고 관리하는 기법에 관한 것이다. 이하 설명하는 기술은 IPv6에 기반을 둔 IP 주소를 사용한다. 이하 설명하는 기술은 DNS 이름을 생성하고 등록하는 과정에서 IPv6 표준에서 재정된 프로토콜을 이용하고자 한다. IPv6 표준에 정의된 프로토콜은 이미 공개된 것이므로 이하 자세한 설명은 생략한다.

[0022] 도 1은 종래 DNS(100)의 구성을 도시한 블록도의 예이다. 종래 DNS(Domain Name System)은 DNS 서버(160)가 핵심적인 구성에 해당한다. 도 1에서는 "www.sample.com"이라는 서비스를 제공하는 서비스 서버(110)를 예로 도시하였다. 서비스 제공자가 서비스 서버(110)를 이용하여 인터넷으로 다른 사용자에게 특정한 서비스를 제공하기 위해서는 서비스 서버(110)에 대한 도메인 이름을 등록해야 한다. 서비스 제공자는 컴퓨터 장치(50)를 통해 DNS 서버(160)에 접근하고, 자신의 서비스 서버(110)에 대한 도메인 이름을 등록한다. DNS 서버(160)는 널리 알려진 바와 같이 도메인 이름과 도메인 이름이 나타내는 실제 IP 주소를 저장하고 관리한다. 도 1에서 DNS 서버(160)

는 서비스 서버(110)에 대한 도메인 이름(www.sample.com) 및 서비스 서버(110)에 대한 IP 주소(214.11.00.124)를 저장한다. 도 1은 IPv4를 사용한 예를 도시하였다. 라우터(150)는 DNS 서버(160) 및 서비스 서버(110)에 대한 데이터 전달을 제어하는 구성이다.

[0023] 이후 서비스를 제공받기 위한 사용자가 클라이언트 장치를 사용하여 도메인 이름으로 서비스 서버에 접속을 시도한다고 가정한다. 클라이언트 장치가 전달한 접속 명령은 DNS 서버(160)에서 도메인 이름에 해당하는 IP 주소를 찾아내고, 해당 IP 주소를 갖는 서비스 서버(110)에 클라이언트 장치가 접속되게 한다.

[0025] 그러나 IoT 환경에서는 수많은 IoT 디바이스에 대한 도메인 이름을 일일이 설정하기가 어려울 수 있다. 따라서 이하 설명하는 기술에서 IoT 디바이스(예, 센서, 가전제품, 신호등, 자동차 등)는 IPv6 네트워크에 참여할 때 자신의 디바이스 정보를 기반으로 DNS 이름을 생성한다. 이렇게 생성된 유일한 DNS 이름은 IoT 디바이스를 위한 DNS 서버에 등록된다. 즉, DNS는 관리자 또는 사용자의 개입 없이 IoT 디바이스에 대한 DNS 이름을 자동으로 생성하여 등록한다. 이후 사용자는 클라이언트 장치(PC, 스마트폰, 태블릿 PC 등)를 이용해서 DNS 서버를 통해 IoT 디바이스에 대한 도메인 이름을 획득하여 IoT 디바이스를 원격 제어할 수 있다. 종래 DNS와는 달리 이름을 자동으로 설정한다는 의미에서 도메인 이름을 설정하고 등록하는 시스템을 이하 DNSNA(DNS Name Autoconfiguration)라고 명명한다.

[0026] DNSNA은 IoT 디바이스를 관리하기 위해 DNS 서버(DNS Server)를 이용하여 IoT 디바이스의 유일한 이름을 생성, 등록, 관리한다. IoT 디바이스가 IPv6 네트워크에 들어올 때 IoT 디바이스는 자신의 디바이스 정보를 기반으로 현재 IPv6 네트워크에 유일한 DNS 이름을 생성한다. 생성된 DNS 이름은 라우터(Router)에 등록되고 IoT 디바이스의 정보는 동적으로 DNS 서버에 등록한다. 사용자는 클라이언트 장치를 이용해 IoT 디바이스 정보를 DNS 서버로부터 수신하여 IoT 디바이스를 원격제어를 할 수 있다.

[0028] 도 2는 DNSNA(200)의 구성을 도시한 블록도의 예이다. 도 2는 IoT 디바이스(210)로 댁내에서 사용하는 냉장고를 예로 도시하였다. IoT 디바이스는 일반적으로 무선 통신을 수행하므로, 도 2는 IoT 디바이스(210)가 댁내에 있는 AP(230)를 통해 네트워크에 연결된 형태를 도시하였다. 라우터(250)는 도 1과 같이 DNS 서버(260) 및 IoT 디바이스(210)에 전달되는 데이터 패킷을 제어한다.

[0029] IoT 디바이스(210)는 네트워크에 최초 접속하는 경우 또는 일정한 요청이 있는 경우 자신이 직접 DNS 이름을 생성한다. 도 2에서 IoT 디바이스(210)는 "refrigerator1.samsung_RH269LP.refrigerator.home"이라는 DNS 이름을 생성하였다. IoT 디바이스가 DNS 이름을 생성하는 과정 및 DNS 이름이 의미하는 내용은 후술한다. IoT 디바이스(210)가 생성한 DNS 이름은 DNS 서버(260)에 저장된다. DNS 서버(260)는 IoT 디바이스(210)가 전달한 DNS 이름과 해당 DNS 이름이 의미하는 IP 주소를 저장한다. 도 2에서 IP 주소는 IPv6에 따른 주소이다. 도 2에 도시한 바와 같이 IPv6에 따른 IP 주소는 콜론으로 구분된 모두 8개의 세그먼트를 갖고, 각 세그먼트는 16bits로 표현된다. IPv6는 128bits를 사용하여 IP 주소를 표현한다.

[0030] 사용자는 클라이언트 장치(80)를 통해 DNS 서버(260)로부터 IoT 디바이스(210)에 대한 DNS 이름을 획득하고, IoT 디바이스(210)에 접근하여 냉장고의 상태를 확인할 수 있다. 또한 사용자는 IoT 디바이스(210)에 접근하여 냉장고를 제어할 수도 있다.

[0032] 도 3은 DNSNA(200)에서 IoT 디바이스가 도메인 이름을 생성하는 과정에 대한 예이다.

[0033] IoT 디바이스(210)는 라우터(250)를 통해 DNSSL(DNS Search list, DNS 검색 리스트) 옵션(option)을 수신한다(①). DNSSL은 IoT 디바이스가 도메인 이름을 등록하는 과정에서 사용하는 DNS 도메인 서픽스(suffix)의 리스트이다. DNSSL에는 복수의 DNS 서픽스가 있을 수 있다. DNSSL 옵션은 IPv6 표준에 따른 RA(Router Advertisement) 옵션 또는 DHCP(Dynamic Host Configuration Protocol) 옵션을 통해 전달된다. IPv6 표준에 따른 RA 옵션은 DNSSL 정보를 IPv6 호스트(IoT 디바이스)에 전달한다. 또한 DHCP 옵션도 DNSSL 정보를 IPv6 호스트(IoT 디바이스)에 전달한다.

[0034] IPv6 호스트인 IoT 디바이스(210)는 수신한 DNSSL 옵션이 유효한지 점검할 수 있다. DNSSL 옵션이 유효하다면 IoT 디바이스(210)는 자신의 DNS 이름을 생성한다(②). IoT 디바이스(210)는 생산 과정에서 부여되는 모델 이름 등과 같은 모델 정보를 보유한다. IoT 디바이스(210)는 모델 정보를 이용하여 DNS 이름을 생성할 수 있다. IoT 디바이스(210)는 DNS 도메인 서픽스에 자신의 디바이스 정보(예, Device Category, Vendor Name, Model Name)을 결합해서 자동으로 설정할 수 있다. 나아가 IoT 디바이스(210)는 DNS 이름은 디바이스 정보에 고유 식별자(unique_id)를 결합할 수 있다. 도 3에서 DNS 이름 중 "refrigerator1"은 동일한 모델의 디바이스를 구별하기 위한 고유 식별자이고, "samsung_RH269LP"는 제조사이름을 포함하는 모델 이름이고, "refrigerator"는 모델 카

테고리이고, "home"은 DNS 도메인 서픽스에 해당한다.

[0035] IoT 디바이스(210)는 생성한 DNS 이름이 도메인 이름(subnet)에서 유일한 DNS 이름인지 여부를 확인하는 것이 바람직하다. IoT 디바이스(210)는 ND(neighbor discovery) 프로토콜에 따라 도메인 이름에 대한 중복 검사를 수행할 수 있다(③). IoT 디바이스(210)는 IPv6 멀티캐스트 주소(Multicast Address)로 DAD(Duplicate Address Detection)을 수행할 수 있다.

[0036] DNS 이름의 IPv6 멀티캐스트 주소는 네트워크 프리픽스(Prefix)에 해당하는 상위 64 비트를 위해 Link-Local Multicast Address Prefix인 ff02::/16를 이용하고, 네트워크 인터페이스(Network Interface) ID에 해당하는 하위 64 비트를 위해 DNS 이름에 대한 해싱값인 128 비트 중에서 MSB(Most Significant Bit)를 포함하는 상위 64비트를 취한다. IoT 디바이스(210)는 ND(neighbor discovery) 프로토콜에 따라 DNS 이름에 해당하는 해싱값을 이용하여 DAD 과정을 수행한다.

[0037] 도메인 이름이 중복된 경우 IoT 디바이스(210)는 종전의 도메인 이름에서 디바이스의 고유 식별자를 새로운 식별자로 변경한 새로운 도메인 이름을 생성하고, 새로운 도메인 이름에 대한 중복 검사를 수행한다. IoT 디바이스(210)가 도메인 이름에서 중복되지 않은 도메인 이름을 생성했다고 확인해야 도메인 이름 생성과정이 종료된다.

[0038] 만약 DNSSL에 복수의 DNS 도메인 서픽스가 있는 경우 IoT 디바이스(210)는 모든 DNS 도메인 서픽스에 대해 각각 도메인 이름 생성 과정을 반복한다.

[0040] 도 4는 DNSNA(200)에서 IoT 디바이스에 대한 도메인 이름을 등록하는 과정에 대한 예이다. 도 3에서 IoT 디바이스(210)는 자신의 DNS 이름을 생성하였고, 도메인 이름에서 중복되지 않는다는 것을 확인한 상태이다.

[0041] 네트워크에 연결된 도메인 이름 수집 장치가 IoT 디바이스(210)에 도메인 이름을 묻고(query), 응답으로 수신한 IoT 디바이스(210)의 도메인 이름은 DNS 서버(260)에 저장하게 된다. 도메인 이름 수집 장치는 네트워크 구성 중 IoT 디바이스(210)의 도메인 이름에 대한 등록 과정을 제어한다. 일반적으로 도메인 이름 수집 장치는 IoT 디바이스(210)의 도메인 이름을 관리하는 장치일 수 있다. 도 4에서는 라우터(250)가 IoT 디바이스(210)에 대한 도메인 이름을 수집한다.

[0042] 라우터(250)은 먼저 IoT 디바이스(210)에 대해 DNS 이름을 요청한다. 라우터(250)는 NI(Node Information) 프로토콜을 이용하여 IoT 디바이스(210)에 DNS 이름을 묻는다(NI query,①). 이 과정은 종래 NI 프로토콜에서 새로운 NI query가 필요하다. 즉, NI 타입에 대한 새로운 코드가 정의되어야 한다. IoT 디바이스(210)는 NI query에 대한 응답으로 자신의 DNS 이름을 전달한다(NI reply,②).

[0043] 한편 NI query는 동일한 도메인 이름에 있는 IoT 디바이스에 멀티캐스트로 전달될 수 있다. 이 경우 도메인 이름에 있는 복수의 IoT 디바이스들은 서로 다른 시간에 NI reply를 하는 것이 바람직하다. 따라서 복수의 IoT 디바이스들은 각각 임의(random)로 지연된 시간을 두고 NI reply를 전송할 수 있다.

[0044] 라우터(250)는 NI reply를 통해 수신한 IoT 디바이스(210)의 DNS 이름을 DNS 서버(260)에 전송하고, DNS 서버(260)는 IoT 디바이스(210)의 DNS 이름을 저장한다(③). 경우에 따라서 DNS 서버(260)에 이미 도메인 이름이 존재하는 IoT 디바이스가 있을 수 있다. 후술하겠지만 IoT 디바이스의 도메인 이름에 위치 정보 등이 들어갈 수 있기 때문에 IoT 디바이스가 이동하면 도메인 이름도 변경될 수 있다. 따라서 이 경우 DNS 서버(260)는 IoT 디바이스에 대한 DNS 이름을 새롭게 수신한 DNS 이름으로 업데이트한다. 라우터(250)는 IoT 디바이스(210)의 도메인 이름과 함께 IPv6의 IP 주소를 수집하여 DNS 서버(260)에 함께 저장한다.

[0046] 도 5는 DNSNA(200)에서 클라이언트 장치가 IoT 디바이스를 제어하는 과정에 대한 예이다. 도 5는 도4와 같이 DNS 서버(260)에 IoT 디바이스(210)에 대한 DNS 이름이 저장된 상태를 전제로 한다.

[0047] 사용자는 스마트폰과 같은 클라이언트 장치(80)를 통해 IoT 디바이스(210)에 대한 정보를 확인하거나, IoT 디바이스(210)를 제어할 수 있다. 클라이언트 장치(80)는 DNS 서버(260)에 접속하여 네트워크에 존재하는 IoT 디바이스(210)에 대한 DNS 이름을 획득한다(①). 도 5에서는 하나의 IoT 디바이스(210)만을 도시하였으나, 네트워크에는 다수의 IoT 디바이스들이 있을 수 있다. 이 경우 클라이언트 장치(80)는 DNS 서버(260)로부터 네트워크에 존재하는 다수의 IoT 디바이스들에 대한 DNS 이름을 획득할 수 있다. 클라이언트 장치(80)는 획득한 DNS 이름을 이용하여 IoT 디바이스(210)에 접근할 수 있다. 이후 클라이언트 장치(80)는 IoT 디바이스(210)인 냉장고의 정보를 확인하거나, 냉장고의 동작을 제어할 수 있다.

[0049] DNS 이름은 IoT 디바이스의 모델 이름, IoT 디바이스의 종류를 나타내는 모델 카테고리, 동일한 모델 이름을 갖

는 IoT 디바이스 마다 서로 다른 디바이스 식별자 및 IoT 디바이스의 네트워크 도메인 중 적어도 하나를 포함할 수 있다. DNS 이름은 IoT 디바이스의 모델과 관련된 정보(모델 정보)를 DNS 이름에 갖는 것이 특징이다.

[0050] 도 6은 댁내에서 사용하는 IoT 디바이스에 대한 DNS 이름에 대한 예이다. 도 6에서 DNS 이름은 IoT 디바이스의 제품 종류(카테고리) 및 도메인 이름으로 구성된다. 도 6은 DNS 이름이 IoT 디바이스의 모델 정보 중 모델 카테고리에 해당하는 정보를 포함한 예이다. 도 6에서 도메인 이름은 "home"이고, 도메인 이름 앞에 위치한 정보가 모델 카테고리에 해당한다. 도 6에서 TV는 "tv.home", 냉장고는 "refrigerator.home", 세탁기는 "washer.home", 통풍팬(fan)은 "fan.home", 난방기구인 히터는 "heater.home", 컴퓨터는 "computer.home", 전화기는 "phone.home", 체중계는 "scale.home"이란 DNS 이름을 갖는다.

[0051] DNS 이름에 도메인 이름 외에 모델 카테고리가 포함되면, DNS 이름이 IoT 디바이스에 대한 정보를 제공하게 된다. 물론 DNS 이름에 포함된 모델 정보가 어떤 의미인지에 대해서는 사전에 정의되고, 공유되어야 할 것이다. 예컨대, 사용자는 자신의 스마트폰을 통해 DNS 서버에 접속하여 DNS 이름을 획득하면, 스마트폰은 사전에 정의된 규약에 따라 모델 카테고리를 기준으로 도 6과 같이 댁내(home)에 어떠한 종류의 IoT 디바이스가 있는지 표시할 수 있다.

[0053] DNS 이름은 아래의 표 1과 같은 형태를 가질 수 있다.

*표 1*

[0054]

| unique id.device model.device category.domain name |
|---|

[0055] unique_id는 도메인 이름에 존재하는 동일한 모델에 대해 서로 다른 고유 식별자를 의미한다. device_model은 IoT 디바이스를 제조한 제조자 또는 판매자가 제공하는 모델 이름을 말한다. 모델 이름에는 제품을 생산한 제조자의 이름이 포함될 수 있다. device_category는 IoT 디바이스의 모델 카테고리를 나타낸다. domain_name은 IoT 디바이스가 갖는 네트워크 도메인의 DNS 서픽스를 의미한다. 물론 DNS 이름을 구성하는 요소의 순서는 표1과 다를 수 있다. 모델 이름 및 모델 카테고리는 사전에 IoT 디바이스에 저장된 정보에 해당한다. 고유 식별자는 DNS 이름 생성 과정에서 IoT 디바이스가 생성하는 정보이다. DNS 서픽스는 DNSSL에 포함되어 전달된 정보이다.

[0056] 예컨대, 도 3에서 IoT 디바이스(210)는 "refrigerator1.samsung_RH269LP. refrigerator.home"이란 DNS 이름을 생성하였다. 이미 설명한 바와 같이 "refrigerator1"은 동일한 모델의 디바이스를 구별하기 위한 고유 식별자이고, "samsung_RH269LP"는 제조사이름을 포함하는 모델 이름이고, "refrigerator"는 모델 카테고리이고, "home"은 DNS 도메인 서픽스에 해당한다.

[0058] 한편 IoT 디바이스의 개체 수가 많을 경우 모델 정보만으로는 IoT 디바이스를 구분하기 어려울 수 있다. 이 경우 DNS 이름은 IoT 디바이스의 위치 정보라는 추가적인 정보를 포함할 수 있다. 장치의 구별 가능성과는 관계없이 DNS 이름에 포함되는 위치 정보는 새로운 정보를 사용자에게 제공할 수 있다. 위치 정보가 포함된 DNS 이름의 예는 아래의 표 2와 같다.

*표 2*

[0059]

| unique id.device model.device category.location of device.domain name |
|---|

[0060] 표 2의 DNS 이름은 표1의 DNS 이름에 location_of_device가 추가되었다. location_of_device는 IoT 디바이스의 위치 정보를 의미한다. 위치 정보는 IoT 디바이스가 사전에 미리 획득한 것을 전제로 한다.

[0061] 위치 정보는 다음과 같이 다양한 방법으로 수집이 가능하다. (1) IoT 디바이스가 GPS 센서와 같인 위치 추적 장치를 갖고 있다면 GPS 좌표와 같은 절대적인 좌표 정보를 사용할 수 있다. GPS 좌표 정보와 같이 위성 신호를 이용하는 경우 주로 실외에 배치되는 IoT 디바이스가 좌표를 획득할 수 있다. 또는 실내 통신 신호(WiFi 등)를 이용하여 실내에서 위치 측위를 하는 시스템이 있는 경우, IoT 디바이스는 실내에서의 위치 정보도 획득할 수

있다.

[0062] (2) IoT 디바이스가 가속도 센서, 지자기 센서 등을 내장하고 있다면 특정 기준 지점을 기준으로 IoT 디바이스가 이동하여 위치한 지점에 대한 상대적인 위치 정보를 생성할 수 있다. 예컨대, 댁내에 배치되는 IoT 디바이스는 댁내의 구조를 사전에 알고 있다는 전제하에 출입문과 같은 기준 위치에서 이동을 시작하여 배치되는 위치를 파악할 수 있다.

[0063] (3) 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. 또한 IoT 디바이스는 통신 가능한 거리에 다수의 위치 정보를 갖고 있는 장치가 3개 이상 있다면 서로 주고받는 신호의 세기를 이용하여, 자신이 주변 장치와의 사이에서 어떤 위치에 있는지 비교적 정확하게 파악할 수도 있다.

[0064] (4) IoT 디바이스에 대한 위치 정보를 획득하는 별도의 장치를 사용할 수도 있다. 예컨대, 사용자가 자신의 실외 또는 실내에서 위치를 측위할 수 있는 장치를 이용하여 근처에 배치된 IoT 디바이스에 IoT 디바이스의 위치 정보를 전달할 수 있다.

[0066] 도 7은 댁내에서 사용하는 IoT 디바이스에 대한 DNS 이름의 다른 예이다. 도 7은 댁내에서 사용하는 IoT 디바이스가 모델 정보 및 위치 정보를 사용하여 DNS 이름을 생성한 예를 도시한다.

[0067] 도 7은 댁내에서의 각 공간(거실, 주방, 침실 등)을 알파벳(A, B, C, D, E 및 F)로 표시하였다. 각 구역에 있는 IoT 디바이스의 DNS 이름은 아래의 표 3과 같다.

*표 3*

[0068]

| 영역 | IoT 디바이스 | DNS 이름 |
|---|---|---|
| A | 에어컨 | air_conditioner1.samsung_AC64.air conditiner.bed_room1.home |
| B | 스마트TV | smart_tv1.samsung_HG32N.smart_tv.living_room.home |
| C | 에어컨 | air_conditioner2.samsung_AC64.air conditiner.bed_room2.home |
| D | 에어컨 | air_conditioner3.samsung_AC64.air conditiner.bed_room3.home |
| E | 냉장고 | refrigerator1.samsung.regrigerator.home |
| F | 세탁기 | washer.home |

[0069] F로 표시된 세탁실에 있는 세탁기의 DNS 이름은 모델 카테고리 및 도메인 이름 이름을 포함한다. E로 표시된 주방에 있는 냉장고의 DNS 이름은 고유 식별자,제조사만 표시된 모델 이름, 모델 카테고리 및 도메인 이름 이름을 포함한다. B로 표시된 거실에 있는 스마트 TV의 DNS 이름은 고유 식별자, 모델 이름, 모델 카테고리, 위치 정보(living_room) 및 도메인 이름 이름을 포함한다.

[0070] A, C 및 D로 표시된 침실에 있는 에어컨도 각각 고유 식별자, 모델 이름, 모델 카테고리, 위치 정보 및 도메인 이름 이름을 DNS 이름으로 갖는다. A, C 및 D로 표시된 침실에 있는 에어컨은 모델 이름이 동일하다. 따라서 고육 식별자가 각각 air conditioner1, air conditioner2 및 air conditioner3로 각각 다르다. 또한 각 에어컨이 위치한 위치 정보도 bed_room1, bed_room2 및 bed_room3으로 서로 다르다. 도 7과 같은 비교적 작은 영역에서 사용되는 IoT 디바이스라면 고유 식별자를 사용하지 않고, 위치 정보만으로도 서로 구분 가능할 수도 있다.

[0071] 사용자는 IoT 디바이스의 상태를 확인할 수 있고, 각 IoT 디바이스를 제어할 수도 있다. 도 7에서 사용자는 댁내에 있는 WiFi AP를 이용하여 인터넷을 통해 댁내에 배치된 IoT 디바이스에 접속할 수 있다.

[0072] 도 8은 스마트폰이 댁내에 위치한 IoT 디바이스에 대한 정보를 표시하는 예이다. 도 8은 도 7의 사용자가 스마트폰을 통해 댁내에 배치된 IoT 디바이스를 확인하는 화면에 대한 예이다. DNS 이름에 위치 정보가 포함된 스마트 TV 및 에어컨들은 정확하게 배치된 위치에 각 IoT 디바이스에 해당하는 아이콘이 표시되었다. 이는 사용자의 클라이언트 장치(스마트폰)이 DNS 서버로부터 DNS 이름을 획득하고, DNS 이름으로부터 각 IoT 디바이스의 위치 정보를 추출했기 때문이다. 그러나 DNS 이름에 위치 정보가 없었던 냉장고 및 세탁기는 정확한 위치를 알 수 없어서 중앙에 점선 역역에 물음표(?)와 함께 표시하였다.

[0073] 사용자는 침실 3에 있는 에어컨을 선택하면 도 8의 하단과 같은 메뉴가 화면에 표시될 수 있다. 스마트폰은 도 8의 하단과 같이 선택한 IoT 디바이스(에어컨)에 대한 개략적인 정보를 출력하고, 에어컨을 제어할 수 있는 메뉴를 출력할 수 있다. 이후 사용자는 제어 메뉴를 통해 에어컨을 제어할 수 있다. 예컨대, (1) 전원을 키거나

끌 수 있고(On/Off), (2) 동작되는 시간을 설정할 수 있고(Timer), (3) 현재 상태(Status)를 확인할 수도 있다.

[0075] 별도로 도시하지 않았지만 IoT 디바이스는 댁내뿐만 아니라 쇼핑몰, 물류 창고와 같은 거대한 공간에서 사용될 수 있다. 예컨대, 물류 창고에 보관된 각 물품은 자신의 정보를 나타내는 센서 장치(IoT 디바이스)가 부착되어 있다고 가정한다. 이 경우 관리자는 각 IoT 디바이스의 DNS 이름을 획득하면, IoT 디바이스의 위치 정보를 획득할 수 있다. 또한 쇼핑몰 같은 곳에서 매장에 배치된 제품에 자신의 정보를 나타내는 센서 장치(IoT 디바이스)가 부착되어 있다고 가정하면, 쇼핑몰 고객은 스마트폰을 통해 자신이 원하는 물건이 쇼핑몰 몇 층 어떤 구역에 있는지 쉽게 파악할 수 있다. 이는 DNS 이름에 모델 정보 및 위치 정보가 포함되어 있기 때문이다. 즉, 별도로 서버에서 모델 정보 및 위치 정보를 관리하지 않아도, 클라이언트 장치가 IPv6의 DNS 서버에서 DNS 이름을 획득하면, 사용자에게 특정한 정보를 제공할 수 있다.

[0077] 도 9 및 도 10은 도로에서 자동차 및 교통 제어 장치에 대한 정보를 획득하는 예를 도시한다. 교통관제 센터가 교통 관리를 위하여 차량 등에 대한 정보를 이용할 수 있다. 또한 지능형 자동차 시스템에서 차량의 자율 주행을 위해 차량 등에 대한 정보를 이용할 수도 있다.

[0078] 도 9는 도로에서 이동하는 차량에 대한 DNS 이름에 대한 예이다. 도 9는 도로를 이동하는 차량(Car 1)과 과속 감시 카메라(Camera1)를 도시하였다. 도 9는 IoT 디바이스의 DNS 이름을 생성하는 과정에 대한 예이다. Car1은 IoT 디바이스에 해당하는 장치를 내장하고 있다. Camera1은 차량들과 통신할 수 있는 장치를 내장하고 있다. Camera1이 라우터의 역할을 할 수도 있다. 물론 일반적으로 과속 감시 카메라는 차량의 번호판을 인식하여 차량을 식별하지만, 영상을 통한 식별 외에 다른 정보를 이용할 수도 있을 것이다.

[0079] Car1이 도로를 이동하면 Camera1가 Car1의 속도를 측정하는 동시에 Car1에 대해 DNS 이름을 묻을 수 있다. Camera1은 DNSSL 옵션을 Car1에 전달하고, Car1은 자신의 DNS 이름을 생성한다.

[0080] Car1이 생성한 DNS 이름은 "car1.benz_GLK.car_suv.road"이다. 여기서 DNS 이름은 고유 식별자(car1), 모델 이름(benz_GLK), 모델 카테고리(car_suv) 및 도메인 이름 이름(road)으로 구성되어 있다. Car1이 생성한 DNS 이름이 중복되지 않는 다면, Camera1은 곧바로 Car1에 대한 DNS 이름을 DNS 서버에 등록할 수 있다. 또는 전술한 바와 같이 Camera1이 NI query를 통해 Car1의 DNS 이름을 획득하고, 이후 DNS 서버에 Car1의 DNS 이름을 등록할 수도 있을 것이다.

[0081] 클라이언트 장치는 DNS 서버에 저장한 차량의 DNS 이름을 이용하여 해당 차량에 대한 관리를 할 수도 있고, 특정한 종류의 차량에 대한 통계적인 정보를 생성할 수도 있다.

[0083] 도 10은 교차로에서의 차량 및 고정 장치에 대한 DNS 이름에 대한 예이다. 도 10에서 교차로를 이동하는 차량 및 교차로 주변에 배치된 교통 장치는 모두 IoT 디바이스에 해당한다. 도 10에서는 교차로에 존재하는 IoT 디바이스에 대한 DNS 이름이 DNS 서버에 이미 등록된 상태라고 가정한다. 도 10은 교차로가 구분하는 영역을 하단의 영역부터 반시계방향으로 Segment1, Segment2, Segment3 및 Segment4로 표시하고 있다.

[0084] RSU는 차량 통신(VANET)을 위해 도로에 설치되는 AP 장치이고, Traffic Light는 교차로에 배치되는 신호등이고, Detector는 이동하는 차량에 대한 정보를 획득하기 위한 센서 장치이다. Controller는 현재 교차로에 위치하는 차량 및 교통 장치에 대한 정보를 나타내고, 제어하는 장치이다. 교차로에 배치되는 차량 및 교통 장치는 기본적으로 AP를 통해 네트워크에 접속한다고 가정한다.

[0085] 도 10에서 4개의 차량(Car1, Car2, Car3 및 Car4)을 도시한다. 도 10은 각 차량에 대한 위치 정보를 X,Y,Z로 표시하였다. RSU가 교차로에 위치하는 IoT 디바이스에 대한 DNS 이름을 요청할 수도 있다. 한편 RSU 자체도 IoT 디바이스에 해당한다. 교차로에서 Detector는 이동하는 차량에 대해 DNS 이름을 요청할 수 있다. 도 10에 도시된 IoT 디바이스 각각에 대한 DNS 이름은 아래의 표 4와 같다.

*표 4*

[0087]

| IoT 디바이스 | DNS 이름 |
|---|---|
| RSU1 | rsu1.cisco_xz43.rsu.road_segment3.road |
| Traffic Light1 | traffic_light1.gov_TL32.traffic light.road_segment1.road |
| Traffic Light2 | traffic_light2.gov_TL32.traffic light.road_segment2.road |
| Detector1 | detector1.gov_LD51.loop_detector.road_segment1.road |
| Detector2 | detector2.gov_LD51.loop_detector.road_segment2.road |
| Car1 | car1.jeep_wrangler.car_suv.x1y1z1.road |

| | |
|---|---|
| Car2 | car2.audi_a435.car_sedan.x2y2z2.road |
| Car3 | car3.lexus_es300h.car_sedan.x3y3z3.road |
| Car4 | car4.bmw_x5.car_suv.x4y4z4.road |

[0088] 도 10의 IoT 디바이스의 DNS 이름은 모두 고유 식별자, 모델 이름, 모델 카테고리, 위치 정보 및 도메인 이름 이름을 포함한다. RSU1을 예로 설명하면 rus1은 고유 식별자이고, cisco_xz43은 모델 이름이고, rsu는 모델 카테고리이고, road_segment3는 위치 정보이고, road는 도메인 이름 이름이다.

[0089] 교차로에서 Detector는 이동하는 차량을 인식하고, 차량에 대한 DNS 이름을 요청할 수 있다. 도 10에서 차량의 DNS 이름에도 위치 정보를 사용하였다. 따라서 차량이 이동하는 경우 차량의 DNS 이름이 변경될 수 있다. 차량은 GPS 장치를 통해 측정되는 좌표 정보를 위치 정보로 삼는다고 가정한다. 예컨대, Detector1이 Car1에게 DNSSL 옵션을 전송하고, Car1이 현재 위치($X_1$, $Y_1$, $Z_1$)을 포함하는 DNS 이름을 생성하여 중복 검사할 수 있다.

이후 Detector1 또는 RUS1이 Car 1의 DNS 이름을 DNS 서버에 업데이트할 수 있다. Detector2는 Car2에 대한 DNS 이름 생성을 요청할 수도 있을 것이다.

[0090] 교차로에 고정 배치된 교통 장치는 배치되면서 자신의 위치 정보를 알고 있다고 가정한다. 따라서 RSU1의 DNS 이름은 road_segment3이라는 위치 정보를 포함하고, Traffic Light1은 road_segment1이라는 위치 정보를 포함하고, Traffic Light2은 road_segment2라는 위치 정보를 포함한다. Detector1은 road_segment1이라는 위치 정보를 포함하고, Detector2은 road_segment2라는 위치 정보를 포함한다.

[0091] 관리자가 교차로에 배치된 controller 또는 자신의 스마트폰으로 현재 교차로의 상태 정보를 확인할 수 있다. 이 경우 controller 또는 스마트폰의 화면에는 각 IoT 디바이스의 DNS 이름에 포함된 정보를 이용하여, 교차로에 각 IoT 디바이스의 종류 및 배치를 표시할 수 있다. 관리자는 controller 또는 스마트폰을 이용하여 교차로의 신호등(Traffic Light1, Traffic Light2)를 제어할 수도 있을 것이다. 관리자는 신호등의 위치 정보를 제공받기 때문에 직감적으로 자신이 제어해야 하는 신호등이 어떤 것인지(Traffic Light1 인지 또는 Traffic Light2 인지)를 식별할 수 있다. 또한 controller 또는 스마트폰에는 교차로에 위치하는 차량도 파악할 수 있기 때문에 관리자는 교통의 흐름을 고려하여 신호등을 적절하게 제어할 수 있다.

[0093] 한편 IoT 디바이스가 일정하게 이동하는 경우가 있다. 예컨대, 도 10에서 설명한 차량, 사용자가 소지하는 센서 장치, 사용자가 소지하는 스마트폰, 건물 내에서 이동하는 로봇 등과 같은 IoT 디바이스가 이에 해당한다. IoT 디바이스가 이동하면 IoT 디바이스가 속하는 도메인 네임 자체가 변동될 수 있다. 이 경우 IoT 디바이스가 이동하여 새로운 도메인 네임(도메인 서픽스)을 갖는 영역에 도달하는 경우, 새로운 DNS 이름을 생성할 수 있다.

[0094] IoT 디바이스가 이동에 따라 새로운 도메인 네임을 포함하는 RA 옵션을 받는 경우 IoT 디바이스는 DNS 동적 업데이트(RFC 2136)를 이용하여 종전의 DNS 이름을 DNS 서버에서 삭제할 수 있다. 특정 IoT 디바이스에 대해서는 적어도 하나의 DNS 이름이 DNA 서버에 존재해야 하므로, IoT 디바이스는 모바일 IPv6의 홈 네트워크에 있는 기본(default) DNS 이름은 삭제하지 않는 것이 바람직하다.

[0096] 전술한 모든 과정에서 메시지 전달은 보안성을 유지하기 위한 다양한 방법을 사용할 수 있다. 예컨대, RFC 4620 "Security Consideration"에 정의된 NI 프로토콜의 보안 이슈와 관련된 내용을 적용할 수 있다. 예컨대, 개인 정보 유출을 막기 위해 위치와 관련된 도메인 이름은 공유키 또는 공개키를 이용하여 암호화할 수 있다. 예를들어 smartphone1.living_room.home이라는 DNS 이름은 암호화되어 smartphone1.xxx.home과 같이 암호화 될 수 있다. xxx는 living_room에 대해 암호화된 문자열이다.

[0098] 본 실시 예 및 본 명세서에 첨부된 도면은 전술한 기술에 포함되는 기술적 사상의 일부를 명확하게 나타내고 있는 것에 불과하며, 전술한 기술의 명세서 및 도면에 포함된 기술적 사상의 범위 내에서 당업자가 용이하게 유추할 수 있는 변형 예와 구체적인 실시 예는 모두 전술한 기술의 권리범위에 포함되는 것이 자명하다고 할 것이다.

## *부호의 설명*

[0099] 50 : 컴퓨터 장치

100 : DNS

110 : 서비스 서버

150 : 라우터

160 : DNS 서버

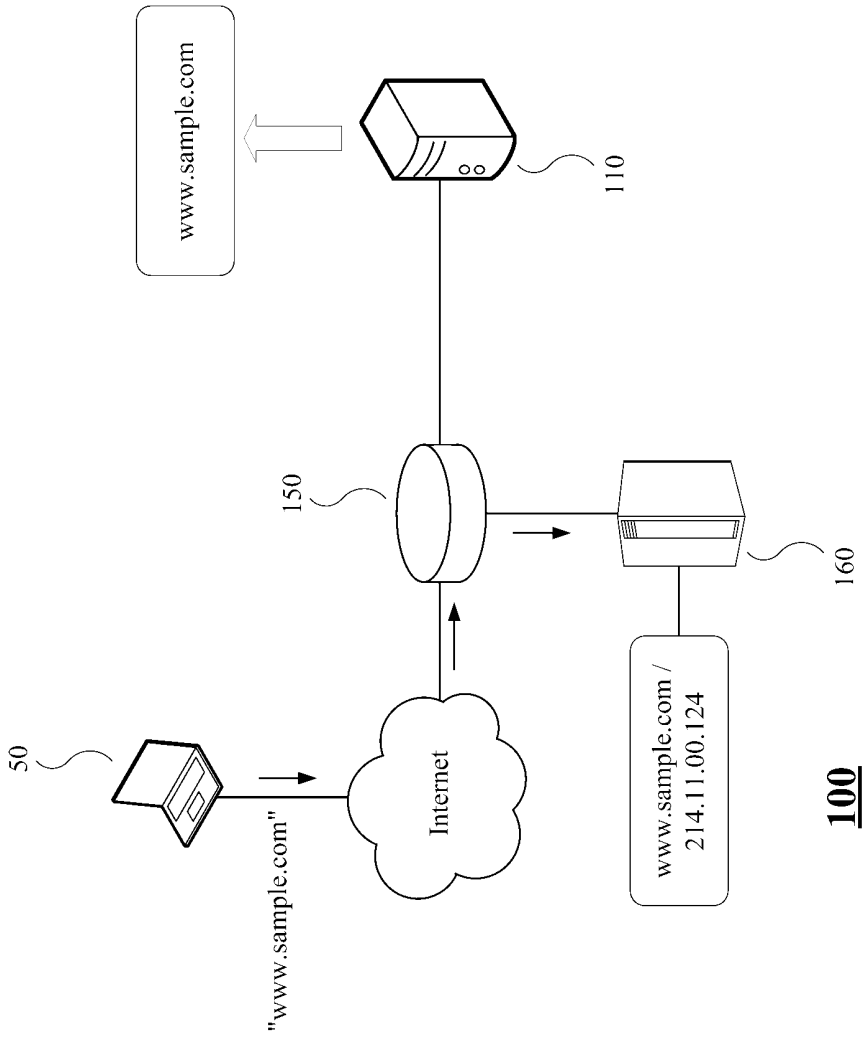
80 : 클라이언트 장치 200 : DNSNA

210 : IoT 디바이스 230 : AP

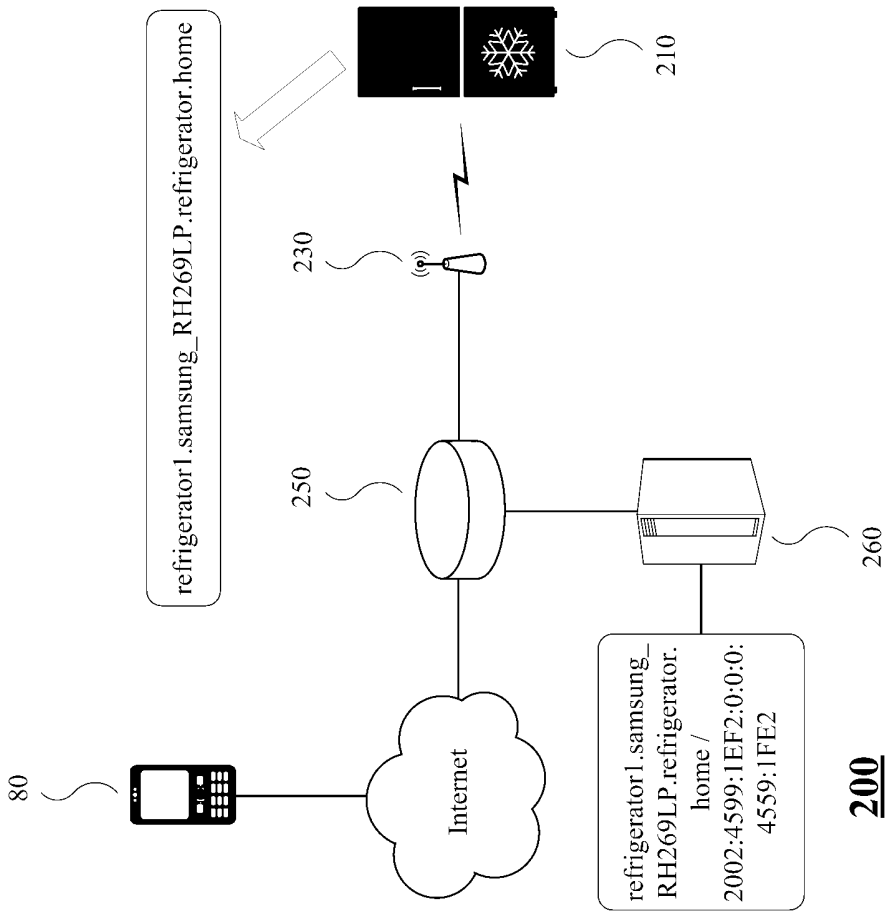
250 : 라우터 260 : DNS 서버

## 도면

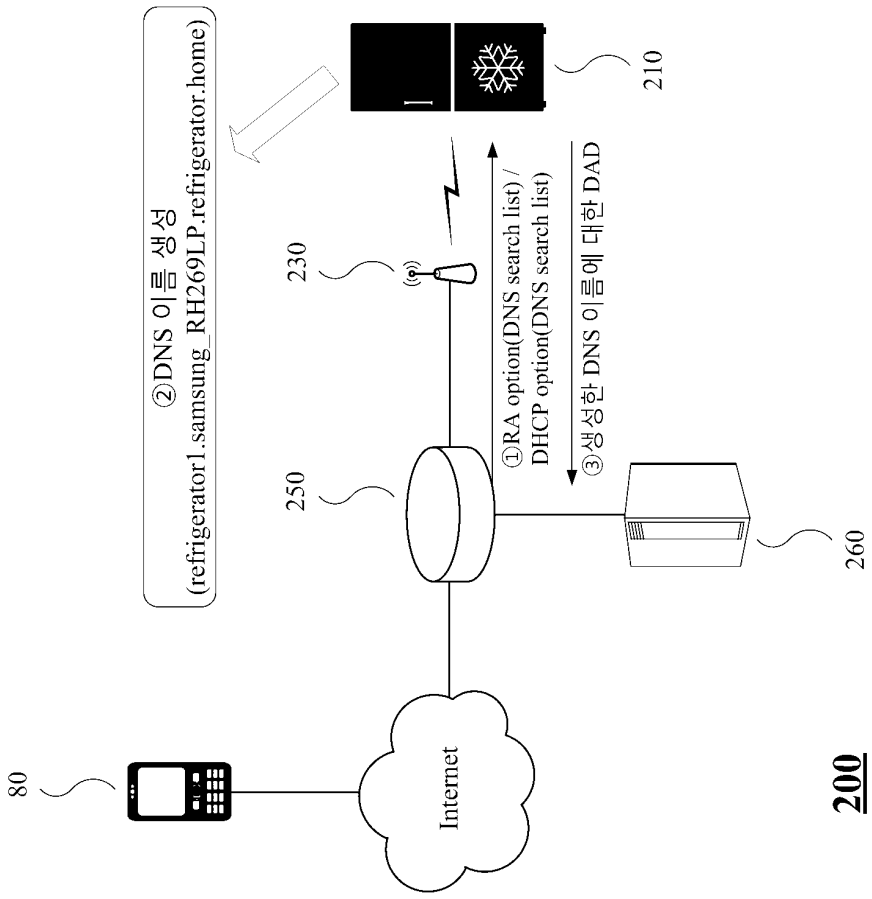
### 도면1



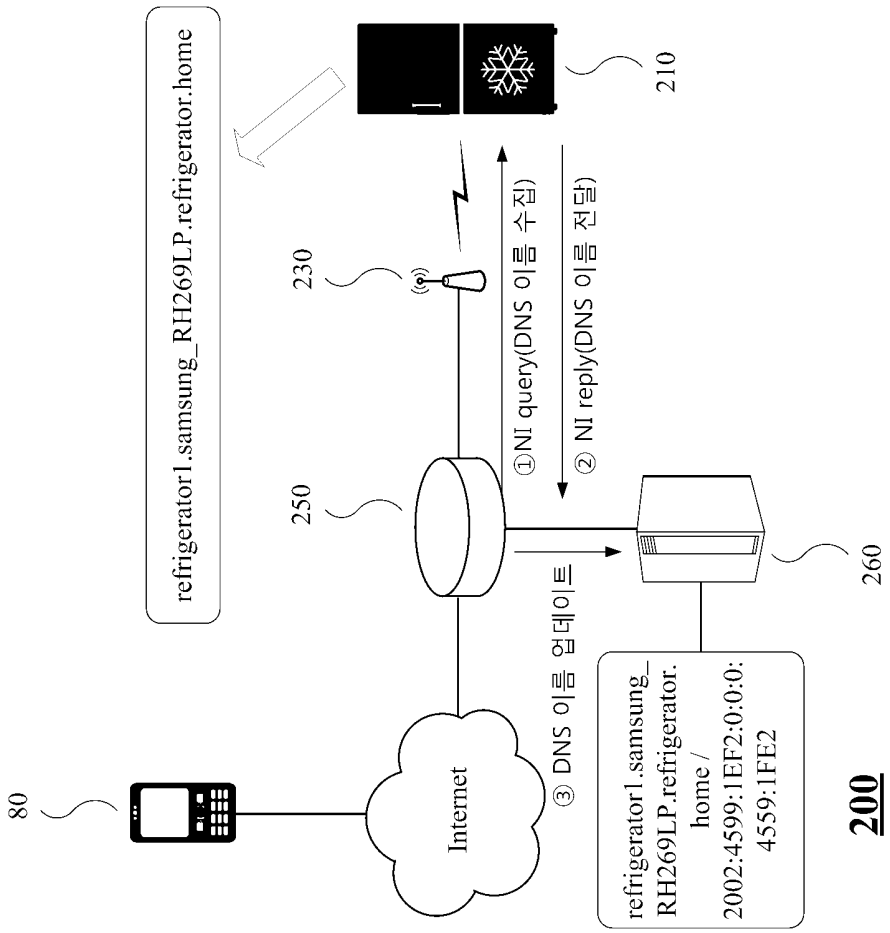
도면2


refrigerator1.samsung_RH269LP.refrigerator.home
210
230
250
260
80
Internet
refrigerator1.samsung_
RH269LP.refrigerator.
home /
2002:4599:1EF2:0:0:0:
4559:1FE2
200

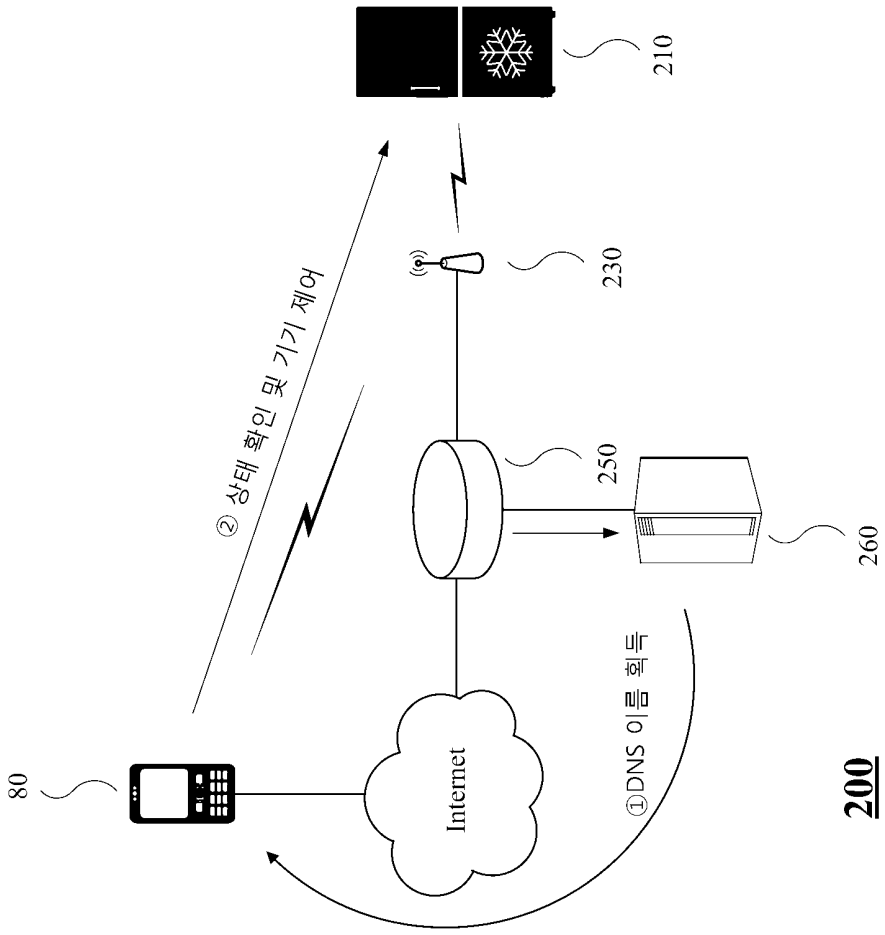
도면3


②DNS 이름 생성
(refrigerator1.samsung_RH269LP.refrigerator.home)
210
230
250
①RA option(DNS search list) /
DHCP option(DNS search list)
③생성한 DNS 이름에 대한 DAD
260
80
Internet
200

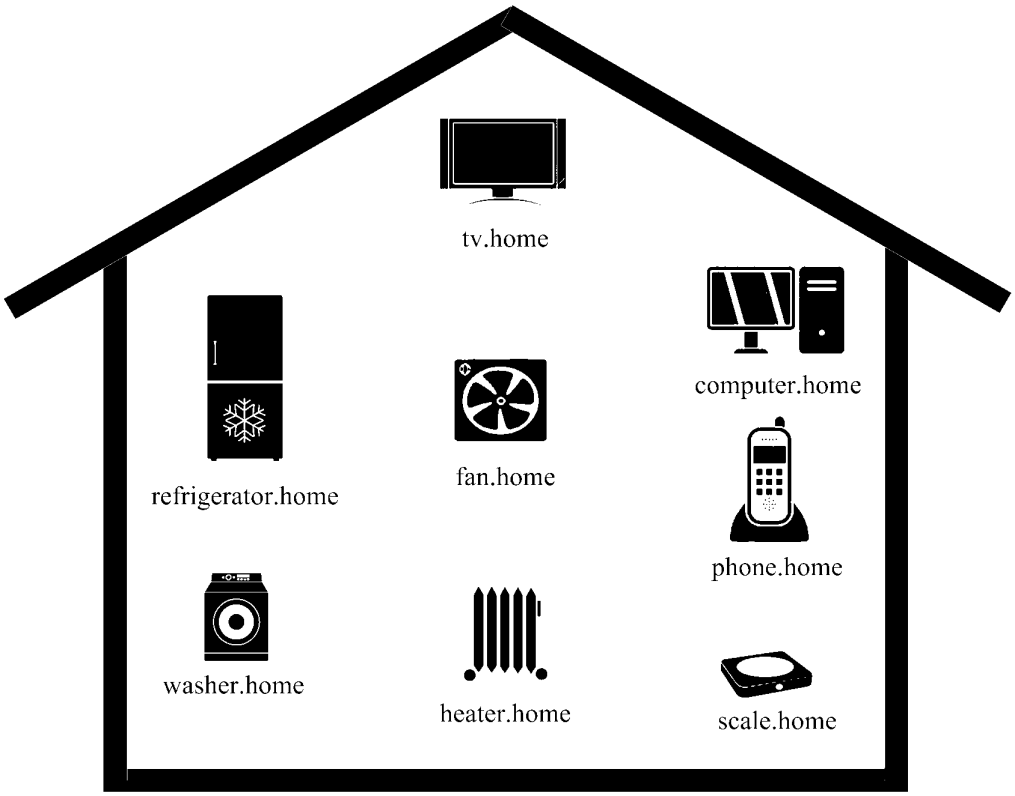
도면4


refrigerator1.samsung_RH269LP.refrigerator.home
210
230
250
260
80
Internet
①NI query(DNS 이름 수집)
② NI reply(DNS 이름 전달)
③ DNS 이름 업데이트
refrigerator1.samsung_RH269LP.refrigerator.home / 2002:4599:1EF2:0:0:0:4559:1FE2
200

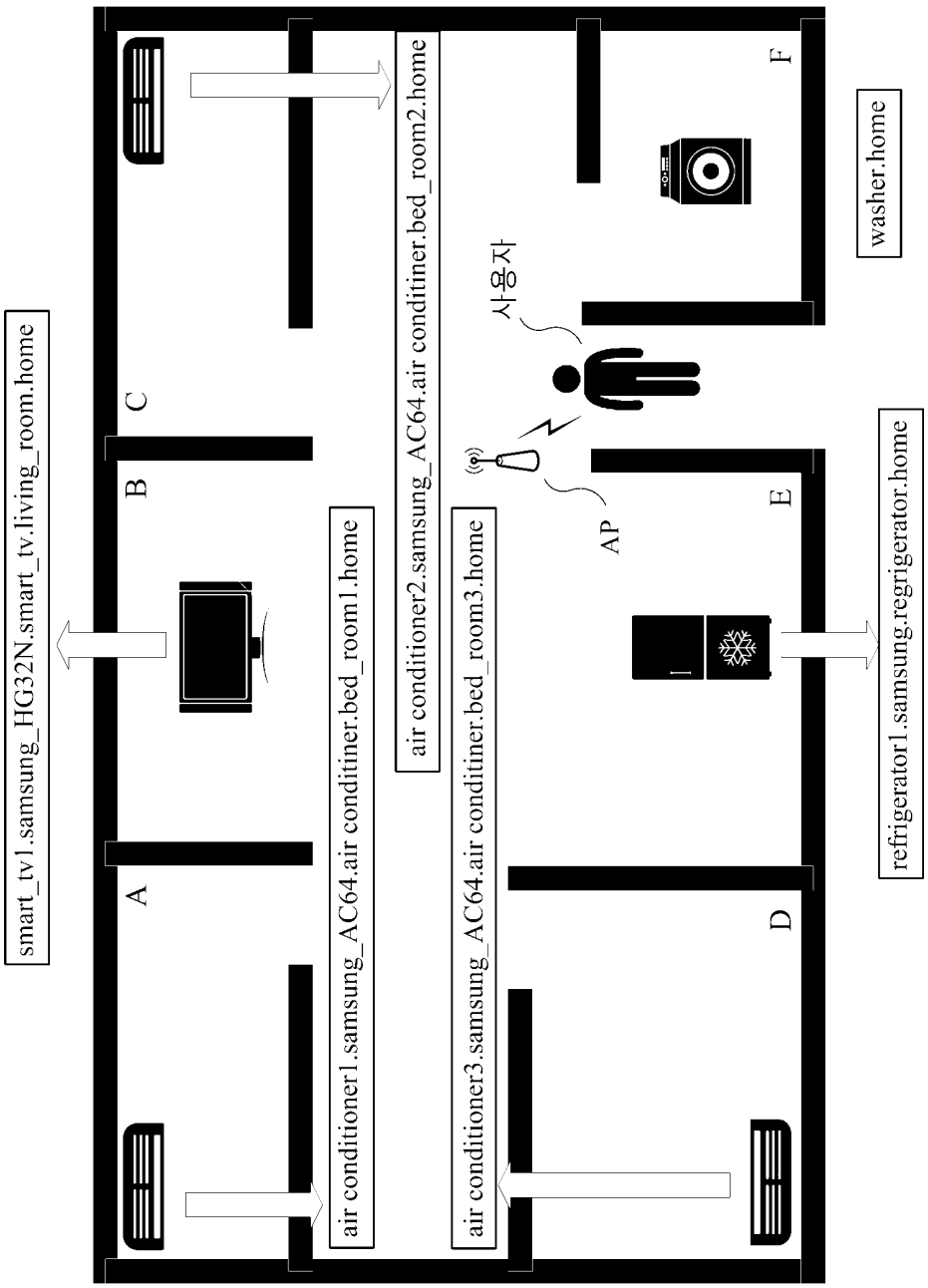
도면5



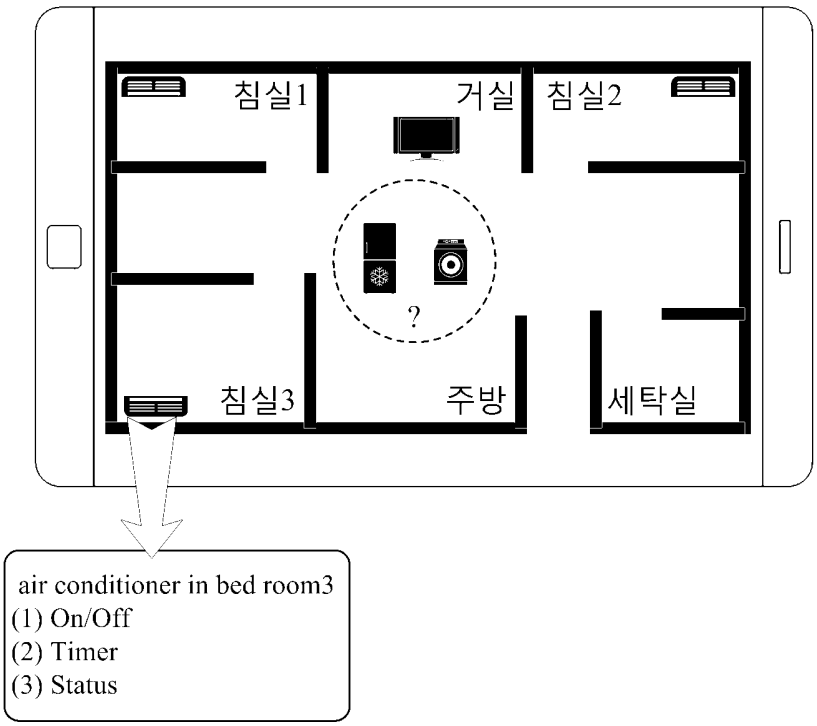
도면6


tv.home
computer.home
refrigerator.home
fan.home
phone.home
washer.home
heater.home
scale.home

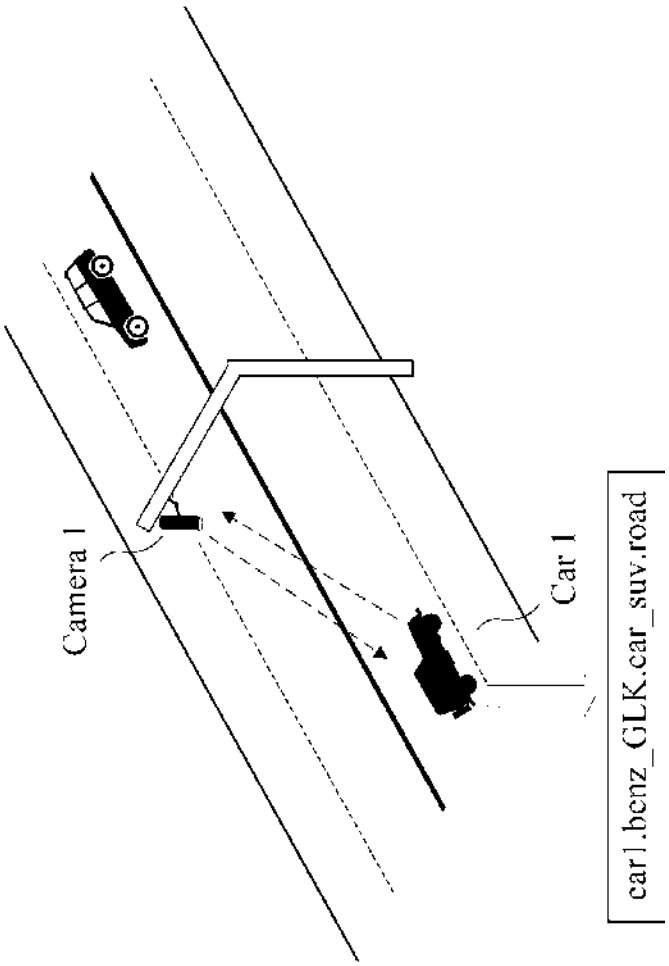
도면7


smart_tv1.samsung_HG32N.smart_tv.living_room.home
air conditioner1.samsung_AC64.air conditiner.bed_room1.home
air conditioner2.samsung_AC64.air conditiner.bed_room2.home
air conditioner3.samsung_AC64.air conditiner.bed_room3.home
refrigerator1.samsung.regrigerator.home
washer.home
AP
사용자
A
B
C
D
E
F

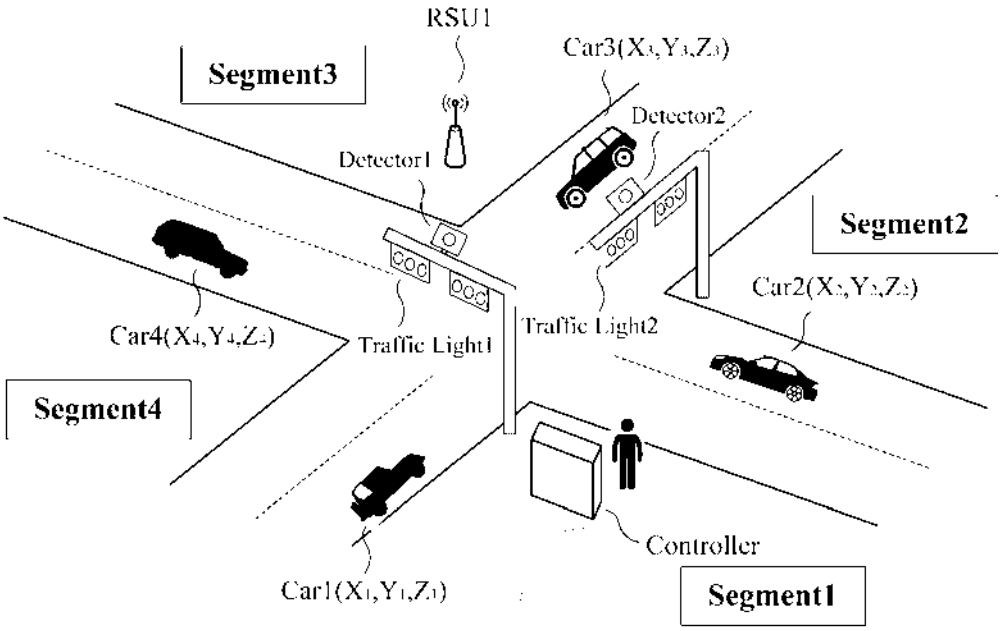
도면8


침실1
거실
침실2
?
침실3
주방
세탁실
air conditioner in bed room3
(1) On/Off
(2) Timer
(3) Status

도면9



도면10




RSU1: rsu1.cisco_xz43.rsu.road_segment3.road
Traffic Light1: traffic_light1.gov_TL32.traffic light.road_segment1.road
Traffic Light2: traffic_light2.gov_TL32.traffic light.road_segment2.road
Detector1: detector1.gov_LD51.loop_detector.road_segment1.road
Detector2: detector2.gov_LD51.loop_detector.road_segment2.road
Car1: car1.jeep_wrangler.car_suv.$x_1y_1z_1$.road
Car2: car2.audi_a435.car_sedan.$x_2y_2z_2$.road
Car3: car3.lexus_es300h.car_sedan.$x_3y_3z_3$.road
Car4: car4.bmw_x5.car_suv.$x_4y_4z_4$.road

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년09월08일
(11) 등록번호 10-1776882
(24) 등록일자 2017년09월04일

(51) 국제특허분류(Int. Cl.)
*H04L 29/12* (2006.01) *H04L 12/12* (2006.01)
*H04L 29/06* (2006.01)
(52) CPC특허분류
*H04L 61/3015* (2013.01)
*H04L 12/12* (2013.01)
(21) 출원번호 10-2016-0135083
(22) 출원일자 2016년10월18일
심사청구일자 2016년10월18일
(56) 선행기술조사문헌
US20140304785 A1*
J. Jeong외 2명, "DNS Name Autoconfiguration for Internet of Things Devices", Network Working Group, IETF, draft-jeong-6man-iot-dns-autoconf-00 (2015.10.14.)*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
**성균관대학교산학협력단**
경기도 수원시 장안구 서부로 2066 (천천동, 성균관대학교내)
(72) 발명자
**정재훈**
부산광역시 금정구 금강로 225, 204동 1501호 (장전동, 벽산블루밍디자인시티)
**이근태**
경기도 수원시 장안구 하률로46번길 22, 제208동 1406호 (천천동, 삼호진덕아파트)
(74) 대리인
**특허법인로얄**

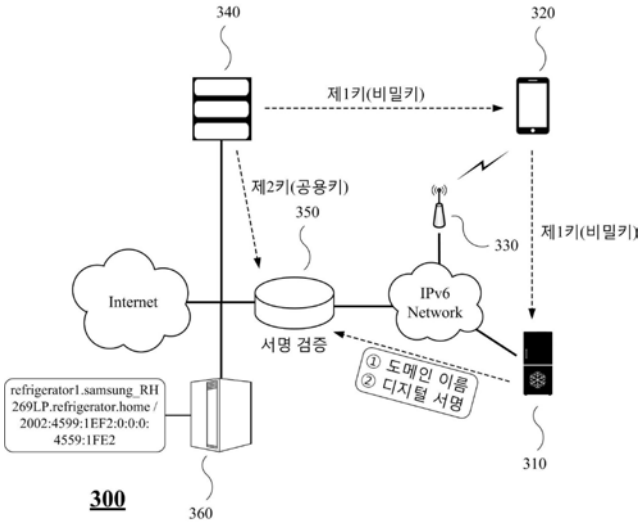
전체 청구항 수 : 총 9 항 심사관 : 박보미

(54) 발명의 명칭 **IoT 디바이스에 대한 보안성 있는 DNS 네이밍 방법 및 보안성 있는 DNS 네이밍 등록을 수행하는 라우터 장치**

## (57) 요 약

IoT 디바이스에 대한 보안성 있는 DNS 네이밍 방법은 이동 단말이 제1 키를 인증 서버로부터 수신하고, 도메인 이름 수집장치가 제2 키를 인증 서버로부터 수신하는 단계, 상기 도메인 이름 수집장치가 IoT 디바이스로부터 모델 정보가 포함된 도메인 이름 및 서명을 수신하는 단계, 상기 도메인 이름 수집장치가 상기 디지털 서명을 상기 제2 키로 복호하여 상기 서명을 검증하는 단계 및 상기 도메인 이름 수집장치가 상기 검증이 성공하면 상기 IoT 디바이스에 대한 DNS 이름을 DNS 서버에 등록하는 단계를 포함한다. 상기 제1 키 및 상기 제2 키는 인증 서버가 상기 IoT 디바이스에 대한 인증을 위해 생성하는 비대칭 키이고, 상기 서명은 상기 IoT 디바이스가 상기 이동 단말로부터 근거리 통신을 통해 전달받은 상기 제1 키를 이용하여 생성한다.

## 대 표 도



(52) CPC특허분류
*H04L 61/2007* (2013.01)
*H04L 61/6059* (2013.01)
*H04L 63/12* (2013.01)
*H04L 67/16* (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 2014R1A1A1006438
부처명 미래창조과학부
연구관리전문기관 한국연구재단
연구사업명 이공분야기초연구사업-(구)일반연구자지원사업(신진연구) 3/3
연구과제명 [EZ]사이버물리시스템 기반의 스마트 도로 서비스 소프트웨어
기 여 율 34/100
주관기관 성균관대학교 산학협력단
연구기간 2016.05.01 ~ 2017.04.30
이 발명을 지원한 국가연구개발사업
과제고유번호 10041244
부처명 미래창조과학부
연구관리전문기관 정보통신기술진흥센터
연구사업명 SW컴퓨팅산업원천기술개발사업(SW)
연구과제명 스마트TV 2.0 소프트웨어 플랫폼
기 여 율 33/100
주관기관 성균관대학교 산학협력단
연구기간 2015.09.01 ~ 2016.08.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711035549
부처명 미래창조과학부
연구관리전문기관 정보통신기술진흥센터
연구사업명 SW중심대학지원사업-SW특성화대학원
연구과제명 SW중심대학(성균관대)
기 여 율 33/100
주관기관 성균관대학교 산학협력단
연구기간 2016.03.01 ~ 2017.02.28

# 명 세 서

## 청구범위

### 청구항 1

인증 서버가 서명용 키인 제1 키를 이동 단말로 전송하고, 상기 제1 키에 대한 검증용 키인 제2 키를 도메인 이름 수집장치로 전송하는 단계;

상기 이동 단말이 근거리 통신을 통해 상기 제1 키를 IoT(Internet of Things) 디바이스로 전달하는 단계;

상기 IoT 디바이스가 상기 제1 키를 이용하여 상기 IoT 디바이스의 도메인 이름의 적어도 일부를 암호화한 디지털 서명을 생성하는 단계;

상기 IoT 디바이스가 상기 디지털 서명, 상기 도메인 이름 및 상기 도메인 이름에 대응되는 IPv6 주소를 상기 도메인 이름 수집장치로 전송하는 단계;

상기 도메인 이름 수집장치가 상기 디지털 서명을 상기 제2 키로 복호하고, 복호 결과가 상기 도메인 이름과 동일한지 판단함으로써 상기 디지털 서명을 검증하는 단계; 및

상기 도메인 이름 수집장치가 상기 검증에 성공하면 상기 IoT 디바이스로부터 수신한 상기 도메인 이름 및 상기 IPv6 주소를 DNS(Domain Name System) 서버에 등록하는 단계; 를 포함하되,

상기 제1 키 및 상기 제2 키는 상기 인증 서버가 상기 IoT 디바이스에 대한 인증을 위해 생성되는 비대칭 키인, IoT 디바이스에 대한 보안성 있는 DNS 네이밍 방법.

### 청구항 2

삭제

### 청구항 3

제1항에 있어서,

상기 도메인 이름은 상기 IoT 디바이스의 모델 이름, 상기 IoT 디바이스의 생산자, 상기 IoT 디바이스의 판매자, 상기 IoT 디바이스의 모델 카테고리, 동일한 모델 이름을 갖는 IoT 디바이스를 구별하는 식별자 및 상기 IoT 디바이스의 네트워크 도메인 중 적어도 하나를 포함하는 IoT 디바이스에 대한 보안성 있는 DNS 네이밍 방법.

### 청구항 4

제1항에 있어서,

상기 도메인 이름은 상기 IoT 디바이스에 대한 위치 정보를 더 포함하는 IoT 디바이스에 대한 보안성 있는 DNS 네이밍 방법.

### 청구항 5

제1항에 있어서,

상기 IoT 디바이스는 IPv6 프로토콜의 ND(neighbor discovery) 프로토콜에 따라 상기 도메인 이름에 대한 중복 검사를 수행하고, 상기 도메인 이름이 중복된 경우 상기 IoT 디바이스는 새로운 식별자를 포함하는 새로운 도메인 이름을 생성하는 IoT 디바이스에 대한 보안성 있는 DNS 네이밍 방법.

### 청구항 6

제1항에 있어서,

상기 도메인 이름 수집장치는 IPv6의 NI(Node Information) 프로토콜에 따라 상기 도메인 이름과 상기 IPv6 주소를 상기 DNS 서버에 등록하는 IoT 디바이스에 대한 보안성 있는 DNS 네이밍 방법.

**청구항 7**

제1항에 있어서,

상기 근거리 통신은 NFC(Near Field Communication), 와이파이, 블루투스 또는 Zigbee 중 적어도 하나인 IoT 디바이스에 대한 보안성 있는 DNS 네이밍 방법.

**청구항 8**

제1항에 있어서,

상기 도메인 이름은 상기 IoT 디바이스가 IPv6 프로토콜에 따른 RA(Router Advertisement) 옵션 또는 DHCP(Dynamic Host Configuration Protocol) 옵션을 통해 DNS 검색 리스트가 포함된 메시지를 수신하면 상기 IoT 디바이스가 자신의 모델 정보 및 위치 정보를 포함하여 생성하는 IoT 디바이스에 대한 보안성 있는 DNS 네이밍 방법.

**청구항 9**

제1항에 있어서,

상기 도메인 이름 수집장치는 IPv6 프로토콜에 따른 NI(Node Information) 질의를 상기 IoT 디바이스에 전송하고, 상기 NI 질의에 대한 NI 응답으로 상기 도메인 이름 및 상기 디지털 서명을 수신하는 IoT 디바이스에 대한 보안성 있는 DNS 네이밍 방법.

**청구항 10**

삭제

**청구항 11**

삭제

**청구항 12**

삭제

**청구항 13**

삭제

**청구항 14**

삭제

**청구항 15**

삭제

**청구항 16**

삭제

**청구항 17**

삭제

**청구항 18**

삭제

**청구항 19**

IoT(Internet of Things) 디바이스로부터 모델 정보가 포함된 도메인 이름, 상기 도메인 이름에 대응되는 IPv6

주소 및 인증 서버가 전달한 서명용 키인 제1 키로 상기 도메인 이름의 적어도 일부를 암호화한 디지털 서명을 수신하고, 상기 인증 서버로부터 수신한 검증용 키인 제2 키로 상기 디지털 서명을 복호하고, 복호 결과가 상기 도메인 이름과 동일한지 판단함으로써 상기 디지털 서명을 검증하고, 상기 검증에 성공하면 상기 IoT 디바이스로부터 수신한 상기 도메인 이름 및 상기 IPv6 주소를 DNS(Domain Name System) 서버에 등록하되,

상기 제1 키 및 상기 제2 키는 상기 인증 서버가 상기 IoT 디바이스에 대한 인증을 위해 생성되는 비대칭 키인, IoT 디바이스에 대한 보안성 있는 DNS 네이밍 등록을 수행하는 라우터 장치.

**청구항 20**

삭제

## 발명의 설명

### 기 술 분 야

[0001] 이하 설명하는 기술은 보안성을 고려한 IoT 디바이스의 DNS 네이밍 기법에 관한 것이다.

### 배 경 기 술

[0002] IoT(Internet of Things)는 최근 가장 주목받고 있는 인터넷 관련 기술이다. IoT를 이용한 서비스는 기본적으로 인터넷에 연결된 매우 많은 디바이스를 통해 일정한 서비스를 제공할 수 있다.

[0003] DNS(Domain Name System)는 인터넷에 연결된 개체에 대한 IP 주소와 도메인 이름을 관리한다. 종래 인터넷 환경에서는 도메인 이름을 사용자가 수동으로 설정하였다.

### 선행기술문헌

#### 특허문헌

[0004] (특허문헌 0001) 한국공개특허 제10-2015-0066475

## 발명의 내용

### 해결하려는 과제

[0005] IoT 환경에서 IoT 디바이스는 매우 많은 개수가 사용된다. 따라서 각 IoT 디바이스에 대한 도메인 이름을 일일이 설정하는 것은 매우 번거로운 일이 된다. mDNS(multicast Domain Name System)과 같이 도메인 이름을 자동으로 부여하기 위한 기술이 있다. 대표적으로 애플(Apple)이 개발한 Bonjour와 같은 프로토콜이 있다. mDNS는 기본적으로 멀티캐스트 기반으로 네트워크에 많은 트래픽을 발생시키고, 인터넷에 연결된 디바이스가 항상 깨어 있는 경우에만 네이밍 요청 메시지에 응답할 수 있다. 또한 애플의 로컬(Local) DNS 네이밍 서비스는 로컬에서 개인(Private) IP 주소로 맵핑되어 인터넷을 통해 원격 제어를 수행하기 어렵다.

[0006] 이하 설명하는 기술은 IoT 디바이스가 자동으로 생성한 도메인 이름을 DNS 서버에 등록하는 기법을 제공하고자 한다. 특히 이하 설명하는 기술은 사용자(Administrator)가 허용한 IoT 디바이스를 인증하여 인증이 성공한 IoT 디바이스의 도메인 이름만을 등록하게 하고자 한다.

### 과제의 해결 수단

[0007] IoT 디바이스에 대한 보안성 있는 DNS 네이밍 방법은 이동 단말이 제1 키를 인증 서버로부터 수신하고, 도메인 이름 수집장치가 제2 키를 인증 서버로부터 수신하는 단계, 상기 도메인 이름 수집장치가 IoT 디바이스로부터 모델 정보가 포함된 도메인 이름 및 서명을 수신하는 단계, 상기 도메인 이름 수집장치가 상기 디지털 서명을 상기 제2 키로 복호하여 상기 서명을 검증하는 단계 및 상기 도메인 이름 수집장치가 상기 검증이 성공하면 상기 IoT 디바이스에 대한 DNS 이름을 DNS 서버에 등록하는 단계를 포함한다.

[0008] 다른 측면에서 IoT 디바이스에 대한 보안성 있는 DNS 네이밍 방법은 이동 단말이 제1 키를 인증 서버로부터 수신하고, 도메인 이름 수집장치가 제2 키를 인증 서버로부터 수신하는 단계, 상기 이동 단말이 IoT 디바이스로부

터 모델 정보가 포함된 도메인 이름을 수신하는 단계, 상기 이동 단말이 상기 제1 키를 이용하여 서명을 생성하는 단계, 상기 도메인 이름 수집 장치가 상기 이동 단말로터 상기 도메인 이름 및 상기 서명을 수신하는 단계, 상기 도메인 이름 수집장치가 상기 서명을 상기 제2 키로 복호하여 상기 서명을 검증하는 단계 및 상기 도메인 이름 수집장치가 상기 검증이 성공하면 상기 IoT 디바이스에 대한 DNS 이름을 DNS 서버에 등록하는 단계를 포함한다. 상기 제1 키 및 상기 제2 키는 인증 서버가 상기 IoT 디바이스에 대한 인증을 위해 생성한 비대칭 키이다.

[0009] IoT 디바이스에 대한 보안성 있는 DNS 네이밍 등록을 수행하는 라우터 장치는 IoT 디바이스로부터 모델 정보가 포함된 도메인 이름 및 인증 서버가 전달한 제1 키로 암호화한 서명을 수신하고, 상기 인증 서버로부터 수신한 제2 키로 상기 서명을 복호하여 상기 서명을 검증하고, 상기 검증이 성공하면 상기 IoT 디바이스에 대한 DNS 이름을 DNS 서버에 등록한다. 상기 제1 키 및 상기 제2 키는 인증 서버가 상기 IoT 디바이스에 대한 인증을 위해 생성하는 비대칭 키이고, 상기 제1 키는 이동 단말이 상기 인증 서버로부터 수신한 것을 근거리 통신을 통해 상기 이동 단말로부터 상기 IoT 디바이스가 전달받은 것이다.

[0010] 다른 측면에서 IoT 디바이스에 대한 보안성 있는 DNS 네이밍 등록을 수행하는 라우터 장치는 이동 단말로부터 IoT 디바이스의 모델 정보가 포함된 도메인 이름 및 인증 서버가 전달한 제1 키로 암호화한 서명을 수신하고, 상기 인증 서버로부터 수신한 제2 키로 상기 서명을 복호하여 상기 서명을 검증하고, 상기 검증이 성공하면 상기 IoT 디바이스에 대한 DNS 이름을 DNS 서버에 등록한다. 상기 제1 키 및 상기 제2 키는 인증 서버가 상기 IoT 디바이스에 대한 인증을 위해 생성하는 비대칭 키이고, 상기 제1 키는 이동 단말이 상기 인증 서버로부터 전달받은 것이다.

## *발명의 효과*

[0011] 이하 설명하는 기술은 IoT 디바이스에 대한 모델 정보 등을 포함하는 도메인 이름을 제공하여 도메인 이름을 기반으로 다양한 서비스 제공이 가능하다. 또한 이하 설명하는 기술은 도메인 이름으로 식별되는 디바이스에 대한 원격 제어가 가능하다. 나아가 이하 설명하는 기술은 보안 요소를 추가하여 인증된 IoT 디바이스에 대한 도메인 등록을 수행한다.

## *도면의 간단한 설명*

[0012] 도 1은 종래 DNS의 구성을 도시한 블록도의 예이다.

도 2는 DNSNA의 구성을 도시한 블록도의 예이다.

도 3은 DNSNA에서 IoT 디바이스가 도메인 이름을 생성하는 과정에 대한 예이다.

도 4는 DNSNA에서 IoT 디바이스에 대한 도메인 이름을 등록하는 과정에 대한 예이다.

도 5는 DNSNA에서 클라이언트 장치가 IoT 디바이스를 제어하는 과정에 대한 예이다.

도 6은 댁내에서 사용하는 IoT 디바이스에 대한 DNS 이름에 대한 예이다.

도 7은 댁내에서 사용하는 IoT 디바이스에 대한 DNS 이름의 다른 예이다.

도 8은 댁내에서 사용하는 IoT 디바이스에 대한 DNS 이름의 또 다른 예이다.

도 9는 SDNSNA에서 도메인 이름을 등록하는 과정에 대한 예이다.

도 10은 SDNSNA에서 도메인 이름을 등록하는 과정에 대한 절차 흐름도의 예이다.

도 11은 SDNSNA에서 도메인 이름을 등록하는 과정에 대한 다른 예이다.

도 12는 SDNSNA에서 도메인 이름을 등록하는 과정에 대한 절차 흐름도의 다른 예이다.

도 13은 도메인 이름을 수신한 이동 단말의 화면을 도시한 예이다.

## *발명을 실시하기 위한 구체적인 내용*

[0013] 이하 설명하는 기술은 다양한 변경을 가할 수 있고 여러 가지 실시례를 가질 수 있는 바, 특정 실시례들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 이하 설명하는 기술을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 이하 설명하는 기술의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하

는 것으로 이해되어야 한다.

[0014] 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 해당 구성요소들은 상기 용어들에 의해 한정되지는 않으며, 단지 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 이하 설명하는 기술의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

[0015] 본 명세서에서 사용되는 용어에서 단수의 표현은 문맥상 명백하게 다르게 해석되지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함한다" 등의 용어는 설시된 특징, 개수, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 의미하는 것이지, 하나 또는 그 이상의 다른 특징들이나 개수, 단계 동작 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 배제하지 않는 것으로 이해되어야 한다.

[0016] 도면에 대한 상세한 설명을 하기에 앞서, 본 명세서에서의 구성부들에 대한 구분은 각 구성부가 담당하는 주기능 별로 구분한 것에 불과함을 명확히 하고자 한다. 즉, 이하에서 설명할 2개 이상의 구성부가 하나의 구성부로 합쳐지거나 또는 하나의 구성부가 보다 세분화된 기능별로 2개 이상으로 분화되어 구비될 수도 있다. 그리고 이하에서 설명할 구성부 각각은 자신이 담당하는 주기능 이외에도 다른 구성부가 담당하는 기능 중 일부 또는 전부의 기능을 추가적으로 수행할 수도 있으며, 구성부 각각이 담당하는 주기능 중 일부 기능이 다른 구성부에 의해 전담되어 수행될 수도 있음은 물론이다.

[0017] 또, 방법 또는 동작 방법을 수행함에 있어서, 상기 방법을 이루는 각 과정들은 문맥상 명백하게 특정 순서를 기재하지 않은 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 과정들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.

[0019] 이하 설명하는 기술은 보안성을 고려하여 DNS 이름을 자동으로 생성하고 관리하는 기법에 관한 것이다. 이하 설명하는 기술은 IPv6에 기반을 둔 IP 주소를 사용한다고 가정한다. 따라서 이하 설명하는 기술은 DNS 이름을 생성하고 등록하는 과정에서 IPv6 표준에서 재정된 프로토콜을 기반으로 설명한다. IPv6 표준에 정의된 프로토콜은 이미 공개된 것이므로 이하 자세한 설명은 생략한다. 다만 이하 설명하는 기술은 IPv4 등과 같은 프로토콜에도 적용이 가능하다.

[0021] 도 1은 종래 DNS(100)의 구성을 도시한 블록도의 예이다. 종래 DNS(Domain Name System)은 DNS 서버(160)가 핵심적인 구성에 해당한다. 도 1에서는 "www.sample.com"이라는 서비스를 제공하는 서비스 서버(110)를 예로 도시하였다. 서비스 제공자가 서비스 서버(110)를 이용하여 인터넷으로 다른 사용자에게 특정한 서비스를 제공하기 위해서는 서비스 서버(110)에 대한 도메인 이름을 등록해야 한다. 서비스 제공자는 컴퓨터 장치(50)를 통해 DNS 서버(160)에 접근하고, 자신의 서비스 서버(110)에 대한 도메인 이름을 등록한다. DNS 서버(160)는 널리 알려진 바와 같이 도메인 이름과 도메인 이름이 나타내는 실제 IP 주소를 저장하고 관리한다. 도 1에서 DNS 서버(160)는 서비스 서버(110)에 대한 도메인 이름(www.sample.com) 및 서비스 서버(110)에 대한 IP 주소(214.11.00.124)를 저장한다. 도 1은 IPv4를 사용한 예를 도시하였다. 라우터(150)는 DNS 서버(160) 및 서비스 서버(110)에 대한 데이터 전달을 제어하는 구성이다.

[0022] 이후 서비스를 제공받기 위한 사용자가 클라이언트 장치를 사용하여 도메인 이름으로 서비스 서버에 접속을 시도한다고 가정한다. 클라이언트 장치가 전달한 접속 명령은 DNS 서버(160)에서 도메인 이름에 해당하는 IP 주소를 찾아내고, 해당 IP 주소를 갖는 서비스 서버(110)에 클라이언트 장치가 접속되게 한다.

[0024] 그러나 IoT 환경에서는 수많은 IoT 디바이스에 대한 도메인 이름을 일일이 설정하기가 어려울 수 있다. 따라서 이하 설명하는 기술에서 IoT 디바이스(예, 센서, 가전제품, 신호등, 자동차 등)는 IPv6 네트워크에 참여할 때 자신의 디바이스 정보를 기반으로 DNS 이름을 생성한다. 이렇게 생성된 유일한 DNS 이름은 IoT 디바이스를 위한 DNS 서버에 등록된다. 이후 사용자는 클라이언트 장치(PC, 스마트폰, 태블릿 PC 등)를 이용해서 DNS 서버를 통해 IoT 디바이스에 대한 도메인 이름을 획득하여 IoT 디바이스를 원격 제어할 수 있다. 종래 DNS와는 달리 이름을 자동으로 설정한다는 의미에서 도메인 이름을 설정하고 등록하는 시스템을 이하 DNSNA(DNS Name Autoconfiguration)라고 명명한다.

[0025] DNSNA은 IoT 디바이스를 관리하기 위해 DNS 서버(DNS Server)를 이용하여 IoT 디바이스의 유일한 이름을 생성, 등록, 관리한다. IoT 디바이스가 IPv6 네트워크에 들어올 때 IoT 디바이스는 자신의 디바이스 정보를 기반으로 현재 IPv6 네트워크에 유일한 DNS 이름을 생성한다. 생성된 DNS 이름은 라우터(Router)에 등록되고 IoT 디바이스의 정보는 동적으로 DNS 서버에 등록한다. 사용자는 클라이언트 장치를 이용해 IoT 디바이스 정보를 DNS 서버

로부터 수신하여 IoT 디바이스를 원격제어를 할 수 있다.

[0027] 도 2는 DNSNA(200)의 구성을 도시한 블록도의 예이다. 도 2는 IoT 디바이스(210)로 댁내에서 사용하는 냉장고를 예로 도시하였다. IoT 디바이스는 일반적으로 무선 통신을 수행하므로, 도 2는 IoT 디바이스(210)가 댁내에 있는 AP(Access Point, 230)를 통해 네트워크에 연결된 형태를 도시하였다. 라우터(250)는 도 1과 같이 DNS 서버(260) 및 IoT 디바이스(210)에 전달되는 데이터 패킷을 제어한다.

[0028] IoT 디바이스(210)는 네트워크에 최초 접속하는 경우 또는 일정한 요청이 있는 경우 자신이 직접 DNS 이름을 생성한다. 도 2에서 IoT 디바이스(210)는 "refrigerator1.samsung_RH269LP.refrigerator.home"이라는 DNS 이름을 생성하였다. IoT 디바이스가 DNS 이름을 생성하고 등록하는 과정 및 DNS 이름이 의미하는 내용은 후술한다. IoT 디바이스(210)가 생성한 DNS 이름은 DNS 서버(260)에 저장된다. DNS 서버(260)는 IoT 디바이스(210)가 전달한 DNS 이름과 해당 DNS 이름이 의미하는 IP 주소를 저장한다. 도 2에서 IP 주소는 IPv6에 따른 주소이다. 도 2에 도시한 바와 같이 IPv6에 따른 IP 주소는 콜론으로 구분된 모두 8개의 세그먼트를 갖고, 각 세그먼트는 16bits로 표현된다. IPv6는 128bits를 사용하여 IP 주소를 표현한다.

[0029] 사용자는 클라이언트 장치(80)를 통해 DNS 서버(260)로부터 IoT 디바이스(210)에 대한 DNS 이름을 획득하고, IoT 디바이스(210)에 접근하여 냉장고의 상태를 확인할 수 있다. 또한 사용자는 IoT 디바이스(210)에 접근하여 냉장고를 제어할 수도 있다.

[0031] 도 3은 DNSNA(200)에서 IoT 디바이스가 도메인 이름을 생성하는 과정에 대한 예이다.

[0032] IoT 디바이스(210)는 라우터(250)를 통해 DNSSL(DNS Search list, DNS 검색 리스트) 옵션(option)을 수신한다(①). DNSSL은 IoT 디바이스가 도메인 이름을 등록하는 과정에서 사용하는 DNS 도메인 서픽스(suffix)의 리스트이다. DNSSL에는 복수의 DNS 서픽스가 있을 수 있다. DNSSL 옵션은 IPv6 표준에 따른 RA(Router Advertisement) 옵션 또는 DHCP(Dynamic Host Configuration Protocol) 옵션을 통해 전달된다. IPv6 표준에 따른 RA 옵션은 DNSSL 정보를 IPv6 호스트(IoT 디바이스)에 전달한다. 또한 DHCP 옵션도 DNSSL 정보를 IPv6 호스트(IoT 디바이스)에 전달한다.

[0033] IPv6 호스트인 IoT 디바이스(210)는 수신한 DNSSL 옵션이 유효한지 점검할 수 있다. DNSSL 옵션이 유효하다면 IoT 디바이스(210)는 자신의 DNS 이름을 생성한다(②). IoT 디바이스(210)는 생산 과정에서 부여되는 모델 이름 등과 같은 모델 정보를 보유한다. IoT 디바이스(210)는 모델 정보를 이용하여 DNS 이름을 생성할 수 있다. IoT 디바이스(210)는 DNS 도메인 서픽스에 자신의 디바이스 정보(예, Device Category, Vendor Name, Model Name)을 결합해서 자동으로 설정할 수 있다. 나아가 IoT 디바이스(210)는 DNS 이름은 디바이스 정보에 고유 식별자(unique_id)를 결합할 수 있다. 도 3에서 DNS 이름 중 "refrigerator1"은 동일한 모델의 디바이스를 구별하기 위한 고유 식별자이고, "samsung_RH269LP"는 제조사이름을 포함하는 모델 이름이고, "refrigerator"는 모델 카테고리이고, "home"은 DNS 도메인 서픽스에 해당한다.

[0034] IoT 디바이스(210)는 생성한 DNS 이름이 도메인 이름(subnet)에서 유일한 DNS 이름인지 여부를 확인하는 것이 바람직하다. IoT 디바이스(210)는 ND(Neighbor Discovery) 프로토콜에 따라 도메인 이름에 대한 중복 검사를 수행할 수 있다(③). IoT 디바이스(210)는 IPv6 멀티캐스트 주소(Multicast Address)로 DAD(Duplicate Address Detection)을 수행할 수 있다.

[0035] DNS 이름의 IPv6 멀티캐스트 주소는 네트워크 프리픽스(Prefix)에 해당하는 상위 64 비트를 위해 Link-Local Multicast Address Prefix인 ff02::/16를 이용하고, 네트워크 인터페이스(Network Interface) ID에 해당하는 하위 64 비트를 위해 DNS 이름에 대한 해싱값인 128 비트 중에서 MSB(Most Significant Bit)를 포함하는 상위 64비트를 취한다. IoT 디바이스(210)는 ND 프로토콜에 따라 DNS 이름에 해당하는 해싱값을 이용하여 DAD 과정을 수행한다.

[0036] 도메인 이름이 중복된 경우 IoT 디바이스(210)는 종전의 도메인 이름에서 디바이스의 고유 식별자를 새로운 식별자로 변경한 새로운 도메인 이름을 생성하고, 새로운 도메인 이름에 대한 중복 검사를 수행한다. IoT 디바이스(210)가 도메인 이름에서 중복되지 않은 도메인 이름을 생성했다고 확인해야 도메인 이름 생성과정이 종료된다.

[0037] 만약 DNSSL에 복수의 DNS 도메인 서픽스가 있는 경우 IoT 디바이스(210)는 모든 DNS 도메인 서픽스에 대해 각각 도메인 이름 생성 과정을 반복한다.

[0039] 도 4는 DNSNA(200)에서 IoT 디바이스에 대한 도메인 이름을 등록하는 과정에 대한 예이다. 도 3에서 IoT 디바이

스(210)는 자신의 DNS 이름을 생성하였고, 도메인 이름에서 중복되지 않는다는 것을 확인한 상태이다.

[0040] 네트워크에 연결된 도메인 이름 수집 장치가 IoT 디바이스(210)에 도메인 이름을 묻고(Query), 응답으로 수신한 IoT 디바이스(210)의 도메인 이름은 DNS 서버(260)에 저장하게 된다. 도메인 이름 수집 장치는 네트워크 구성 중 IoT 디바이스(210)의 도메인 이름에 대한 등록 과정을 제어한다. 일반적으로 도메인 이름 수집 장치는 IoT 디바이스(210)의 도메인 이름을 관리하는 장치일 수 있다. 도 4에서는 라우터(250)가 IoT 디바이스(210)에 대한 도메인 이름을 수집한다.

[0041] 라우터(250)은 먼저 IoT 디바이스(210)에 대해 DNS 이름을 요청한다. 라우터(250)는 NI(Node Information) 프로토콜을 이용하여 IoT 디바이스(210)에 DNS 이름을 묻는다(NI query,①). 이 과정은 종래 NI 프로토콜에서 새로운 NI query가 필요하다. 즉, NI 타입에 대한 새로운 코드가 정의되어야 한다. IoT 디바이스(210)는 NI query에 대한 응답으로 자신의 DNS 이름을 전달한다(NI reply,②).

[0042] 한편 NI query는 동일한 서브넷에 있는 IoT 디바이스에 멀티캐스트로 전달될 수 있다. 이 경우 도메인 이름에 있는 복수의 IoT 디바이스들은 서로 다른 시간에 NI reply를 하는 것이 바람직하다. 따라서 복수의 IoT 디바이스들은 각각 임의(Random)로 지연된 시간을 두고 NI reply를 전송할 수 있다.

[0043] 라우터(250)는 NI reply를 통해 수신한 IoT 디바이스(210)의 DNS 이름을 DNS 서버(260)에 전송하고, DNS 서버(260)는 IoT 디바이스(210)의 DNS 이름을 저장한다(③). 경우에 따라서 DNS 서버(260)에 이미 도메인 이름이 존재하는 IoT 디바이스가 있을 수 있다. 후술하겠지만 IoT 디바이스의 도메인 이름에 위치 정보 등이 들어갈 수 있기 때문에 IoT 디바이스가 이동하면 도메인 이름도 변경될 수 있다. 따라서 이 경우 DNS 서버(260)는 IoT 디바이스에 대한 DNS 이름을 새롭게 수신한 DNS 이름으로 업데이트한다. 라우터(250)는 IoT 디바이스(210)의 도메인 이름과 함께 IPv6의 IP 주소를 수집하여 DNS 서버(260)에 함께 저장한다.

[0045] 도 5는 DNSNA(200)에서 클라이언트 장치가 IoT 디바이스를 제어하는 과정에 대한 예이다. 도 5는 도4와 같이 DNS 서버(260)에 IoT 디바이스(210)에 대한 DNS 이름이 저장된 상태를 전제로 한다.

[0046] 사용자는 스마트폰과 같은 클라이언트 장치(80)를 통해 IoT 디바이스(210)에 대한 정보를 확인하거나, IoT 디바이스(210)를 제어할 수 있다. 클라이언트 장치(80)는 DNS 서버(260)에 접속하여 네트워크에 존재하는 IoT 디바이스(210)에 대한 DNS 이름을 획득한다(①). 도 5에서는 하나의 IoT 디바이스(210)만을 도시하였으나, 네트워크에는 다수의 IoT 디바이스들이 있을 수 있다. 이 경우 클라이언트 장치(80)는 DNS 서버(260)로부터 네트워크에 존재하는 다수의 IoT 디바이스들에 대한 DNS 이름을 획득할 수 있다. 클라이언트 장치(80)는 획득한 DNS 이름을 이용하여 IoT 디바이스(210)에 접근할 수 있다. 이후 클라이언트 장치(80)는 IoT 디바이스(210)인 냉장고의 정보를 확인하거나, 냉장고의 동작을 제어할 수 있다.

[0048] DNS 이름은 IoT 디바이스의 모델 이름, IoT 디바이스의 종류를 나타내는 모델 카테고리, 동일한 모델 이름을 갖는 IoT 디바이스 마다 서로 다른 디바이스 식별자 및 IoT 디바이스의 네트워크 도메인 중 적어도 하나를 포함할 수 있다. DNS 이름은 IoT 디바이스의 모델과 관련된 정보(모델 정보)를 DNS 이름에 갖는 것이 특징이다.

[0049] 도 6은 댁내에서 사용하는 IoT 디바이스에 대한 DNS 이름에 대한 예이다. 도 6에서 DNS 이름은 IoT 디바이스의 제품 종류(카테고리) 및 도메인 이름으로 구성된다. 도 6은 DNS 이름이 IoT 디바이스의 모델 정보 중 모델 카테고리에 해당하는 정보를 포함한 예이다. 도 6에서 도메인 이름은 "home"이고, 도메인 이름 앞에 위치한 정보가 모델 카테고리에 해당한다. 도 6에서 TV는 "tv.home", 냉장고는 "refrigerator.home", 세탁기는 "washer.home", 통풍팬(fan)은 "fan.home", 난방기구인 히터는 "heater.home", 컴퓨터는 "computer.home", 전화기는 "phone.home", 체중계는 "scale.home"이란 DNS 이름을 갖는다.

[0050] DNS 이름에 도메인 이름 외에 모델 카테고리가 포함되면, DNS 이름이 IoT 디바이스에 대한 정보를 제공하게 된다. 물론 DNS 이름에 포함된 모델 정보가 어떤 의미인지에 대해서는 사전에 정의되고, 공유되어야 할 것이다. 예컨대, 사용자는 자신의 스마트폰을 통해 DNS 서버에 접속하여 DNS 이름을 획득하면, 스마트폰은 사전에 정의된 규약에 따라 모델 카테고리를 기준으로 도 6과 같이 댁내(home)에 어떠한 종류의 IoT 디바이스가 있는지 표시할 수 있다.

[0052] DNS 이름은 아래의 표 1과 같은 형태를 가질 수 있다.

표 1

[0053]

| unique_id.device_model.device_category.domain_name |
|---|

[0055] unique_id는 도메인 이름에 존재하는 동일한 모델에 대해 서로 다른 고유 식별자를 의미한다. device_model은 IoT 디바이스를 제조한 제조자 또는 판매자가 제공하는 모델 이름을 말한다. 모델 이름에는 제품을 생산한 제조자의 이름이 포함될 수 있다. device_category는 IoT 디바이스의 모델 카테고리를 나타낸다. domain_name은 IoT 디바이스가 갖는 네트워크 도메인의 DNS 서픽스를 의미한다. 물론 DNS 이름을 구성하는 요소의 순서는 표1과 다를 수 있다. 모델 이름 및 모델 카테고리는 사전에 IoT 디바이스에 저장된 정보에 해당한다. 고유 식별자는 DNS 이름 생성 과정에서 IoT 디바이스가 생성하는 정보이다. DNS 서픽스는 DNSSL에 포함되어 전달된 정보이다.

[0056] 예컨대, 도 3에서 IoT 디바이스(210)는 "refrigerator1.samsung_RH269LP. refrigerator.home"이란 DNS 이름을 생성하였다. 이미 설명한 바와 같이 "refrigerator1"은 동일한 모델의 디바이스를 구별하기 위한 고유 식별자이고, "samsung_RH269LP"는 제조사이름을 포함하는 모델 이름이고, "refrigerator"는 모델 카테고리이고, "home"은 DNS 도메인 서픽스에 해당한다.

[0058] 한편 IoT 디바이스의 개체 수가 많을 경우 모델 정보만으로는 IoT 디바이스를 구분하기 어려울 수 있다. 이 경우 DNS 이름은 IoT 디바이스의 위치 정보라는 추가적인 정보를 포함할 수 있다. 장치의 구별 가능성과는 관계없이 DNS 이름에 포함되는 위치 정보는 새로운 정보를 사용자에게 제공할 수 있다. 위치 정보가 포함된 DNS 이름의 예는 아래의 표 2와 같다.

표 2

[0059]

| unique_id.device_model.device_category.location_of_device.domain_name |
|---|

[0060] 표 2의 DNS 이름은 표1의 DNS 이름에 location_of_device가 추가되었다. location_of_device는 IoT 디바이스의 위치 정보를 의미한다. 위치 정보는 IoT 디바이스가 사전에 미리 획득한 것을 전제로 한다.

[0061] 위치 정보는 다음과 같이 다양한 방법으로 수집이 가능하다. (1) IoT 디바이스가 GPS 센서와 같은 위치 추적 장치를 갖고 있다면 GPS 좌표와 같은 절대적인 좌표 정보를 사용할 수 있다. GPS 좌표 정보와 같이 위성 신호를 이용하는 경우 주로 실외에 배치되는 IoT 디바이스가 좌표를 획득할 수 있다. 또는 실내 통신 신호(WiFi 등)를 이용하여 실내에서 위치 측위를 하는 시스템이 있는 경우, IoT 디바이스는 실내에서의 위치 정보도 획득할 수 있다.

[0062] (2) IoT 디바이스가 가속도 센서, 지자기 센서 등을 내장하고 있다면 특정 기준 지점을 기준으로 IoT 디바이스가 이동하여 위치한 지점에 대한 상대적인 위치 정보를 생성할 수 있다. 예컨대, 댁내에 배치되는 IoT 디바이스는 댁내의 구조를 사전에 알고 있다는 전제하에 출입문과 같은 기준 위치에서 이동을 시작하여 배치되는 위치를 파악할 수 있다.

[0063] (3) 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. 또한 IoT 디바이스는 통신 가능한 거리에 다수의 위치 정보를 갖고 있는 장치가 3개 이상 있다면 서로 주고받는 신호의 세기를 이용하여, 자신이 주변 장치와의 사이에서 어떤 위치에 있는지 비교적 정확하게 파악할 수도 있다.

[0064] (4) IoT 디바이스에 대한 위치 정보를 획득하는 별도의 장치를 사용할 수도 있다. 예컨대, 사용자가 자신의 실외 또는 실내에서 위치를 측위할 수 있는 장치를 이용하여 근처에 배치된 IoT 디바이스에 IoT 디바이스의 위치 정보를 전달할 수 있다.

[0066] 도 7은 댁내에서 사용하는 IoT 디바이스에 대한 DNS 이름의 다른 예이다. 도 7은 댁내에서 사용하는 IoT 디바이

스가 모델 정보 및 위치 정보를 사용하여 DNS 이름을 생성한 예를 도시한다.

[0067] 도 7은 댁내에서의 각 공간(거실, 주방, 침실 등)을 알파벳(A, B, C, D, E 및 F)로 표시하였다. 각 구역에 있는 IoT 디바이스의 DNS 이름은 아래의 표 3과 같다.

표 3

[0069]

| 영역 | IoT 디바이스 | DNS 이름 |
|---|---|---|
| A | 에어컨 | air_conditioner1.samsung_AC64.air conditiner.bed_room1.home |
| B | 스마트TV | smart_tv1.samsung_HG32N.smart_tv.living_room.home |
| C | 에어컨 | air_conditioner2.samsung_AC64.air conditiner.bed_room2.home |
| D | 에어컨 | air_conditioner3.samsung_AC64.air conditiner.bed_room3.home |
| E | 냉장고 | refrigerator1.samsung.regrigerator.home |
| F | 세탁기 | washer.home |

[0070] F로 표시된 세탁실에 있는 세탁기의 DNS 이름은 모델 카테고리 및 도메인 이름 이름을 포함한다. E로 표시된 주방에 있는 냉장고의 DNS 이름은 고유 식별자,제조사만 표시된 모델 이름, 모델 카테고리 및 도메인 이름 이름을 포함한다. B로 표시된 거실에 있는 스마트 TV의 DNS 이름은 고유 식별자, 모델 이름, 모델 카테고리, 위치 정보(living_room) 및 도메인 이름 이름을 포함한다.

[0071] A, C 및 D로 표시된 침실에 있는 에어컨도 각각 고유 식별자, 모델 이름, 모델 카테고리, 위치 정보 및 도메인 이름 이름을 DNS 이름으로 갖는다. A, C 및 D로 표시된 침실에 있는 에어컨은 모델 이름이 동일하다. 따라서 고유 식별자가 각각 air conditioner1, air conditioner2 및 air conditioner3로 각각 다르다. 또한 각 에어컨이 위치한 위치 정보도 bed_room1, bed_room2 및 bed_room3으로 서로 다르다. 도 7과 같은 비교적 작은 영역에서 사용되는 IoT 디바이스라면 고유 식별자를 사용하지 않고, 위치 정보만으로도 서로 구분 가능할 수도 있다.

[0072] 사용자는 IoT 디바이스의 상태를 확인할 수 있고, 각 IoT 디바이스를 제어할 수도 있다. 도 7에서 사용자는 댁내에 있는 WiFi AP를 이용하여 인터넷을 통해 댁내에 배치된 IoT 디바이스에 접속할 수 있다.

[0074] 전술한 DNS 이름은 장치의 모델 이름 내지 모델 카테고리를 포함하였다. DNS 이름은 모델 이름 대신에 장치를 특정하기 위한 다른 정보를 포함할 수도 있다. 예컨대, DNS 이름은 아래의 표 4와 같이 모델 이름 내지 모델 카테고리 대신에 object_identifier를 포함할 수 있다. object_identifier는 국제표준화기구인 ISO/IEC이 객체에 대한 ID를 위해 ISO/IEC 9834-1:2012를 제정하였고, 사물 통신을 위한 사실 표준화 단체인 oneM2M에서 사물을 식별하는데 사용하자고 제안한 정보이다. object_identifier는 M2M 노드 ID(M2M node indication ID), 제조사 ID(Manufacturer ID), 모델 ID(Model ID), 시리얼 ID(Serial Number ID)를 포함한다. M2M 노드 ID는 M2M 노드를 관리하는 주체(예컨대, 국가, 조직 등)에 대한 식별자를 포함한다. object_identifier은 제조사 식별자, 모델 식별자 및 제품에 대한 시리얼 정보를 포함한다. 결국 object_identifier도 IoT 디바이스에 대한 모델 정보를 포함한다.

표 4

[0076]

| unique_id.object_identifier.domain_name |
|---|

[0077] unique_id는 동일한 도메인 서픽스(domain_name)에 존재하는 동일한 모델에 대해 서로 다른 고유 식별자를 의미한다. domain_name은 IoT 디바이스가 갖는 네트워크 도메인의 DNS 서픽스를 의미한다. 물론 DNS 이름을 구성하는 요소의 순서는 표1과 다를 수 있다.

[0079] 나아가 DNS 이름은 아래의 표 5와 같이 object_identifier와 함께 IoT 디바이스의 위치 정보를 더 포함할 수 있다.

표 5

[0081]

| unique_id.object_identifier.location_of_device.domain_name |
|---|

[0082] 표 5의 DNS 이름은 표4의 DNS 이름에 location_of_device가 추가되었다. location_of_device는 IoT 디바이스의 위치 정보를 의미한다. 위치 정보는 IoT 디바이스가 사전에 미리 획득한 것을 전제로 한다.

[0084] 도 8은 댁내에서 사용하는 IoT 디바이스에 대한 DNS 이름의 또 다른 예이다. 도 8은 댁내에서 사용하는 IoT 디바이스가 object_identifier 및 위치 정보를 사용하여 DNS 이름을 생성한 예를 도시한다. 도 8은 댁내에서의 각 공간(거실, 주방, 침실 등)을 알파벳(A, B, C, D, E 및 F)로 표시하였다. 각 구역에 있는 IoT 디바이스의 DNS 이름은 아래의 표 6과 같다.

표 6

[0085]

| 영역 | IoT 디바이스 | DNS 이름 |
|---|---|---|
| A | 에어컨 | air_conditioner1.object_identifier2.bed_room1.home |
| B | 스마트TV | smart_tv1.object_identifier1.living_room.home |
| C | 에어컨 | air_conditioner2.object_identifier3.bed_room2.home |
| D | 에어컨 | air_conditioner3.object_identifier4.bed_room3.home |
| E | 냉장고 | refrigerator1.object_identifier5.kitchen.home |
| F | 세탁기 | washer1.object_identifier6.laundry_room.home |

[0086] A, C 및 D로 표시된 침실에 있는 에어컨의 DNS 이름은 각각 고유 식별자, object_identifier, 위치 정보 및 도메인 서픽스(home)를 갖는다. A, C 및 D로 표시된 침실에 있는 에어컨은 모델 이름이 동일하다. 따라서 고육 식별자가 각각 air_conditioner1, air_conditioner2 및 air_conditioner3로 각각 다르다. 또한 각 에어컨이 위치한 위치 정보도 bed_room1, bed_room2 및 bed_room3으로 서로 다르다. 도 8과 같은 비교적 작은 영역에서 사용되는 IoT 디바이스라면 고유 식별자를 사용하지 않고, 위치 정보만으로도 서로 구분 가능할 수도 있다.

[0087] B로 표시된 영역에 있는 스마트 TV의 DNS 이름은 고유 식별자(smart_tv1), object_identifier, 위치 정보(living_room) 및 도메인 서픽스(home)를 포함한다. E로 표시된 영역에 있는 냉장고의 DNS 이름은 고유 식별자(refrigerator1), object_identifier, 위치 정보(living_room) 및 도메인 서픽스(home)를 포함한다. F로 표시된 영역에 있는 세탁기의 DNS 이름은 고유 식별자(washer1), object_identifier, 위치 정보(laundry_room) 및 도메인 서픽스(home)를 포함한다.

[0088] 사용자는 IoT 디바이스의 상태를 확인할 수 있고, 각 IoT 디바이스를 제어할 수도 있다. 도 8에서 사용자는 댁내에 있는 AP를 이용하여 인터넷을 통해 댁내에 배치된 IoT 디바이스에 접속할 수 있다. 또는 사용자는 댁내에 있는 AP를 통해 직접 근거리 통신을 통해 IoT 디바이스에 접속할 수 있다.

[0090] 이하 전술한 DNSNA에 기반하여 보안성을 갖는 도메인 이름 생성 및 등록 과정에 대해 설명한다. 이하 보안성을 갖는 DNSNA를 SDNSNA(Secure DNSNA)라고 명명한다.

[0091] 도 9는 SDNSNA(300)에서 도메인 이름을 등록하는 과정에 대한 예이다. SDNSNA(300)는 IoT 디바이스(310), 이동 단말(320), 인증 서버(340), 라우터(350) 및 DNS 서버(360)를 포함한다.

[0092] 인증 서버(340)는 IoT 디바이스(310)에 대한 인증을 수행하기 위한 비대칭키 쌍을 생성한다. SDNSNA는 기본적으로 디지털 서명(Digital Signature) 방식에 기반한 인증을 수행한다고 가정한다. 디지털 서명 방식은 DSA(Digital Signature Algorithm), RSA(Rivest-Shamir-Adleman) 방식, ElGamal 방식, Rabin 방식 등이 있다. 물론 SDNSNA는 다양한 방식의 다른 인증 기법을 사용할 수도 있다.

[0093] 이동 단말(320)은 보안성이 보장된 채널을 통해 인증 서버(340)에 접속한다. 예컨대, 이동 단말(320)은 인증 서버(340)과 TSL, SSH 등과 같은 기법으로 보안성이 보장된 채널을 수립한다. 이동 단말(320)은 IoT 디바이스(310)가 위치한 영역의 AP(330)를 통해 인증 서버(340)에 접속할 수 있다. 또는 이동 단말(320)은 별도의 이동 통신 네트워크를 통해 인증 서버(340)에 접속할 수도 있다. 이동 단말(320)은 인증 서버(340)가 생성한 키 중 비밀키에 해당하는 제1 키(서명용 키, Signing Key)를 전달받는다. 이동 단말(320)은 제1 키를 IoT 디바이스

(310)에 전달한다. 이동 단말(320)은 근거리 통신을 통해 제1 키를 IoT 디바이스(310)에 전달한다. 즉 사용자가 이동 단말(320)을 소지하고, IoT 디바이스(310)와 통신 가능한 거리에 접근하여 제1 키를 IoT 디바이스(310)에 전달한다.

[0094] 도 9에서 IoT 디바이스(310)는 일 예로 냉장고를 도시하였다. 전술한 바와 같이 IoT 디바이스(310)는 도메인 이름을 생성한다. 추가적으로 IoT 디바이스(310)는 수신한 제1 키를 이용하여 디지털 서명을 생성한다. IoT 디바이스(310)는 도메인 이름, 도메인 이름에 대응하는 IPv6 주소 및 디지털 서명을 라우터(350)에 전달한다.

[0095] 라우터(350)는 인증 서버(340)로부터 공용키에 해당하는 제2 키(검증용 키, Verification Key)를 수신한다. 라우터(350)는 제2 키를 이용하여 IoT 디바이스(310)로부터 수신한 서명을 검증한다. 라우터(350)는 전술한 도메인 이름 수집 장치에 해당한다.

[0096] SDNSNA에서 이용하는 인증 방식에 따라 구체적인 알고리즘이 상이할 수 있다. 디지털 서명 방식을 기준으로 설명하면 (1) IoT 디바이스(310)는 도메인 이름 전체 또는 일부를 제1 키로 암호화하여 서명을 생성한다. 라우터(350)는 제2 키로 서명을 복호하고, 복호한 메시지와 IoT 디바이스(310)로부터 수신한 도메인 이름이 일치하는지 확인한다. 라우터(350)는 복호한 메시지와 도메인 이름이 동일하면 검증이 성공했다고 판단한다. (2) 또는 IoT 디바이스(310)는 도메인 이름 전체 또는 일부를 해싱한 값을 제1 키로 암호화하여 서명을 생성한다. 라우터(350)는 IoT 디바이스(310)에서 사용한 동일한 해싱 함수로 수신한 도메인 이름 전체 또는 일부를 해싱한다. 라우터(350)는 제2 키로 서명을 복호하고, 복호한 값과 사전에 도메인 이름을 해싱한 값이 일치하는지 확인한다. 라우터(350)는 복호한 값과 도메인 이름을 해싱한 값이 동일하면 검증이 성공했다고 판단한다.

[0097] 라우터(350)는 서명에 대한 검증이 성공하면 해당 도메인 이름 및 IPv6 주소를 DNS 서버(360)에 등록한다.

[0099] 도 10은 SDNSNA에서 도메인 이름을 등록하는 과정(400)에 대한 절차 흐름도의 예이다.

[0100] 이동 단말(320)은 인증 서버(340)와 보안 채널을 수립한다(401). 이동 단말(320)은 인증 서버(340)에 IoT 디바이스(310)의 DNS 이름 등록을 위한 인증 정보를 요청한다(411). 인증 서버(340)는 전술한 비대칭키 쌍(제1 키 및 제2 키)를 생성한다(412). 인증 서버(340)는 생성한 제1 키(서명용 키)를 이동 단말(320)에 전달한다(413). 또한 인증 서버(340)는 생성한 제2 키(검증용 키)를 라우터(350)에 전달한다(414).

[0101] 이동 단말(320)은 근거리 통신으로 제1 키를 IoT 디바이스(310)에 전달한다. 근거리 통신은 IEEE 802.11 기반의 통신, IEEE 802.15 기반의 통신 기법 등을 포함한다. 예컨대, 근거리 통신은 NFC, 와이파이, 블루투스 또는 Zigbee 중 적어도 하나에 해당할 수 있다.

[0102] IoT 디바이스(310)는 자신의 DNS 네임을 생성하기 위해서 라우터(350)로부터 IPv6 프로토콜에 따른 RA 옵션 또는 DHCP 옵션을 통해 DNS 검색 리스트가 포함된 메시지를 수신하여 자신의 디바이스 정보와 수신한 DNS 검색 리스트에 있는 DNS 서픽스를 결합하여 DNS 이름을 생성할 수 있다. IoT 디바이스(310)는 수신한 제1 키를 이용하여 전술한 바와 같이 DNS 등록 메시지에 대해 디지털 서명을 생성할 수 있다. 즉, IoT 디바이스(310)는 DNS 이름과 IPv6 주소를 포함한 DNS 등록 메시지에 대해 디지털 서명을 생성한다(431).

[0103] 한편 IoT 디바이스(350)는 IPv6 프로토콜의 ND 프로토콜에 따라 상기 도메인 이름에 대한 중복 검사를 수행할 수 있다. 도메인 이름이 중복된 경우 전술한 바와 같이 IoT 디바이스(350)는 새로운 식별자를 포함하는 새로운 도메인 이름을 생성할 수 있다.

[0104] 이후 라우터(350)는 NI 프로토콜을 이용하여 IoT 디바이스(310)에 DNS 이름을 묻는다(441). 이 과정은 종래 NI 프로토콜에서 새로운 NI query가 필요하다. 즉, NI 타입에 대한 새로운 코드가 정의되어야 한다. IoT 디바이스(310)는 NI query에 대한 응답으로 자신의 DNS 이름가 IPv6 주소를 포함하는 NI reply를 생성하고, 이 메시지에 대해 디지털 서명을 생성한 뒤 NI reply를 라우터(350) 전달한다(442).

[0105] 한편 NI query는 동일한 서브넷에 있는 IoT 디바이스에 멀티캐스트로 전달될 수 있다. 이 경우 도메인 이름에 있는 복수의 IoT 디바이스들은 서로 다른 시간에 NI reply를 하는 것이 바람직하다. 따라서 복수의 IoT 디바이스들은 각각 임의로 지연된 시간을 두고 NI reply를 전송할 수 있다.

[0106] 라우터(350)는 인증 서버(340)로부터 수신한 제2 키를 이용하여 IoT 디바이스(310)가 전달한 디지털 서명을 검증한다(451). 서명에 대한 검증이 성공하면 라우터(350)는 IoT 디바이스(310)의 도메인 이름과 함께 IPv6의 IP 주소를 DNS 서버(360)에 저장한다(461).

[0107] 한편 IoT 디바이스(310)는 이동할 수 있다. 예컨대, IoT 디바이스(310)가 차량일 경우 위치가 변경될 수 있다.

이 경우 도메인 이름이 변경될 수 있다. 전술한 바와 같이 도메인 이름에 위치 정보 등이 포함될 수 있기 때문이다. 종래 DNS 서버(360)에 도메인 이름이 등록되어 있는 IoT 디바이스인 경우, 라우터(350)는 새로운 도메인 이름을 업데이트(DNS 업데이트)한다.

[0110] 도 11은 SDNSNA(500)에서 도메인 이름을 등록하는 과정에 대한 다른 예이다. 즉 이동단말이 다수의 IoT 디바이스들의 DNS 정보를 라우터 대신에 이동 라우터로서 수집하고 라우터에게 요청하여 DNS 서버에 해당 정보들을 일괄적으로 등록하는 과정을 기술하고자 한다.

[0111] SDNSNA(500)는 IoT 디바이스(510), 이동 단말(520), 인증 서버(540), 라우터(550) 및 DNS 서버(560)를 포함한다. 도 11은 복수의 IoT 디바이스(510A, 510B 및 510C)를 도시한다.

[0112] 인증 서버(540)는 IoT 디바이스(510)에 대한 인증을 수행하기 위한 비대칭키 쌍을 생성한다. SDNSNA는 기본적으로 디지털 서명 방식에 기반한 인증을 수행한다고 가정한다.

[0113] 이동 단말(520)은 보안성이 보장된 채널을 통해 인증 서버(540)에 접속한다. 예컨대, 이동 단말(520)은 인증 서버(340)과 TSL, SSH 등과 같은 기법으로 보안성이 보장된 채널을 수립한다. 이동 단말(520)은 IoT 디바이스(510)가 위치한 영역의 AP(530)을 통해 인증 서버(540)에 접속할 수 있다. 또는 이동 단말(520)은 별도의 이동통신 네트워크를 통해 인증 서버(540)에 접속할 수도 있다. 이동 단말(520)은 인증 서버(540)가 생성한 키 중 비밀키에 해당하는 제1 키(서명용 키)를 전달받는다.

[0114] 이동 단말(520)은 근거리 통신을 통해 IoT 디바이스(510)로부터 도메인 이름을 수집한다. 이동 단말(520)은 NI 질의 메시지를 IoT 디바이스(510)에 전달하여 도메인 이름을 요청한다. 도 11에서 IoT 디바이스(510)는 일 예로 냉장고(510A), 세탁기(510B) 및 TV(510C)를 도시하였다. 전술한 바와 같이 IoT 디바이스(510)는 도메인 이름을 생성한다. IoT 디바이스(310)는 NI 응답(NI reply)으로 생성한 도메인 이름 및 도메인 이름에 대응하는 IPv6 주소를 이동 단말(520)에 전달한다. 이동 단말(520)은 제1 키를 이용하여 도메인 이름을 등록하고자 하는 특정 IoT 디바이스에 대한 디지털 서명을 생성한다.

[0115] 라우터(550)는 NI 프로토콜을 이용하여 이동 단말(520)에게 DNS 이름을 요청한다. 이 과정은 종래 NI 프로토콜에서 새로운 NI query가 필요하다. 즉, NI 타입에 대한 새로운 코드가 정의되어야 한다. 이동 단말(520)은 NI query에 대한 응답으로 특정 IoT 디바이스에 대한 DNS 이름, IPv6 주소 및 디지털 서명을 라우터(550)에 전달한다.

[0116] 라우터(550)는 인증 서버(540)로부터 공용키에 해당하는 제2 키(검증용 키)를 수신한다. 라우터(550)는 제2 키를 이용하여 이동 단말(520)로부터 수신한 서명을 검증한다. 라우터(550)는 서명에 대한 검증이 성공하면 해당 도메인 이름 및 IPv6 주소를 DNS 서버(660)에 등록한다.

[0118] 도 12는 SDNSNA에서 도메인 이름을 등록하는 과정(600)에 대한 절차 흐름도의 다른 예이다.

[0119] 이동 단말(520)은 인증 서버(540)와 보안 채널을 수립한다(601). 이동 단말(520)은 인증 서버(540)에 IoT 디바이스(510)의 DNS 이름 등록을 위한 인증 정보를 요청한다(611). 인증 서버(540)는 전술한 비대칭키 쌍(제1 키 및 제2 키)를 생성한다(612). 인증 서버(540)는 생성한 제1 키(서명용 키)를 이동 단말(520)에 전달한다(613). 또한 인증 서버(540)는 생성한 제2 키(검증용 키)를 라우터(550)에 전달한다(614).

[0120] IoT 디바이스(510)는 라우터(550)로부터 IPv6 프로토콜에 따른 RA 옵션 또는 DHCP 옵션을 통해 DNS 검색 리스트가 포함된 메시지를 수신하면 DNS 이름을 생성할 수 있다. IoT 디바이스(510)는 DNS 이름을 생성한다(621).

[0121] 이동 단말(520)은 IoT 디바이스의 DNS 네임 수집을 위한 이동 라우터로서 근거리 통신으로 IoT 디바이스(510)에 NI 질의로 도메인 이름을 요청한다(631). 이 과정은 종래 NI 프로토콜에서 새로운 NI query가 필요하다. 즉, NI 타입에 대한 새로운 코드가 정의되어야 한다. 한편 NI query는 동일한 서브넷에 있는 IoT 디바이스에 멀티캐스트로 전달될 수 있다. 이 경우 도메인 이름에 있는 복수의 IoT 디바이스(510)들은 서로 다른 시간에 NI reply를 하는 것이 바람직하다. 따라서 복수의 IoT 디바이스들은 각각 임의로 지연된 시간을 두고 NI reply를 전송할 수 있다. IoT 디바이스(310)는 NI query에 대한 응답으로 자신의 DNS 이름, IPv6 주소 및 디지털 서명을 전달한다(442).

[0122] 근거리 통신은 IEEE 802.11 기반의 통신, IEEE 802.15 기반의 통신 기법 등을 포함한다. 예컨대, 근거리 통신은 NFC, 와이파이, 블루투스 또는 Zigbee 중 적어도 하나에 해당할 수 있다. IoT 디바이스(510)는 NI 응답으로 생성한 DNS 이름 및 IPv6 주소를 이동 단말(520)에 전달한다(632).

[0123] 라우터(350)는 NI 프로토콜을 이용하여 이동 단말(520)에 DNS 이름을 묻는다(641). 이 과정은 종래 NI 프로토콜에서 새로운 NI query가 필요하다. 즉, NI 타입에 대한 새로운 코드가 정의되어야 한다. 이동 단말(520)은 요청받은 또는 사용자로부터 선택된 특정 IoT 디바이스에 대한 디지털 서명을 생성한다(642). 이동 단말(520)는 인증 서버(540)로부터 수신한 제1 키를 이용하여 특정 도메인 이름을 기준으로 디지털 서명을 생성한다. 이동 단말(520)은 NI query에 대한 응답으로 특정 IoT 디바이스에 대한 DNS 이름, IPv6 주소 및 디지털 서명을 라우터(550)에 전달한다(643).

[0124] 라우터(550)는 인증 서버(540)로부터 수신한 제2 키를 이용하여 이동 단말(520)이 전달한 디지털 서명을 검증한다(651). 서명에 대한 검증이 성공하면 라우터(550)는 IoT 디바이스(510)의 도메인 이름과 함께 IPv6의 IP 주소를 DNS 서버(560)에 저장한다(661).

[0125] 한편 IoT 디바이스(510)는 이동할 수 있다. 예컨대, IoT 디바이스(510)가 차량일 경우 위치가 변경될 수 있다. 이 경우 도메인 이름이 변경될 수 있다. 전술한 바와 같이 도메인 이름에 위치 정보 등이 포함될 수 있기 때문이다. 종래 DNS 서버(560)에 도메인 이름이 등록되어 있는 IoT 디바이스인 경우, 라우터(550)는 새로운 도메인 이름을 업데이트(DNS 업데이트)한다.

[0126] 특정 IoT 디바이스에 대한 도메인 이름이 중복되는 경우 라우터(550)는 DNS 서버(560)로부터 도메인 이름 등록 실패 메시지를 수신할 수 있다(671). 이 경우 라우터(550)는 도메인 이름 등록 실패에 대한 정보를 NI 질의로 이동 단말(520) 및/또는 IoT 디바이스(510)에 전달할 수 있다(672, 673). 등록 실패 메시지를 수신한 이동 단말(520)은 다시 IoT 디바이스(510)에 도메인 이름을 요청할 수 있고, IoT 디바이스(510)는 자신의 식별자를 변경한 새로운 도메인 이름을 이동 단말(520)에 전송할 수 있다. 이후 전술한 과정을 거쳐 도메인 이름 등록을 수행하게 된다.

[0128] 이동 단말(520)은 복수의 IoT 디바이스(510)에 대해 NI 질의로 도메인 이름 생성을 요청할 수 있다. 이 경우 이동 단말(520)은 복수의 IoT 디바이스(510)에 대한 도메인 이름 목록을 저장하고 관리할 수 있다. 도 13은 도메인 이름을 수신한 이동 단말의 화면을 도시한 예이다. 도 13은 이동 단말(520)의 화면에 3개의 도메인 이름을 출력하였다. 라우터(550)가 도메인 이름 등록을 위한 요청을 한 경우 사용자는 이동 단말(520)에 표시된 목록 중 특정 도메인 이름을 선택하여 이후 도메인 이름 등록 절차를 진행할 수 있다. 도메인 이름에 모델 정보, 위치 정보 등이 포함되기 때문에 사용자는 특정한 IoT 디바이스를 식별할 수 있다. 도 13은 사용자가 거실(living room)에 있는 스마트 TV(smart tv)를 선택한 예를 도시한다(굵은 실선으로 표시하였음). 이후 이동 단말(520)은 선택된 도메인 이름에 대한 디지털 서명을 생성한다. 이동 단말(520)은 선택된 도메인 이름, IPv6 주소 및 생성한 디지털 서명을 라우터(550)에 전달한다.

[0130] 본 실시례 및 본 명세서에 첨부된 도면은 전술한 기술에 포함되는 기술적 사상의 일부를 명확하게 나타내고 있는 것에 불과하며, 전술한 기술의 명세서 및 도면에 포함된 기술적 사상의 범위 내에서 당업자가 용이하게 유추할 수 있는 변형 예와 구체적인 실시례는 모두 전술한 기술의 권리범위에 포함되는 것이 자명하다고 할 것이다.

## *부호의 설명*

[0131] 50 : 컴퓨터 장치

100 : DNS

110 : 서비스 서버

150 : 라우터

160 : DNS 서버

80 : 클라이언트 장치

200 : DNSNA

210 : IoT 디바이스

230 : AP

250 : 라우터

260 : DNS 서버

300 : SDNSNA

310 : IoT 디바이스

320 : 이동 단말

330 : AP

340 : 인증 서버

350 : 라우터

360 : DNS 서버

500 : SDNSNA

510 : IoT 디바이스

520 : 이동 단말

530 : AP

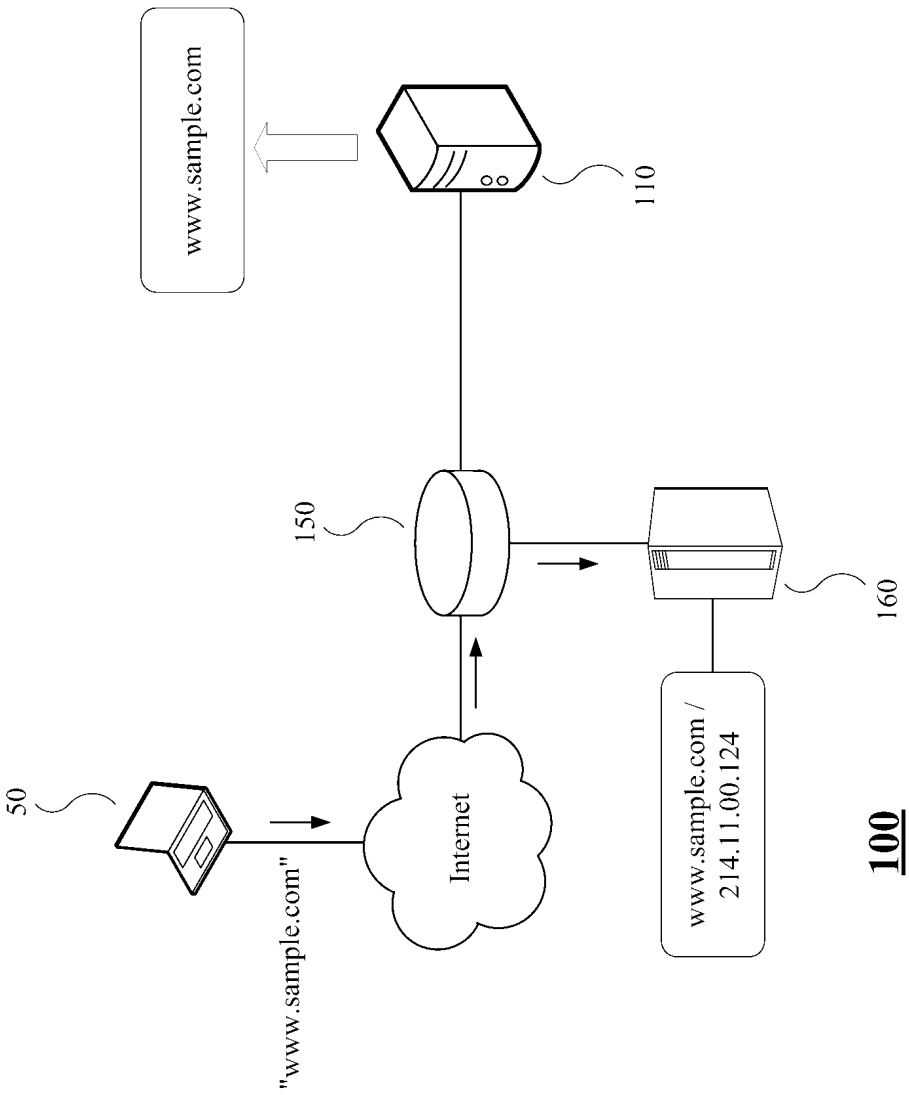
540 : 인증 서버

550 : 라우터

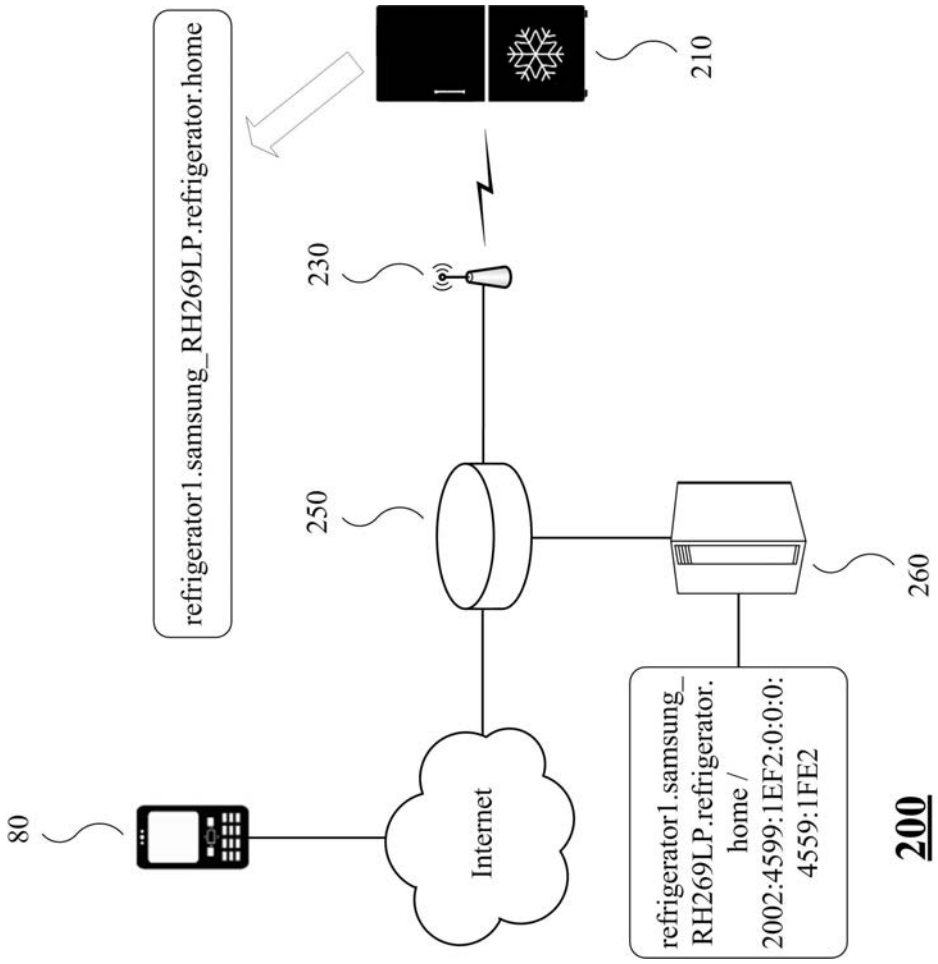
560 : DNS 서버

도면

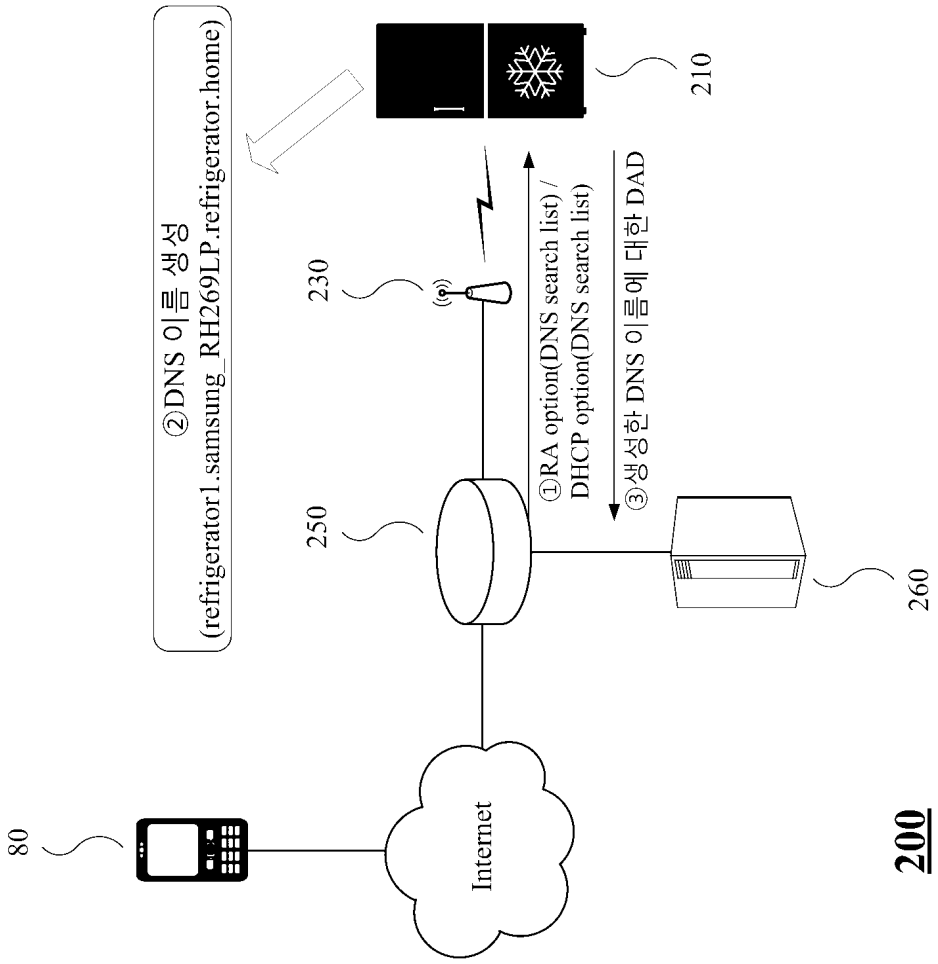
도면1


www.sample.com
110
150
160
50
Internet
"www.sample.com"
www.sample.com /
214.11.00.124
100

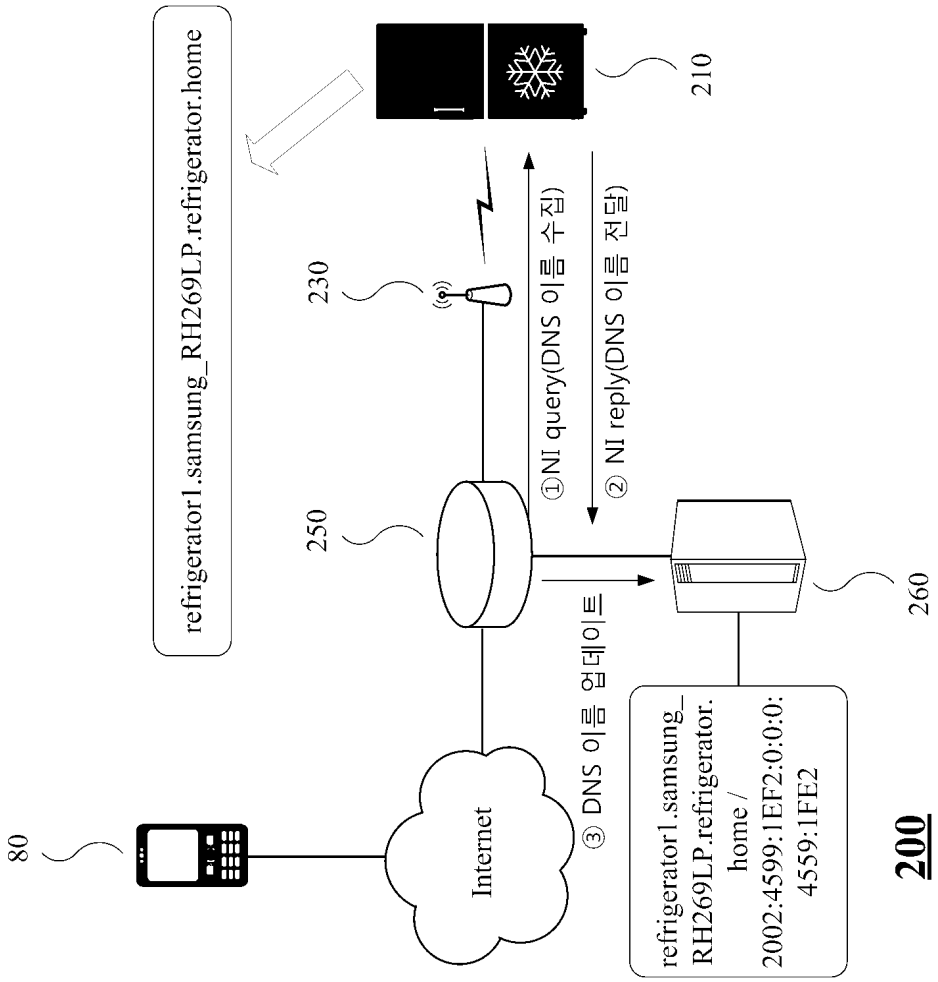
도면2


refrigerator1.samsung_RH269LP.refrigerator.home
210
230
250
260
80
Internet
refrigerator1.samsung_
RH269LP.refrigerator.
home /
2002:4599:1EF2:0:0:0:
4559:1FE2
200

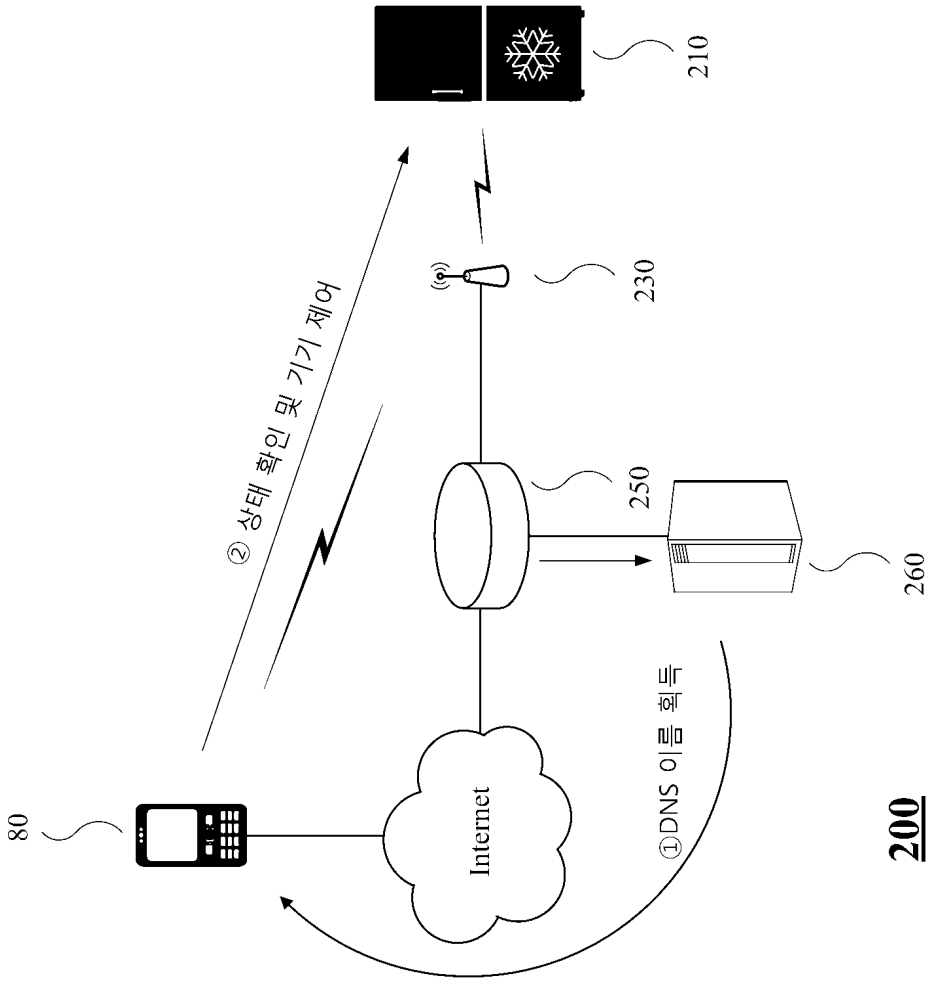
도면3


②DNS 이름 생성
(refrigerator1.samsung_RH269LP.refrigerator.home)
210
230
250
①RA option(DNS search list) /
DHCP option(DNS search list)
③생성한 DNS 이름에 대한 DAD
260
80
Internet
200

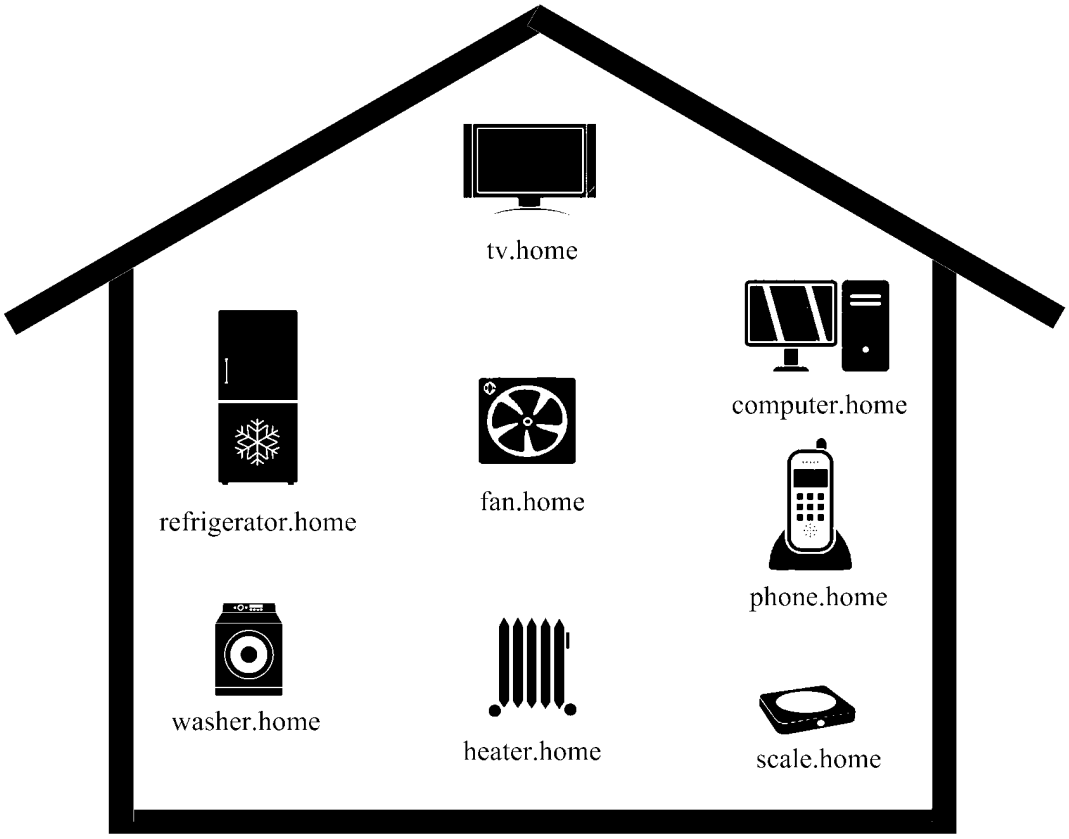
도면4


refrigerator1.samsung_RH269LP.refrigerator.home
210
230
250
260
80
Internet
① NI query(DNS 이름 수집)
② NI reply(DNS 이름 전달)
③ DNS 이름 업데이트
refrigerator1.samsung_RH269LP.refrigerator.home / 2002:4599:1EF2:0:0:0:4559:1FE2
200

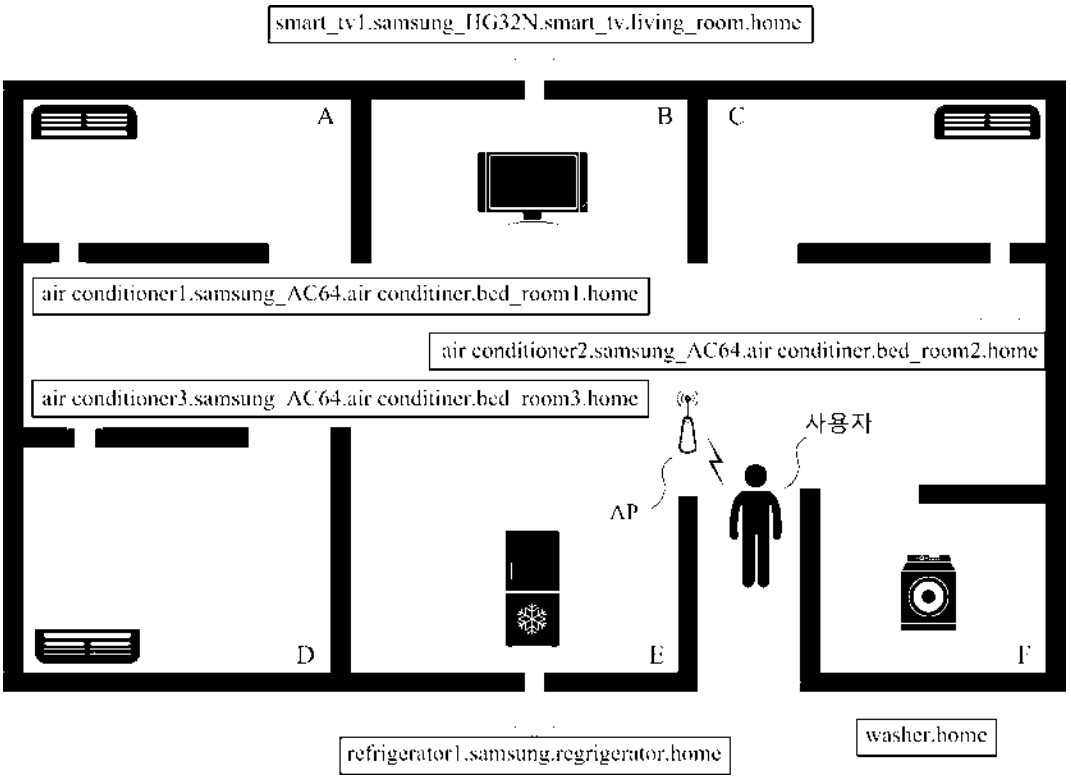
도면5



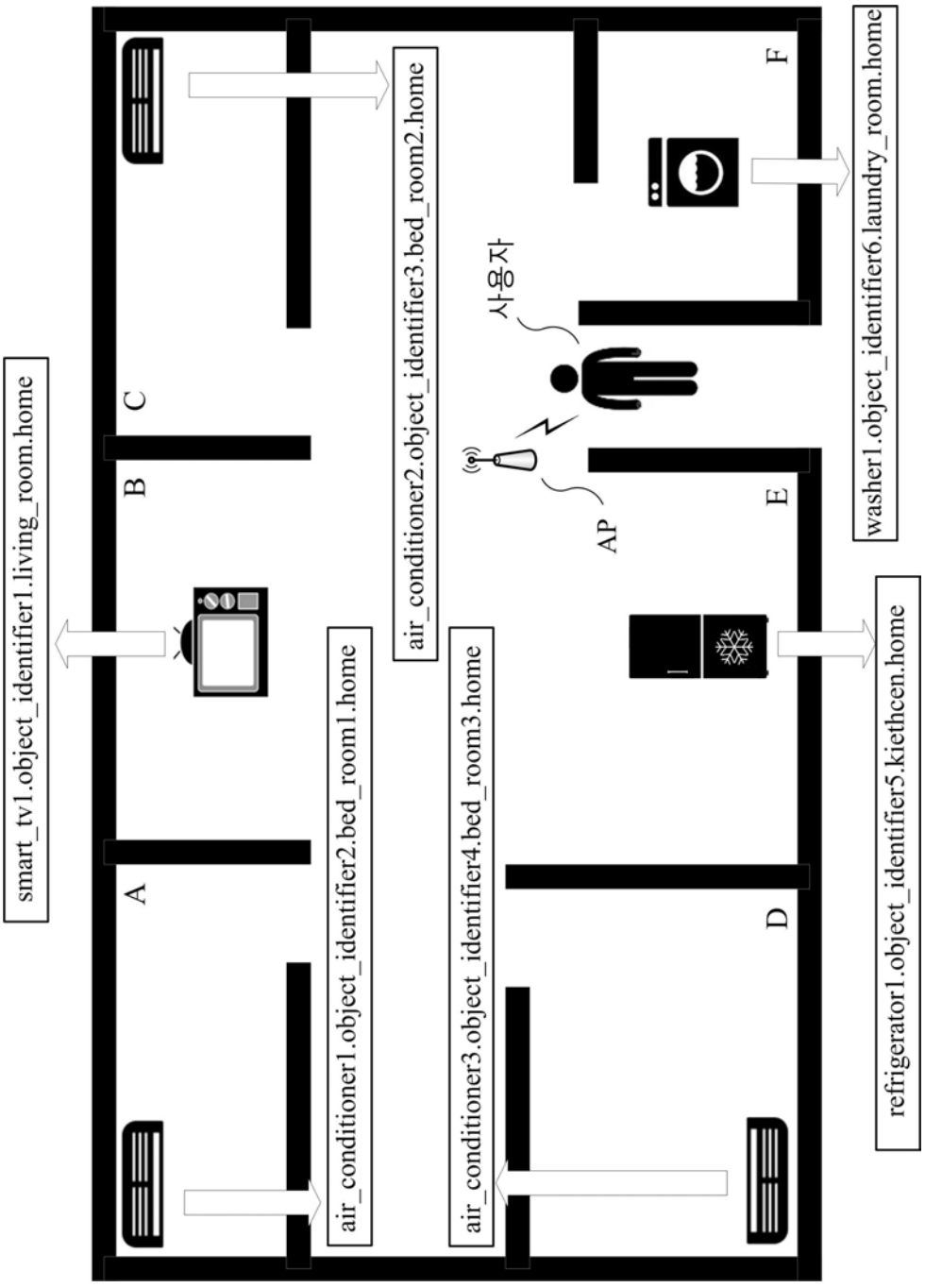
도면6


tv.home
computer.home
refrigerator.home
fan.home
phone.home
washer.home
heater.home
scale.home

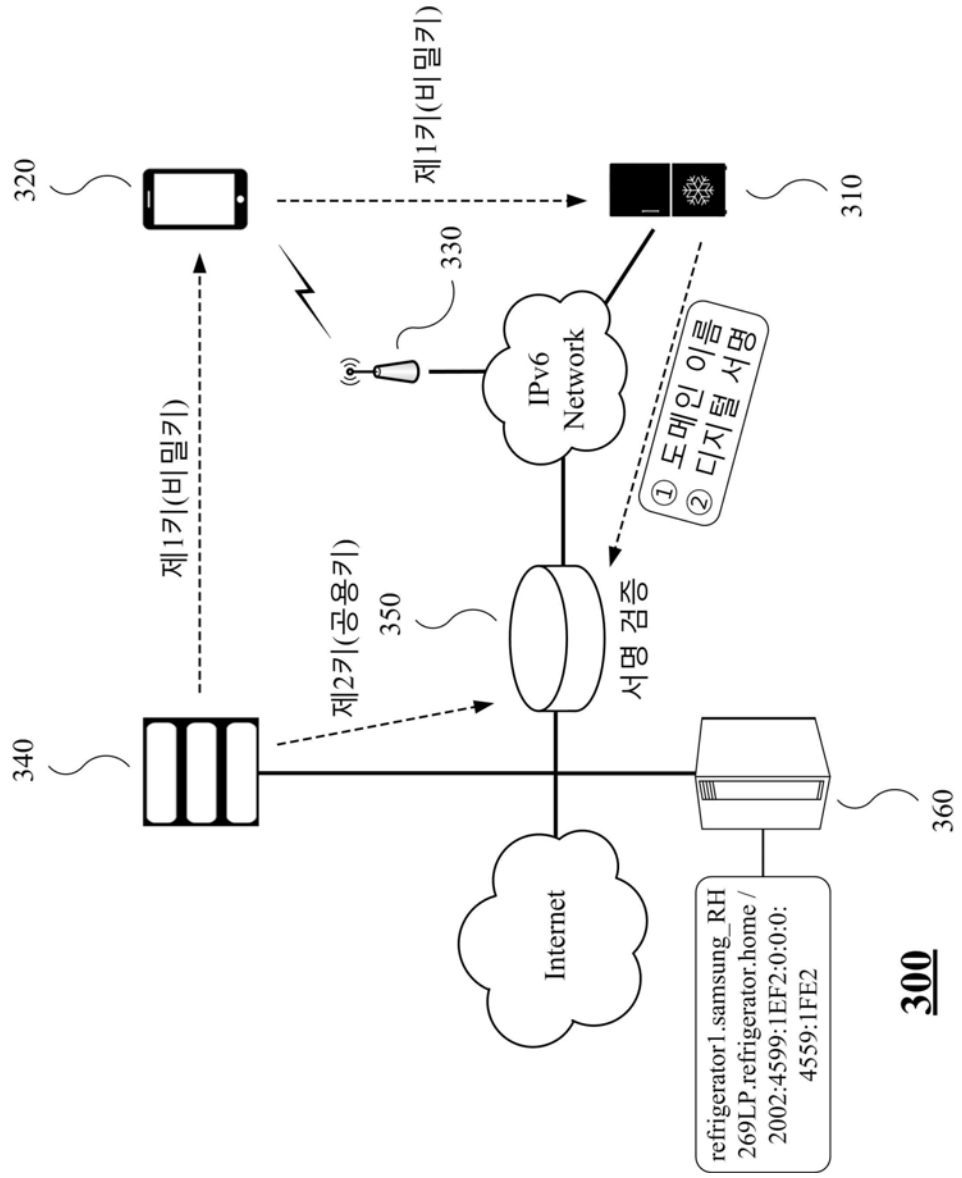
도면7


smart_tv1.samsung_HG32N.smart_tv.living_room.home
A
B
C
air conditioner1.samsung_AC64.air conditiner.bed_room1.home
air conditioner2.samsung_AC64.air conditiner.bed_room2.home
air conditioner3.samsung_AC64.air conditiner.bed_room3.home
사용자
AP
D
E
F
refrigerator1.samsung.regrigerator.home
washer.home

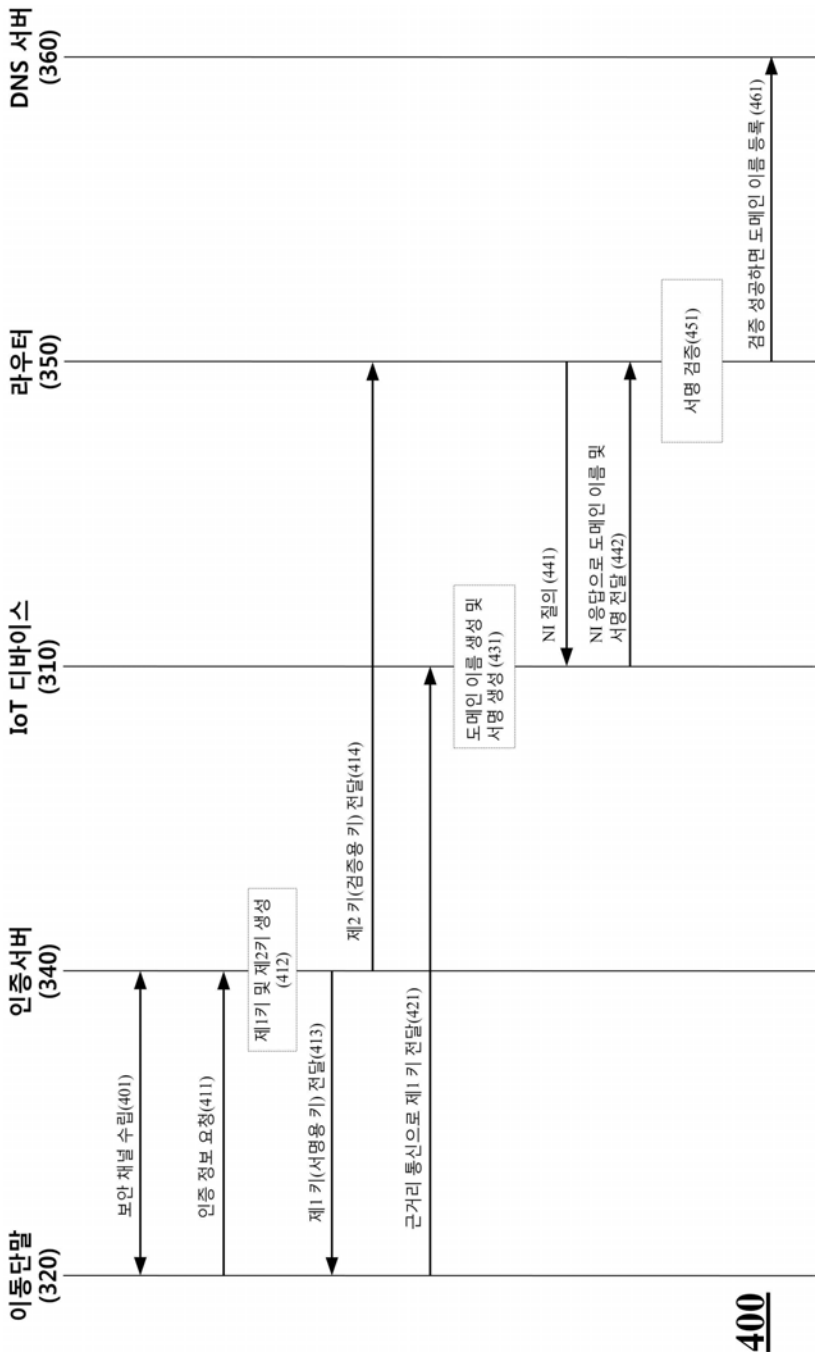
도면8


smart_tv1.object_identifier1.living_room.home
air_conditioner1.object_identifier2.bed_room1.home
air_conditioner2.object_identifier3.bed_room2.home
air_conditioner3.object_identifier4.bed_room3.home
refrigerator1.object_identifier5.kiethcen.home
washer1.object_identifier6.laundry_room.home
A
B
C
D
E
F
AP
사용자

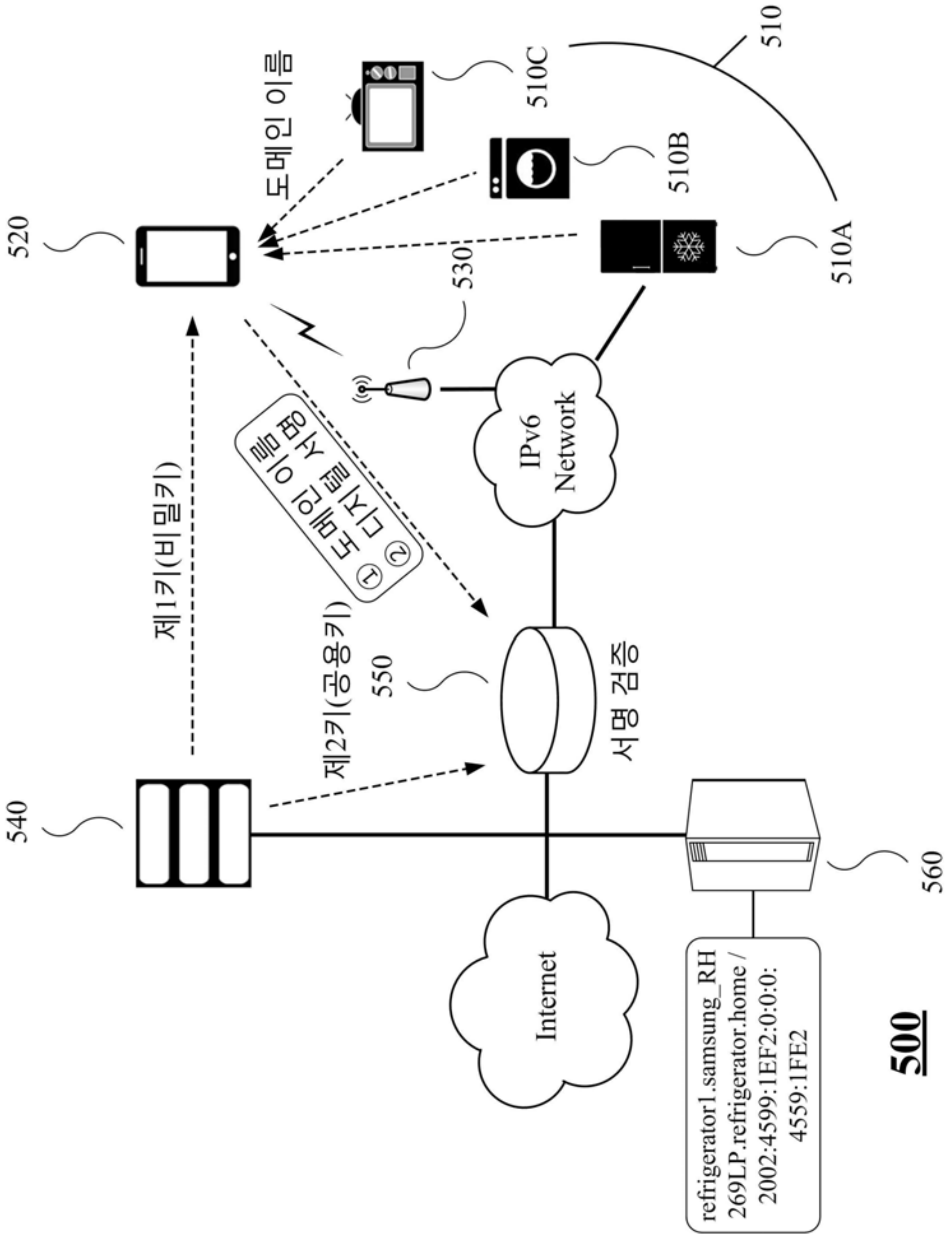
도면9


320
제1키(비밀키)
310
330
IPv6 Network
① 도메인 이름
② 디지털 서명
제1키(비밀키)
제2키(공용키)
350
서명 검증
340
360
Internet
refrigerator1.samsung_RH269LP.refrigerator.home / 2002:4599:1EF2:0:0:0:4559:1FE2
300

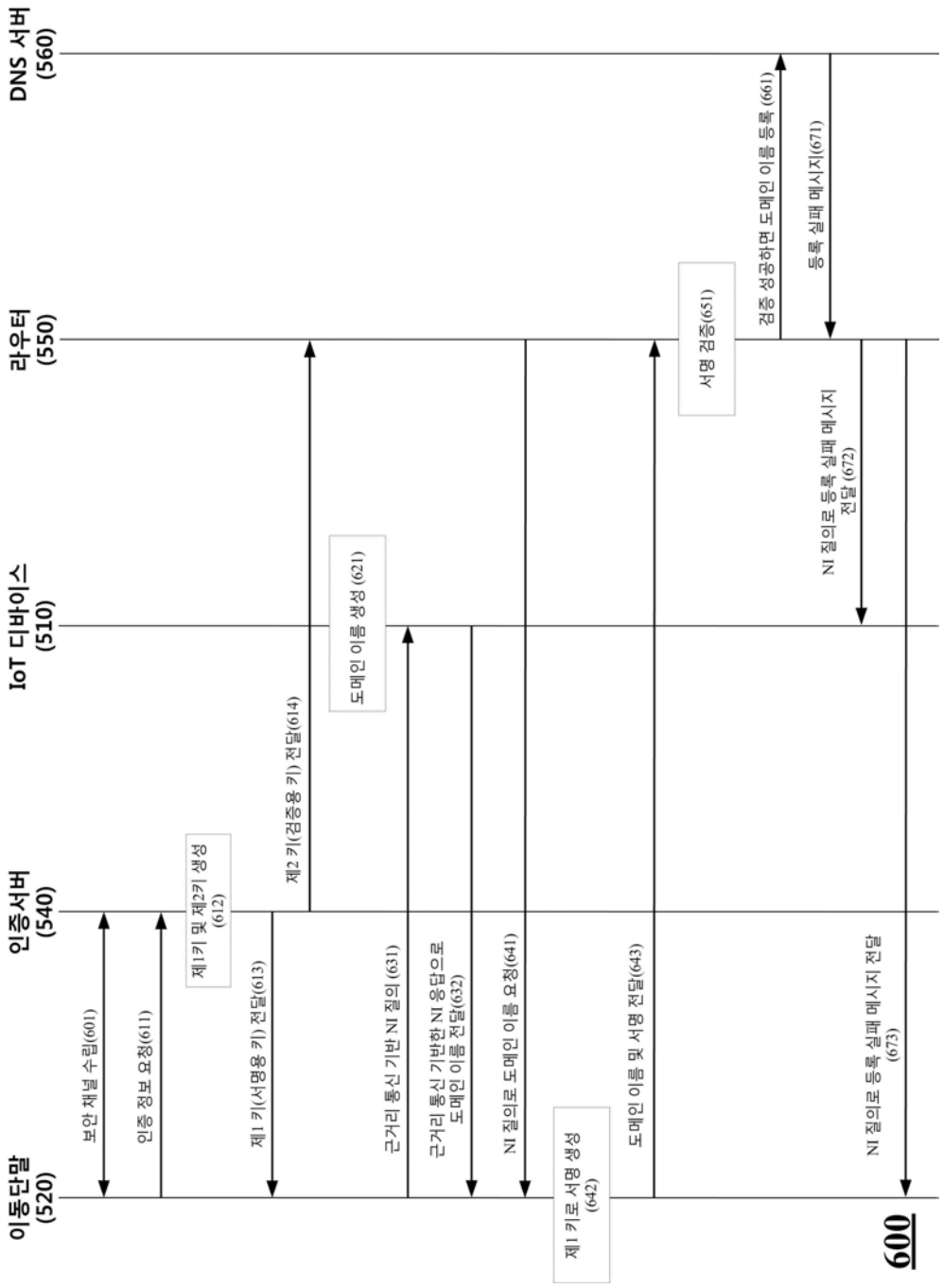
도면10


이동단말 (320)
인증서버 (340)
IoT 디바이스 (310)
라우터 (350)
DNS 서버 (360)
보안 채널 수립(401)
인증 정보 요청(411)
제1키 및 제2키 생성 (412)
제1 키(서명용 키) 전달(413)
제2 키(검증용 키) 전달(414)
근거리 통신으로 제1 키 전달(421)
도메인 이름 생성 및 서명 생성 (431)
NI 질의 (441)
NI 응답으로 도메인 이름 및 서명 전달 (442)
서명 검증(451)
검증 성공하면 도메인 이름 등록 (461)
400

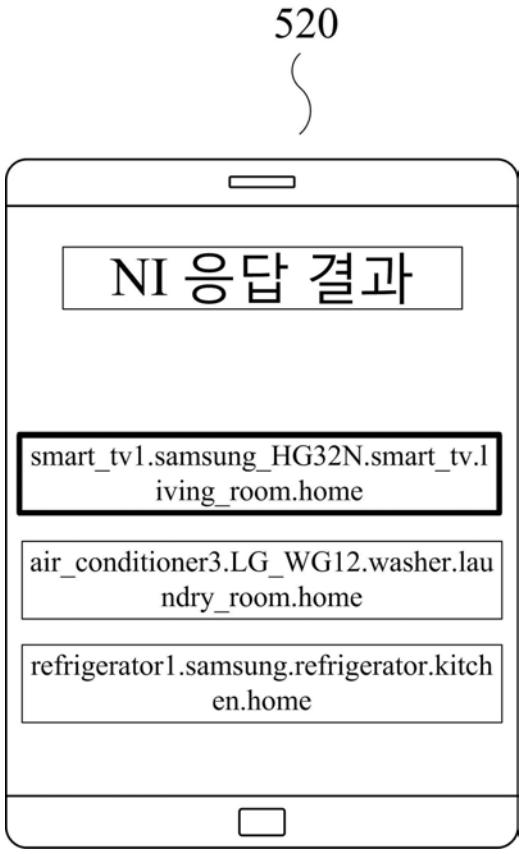
도면11


520
510
510C
510B
510A
도메인 이름
530
IPv6 Network
제1키(비밀키)
① 도메인 이름
② 디지털 서명
제2키(공용키)
550
서명 검증
540
560
Internet
refrigerator1.samsung_RH
269LP.refrigerator.home /
2002:4599:1EF2:0:0:0:
4559:1FE2
500

도면12


이동단말 (520)
인증서버 (540)
IoT 디바이스 (510)
라우터 (550)
DNS 서버 (560)
보안 채널 수립(601)
인증 정보 요청(611)
제1키 및 제2키 생성 (612)
제1 키(서명용 키) 전달(613)
제2 키(검증용 키) 전달(614)
도메인 이름 생성 (621)
근거리 통신 기반 NI 질의 (631)
근거리 통신 기반한 NI 응답으로 도메인 이름 전달(632)
NI 질의로 도메인 이름 요청(641)
제1 키로 서명 생성 (642)
도메인 이름 및 서명 전달(643)
서명 검증(651)
검증 성공하면 도메인 이름 등록 (661)
등록 실패 메시지(671)
NI 질의로 등록 실패 메시지 전달 (672)
NI 질의로 등록 실패 메시지 전달 (673)
600

도면13


520
NI 응답 결과
smart_tv1.samsung_HG32N.smart_tv.living_room.home
air_conditioner3.LG_WG12.washer.laundry_room.home
refrigerator1.samsung.refrigerator.kitchen.home