



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0096632
(43) 공개일자 2017년08월24일

(51) 국제특허분류(Int. Cl.)
H04L 29/12 (2006.01) G06F 9/44 (2006.01)
H04L 29/06 (2006.01) H04L 29/08 (2006.01)
H04W 4/00 (2009.01)
(52) CPC특허분류
H04L 61/3025 (2013.01)
G06F 9/4416 (2013.01)
(21) 출원번호 10-2017-7019852
(22) 출원일자(국제) 2015년12월17일
심사청구일자 없음
(85) 번역문제출일자 2017년07월17일
(86) 국제출원번호 PCT/GB2015/054055
(87) 국제공개번호 WO 2016/097743
국제공개일자 2016년06월23일
(30) 우선권주장
1422669.0 2014년12월18일 영국(GB)

(71) 출원인
암, 리미티드
영국 캠브리지 씨비1 9엔제이 풀본로드 110
(72) 발명자
세이신, 사이먼
영국 씨비1 9엔제이 캠브리지 체리 힌턴 풀본 로드 110
데이비드, 노버트
영국 씨비1 9엔제이 캠브리지 체리 힌턴 풀본 로드 110
사니발라, 미코
영국 씨비1 9엔제이 캠브리지 체리 힌턴 풀본 로드 110
(74) 대리인
양영준, 백만기

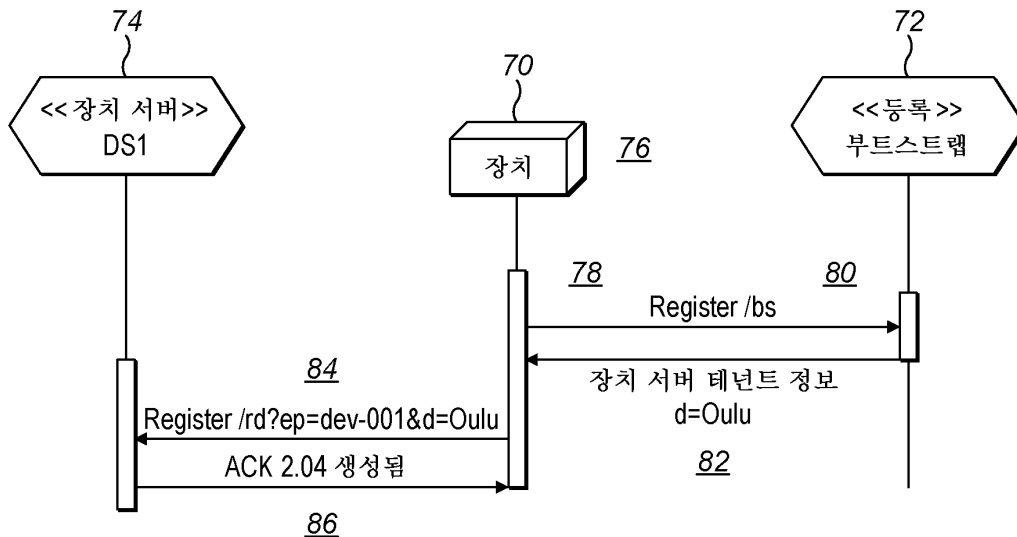
전체 청구항 수 : 총 18 항

(54) 발명의 명칭 장치들에의 테넌시의 할당

(57) 요약

네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법은 장치 식별자를 부트스트랩 서버에 전송하는 단계를 포함한다. 이 방법은 장치에서, 장치가 장치 서버에 등록할 수 있게 하는 장치 서버 주소를 수신하는 단계를 더 포함한다. 테넌시는 장치 서버 주소를 갖는 장치에 할당된다.

대표도 - 도5a



(52) CPC특허분류

H04L 63/0823 (2013.01)

H04L 67/34 (2013.01)

H04W 4/001 (2013.01)

H04W 4/005 (2013.01)

명세서

청구범위

청구항 1

네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시(tenancy)를 할당하는 방법으로서,
장치 식별자를 부트스트랩 서버에 전송하는 단계;
상기 장치에서, 상기 장치가 장치 서버에 등록할 수 있게 하는 장치 서버 주소를 수신하는 단계를 포함하고,
상기 테넌시는 상기 장치 서버 주소를 갖는 상기 장치에 할당되는, 방법.

청구항 2

제1항에 있어서, 상기 장치 서버 주소를 갖는 인증서 공통 이름(certificate common name) 내에 테넌시를 할당하는 단계를 포함하는, 방법.

청구항 3

제2항에 있어서, 상기 인증서 공통 이름은 상기 장치 서버의 호스트 및 도메인 이름인, 방법.

청구항 4

제1항 내지 제3항 중 어느 한 항에 있어서, 상기 장치 식별자는 상기 장치의 IMEI, 일련 번호 또는 논리적 이름 중에서 선택된 장치 이름을 포함하는, 방법.

청구항 5

제1항에 있어서, 상기 부트스트랩 서버는 도메인들을 장치들과 매칭시키는 데이터에 액세스하고 상기 데이터에 대한 참조에 기초하여 테넌시를 할당하는, 방법.

청구항 6

제5항에 있어서, 부트스트래핑 전에 상기 데이터를 데이터 패킷으로 상기 부트스트랩 서버에 통신하는 단계를 포함하는, 방법.

청구항 7

제6항에 있어서, 상기 통신은 대역 외 통신(out of band communication)인, 방법.

청구항 8

네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법으로서,
부트스트랩 서버에서 장치 식별자를 수신하는 단계;
상기 장치에, 상기 장치가 장치 서버에 등록할 수 있게 하는 장치 서버 주소를 전송하는 단계; 및
상기 장치 서버 주소를 갖는 상기 장치에 테넌시를 할당하는 단계를 포함하는 방법.

청구항 9

제8항에 있어서, 상기 부트스트랩 서버는 도메인들을 장치들과 매칭시키는 데이터에 액세스하고 상기 데이터에 대한 참조에 기초하여 테넌시를 할당하는, 방법.

청구항 10

제9항에 있어서, 부트스트래핑 전에 상기 데이터를 데이터 패킷으로 상기 부트스트랩 서버에 통신하는 단계를

포함하는, 방법.

청구항 11

제10항에 있어서, 상기 통신은 대역 외 통신인, 방법.

청구항 12

제8항에 있어서, 상기 장치 서버 주소를 갖는 상기 장치에 테넌시를 할당하는 단계는 상기 장치 서버 주소를 갖는 인증서 공통 이름 내에 테넌시를 할당하는 단계를 포함하는, 방법.

청구항 13

제12항에 있어서, 상기 인증서 공통 이름은 상기 장치 서버의 호스트 및 도메인 이름인, 방법.

청구항 14

제8항에 있어서, 상기 장치 식별자는 상기 장치의 IMEI, 일련 번호 또는 논리적 이름 중에서 선택된 장치 이름을 포함하는, 방법.

청구항 15

네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법으로서,

부트스트랩 서버에서 장치 식별자를 수신하는 단계;

상기 장치에, 상기 장치가 장치 서버에 등록할 수 있게 하는 장치 서버 주소를 전송하는 단계;

상기 부트스트랩 서버에서, 상기 장치 서버로부터 상기 장치에 대한 테넌시의 요청을 수신하여 상기 장치에 테넌시를 할당하는 단계

를 포함하는 방법.

청구항 16

제15항에 있어서,

상기 장치 식별자를 상기 부트스트랩 서버에 전송하는 단계를 더 포함하는, 방법.

청구항 17

제15항에 있어서,

상기 장치에서, 상기 장치 서버 주소를 수신하는 단계를 더 포함하는, 방법.

청구항 18

제15항에 있어서, 상기 장치 서버에의 상기 장치의 등록시에, 상기 장치 서버는, 상기 부트스트랩 서버에, 상기 장치에 대한 테넌시의 상기 요청을 전송하는, 방법.

발명의 설명

기술 분야

[0001] 관련 출원(들)에 대한 상호 참조

[0002] 본 출원은 2014년 12월 18일자로 출원된 "ASSIGNMENT OF TENANCY TO DEVICES"라는 제목의 영국 특허 출원 제 1422669.0호에 대한 우선권을 주장하며, 해당 출원은 이로써 그 전체 내용이 참고로 포함된다.

[0003] 기술분야

[0004] 본 개시내용의 기술들은 일반적으로 엔드 노드들, 엔드포인트들 또는 사물 인터넷(Internet of Things, IoT) 장치들로서 일반적으로 알려진 장치들에의 테넌시(tenancy)의 할당에 관한 것이다.

배경 기술

- [0005] 사물 인터넷은 IP 사용이 가능하고 인터넷 연결되는 장치들 및 네트워크들과 함께, 그 장치들을 모니터링하고 제어하는 인터넷 서비스들을 포함한다. 인터넷에 연결된 이러한 IP 사용이 가능한 장치들은 엔드 노드, 엔드포인트 또는 IoT 장치라고 불릴 수 있으며, 몇 가지만 예를 들면 센서, 머신, 액티브 포지셔닝 태그, 무선 주파수 식별(RFID) 판독기 및 건물 자동화 장비를 포함할 수 있다.
- [0006] 프로그램들, 컴퓨터들 및 M2M(Machine-to-Machine) 간의 데이터 교환은 필수 요소이다. 상이한 프로그램들, 컴퓨터들 및 프로세서들이 상이한 환경들에서 사용된다. 인터넷에서, 전송 제어 프로토콜/인터넷 프로토콜(TCP/IP)은 통신에 사용되는 기본 프로토콜이다. TCP/IP는 패킷들에서 전송될 데이터를 어셈블 및 디어셈블하는 것을 담당한다. IP는 패킷들이 올바른 대상에 전달되도록 주소 지정을 처리하고 TCP/IP 위에서, HTTP(Hypertext Transfer Protocol)가 클라이언트/서버 프로토콜로서 사용됨으로써 프로그램이 또 다른 HTTP 메시지로 응답하는 서버에 HTTP 요청을 전송할 수 있다.
- [0007] 무선 임베디드 인터넷(Wireless Embedded Internet)은 사물 인터넷의 서브세트이며 일반적으로 리소스가 제한된 임베디드 장치들로 대표되는데, 이 장치들은 종종 배터리로 전원이 공급되고 저전력, 저 대역폭 무선 네트워크에 의해 인터넷에 연결된다.
- [0008] 현재의 웹 서비스 기술들은 높은 전력 및 큰 대역폭 요구들을 가지며 이들 중 어느 것도 저전력 네트워크와 호환되지 않는다. 최근에는, 저전력 무선 네트워크를 위해 바이너리 웹 서비스 프로토콜이 개발되었다. 바이너리 웹 서비스 솔루션은 적합한 웹 서비스 프로토콜(예를 들어 단순화된 HTTP 또는 제한된 애플리케이션 프로토콜(Constrained Application Protocol, CoAP)와 같은 바이너리 웹 서비스 프로토콜) 및 효율적인 콘텐츠 인코딩(예를 들어 EXI(Efficient XML Interchange), 바이너리 XML 또는 FI(Fast Infoset))의 사용을 포함한다.
- [0009] M2M(Machine-to-Machine) 통신이 널리 적용되는 네트워크 기술의 예는 IEEE 802.15.4 기반 임베디드 및 센서 네트워크와 같은 저전력 무선 네트워크이다. 더 최근에는, M2M 장치들이 IP사용이 가능하게 됨에 따라, 시스템들은 네트워킹 프로토콜로서 IP를 사용함으로써 더 개방적이 되었다.
- [0010] IEEE 802.15.4의 도입에 이어서, 그러한 무선 임베디드 링크들에 대한 IP 적응을 표준화하기 위한 다른 표준들이 개발되었다. 예를 들어, 저전력 무선 표준을 통한 IPv6(IPv6 over Low Power Wireless Standard, 6LoWPAN)는 적응 계층 및 관련 프로토콜들의 최적화를 통해 간단한 임베디드 장치들에서 저전력, 저속 무선 네트워크를 통한 IPv6의 효율적인 사용을 가능하게 하는 표준들의 세트이다.
- [0011] OMA(Open Mobile Alliance) LWM2M(Lightweight Machine-to-Machine)은 6LoWPAN에 적용 가능한 표준이며 제한된 셀룰러 및 M2M 장치들에 초점을 맞추고 있다. LWM2M(Lightweight Machine-to-Machine) 부트스트랩 프로세스는 LWM2M 클라이언트들이 하나 이상의 LWM2M 서버에 등록을 수행할 수 있도록 해당 클라이언트들에 부트스트랩 인터페이스를 통해 필수 정보를 제공하는 데 사용된다. 부트스트래핑 동안, 장치(테넌트)는 클라우드 기반 서버(호스트)에 등록하여, 도메인을 가로질러 애플리케이션들에 액세스할 수 있다. 도메인은 장치들의 논리적 그룹일 수 있으며 도메인을 DNS(Domain Name System)로 내보낼 때 도메인 값은 일반적으로 DNS 도메인 이름에 해당한다.
- [0012] 테넌시(tenancy)는 단일 테넌시 또는 다중 테넌시로 분류될 수 있다. 클라우드 기반 서버가 단독 테넌트 또는 단일 엔터티에 도메인을 제공하는 경우, 이는 단일 테넌시로 알려져 있다. 클라우드 기반 서버는 다수의 엔터티가 상이한 목적들로 상이한 장치들 또는 리소스들을 소유한다는 의미에서 관련이 없는 엔터티들을 포함하는 다수의 엔터티 또는 다수의 테넌트에 동일하거나 유사한 애플리케이션들을 제공할 수 있다. 이러한 경우 이는 다중 테넌시로 알려져 있고 전형적으로 다수의 테넌트를 갖는 클라우드 기반 서버는 다수의 장치를 소유하는 다수의 고객을 가질 것이다.
- [0013] 현재, 장치는 그의 테넌트 이름으로 부트스트래핑하기 전에 사전 구성되며 따라서 장치는 장치 서버에의 등록에 어느 테넌트 이름을 사용할지를 사전에 안다. 이러한 사전 구성은 시간과 리소스 모두에서 값비싼 오버헤드일 수 있다.

발명의 내용

- [0014] 일례에서, 네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법은 장치 식별자를 부트스트랩 서버에 전송하는 단계를 포함한다. 이 방법은 장치에서, 장치가 장치 서버에 등록할 수 있게 하는

장치 서버 주소를 수신하는 단계를 더 포함한다. 테넌시는 장치 서버 주소를 갖는 장치에 할당된다.

- [0015] 또 다른 예에서, 네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법은 부트스트랩 서버에서 장치 식별자를 수신하는 단계; 및 장치로, 장치가 장치 서버에 등록할 수 있게 하는 장치 서버 주소를 전송하는 단계를 포함한다. 이 방법은 장치 서버 주소를 갖는 장치에 테넌시를 할당하는 단계를 더 포함한다.
- [0016] 또 다른 예에서, 네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법은 장치 식별자를 부트스트랩 서버에 전송하는 단계; 및 장치에서, 장치가 장치 서버에 등록할 수 있게 하는 장치 서버 주소를 수신하는 단계를 포함한다. 장치 서버에의 장치의 등록시에, 장치 서버는 부트스트랩 서버와 통신하여 장치에 대한 테넌시의 할당을 요청한다. 이 방법은 장치에 테넌시를 할당하는 단계를 더 포함한다.
- [0017] 또 다른 예에서, 네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법은 부트스트랩 서버에서 장치 식별자를 수신하는 단계; 및 장치로, 장치가 장치 서버에 등록할 수 있게 하는 장치 서버 주소를 전송하는 단계를 포함한다. 이 방법은 부트스트랩 서버에서 장치 서버로부터 장치에 대한 테넌시의 요청을 수신하여 장치에 테넌시를 할당하는 단계를 더 포함한다.
- [0018] 또 다른 예에서, 네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법은 장치 식별자를 부트스트랩 서버에 전송하는 단계; 및 장치에서, 장치가 장치 서버에 등록할 수 있게 하는 장치 서버 주소를 수신하는 단계를 포함한다. 장치 서버는 장치 서버에의 장치의 등록 전에 장치에 대한 테넌시의 할당을 제공받는다.
- [0019] 또 다른 예에서, 네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법은 부트스트랩 서버에서 장치 식별자를 수신하는 단계; 장치에 대한 테넌시의 할당을 장치 서버에 전송하는 단계; 및 장치로, 장치가 장치 서버에 등록할 수 있게 하는 장치 서버 주소를 전송하는 단계를 포함한다. 장치 서버는 장치 서버에의 장치의 등록 전에 장치에 대한 테넌시의 할당을 제공받는다.
- [0020] 또 다른 예에서, 네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법은 부트스트랩 서버에서 장치 식별자를 수신하는 단계; 부트스트랩 서버로부터 장치 식별자를 장치 서버에 제공하는 단계; 및 부트스트랩 서버로부터 상기 장치로, 장치가 장치 서버에 등록할 수 있게 하는 정보를 제공하는 단계를 포함한다. 장치 서버는 장치 서버에의 장치의 등록 전에 장치에 대한 테넌시의 할당을 제공받는다.
- [0021] 또 다른 예에서, 네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법은 부트스트랩 서버로부터 장치 식별자를 수신하는 단계; 장치로부터, 장치를 장치 서버에 등록하기 위한 요청을 수신하는 단계; 및 장치 식별자에 기초하여 테넌시를 할당하는 단계를 포함한다.

도면의 간단한 설명

- [0022] 도 1은 LWM2M 배치 시나리오의 개략도이다.
- 도 2는 서버 및 클라이언트를 갖는 LWM2M 아키텍처의 개략도이다.
- 도 3은 LWM2M 객체 모델의 개략도이다.
- 도 4a 및 도 4b는 각각 클라이언트에 의해 개시되는 부트스트랩(도 4a) 및 서버에 의해 개시되는 부트스트랩(도 4b)의 개략도들이다.
- 도 5a 및 도 5b는 제1 실시예에 따른 테넌시를 할당하는 방법의 개략도들이다.
- 도 6a 및 도 6b는 제2 실시예에 따른 테넌시를 할당하는 방법의 개략도들이다.
- 도 7a 및 도 7b는 제3 실시예에 따른 테넌시를 할당하는 방법의 개략도들이다.
- 도 8a 및 도 8b는 제4 실시예에 따른 테넌시를 할당하는 방법의 개략도들이다.
- 도 9a 및 도 9b는 제5 실시예에 따른 테넌시를 할당하는 방법의 개략도들이다.

발명을 실시하기 위한 구체적인 내용

- [0023] 본 개시내용의 기술들은 LWM2M 부트스트랩 프로세스에 부가적인 특징들을 제공한다. 이 기술들은 장치가 테넌트 이름으로 사전 구성되지 않고 부트스트랩 프로세스의 일부로서 장치에 단일 또는 다중 테넌시가 할당될 수

있게 한다.

- [0024] 이하, 실시예들을 단지 예로서 첨부 도면들을 참조하여 설명한다. 따라서, 도 1을 참조하면, M2M 엔드포인트 장치(10)에 대한 배치 시나리오가 가로등 및 전기 계량기로부터 가정 및 빌딩 자동화까지 다양한 시장 부문들에 걸쳐 도시되어 있다. 엔드포인트 장치(10)는 저전력 무선 네트워크를 통한 제한된 애플리케이션 프로토콜 (CoAP)과 같은 개방 IETF 표준들에 기초하여 LWM2M 서버(16)와 인터페이스(12)할 수 있다. 이러한 저전력 무선 네트워크 중 하나는 저전력 무선 표준을 통한 IPv6 (6LoWPAN)(14)를 사용하는 IEEE 802.15.4 기반 임베디드 및 센서 네트워크이다. LWM2M 서버(16)는 사설 서버, 사설 클라우드 또는 공용 클라우드 환경(19)으로부터 M2M 애플리케이션들의 배치를 가능하게 하는 더 넓은 월드 와이드 웹 네트워크(18)와 인터페이스할 수 있다.
- [0025] 도 2 및 도 3은 M2M 엔드포인트 장치(10)상의 클라이언트(22)와 LWM2M 서버(16)상의 서버 소프트웨어 간에 사용하기 위한 애플리케이션 계층 통신 프로토콜을 기술적으로 정의하는 LWM2M 아키텍처(20)를 도시한다. 클라이언트(22)는 장치(10)에 상주하고 소프트웨어 라이브러리 또는 모듈 또는 장치(10)의 내장 기능으로서 통합된다. 서버(16)는 사설 또는 공용 데이터 센터에 위치하며 M2M 서비스 제공자, 네트워크 서비스 제공자 또는 애플리케이션 서비스 제공자에 의해 호스팅될 수 있다. 서버와 클라이언트 간에 4개의 논리적인 인터페이스, 즉 미리 프로비저닝되거나 클라이언트/서버에 의해 개시되는 부트스트래핑(24); 클라이언트(22) 및 그의 객체들(50)을 등록하기 위한 등록(26); 서버(16)가 객체(50) 또는 리소스(48)에 액세스하는 것을 가능하게 하는 객체/리소스 액세스 또는 장치 관리(28); 및 새로운 리소스(48) 값들을 갖는 통지를 위한 정보 보고(30)가 정의된다.
- [0026] LWM2M 프로토콜 스택(32)은 사용자 데이터그램 프로토콜(UDP) 및 단문 메시지 서버(SMS) 베어러들(36)에 대한 기본 전송 프로토콜로서 제한된 애플리케이션 프로토콜(CoAP)(34)을 사용한다. CoAP 프로토콜(34)은 메시지 헤더, 요청/응답 코드, 메시지 옵션 및 재전송 메커니즘을 정의한다. CoAP 프로토콜(34)은 리소스가 제한된 장치들(10)상의 표현 상태 전송(Representational State Transfer, RESTful) 애플리케이션 프로그래밍 인터페이스(API)에 대한 HTTP의 대안이며, HTTP의 메서드들에 매핑될 수 있는 GET, POST, PUT, DELETE의 메서드들을 지원한다. HTTP와 달리, CoAP 메시지들은 UDP와 같은 데이터그램 지향 전송을 통해 CoAP 엔드포인트들 간에 비동기식으로 교환된다. CoAP 메시지들은 기능이 4 바이트 오버헤드로 시작될 수 있도록 바이너리 형식으로 인코딩된다. LWM2M은 CoAP와의 UDP 바인딩을 필수로 정의하는 반면, CoAP와의 SMS 바인딩은 선택 사항이고, 이는 사실상 LWM2M 클라이언트(22)와 서버(16) 간의 상호 작용이 SMS와 UDP(36) 양자를 통해 발생할 수 있음을 의미한다.
- [0027] LWM2M 아키텍처(20)는 보안 프로토콜들을 사용하여 데이터그램 전송 계층 보안(Datagram Transport Layer Security, DTLS)(38)을 사용하여 클라이언트(22)와 서버(16) 간의 통신을 보안한다. DTLS(38)은 모든 효율적인 페이로드들(40)에 대해 LWM2M 서버(16)와 LWM2M 클라이언트(22) 간의 보안 채널을 제공하는 데 사용된다. 효율적인 페이로드들(40)은 개별 리소스들에 대한 일반 텍스트 및 리소스 배치들(객체 또는 리소스 어레이들)에 대한 바이너리 TLV 또는 JSON일 수 있다. DTLS(38) 보안 모드들은 임베디드 장치들(10)을 지원하기 위해 사전 공유 키와 공개 키 기술 둘 다를 포함한다.
- [0028] M2M 애플리케이션(42) 개발자는 M2M 웹 애플리케이션(44) 및 M2M 장치 관리 애플리케이션(46)을 통해 원격으로 장치(10)를 관리할 수 있다.
- [0029] 도 3을 참조하면, 도 2에 도시된 클라이언트(22)가 더 상세히 도시되어 있다. LWM2M 클라이언트(22)에 의해 이용 가능한 각각의 정보는 리소스(48)이다. 리소스(48)는 관독, 기입 또는 실행될 수 있고 다음과 같은 URI를 이용해 액세스되는 정보이다:
- [0030] • /{객체 ID}/{객체 인스턴스}/{리소스 ID}, 예를 들어 /3/0/1 (장치 객체, 제조업체 리소스).
- [0031] 리소스들(48)은 추가로 객체들(50)로 논리적으로 편성된다. 각각의 LWM2M 클라이언트(22)는 각각이 객체(50)에 속하는 임의의 수의 리소스(48)를 가질 수 있다. 예를 들어, 펌웨어 객체(52)는 펌웨어 업데이트 목적으로 사용되는 모든 리소스(54)를 포함할 수 있다.
- [0032] 도 3에 도시되지는 않았지만, 장치 관리 목적을 위한 객체들의 세트는 다음을 포함한다:
- [0033] • 관리 서버들(16)과 장치(10)상의 클라이언트(22) 간의 보안 양태들을 처리하기 위한 보안 객체;
- [0034] • 관리 서버(16)와 관련된 데이터 및 기능들을 정의하는 서버 객체;
- [0035] • 수 개의 허용되는 관리 서버들(16) 각각에 대해 해당 서버들(16)이 클라이언트(22)상의 각각의 데이터 객체

에 대해 갖는 액세스 권한들을 정의하는 액세스 제어 객체;

- [0036] • 장치(10) 특정 정보와 관련된 M2M 장치(10)상의 리소스들을 상세히 나타내는 장치 객체;
- [0037] • M2M 장치(10)의 현재 위치에 대한 정보를 제공하는 리소스들을 그룹화하는 위치 객체;
- [0038] • 네트워크 연결의 상태를 모니터링하는 데 도움이 되는 M2M 장치(10)상의 리소스들을 함께 그룹화하는 연결 객체;
- [0039] • 기존 네트워크 연결에 대한 통계 정보를 보유하는 M2M 장치(10)상의 리소스들을 함께 그룹화하는 연결 통계 객체.
- [0040] 예로서, 장치 객체는 제조업체, 모델, 전력 정보, 사용 가능 메모리 및 에러 정보와 같은 장치 정보의 원격 검색을 허용한다. 또한 장치 객체는 원격 재부팅 또는 초기화(factory reset)의 개시를 위한 리소스를 제공한다.
- [0041] 도 4a 및 도 4b를 참조하여 더 상세히 설명되는 바와 같이, LWM2M 표준은 서버(16)가 서버(16)에 등록할 장치(10)의 키잉, 액세스 제어 및 구성을 관리할 수 있게 하는 프로비저닝 및 부트스트래핑(24) 기능을 정의한다. LWM2M 부트스트랩 서버는 클라이언트(22)가 하나 이상의 서버(16)에 그의 등록을 수행할 수 있도록 클라이언트(22)에 부트스트랩 인터페이스(24)를 통해 정보를 제공하는 데 사용된다. 부트스트랩 시퀀스 후에, 클라이언트(22)는 서버 계정 및 옵션으로 액세스 제어 또는 연결 객체와 같은 일부 다른 객체 인스턴스를 갖는다.
- [0042] 완전한 프로비저닝 및 키 관리를 위한 부트스트래핑은 플래시로부터와 같은 미리 구성된 부트스트래핑; SIM 카드로부터와 같은 스마트 카드 부트스트래핑 또는 초기 설정 또는 재구성을 위한 부트스트랩 서버의 사용 및 클라이언트에 의해 개시되는 부트스트래핑 및 서버에 의해 개시되는 부트스트래핑 모드들에서의 키잉일 수 있다.
- [0043] 도 4a에 도시된 바와 같은 그의 가장 단순화된 형태에서 클라이언트에 의해 개시되는 부트스트랩은 LWM2M 클라이언트(62)로부터의 엔드포인트 장치에 대한 부트스트랩에 대한 요청 단계(60)를 포함한다. 이에 응답하여, LWM2M 부트스트랩 서버(64)는 LWM2M 클라이언트(62)에 LWM2M 서버 객체들의 프로비전(66)을 제공한다. 모든 경우에, 부트스트랩 서버(64)는 OMA LWM2M에 따른 장치 서버의 정보 - 최소한, 장치 서버의 IP 주소 및 보안 모드를 제공한다.
- [0044] 도 4b를 참조하면 서버에 의해 개시되는 부트스트랩은 단지 LWM2M 부트스트랩 서버(64)로부터 LWM2M 클라이언트(62)로의 LWM2M 서버 객체들의 프로비전(66)을 포함한다.
- [0045] 도 5a는 본 기술들에 따른 테넌시 이름의 프로비저닝을 포함하는 부트스트랩 프로세스의 개략도이다. 부트스트랩 프로세스는 장치(70), 부트스트랩 서버(72) 및 장치 서버(74)를 포함한다. 부트스트랩 서버(72) 및 장치 서버(74)는 동일한 또는 상이한 장치들일 수 있다. 예를 들어, 부트스트랩 서버(72)는 제1 장치일 수 있고, 장치 서버(74)는 제2의 상이한 장치일 수 있다. 다른 예들에서, 부트스트랩 서버(72) 및 장치 서버(74)는 동일한 장치일 수 있고, 본 명세서에서 부트스트랩 서버(72) 및 장치 서버(74)에 속한다고 생각되는 기능은 예를 들어 동일한 장치의 하나 이상의 프로세서에서 실행되는 별개의 애플리케이션들로서 구현된다. 도 5b를 참조하면, 부트스트래핑 프로세스는 단계 76에서 장치(70)의 초기 파워 온을 포함한다. 그 후 장치(70)는 부트스트랩 프로세스를 시작하기 위해 부트스트랩 서버와 연결할 수 있다. 단계 78에서, 예를 들어 에지 라우터상의 디폴트 위치를 가정하는 것 또는 애니캐스트 주소를 부트스트랩 서버(72)에 할당하는 것을 포함하여 부트스트랩 서버(72)를 발견하기 위한 몇몇의 메커니즘이 존재할 수 있다. 부트스트랩 서버들(72)은 배치에 따라 다른 방법들에 의해 발견 가능할 수 있다.
- [0046] 단계 78 및 발견에 이어서 장치(70)는 단계 80에서 부트스트랩 서버(72)에 등록할 수 있다.
- [0047] 전형적으로, 부트스트랩 서버는 POST/bs?ep=temperature_sensor001과 같은 장치(70)로부터의 POST를 수락한다. 본 기술에서, 장치(70)가 속하는 도메인은 POST에 포함되지 않는다. 부트스트랩 서버(72)는 도메인들을 장치들과 매칭시키는 리스트를 유지하거나 그 리스트에 액세스하며, 따라서 장치(70)는 리스트에서 식별될 수 있다. 예를 들어, 장치(70) 이름은 단계 80 동안 장치를 고유하게 식별하는 문자열의 형태로 장치(70)로부터 부트스트랩 서버(72)에 전송될 수 있다. 그러한 식별의 예로는 IMEI(International Mobile Station Equipment Identity) 번호를 포함한다.
- [0048] 단계 82에서, 부트스트랩 서버(72)는 장치(70)가 속한다고 생각되는 도메인의 형태로 장치 서버(74) 테넌트 정보를 반환한다. 그 후 장치(70)는 단계 84에서, Register POST/rd?ep=temperature_sensor001&d=Oulu의 형태로

POST를 사용하여 장치 서버(74)에 연락함으로써 도메인에 등록할 수 있다. 장치 서버(74)는 단계 86에서 요청을 수락하고 ACK(Acknowledge) 2.04로 응답할 수 있고 장치(70)에 추가 정보를 PUT하기 위해 WRITE 동작들을 수행할 수 있다. 따라서, 부트스트래핑 프로세스의 일부로서 장치는 POST/rd?ep=temperature_sensor001&d=tenant_name(여기서 d는 도메인 또는 테넌트 이름임)과 유사한 요청을 사용하여 서버에 등록할 수 있다.

[0049] 도 6a는 본 기술들에 따른 테넌트 이름의 프로비저닝을 포함하는 부트스트랩 프로세스의 개략도이다. 동일한 부분들 및 프로세스 단계들을 식별하기 위해 동일한 참조 번호들이 사용될 것이다. 부트스트랩 프로세스는 장치(70), 부트스트랩 서버(72) 및 장치 서버(74)를 포함한다. 도 6b를 참조하면, 부트스트래핑 프로세스는 단계 76에서 장치(70)의 초기 파워 온을 포함한다. 그 후 장치(70)는 부트스트랩 프로세스를 시작하기 위해 부트스트랩 서버와 연결할 수 있다. 단계 78에서, 예를 들어 에지 라우터상의 디폴트 위치를 가정하는 것 또는 애니캐스트 주소를 부트스트랩 서버(72)에 할당하는 것을 포함하여 부트스트랩 서버(72)를 발견하기 위한 몇몇의 메커니즘이 존재할 수 있다. 부트스트랩 서버들(72)은 배치에 따라 다른 방법들에 의해 발견 가능할 수 있다.

[0050] 단계 78 및 발견에 이어서 장치(70)는 단계 80에서 부트스트랩 서버(72)에 등록할 수 있다. 전형적으로, 부트스트랩 서버는 POST/bs?ep=temperature_sensor001과 같은 장치(70)로부터의 POST를 수락한다. 본 기술에서, 장치(70)가 속하는 도메인은 POST에 포함되지 않는다. 부트스트랩 서버(72)는 장치(70)가 포함되는 도메인들의 리스트를 유지하고, 장치(70)는 다양한 방식으로 리스트에 대해 식별될 수 있다. 예를 들어, 장치(70) 이름은 장치를 고유하게 식별하는 문자열의 형태로 전송될 수 있다. 그러한 식별의 예로는 IMEI(International Mobile Station Equipment Identity) 번호를 포함한다.

[0051] 단계 88에서, 프로세스는 도 5b와 관련하여 설명된 것과 상이한데 그 차이점은 단계 88에서 부트스트랩 서버(72)는 인증서 공통 이름(CN)에만 테넌트 이름을 갖는 장치 서버 정보 인증서를 반환한다는 점이다. 인증서 공통 이름은 전형적으로 장치 서버(74)의 호스트 및 도메인 이름이며, www.arm.com/oulu/와 같은 URI의 형식으로 이루어질 수 있다.

[0052] 따라서 단계 90에서, 장치(70)는 테넌트 이름을 인식하지 못하지만 장치 서버(74)에 등록을 진행하는데 그 이유는 부트스트랩 서버(72)가 CN 필드에 테넌트 이름을 포함하는 인증서를 제공했기 때문이다. 등록은 인증서 CN=domain/temperature_sensor001를 갖는 Register POST/rd?ep=temperature_sensor001과 같은 POST 요청의 형태로 이루어질 수 있다. 단계 92에서, 장치 서버(74)는 인증서 CN 필드로부터 테넌트 이름을 취하여 그 요청을 수락하고 ACK 2.04로 응답할 수 있고 장치(70)에 추가 정보를 PUT하기 위해 WRITE 동작들을 수행할 수 있다.

[0053] 도 7a는 본 기술들에 따른 테넌트 이름의 프로비저닝을 포함하는 부트스트랩 프로세스의 개략도이다. 동일한 부분들 및 프로세스 단계들을 식별하기 위해 동일한 참조 번호들이 사용될 것이다.

[0054] 부트스트랩 프로세스는 장치(70), 부트스트랩 서버(72) 및 장치 서버(74)를 포함한다. 도 7b를 참조하면, 부트스트래핑 프로세스는 단계 76에서 장치(70)의 초기 파워 온을 포함한다. 그 후 장치(70)는 부트스트랩 프로세스를 시작하기 위해 부트스트랩 서버와 연결할 수 있다. 단계 78에서, 예를 들어 에지 라우터상의 디폴트 위치를 가정하는 것 또는 애니캐스트 주소를 부트스트랩 서버(72)에 할당하는 것을 포함하여 부트스트랩 서버(72)를 발견하기 위한 몇몇의 메커니즘이 존재할 수 있다. 부트스트랩 서버들(72)은 배치에 따라 다른 방법들에 의해 발견 가능할 수 있다.

[0055] 단계 78 및 발견에 이어서 장치(70)는 단계 80에서 부트스트랩 서버(72)에 등록할 수 있다. 전형적으로, 부트스트랩 서버는 POST/bs?ep=temperature_sensor001과 같은 장치(70)로부터의 POST를 수락한다. 본 기술에서, 장치(70)가 속하는 도메인은 POST에 포함되지 않는다.

[0056] 단계 94에서, 부트스트랩 서버(72)는 장치 서버(74) 정보를 장치(70)에 반환하지만, 도메인 테넌트 정보는 반환하지 않는다. 최소한 OMA LWM2M 표준에 따라 부트스트랩 서버(72)는 장치 서버(74)의 IP 주소 및 그의 보안 모드를 장치(70)에 제공한다. 따라서, 장치(70)는 단계 96에서 POST rd?ep=temperature_sensor001 메시지를 사용하여 장치 서버(74)에 등록할 수 있다.

[0057] 단계 98에서, 장치 서버(74)는 장치(70)에 대한 테넌트 룩업(tenant look-up)을 구현할 수 있다. 그렇게 하면, 장치 서버(74)는 어느 도메인이 장치(70)에 할당되었는지를 문의하기 위해 부트스트랩 서버(72)와 통신한다. 장치 서버(74)는 장치(70)가 예를 들어 단계 96에서 장치 서버에 등록할 때 장치(70)에 의해 전송된 임의의 메시지에서부터 부트스트랩 서버(72)의 룩업 주소를 획득할 수 있다. 대안적으로, 장치 서버(74)는 부트스트랩 서버(72)의 주소를 인식할 수 있거나 전술한 바와 같은 발견 프로토콜을 이용할 수 있다.

- [0058] 장치 서버(74) 및 부트스트랩 서버(72)는 통신 채널을 통해 서로 통신하고 네트워크에서 채널은 제어 채널 및 데이터 채널로 분할될 수 있다. 전형적으로, 제어 채널은 부트스트랩 프로토콜을 위해 사용되고 데이터 채널은 다수의 제어 또는 데이터 채널을 지원할 수 있는 네트워크상의 임의의 노드와 통상의 네트워크 동작 중에 사용된다. 제어 채널과 데이터 채널이 동일한 경우, 부트스트랩은 대역 내(In Band)(IB)에서 수행된다. 제어 채널과 데이터 채널이 상이한 경우, 부트스트랩은 대역 외(Out of Band)(OOB)에서 수행된다. 본 기술들에서, 통신은 대역 내 또는 대역 외에서 수행될 수 있다.
- [0059] 단계 100에서, 부트스트랩 서버(72)는 요청을 수락하는 장치 서버(74)에 테넌트 이름을 반환하고 ACK 2.04로 응답할 수 있고 단계 102에서 등록이 완료되도록 하기 위해 장치(70)에 추가 정보를 PUT하기 위해 WRITE 동작들을 수행할 수 있다.
- [0060] 도 8a는 본 기술들에 따른 테넌트 이름의 프로비저닝을 포함하는 부트스트랩 프로세스의 개략도이다. 동일한 부분들 및 프로세스 단계들을 식별하기 위해 동일한 참조 번호들이 사용될 것이다.
- [0061] 부트스트랩 프로세스는 장치(70), 부트스트랩 서버(72) 및 장치 서버(74)를 포함한다. 도 8b를 참조하면, 부트스트래핑 프로세스는 단계 76에서 장치(70)의 초기 파워 온을 포함한다. 그 후 장치(70)는 부트스트랩 프로세스를 시작하기 위해 부트스트랩 서버(72)와 연결할 수 있다. 단계 78에서, 예를 들어 에지 라우터상의 디폴트 위치를 가정하는 것 또는 애니캐스트 주소를 부트스트랩 서버(72)에 할당하는 것을 포함하여 부트스트랩 서버(72)를 발견하기 위한 몇몇의 메커니즘이 존재할 수 있다. 부트스트랩 서버들(72)은 배치에 따라 다른 방법들에 의해 발견 가능할 수 있다.
- [0062] 단계 78 및 발견에 이어서 장치(70)는 단계 80에서 부트스트랩 서버(72)에 등록할 수 있다. 전형적으로, 부트스트랩 서버는 POST/bs?ep=temperature_sensor001과 같은 장치(70)로부터의 POST를 수락한다. 본 기술에서, 장치(70)가 속하는 도메인은 POST에 포함되지 않는다. 부트스트랩 서버(72)는 장치(70)가 포함되는 도메인들의 리스트를 유지하고, 장치(70)는 다양한 방식으로 리스트에 대해 식별될 수 있다. 예를 들어, 장치(70) 이름은 장치를 고유하게 식별하는 문자열의 형태로 전송될 수 있다. 그러한 식별의 예로는 IMEI(International Mobile Station Equipment Identity) 번호를 포함한다.
- [0063] 단계 104에서, 프로세스는 전술한 것과 상이한데 그 차이점은 단계 104에서 부트스트랩 서버(72)는 장치 서버(74)에 장치(70)에 대한 테넌트 이름을 전달한다는 점이다. 단계 106에서, 부트스트랩 서버(72)는 장치 서버(74) 정보를 장치(70)에 제공한다. 최소한 OMA LWM2M 표준에 따라, 부트스트랩 서버(72)는 장치 서버(74)의 IP 주소 및 그의 보안 모드를 장치(70)에 제공한다. 따라서, 장치(70)는 단계 108에서 POST rd?ep=temperature_sensor001 메시지를 사용하여 장치 서버(74)에 등록할 수 있다. 장치 서버(74)에의 등록시에, 단계 104에서 제공된 장치(70) 이름을 참조하여 장치 서버(74)에 의해 테넌트 이름이 분석되고 ACK될 수 있다.
- [0064] 단계 110에서, 장치 서버(74)는 ACK 2.04로 응답할 수 있고 단계 110에서 등록이 완료되도록 하기 위해 장치(70)에 추가 정보를 PUT하기 위해 WRITE 동작들을 수행할 수 있다.
- [0065] 도 9a는 본 기술들에 따른 테넌트 이름의 프로비저닝을 포함하는 부트스트랩 프로세스의 개략도이다. 동일한 부분들 및 프로세스 단계들을 식별하기 위해 동일한 참조 번호들이 사용될 것이다. 부트스트랩 프로세스는 장치(70), 부트스트랩 서버(72) 및 장치 서버(74)를 포함한다. 도 9b를 참조하면, 부트스트래핑 프로세스는 단계 76에서 장치(70)의 초기 파워 온을 포함한다. 그 후 장치(70)는 부트스트랩 프로세스를 시작하기 위해 부트스트랩 서버와 연결할 수 있다. 단계 78에서, 예를 들어 에지 라우터상의 디폴트 위치를 가정하는 것 또는 애니캐스트 주소를 부트스트랩 서버(72)에 할당하는 것을 포함하여 부트스트랩 서버(72)를 발견하기 위한 몇몇의 메커니즘이 존재할 수 있다. 부트스트랩 서버들(72)은 배치에 따라 다른 방법들에 의해 발견 가능할 수 있다.
- [0066] 단계 78 및 발견에 이어서 장치(70)는 단계 80에서 부트스트랩 서버(72)에 등록할 수 있다. 전형적으로, 부트스트랩 서버는 POST/bs?ep=temperature_sensor001과 같은 장치(70)로부터의 POST를 수락한다. 본 기술에서, 장치(70)가 속하는 도메인은 POST에 포함되지 않는다.
- [0067] 도 9a 및 도 9b에 설명된 바와 같은 본 기술에서, 부트스트랩 서버(72)는 장치(70)에 관한 어떤 정보도 보유하지 않고, 장치 서버(74)가 장치(70)에 의해 제공된 이름을 사용하여 관련 정보를 GET할 수 있도록 하기 위해 단계 112에서 장치 서버(74)와 통신하도록 강요된다.
- [0068] 따라서 단계 112에서, 부트스트랩 서버(72)는 장치 서버(74)가 장치(70)에 관련된 테넌트 이름, 인증서 및 구성

정보를 획득하도록 하기 위해 장치 서버(74)에 장치(70) 이름을 제공한다. 본 기술은 장치 서버(74) 인증서 및 정보가 부트스트랩 서버(72)와 같은 제3자에 의해서가 아니라 장치 서버(74) 자체에 의해 제어될 수 있게 한다.

[0069] 단계 114에서, 부트스트랩 서버(72)는 장치 서버 정보를 장치(70)에 제공한다. 최소한 OMA LWM2M 표준에 따라 부트스트랩 서버(72)는 장치(70)에 제공하고 장치는 장치 서버(74)에 등록을 진행한다. 등록은 Register POST/rd?ep=temperature_sensor001과 같은 POST 요청의 형태로 이루어질 수 있고 장치 서버(74)는 단계 118에서 ACK 2.04로 응답할 수 있고 장치(70)에 추가 정보를 PUT하기 위해 WRITE 동작들을 수행할 수 있다.

[0070] 본 기술분야의 기술자들은 전술한 내용이 최선의 모드로 간주되는 것을 설명하였고, 적절한 경우 본 기술들을 수행하는 다른 모드들을 설명하였지만, 이러한 기술들은 바람직한 실시예에 대한 이 설명에서 개시된 특정 구성들 및 방법들로 한정되어서는 안된다는 것을 알 것이다. 본 기술분야의 기술자들은 이 기술들이 광범위한 응용들을 가지며, 실시예들은 첨부된 청구항들에서 정의된 본 발명의 개념을 벗어나지 않고 광범위한 수정들을 취할 수 있음을 인지할 것이다.

[0071] 일 양태에서, 본 기술들은 네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법을 제공하는데, 이 방법은: 장치 식별자를 부트스트랩 서버에 전송하는 단계; 상기 장치에서, 상기 장치가 장치 서버에 등록할 수 있게 하는 장치 서버 주소를 수신하는 단계를 포함하고, 상기 테넌시는 상기 장치 서버 주소를 갖는 상기 장치에 할당된다. 이러한 기술은 부트스트랩 서버에 의해 장치에 테넌시가 할당될 수 있게 하여 장치가 테넌시로 프로그래밍되지 않고도 초기에 조작될 수 있다.

[0072] 상기 방법은 상기 장치 서버 주소를 갖는 인증서 공통 이름 내에 테넌시를 할당하는 단계를 포함할 수 있다. 이러한 방식으로 테넌시를 할당하는 것은 인증서 공통 이름이 장치 서버의 호스트 및 도메인 이름인 것을 포함할 수 있다.

[0073] 장치는 장치의 IMEI, 일련 번호 또는 논리적 이름 중에서 선택된 장치 이름을 사용하여 자신을 식별할 수 있다.

[0074] 또 다른 양태에서, 본 기술들은 네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법을 제공하는데, 이 방법은: 부트스트랩 서버에서 장치 식별자를 수신하는 단계; 상기 장치로, 상기 장치가 장치 서버에 등록할 수 있게 하는 장치 서버 주소를 전송하는 단계; 및 상기 장치 서버 주소를 갖는 상기 장치에 테넌시를 할당하는 단계를 포함한다.

[0075] 부트스트랩 서버는 도메인들을 장치들과 매칭시키는 데이터에 액세스할 수 있고, 이로써 데이터에 대한 참조에 기초하여 테넌시를 할당할 수 있다. 이 방법은 부트스트래핑 전에 상기 데이터를 데이터 패킷으로 상기 부트스트랩 서버에 통신하는 단계를 포함할 수 있다. 이 통신은 대역 외 통신에서 발생할 수 있다.

[0076] 또 다른 양태에서, 본 기술들은 네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법을 제공하는데, 이 방법은: 장치 식별자를 부트스트랩 서버에 전송하는 단계; 상기 장치에서, 상기 장치가 장치 서버에 등록할 수 있게 하는 장치 서버 주소를 수신하는 단계 - 상기 장치 서버에의 상기 장치의 등록시에, 상기 장치 서버는 상기 부트스트랩 서버와 통신하여 상기 장치에 대한 테넌시의 할당을 요청함 -; 및 이로써 상기 장치에 테넌시를 할당하는 단계를 포함한다.

[0077] 또 다른 양태에서, 본 기술들은 네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법을 제공하는데, 이 방법은: 부트스트랩 서버에서 장치 식별자를 수신하는 단계; 상기 장치로, 상기 장치가 장치 서버에 등록할 수 있게 하는 장치 서버 주소를 전송하는 단계; 상기 부트스트랩 서버에서, 상기 장치 서버로부터 상기 장치에 대한 테넌시의 요청을 수신하여 상기 장치에 테넌시를 할당하는 단계를 포함한다.

[0078] 또 다른 양태에서, 본 기술들은 네트워크 내의 장치와 서버 사이의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법을 제공하는데, 이 방법은: 장치 식별자를 부트스트랩 서버에 전송하는 단계; 상기 장치에서, 상기 장치가 장치 서버에 등록할 수 있게 하는 장치 서버 주소를 수신하는 단계를 포함하고, 상기 장치 서버는 상기 장치 서버에의 상기 장치의 등록 전에 상기 장치에 대한 테넌시의 할당을 제공받는다.

[0079] 또 다른 양태에서, 본 기술들은 네트워크 내의 장치와 서버 사이의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법을 제공하는데, 이 방법은: 부트스트랩 서버에서 장치 식별자를 수신하는 단계; 장치 서버로, 상기 장치에 대한 테넌시의 할당을 전송하는 단계; 상기 장치로, 상기 장치가 상기 장치 서버에 등록할 수 있게 하는 장치 서버 주소를 전송하는 단계를 포함하고, 상기 장치 서버는 상기 장치 서버에의 상기 장치의 등록 전에 상기 장치에 대한 테넌시의 할당을 제공받는다.

[0080] 또 다른 양태에서, 본 기술들은 네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는

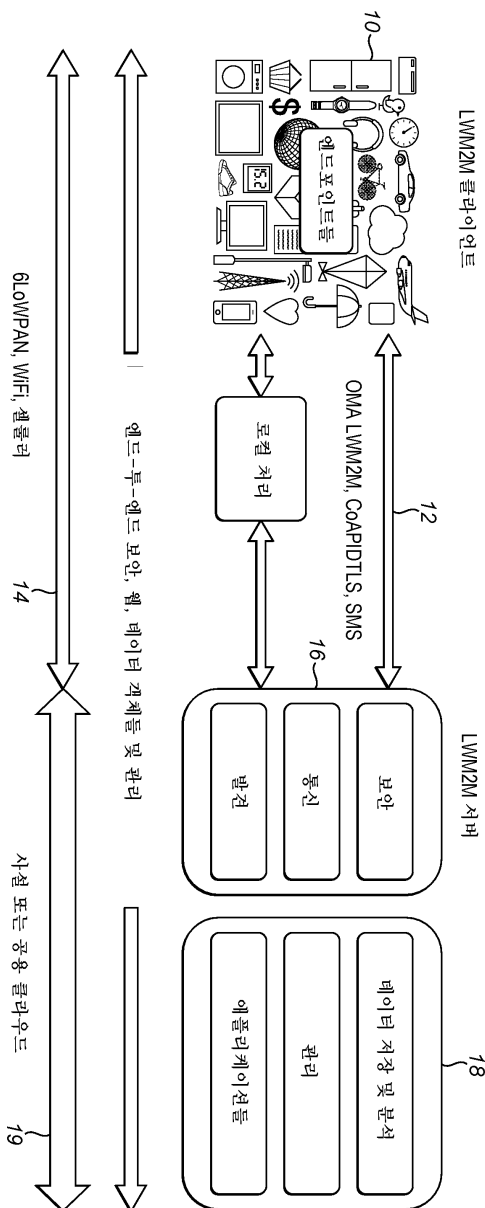
방법을 제공하는데, 이 방법은: 부트스트랩 서버에서 장치 식별자를 수신하는 단계; 상기 부트스트랩 서버로부터, 상기 장치 식별자를 장치 서버에 제공하는 단계; 및 상기 부트스트랩 서버로부터 상기 장치로, 상기 장치가 상기 장치 서버에 등록할 수 있게 하는 정보를 제공하는 단계를 포함하고, 상기 장치 서버는 상기 장치 서버에의 상기 장치의 등록 전에 상기 장치에 대한 테넌시의 할당을 제공받는다.

[0081]

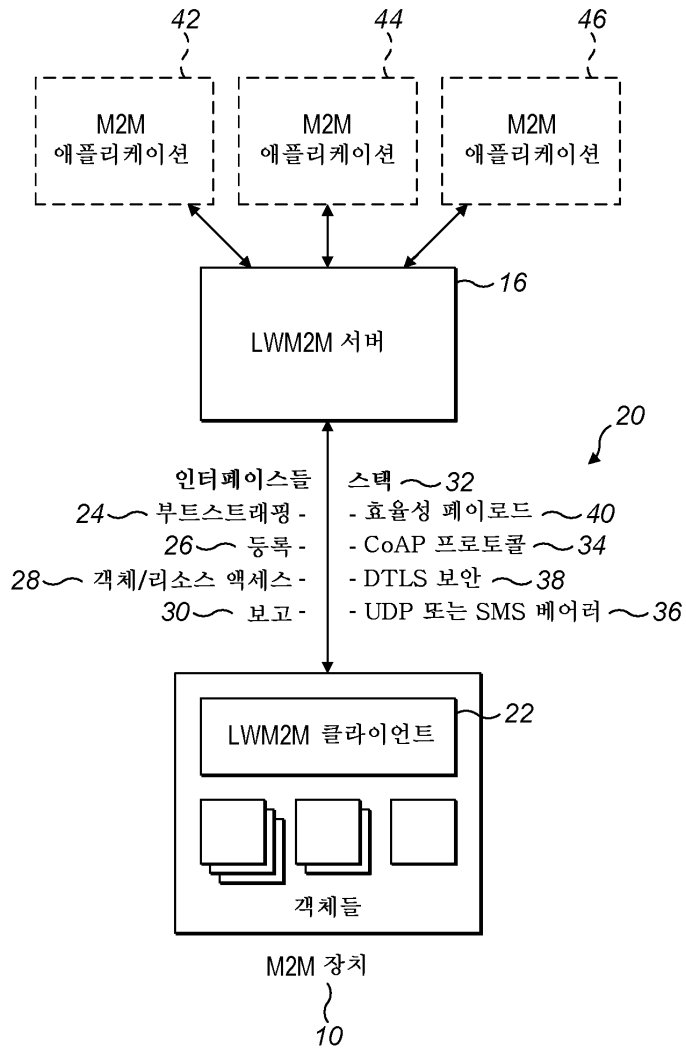
또 다른 양태에서, 본 기술들은 네트워크 내의 장치와 서버 간의 부트스트래핑 동안 장치에 테넌시를 할당하는 방법을 제공하는데, 이 방법은: 부트스트랩 서버로부터 장치 식별자를 수신하는 단계; 장치로부터, 장치 서버에 상기 장치를 등록하기 위한 요청을 수신하는 단계; 및 상기 장치 식별자에 기초하여 테넌시를 할당하는 단계를 포함한다. 장치 서버는 도메인들을 장치들과 매칭시키는 데이터에 액세스할 수 있고 이로써 그 데이터에 대한 참조에 기초하여 테넌시를 할당한다.

도면

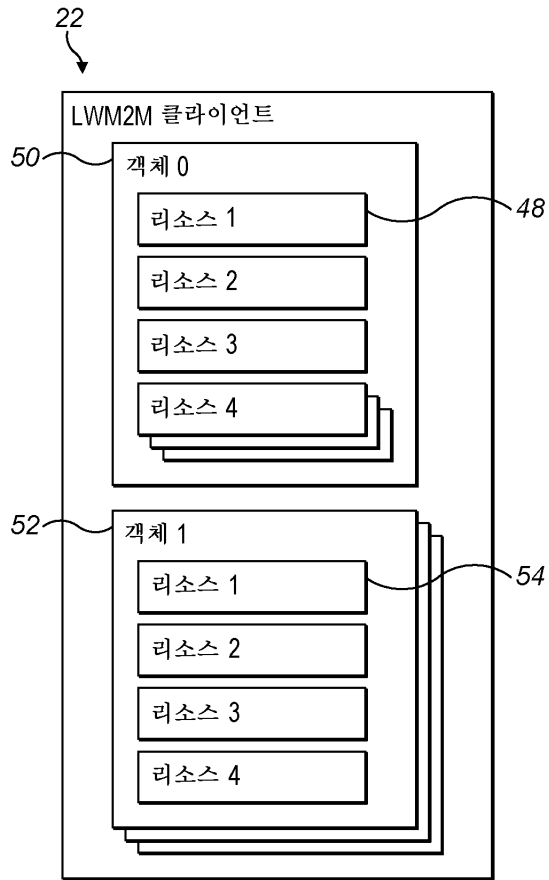
도면1



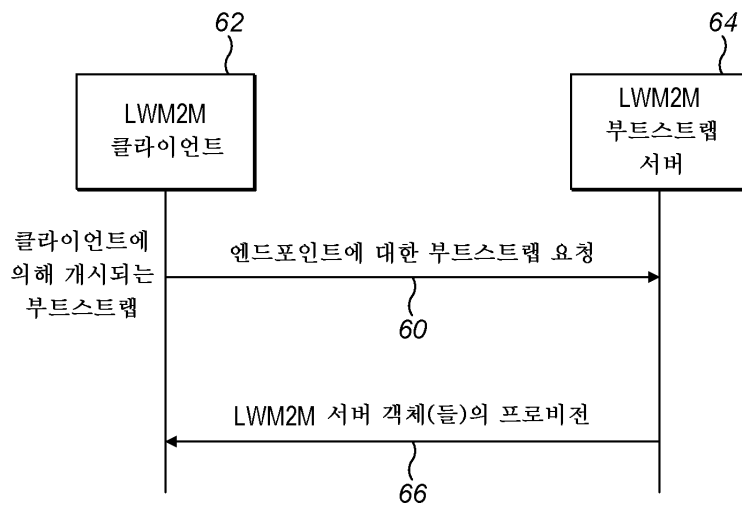
도면2



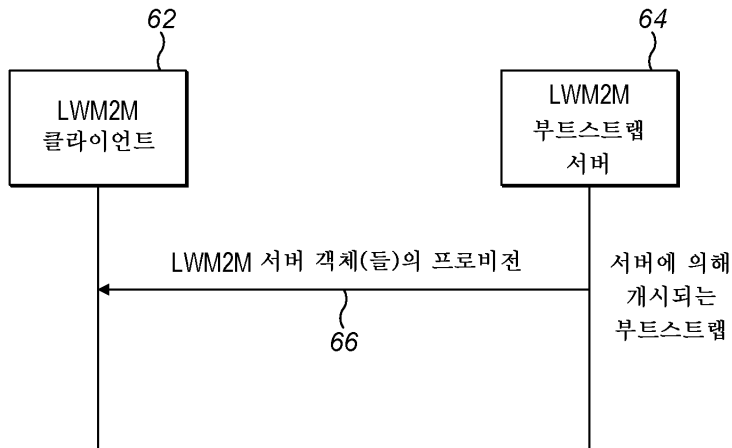
도면3



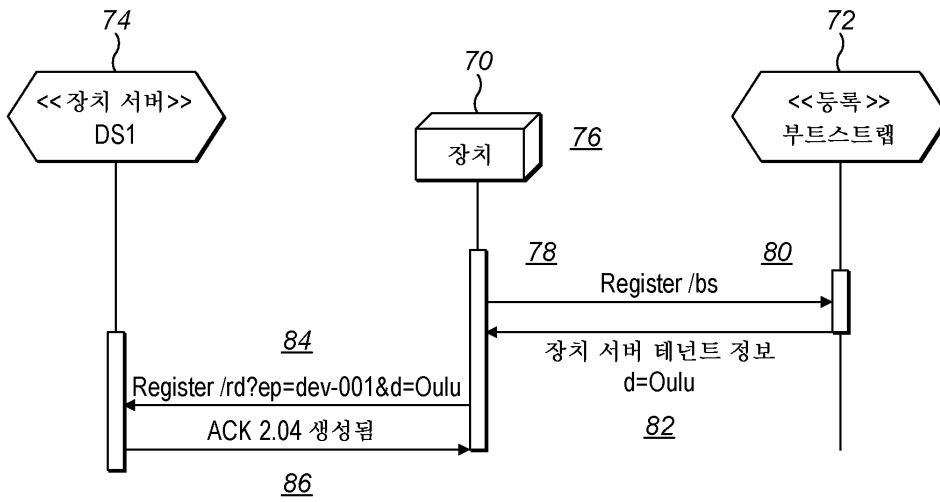
도면4a



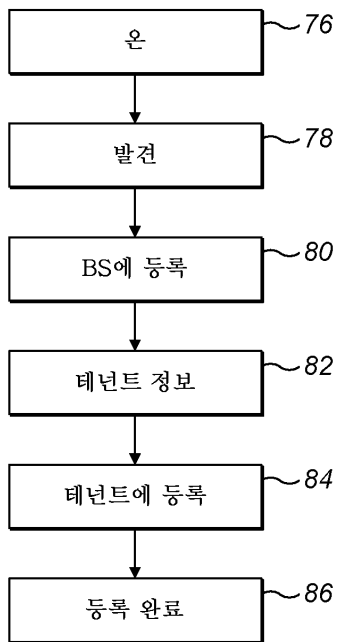
도면4b



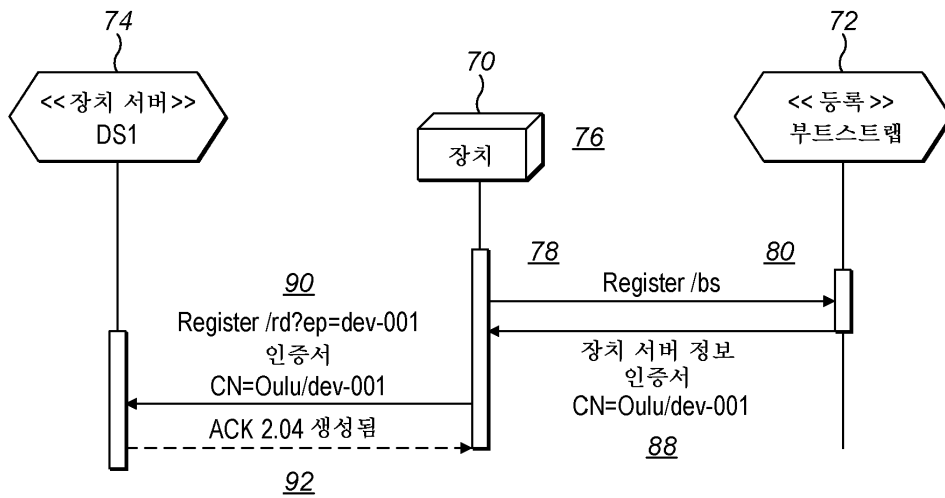
도면5a



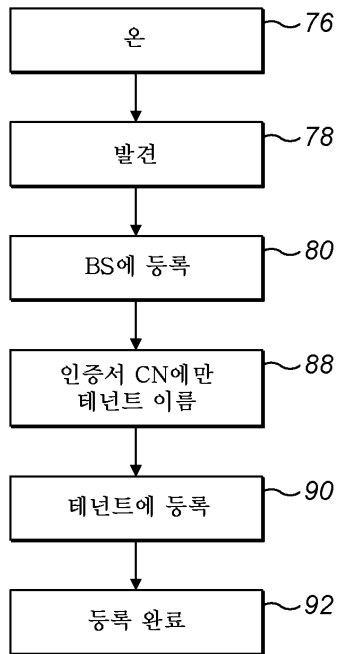
도면5b



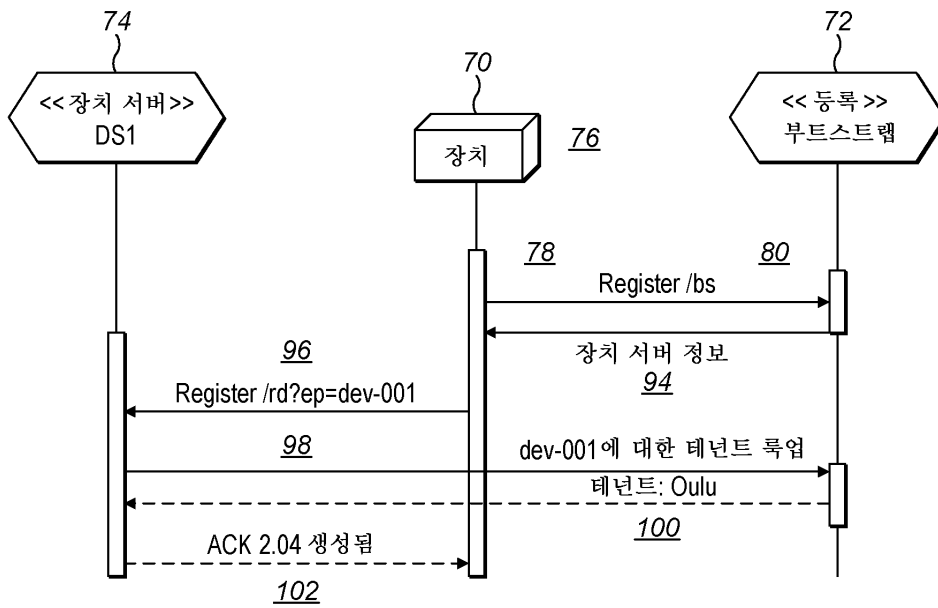
도면6a



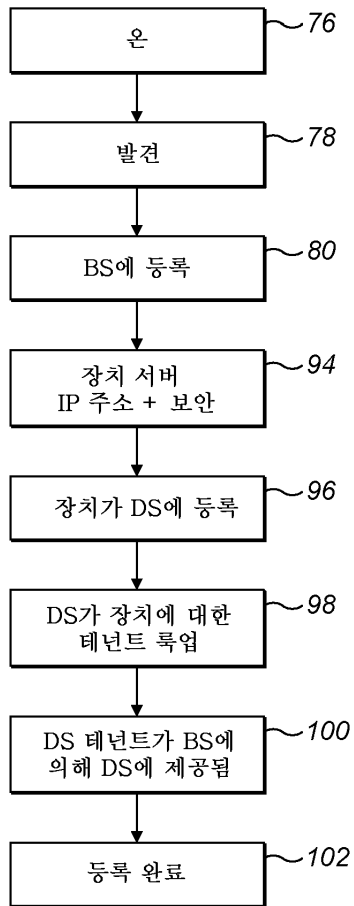
도면6b



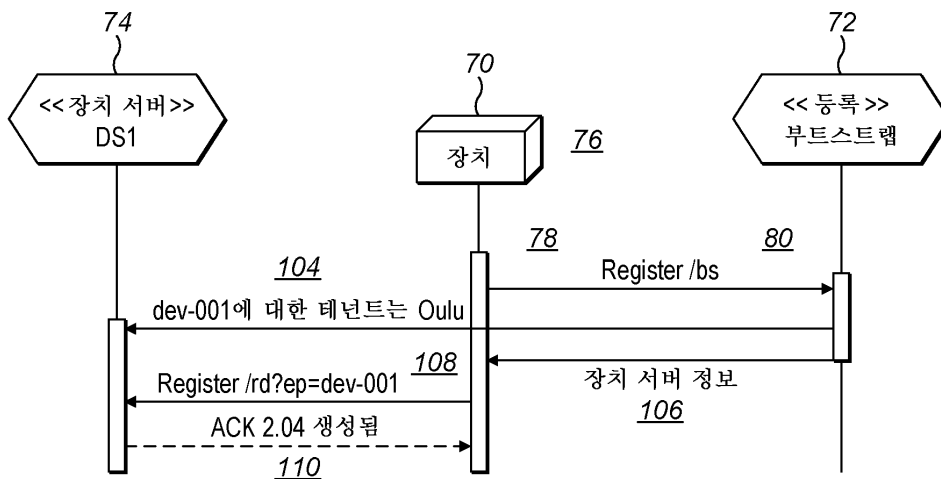
도면7a



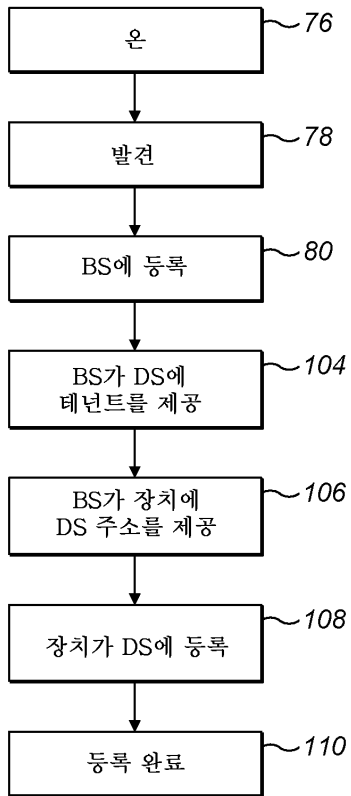
도면7b



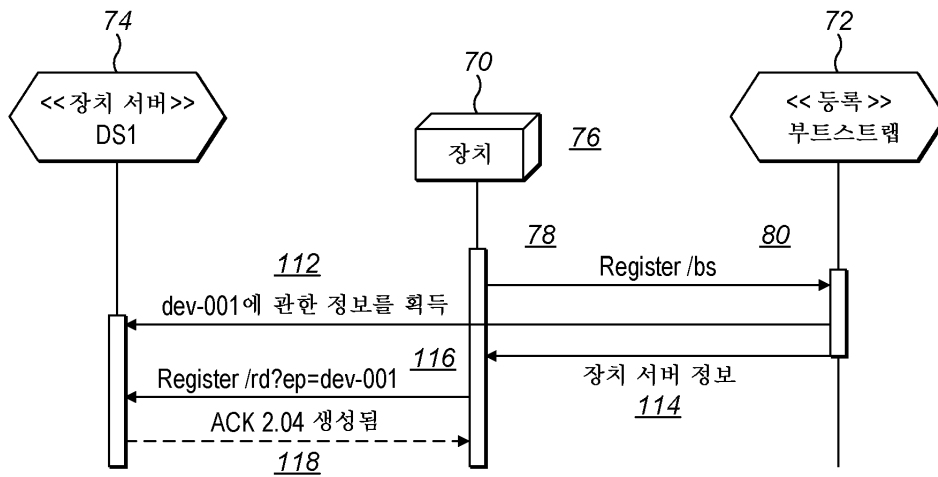
도면8a



도면8b



도면9a



도면9b

