



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0075808  
(43) 공개일자 2017년07월03일

(51) 국제특허분류(Int. Cl.)  
*G06F 9/455 (2006.01) G06F 9/50 (2006.01)*  
*H04L 29/08 (2006.01) H04W 4/00 (2009.01)*

(52) CPC특허분류  
*G06F 9/45558 (2013.01)*  
*G06F 9/50 (2013.01)*

(21) 출원번호 10-2017-7017055(분할)

(22) 출원일자(국제) 2014년05월08일  
심사청구일자 없음

(62) 원출원 특허 10-2015-7034701  
원출원일자(국제) 2014년05월08일  
심사청구일자 2015년12월04일

(85) 번역문제출일자 2017년06월21일

(86) 국제출원번호 PCT/US2014/037303

(87) 국제공개번호 WO 2014/182900  
국제공개일자 2014년11월13일

(30) 우선권주장  
61/820,966 2013년05월08일 미국(US)

(71) 출원인  
콘비다 와이어리스, 엘엘씨  
미국 19809-3727 브라운스빌 월밍턴 스위트 300  
밸리뷰 파크웨이 200

(72) 별명자  
왕, 총강  
미국 08540 뉴저지주 프린스턴 칼라일 코트 9  
시드, 데일, 엔.  
미국 18104 펜실베이니아주 앤더슨 뉴스 36번  
스트리트 229  
동, 리준  
미국 92130 캘리포니아주 샌디에고 아프리칸 홀리  
트레일 6085

(74) 대리인  
양영준, 백만기, 정은진

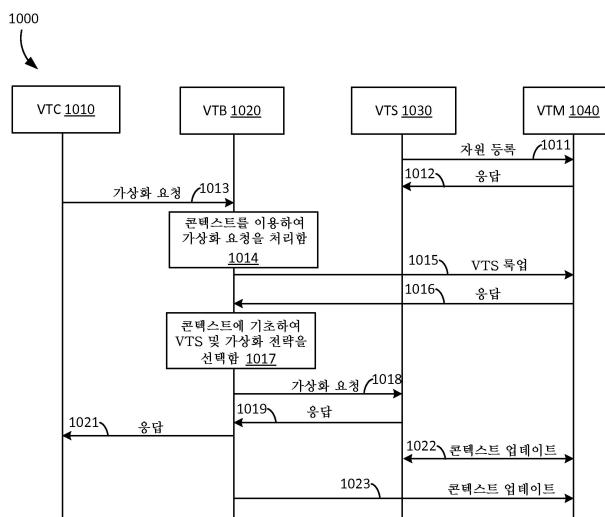
전체 청구항 수 : 총 1 항

(54) 발명의 명칭 가상화 브로커 및 콘텍스트 정보를 이용한 자원들의 가상화를 위한 방법 및 장치

### (57) 요약

접속된 엔티티들의 네트워크에서의 가상화 브로커에서 제1 엔티티로부터 제1 가상화 요청을 수신하는 것 - 제1 가상화 요청은 제1 엔티티와 연관된 콘텍스트 정보를 포함함 -; 가상화 브로커로부터 가상화 관리자로 가상화 서버 식별자에 대한 요청을 송신하는 것; 및 가상화 관리자로부터 가상화 서버 식별자를 수신하는 것에 의해, 자원들을 가상화하기 위한 방법 및 엔티티들이 기재되어 있다. 제2 가상화 요청이 가상화 서버 식별자와 연관된 가상화 서버로 송신될 수 있고, 자원이 가상화되었다는 표시를 포함하는 제1 응답이 가상화 서버로부터 수신될 수 있다. 이 표시를 포함하는 제2 응답이 가상화 브로커로부터 제1 엔티티로 송신될 수 있다. 이 방법은 사물 인터넷(IoT)의 콘텍스트에서 적용된다.

### 대 표 도 - 도10



(52) CPC특허분류

*HO4L 67/10* (2013.01)

*HO4L 67/148* (2013.01)

*HO4L 67/16* (2013.01)

*HO4L 67/18* (2013.01)

*HO4W 4/005* (2013.01)

---

## 명세서

### 청구범위

#### 청구항 1

제1항에 기재된 장치 및 방법.

### 발명의 설명

#### 기술 분야

[0001] 관련 출원들에 대한 상호 참조

[0002] 본 출원은, 2013년 5월 8일자로 출원되었으며 발명의 명칭이 "METHOD AND APPARATUS FOR IOT VIRTUALIZATION SERVICES"인 미국 가특허 출원 제61/820,966호의 혜택을 주장하며, 이 가특허 출원의 내용은 참고로 본 명세서에 포함된다.

### 배경기술

[0003] M2M(machine-to-machine) 기술들은 디바이스들이 유선 및 무선 통신 시스템들을 사용하여 서로 보다 직접적으로 통신하는 것을 허용한다. M2M 기술들은, 고유하게 식별가능한 객체들, 및 인터넷과 같은 네트워크를 통해 그리고 서로 통신하는 이러한 객체들의 가상 표현들의 시스템인 사물 인터넷(IoT: Internet of Things)의 추가 실현을 가능하게 한다. IoT는 평범한 일상적인 객체들, 예컨대 식료품점에서의 상품들 또는 홈에서의 어플라이언스들과의 통신을 용이하게 할 수 있고, 그에 의해 이러한 객체들의 지식을 향상시킴으로써 비용 및 낭비를 감소시킬 수 있다. 예를 들어, 상점들은, 재고에 있을 수 있거나 판매되었을 수 있는 객체들과 통신하거나 이러한 객체들로부터 데이터를 획득할 수 있는 것에 의해, 매우 정밀한 재고 데이터를 유지할 수 있다.

[0004] IoT는 객체들, 디바이스들, 게이트웨이들, 라우터들, 서버들, 사용자들 및 애플리케이션들과 같은 각종 물리적 엔티티들 또는 "사물들"을 포함한다. 각각의 IoT 엔티티는 가상화될 수 있는 상이한 프로토콜 계층들 상의 다양한 물리 자원들을 가질 수 있다. 프로토콜(예를 들어, MAC(media access control) 프로토콜 또는 라우팅 프로토콜)은 가상화를 통해 물리적 엔티티가 동적으로 지원할 수 있는 프로토콜들과 연관된 수개의 물리 자원 중 하나일 수 있다. 물리적 IoT 엔티티 자체도 또한 자원일 수 있다. 배치된 IoT 시스템 또는 물리적 IoT 엔티티는 그것의 구성 및 요건에 기초하여 동시에 또는 상이한 시간에 상이한 애플리케이션들을 지원할 수 있다. 예를 들어, 도시에 배치된 대규모 옥외 무선 이종 센서 네트워크는 공기 품질 모니터링 애플리케이션, 트래픽 모니터링 애플리케이션, 공중 안전 관리 애플리케이션 등을 지원할 수 있다. 다른 예에서, 홈에 설치된 물리적 IoT 게이트웨이는 원격 헬스케어 애플리케이션, 하우스 모니터링 애플리케이션 및 스마트 어플라이언스 애플리케이션과 같은 수개의 애플리케이션을 지원할 수 있다. 결과적으로, 다수의 애플리케이션 및/또는 사용자 사이에 이러한 자원들을 공유하는 것을 지원하기 위해 가상화가 이용될 수 있다.

[0005] IoT 엔티티들에 의한 자원들로의 액세스는 IoT의 성공적인 동작에 필수적이다. 다양한 IoT 엔티티들에 의해 필요할 수 있는 물리적 및 소프트웨어 자원들은, IoT 엔티티들이 자원들의 기저의 물리적 및/또는 소프트웨어 구현들의 지식을 가질 필요 없이 자원들에 대한 인터페이스들을 통해 이러한 자원들과 상호작용할 수 있는 방식으로 표현될 수 있다. IoT에서, 이것은 자원 "가상화"로 지칭될 수 있으며, 이용되는 자원들은 예를 들어 논리 인터페이스들의 이용을 통하여 요청 IoT 엔티티들에 서비스로서 제공될 수 있다. 가상화된 자원들의 예들은, 클라우드 제공자들이 사용자들에 대해 물리 또는 가상 머신으로서 클라우드 인프라스트럭처를 제공하는 IaaS(infrastructure as a service), 클라우드 제공자들이 클라우드 컴퓨팅 플랫폼(예를 들어, 오퍼레이팅 시스템, 프로그래밍 언어 실행 환경, 데이터베이스 및 웹 서버)을 제공하는 PaaS(platform as a service), 클라우드 제공자들이 클라우드에서 소프트웨어를 구동하고 사용자들로부터 소프트웨어에 대한 입력을 수신하고 사용자들에게 소프트웨어의 출력을 제공하는 SaaS(software as a service), 및 다른 가상화 가능 서비스들, 예컨대 SaaS(storage as a service), DaaS(data as a service) 등을 포함할 수 있다.

[0006] 그러나, 기존의 가상화 기술들은 서비스 계층 공통 기능성들이 아니며, IoT 시스템들에 대해 구체적으로 정의되지 않는다. 서비스 계층 가상화 기능성들은 ETSI TC M2M, TS 102 690(Architecture), 유럽 IoT-A 프로젝트,

유럽 iCore 프로젝트 및 유럽 버틀러 프로젝트(Europe Butler Project)에서 정의된 바와 같은 기존의 M2M/IoT 아키텍처들 및 서비스 능력들에서 충분히 개발되지 않는다. 관련 기술분야에서는, 공통 서비스로서 새로운 서비스 계층 가상화 기능성들을 포함하며, 효율적이고 융통성있는 IoT 자원 가상화의 통합을 지원할 수 있는 시스템들 및 방법들이 필요하다.

### 발명의 내용

[0007]

본 명세서에 개시된 실시예들은, 접속된 엔티티들의 네트워크에서의 가상화 브로커에서 제1 엔티티로부터 제1 가상화 요청을 수신하는 것 - 제1 가상화 요청은 제1 엔티티와 연관된 콘텍스트 정보를 포함함 -; 가상화 브로커로부터 가상화 관리자로 가상화 서버 식별자에 대한 요청을 송신하는 것; 및 가상화 관리자로부터 가상화 서버 식별자를 수신하는 것에 의해, 자원들을 가상화하기 위한 방법들을 포함한다. 제2 가상화 요청이 가상화 서버 식별자와 연관된 가상화 서버로 송신될 수 있고, 자원이 가상화되었다는 표시를 포함하는 제1 응답이 가상화 서버로부터 수신될 수 있다. 이 표시를 포함하는 제2 응답이 가상화 브로커로부터 제1 엔티티로 송신될 수 있다.

[0008]

본 명세서에 개시된 실시예들은 접속된 엔티티들의 네트워크에서의 가상화 관리자를 더 포함하는데, 이 가상화 관리자는, 저장된 명령어들을 실행할 때, 가상화 서버와 연관된 자원들을 등록하라는 요청을 수신하는 것; 자원들 및 가상화 서버와 연관된 콘텍스트 정보를 저장하는 것; 가상화 정책들을 가상화 서버로 푸시하는 것; 가상화 클라이언트 또는 가상화 브로커로부터의 루업 이력(lookup history)을 유지하는 것, 가상화 과금 레코드들을 발생시키는 것, 및 추천된 가상화 서버 식별자에 대한 요청을 가상화 브로커 또는 가상화 클라이언트로부터 수신하는 것 - 추천된 가상화 서버 식별자에 대한 요청은 콘텍스트 정보를 포함함 - 을 포함하는 동작들을 실시한다. 가상화 서버는 콘텍스트 정보에 기초하여 선택될 수 있고, 가상화 서버 식별자는 가상화 브로커 또는 가상화 클라이언트로 송신될 수 있다.

[0009]

본 명세서에 개시된 실시예들은 접속된 엔티티들의 네트워크에서의 가상화 서버 엔티티를 더 포함하는데, 이 가상화 서버 엔티티는, 저장된 명령어들을 실행할 때, 가상 자원을 생성하라는 요청을 제1 엔티티로부터 수신하는 것; 가상 자원을 생성하라는 요청을 수락하기로 결정하는 것; 가상 자원을 생성하는 것; 그 자신 및 물리 자원들을 가상화 관리 엔티티에 등록하는 것; 가상화 콘텍스트 정보를 가상화 관리 엔티티에 보고하는 것; 가상화 관리 엔티티로부터의 가상화 정책들에 액세스하는 것; 이전에 생성된 가상 자원을 조정하는 것; 이전에 생성된 가상 자원들을 결합하는 것; 및 가상 자원이 생성되었다는 것을 나타내는 응답을 제1 엔티티로 송신하는 것을 포함하는 동작들을 실시한다.

[0010]

이 개요는 상세한 설명에서 추가로 후술되는 개념들의 선택을 단순화된 형태로 소개하기 위해 제공된다. 이 개요는 청구 대상의 주요 특징들 또는 필수 특징들을 식별하는 것으로 의도되지도 않으며, 청구 대상의 범위를 제한하기 위해 이용되는 것으로 의도되지도 않는다. 또한, 청구 대상은 본 개시물의 임의의 부분에서 언급된 임의의 또는 모든 단점을 해결하는 한정들에 제한되지는 않는다.

### 도면의 간단한 설명

[0011]

도 1은 실시예에 따라 IoT 가상화 서비스들이 구현될 수 있는 예시적인 네트워크 구성을 도시한다.

도 2는 실시예에 따라 IoT 가상화 서비스들이 구현될 수 있는 예시적인 IoT 구성을 도시한다.

도 3은 실시예에 따라 IoT 가상화 서비스들이 구현될 수 있는 다른 예시적인 IoT 구성을 도시한다.

도 4는 실시예에 따른 예시적인 프로토콜 스택을 도시한다.

도 5는 실시예에 따라 IoT 가상화 서비스들이 구현될 수 있는 다른 예시적인 IoT 구성을 도시한다.

도 6은 실시예에 따라 IoT 가상화 서비스들이 구현될 수 있는 다른 예시적인 IoT 구성을 도시한다.

도 7은 실시예에 따라 IoT 가상화 서비스들이 구현될 수 있는 다른 예시적인 IoT 구성을 도시한다.

도 8은 실시예에 따라 IoT 가상화 서비스들이 구현될 수 있는 다른 예시적인 IoT 구성을 도시한다.

도 9는 실시예에 따라 IoT 가상화 서비스들이 구현될 수 있는 다른 예시적인 IoT 구성을 도시한다.

도 10은 실시예에 따른 IoT 가상화 서비스들을 위해 이용될 수 있는 예시적인 신호 흐름을 도시한다.

도 11은 실시예에 따른 IoT 가상화 서비스들을 위해 이용될 수 있는 다른 예시적인 신호 흐름을 도시한다.

도 12는 실시예에 따른 IoT 가상화 서비스들을 위해 이용될 수 있는 다른 예시적인 신호 흐름을 도시한다.

도 13은 실시예에 따른 IoT 가상화 서비스들을 위해 이용될 수 있는 다른 예시적인 신호 흐름을 도시한다.

도 14는 실시예에 따른 IoT 가상화 서비스들을 위해 이용될 수 있는 다른 예시적인 신호 흐름을 도시한다.

도 15는 실시예에 따른 IoT 가상화 서비스들을 위해 이용될 수 있는 다른 예시적인 신호 흐름을 도시한다.

도 16은 실시예에 따라 IoT 가상화 서비스들이 구현될 수 있는 예시적인 네트워크 구성을 도시한다.

도 17은 실시예에 따라 IoT 가상화 서비스들이 구현될 수 있는 다른 예시적인 네트워크 구성을 도시한다.

도 18a는 하나 이상의 개시된 실시예가 구현될 수 있는 예시적인 M2M(machine-to-machine) 또는 사물 인터넷 (IoT) 통신 시스템의 시스템 도면이다.

도 18b는 도 18a에 도시된 M2M/IoT 통신 시스템 내에서 이용될 수 있는 예시적인 아키텍처의 시스템 도면이다.

도 18c는 도 18a에 도시된 통신 시스템 내에서 이용될 수 있는 예시적인 M2M/IoT 단말 또는 게이트웨이 디바이스의 시스템 도면이다.

도 18d는 도 18a의 통신 시스템의 양태들이 구현될 수 있는 예시적인 컴퓨팅 시스템의 블록도이다.

### 발명을 실시하기 위한 구체적인 내용

[0012]

본 명세서에 제시된 실시예들은 REST(representational state transfer) 아키텍처에 관하여 설명될 수 있으며, 설명된 컴포넌트들 및 엔티티들은 REST 아키텍처(REST적인 아키텍처)의 제약들에 따른다. REST적인 아키텍처는, 이용되는 통신 프로토콜 또는 물리적 컴포넌트 구현에 관하여 설명되는 것보다는, 아키텍처에서 이용되는 컴포넌트들, 엔티티들, 커넥터들 및 데이터 요소들에 적용된 제약들에 관하여 설명된다. 따라서, 컴포넌트들, 엔티티들, 커넥터들 및 데이터 요소들의 역할들과 기능들이 설명될 것이다. REST적인 아키텍처에서, 고유하게 어드레싱 가능한 자원들의 표현들이 엔티티들 사이에 전송될 수 있다. 관련 기술분야의 통상의 기술자라면, 본 실시예들의 구현들은 본 개시물의 범위를 유지하면서 변경될 수 있다는 것을 인식할 것이다. 관련 기술분야의 통상의 기술자라면, 개시된 실시예들은 예시적인 실시예들을 설명하기 위해 본 명세서에서 이용되는 ETSI M2M 아키텍처를 이용하는 구현들에 제한되지는 않는다는 것을 또한 인식할 것이다. 개시된 실시예들은 다른 아키텍처들 및 시스템들, 예컨대 oneM2M 및 다른 M2M 시스템들 및 아키텍처들에서 구현될 수 있다.

[0013]

도 1은 본 개시물의 예시적인 실시예를 도시한다. 시스템(100)은 에지 네트워크(110)를 포함할 수 있고, 이 에지 네트워크 내에 IoT 엔티티(130) 및 IoT 게이트웨이(140)가 존재할 수 있다. IoT 엔티티(130)는 물리 자원(132)을 가질 수 있다. 물리 자원(132)은, 스토리지, 계산 또는 CPU, 통신 대역폭, 다른 제공 서비스들, 센서들, 액추에이터들 등을 포함하지만 이에 제한되지는 않는 임의의 물리 자원일 수 있다. 물리 자원(132)은, 서비스 계층 인터페이스(135)를 통해 액세스가능하고 가상화 서비스(131)를 통해 제공되는 가상 자원(133)으로서 다른 IoT 엔티티들에 의한 이용을 위해 이용가능해질 수 있다. 마찬가지로, IoT 게이트웨이(140)는 물리 자원(142)을 가질 수 있는데, 이 물리 자원도 또한 임의의 물리 자원일 수 있으며, 서비스 계층 인터페이스(135)를 통해 액세스가능하고 가상화 서비스(141)를 통해 제공되는 가상 자원(143)으로서 다른 IoT 엔티티들에 의한 이용을 위해 이용가능해질 수 있다. IoT 엔티티(130) 및 IoT 게이트웨이(140) 각각은 다른 것의 가상 자원들을 이용할 수 있다는 점에 유의한다. 가상 자원에 대한 액세스를 제공하는 논리 인터페이스를 통한 상호작용의 방향에 대해 제한이 고려되지는 않는다. 즉, 많은 실시예에서, IoT 엔티티(130)는 서비스(141)를 이용할 수 있으며, IoT 게이트웨이(140)는 서비스(131)를 이용할 수 있다.

[0014]

가상화된 자원들은 또한 네트워크들에 걸쳐 이용가능할 수 있다. 코어 네트워크(120)에 위치된 IoT 라우터(150)는 물리 자원(152)을 가질 수 있는데, 이 물리 자원도 또한 임의의 물리 자원일 수 있으며, 서비스 계층 인터페이스(145)를 통해 IoT 게이트웨이(140)와 같은 에지 네트워크(110)에서의 엔티티들에 의해 액세스가능하고 가상화 서비스(151)를 통해 제공되는 가상 자원(153)으로서 다른 IoT 엔티티들에 의한 이용을 위해 이용가능해질 수 있다. 여기서도, IoT 라우터(150) 및 IoT 게이트웨이(140) 각각은 다른 것의 가상 자원들을 이용할 수 있다. 코어 네트워크(120)에 위치된 IoT 서버(160)는 물리 자원(162)을 가질 수 있는데, 이 물리 자원도 또한 임의의 물리 자원일 수 있으며, 서비스 계층 인터페이스(155)를 통해 IoT 라우터(150)와 같은 코어 네트워크(120)에서의 엔티티들에 의해 액세스가능하고 가상화 서비스(161)를 통해 제공되는 가상 자원(163)으로서 다른 IoT 엔티티들에 의한 이용을 위해 이용가능해질 수 있다. IoT 라우터(150) 및 IoT 서버(160) 각각은 다른 것의 가상 자원들을 이용할 수 있다. IoT 사용자(170)는 서비스 계층 인터페이스(165)를 통해 가상화된 서비스들에

액세스하기 위해 가상화 서비스(171)를 이용할 수 있다. IoT 사용자(170)는, 서비스 계층 인터페이스(165)를 통해 IoT 서버(160)의 서비스(161)를 이용하는 것과 같이, 다른 IoT 엔티티들의 물리 자원들을 나타내는 서비스들을 또한 이용할 수 있다. 이러한 상호작용 및 가상화의 부가적인 상세는 본 명세서에 제시된다.

[0015] 도 2는 본 개시물의 다른 예시적인 실시예를 도시한다. 시스템(200)은 사용자로부터 가상 자원을 생성하라는 요청을 수락하도록 구성될 수 있는 IoT 게이트웨이(210)를 포함할 수 있다. 이러한 요청에 응답하여, IoT 게이트웨이(210)는, 물리 자원(212)과 연관되며 가상화 서비스(211)를 통해 이용가능해지는 가상 자원(213)을 생성할 수 있다. IoT 사용자(230)는, 그것의 가상화 서비스(231)를 통해, 서비스 계층 인터페이스(225)를 경유하여 IoT 서버(220)의 가상화 서비스(221)에 가상 자원이 생성되는 것을 요청할 수 있다. IoT 서버(220)는 이러한 요청에 대한 적절한 수신자가 IoT 게이트웨이(210)라고 결정할 수 있으며, 이 요청을 서비스 계층 인터페이스(215)를 통해 IoT 게이트웨이(210)의 서비스(211)에 포워딩할 수 있다. 이에 응답하여, IoT 게이트웨이(210)는 요청된 가상 자원을 생성할 수 있다.

[0016] 도 3은 본 개시물의 다른 예시적인 실시예를 도시한다. 시스템(300)은 IoT 엔티티(310)를 포함할 수 있고, 이 엔티티는 온도 모니터링을 위해 홈에 설치된 센서 노드 등과 같은 임의의 IoT 엔티티일 수 있으며, 그 자신의 자원들이 가상화되어 서비스로서 다른 IoT 엔티티들에 이용가능해지는 것을 요청할 수 있다. IoT 엔티티(310)는, 그것의 가상화 서비스(311)를 통해, IoT 서버(320)의 가상화 서비스(321)가 가상화 서비스(311)를 이용가능하게 하는 것을 서비스 계층 인터페이스(315)를 경유하여 요청할 수 있다. 이에 응답하여, IoT 서버(320)는 일부 실시예들에서는 가상화 서비스(321)를 통해 가상화 서비스(311)를 광고하거나 또는 다른 방식으로 이 가상화 서비스를 이용가능하게 할 수 있다. 서비스(321)는 IoT 서버(320)의 물리 자원(322)과 연관된 가상 자원(323)에 대한 가상 자원 인터페이스를 또한 제공할 수 있다. 도 1, 도 2 및 도 3에 설명된 시스템들 각각에서, 그리고 본 명세서에 설명된 나머지 실시예에서, 생성된 가상 자원들은 다른 IoT 엔티티들, 예컨대 IoT 사용자들, IoT 서버들, IoT 라우터들, IoT 게이트웨이들, IoT 사물들/디바이스들 등에 의해 액세스되거나 소비될 수 있다.

[0017] 도 4는 예시적인 IoT 엔티티(400)에 구현된 프로토콜 스택에서의 예시적인 가상화 서비스들의 예시적인 위치를 도시한다. 프로토콜 스택은 물리 계층(410), MAC 계층(420), 네트워크 계층(430), 메시징 및/또는 전송 계층(440), 서비스 계층(450) 및 애플리케이션 계층(460)을 포함할 수 있다. 가상화 서비스들은 메시징/전송 계층(440)의 위에 또는 서비스 계층(450)에 존재할 수 있다. 따라서, 예시적인 가상화 관련 커맨드들 및 메시지들은 HTTP, TCP, CoAP, UDP, XMPP 등과 같은 상이한 메시징 및/또는 전송 프로토콜들의 페이로드에 포함될 수 있다.

[0018] 예시적인 가상화 서비스들에서, 물리 자원은 상이한 사용자들에 의해 각각 공유될 수 있는 다수의 가상 자원을 발생시킬 수 있다. 본 명세서에 제시된 바와 같은 하나 이상의 가상화 서비스에 의한 이러한 가상 자원들의 생성 시에, 각각의 가상 자원은 투명한 방식으로 요청되고, 액세스되고 또는 다양한 사용자들에게 할당될 수 있다. 예를 들어, 네트워크 제공자는 도로들을 따라 또는 커뮤니티에 근접하거나 그 주위의 다른 곳에 액세스 포인트들 및 게이트웨이들과 같은 수개의 타입의 네트워크 자원들을 배치할 수 있다. 실시예에서, 이러한 물리적 액세스 포인트들은 상이한 IoT 서비스 제공자들 및/또는 다른 타입의 서비스 제공자들에 의해 공유될 수 있다. 개시된 가상화 서비스들은 물리 자원 제공자와 서비스 제공자 사이의 효율적이고 융통성있는 가상화 관리를 가능하게 할 수 있다. 이러한 관리는 다른 기능들 중에서 가상 자원 생성, 모니터링 및 소거(cancellation)를 포함할 수 있다. 물리적 액세스 포인트 제공자들은, 제공된 물리적 액세스 포인트가 가상화 커맨드들을 수신하고 대응하는 가상 자원들을 생성할 수 있도록 개시된 가상화 기능들(예를 들어, 본 명세서에 설명된 바와 같은 가상화 서버)을 지원할 수 있다.

[0019] 예를 들어, 실시예에서, 게이트웨이 또는 액세스 포인트는 센서 네트워크를 인터넷에 접속할 수 있다. 센서 네트워크에서, 각각의 센서 노드는 IoT 디바이스, 예컨대 온도 센서 또는 압력 센서일 수 있다. 센서 네트워크는, 본 명세서에 제시된 실시예들을 이용하여, IoT 서비스 제공자에 의해 배치 및 소유되거나, 또는 네트워크 제공자에 의해 소유 및/또는 배치되고, IoT 서비스 제공자에 의해 액세스되거나 다른 방식으로 이용될 수 있다.

[0020] 도 5는 각종 표준들과 함께 이용될 수 있는 예시적인 가상화 서비스 아키텍처(500)를 도시한다. 고려된 표준 애플리케이션들 중 일부는 개시된 가상화 서비스들에 대한 애플리케이션 예들을 제공하는 이하의 표 1에 열거되어 있다.

표 1

[0021]

| 표준들         | 가능한 애플리케이션들  |
|-------------|--|
| ETSI TC M2M | 예시적인 아키텍처 및 가상화 서비스 기능들(즉, VTC, VTS, VTB 및 VTM)은 M2M 서비스 능력 계층(SCL: service capability layer)에 통합될 수 있다. 예를 들어, VTM은 네트워크 SCL(NSCL)과 통합할 수 있고, VTB는 게이트웨이 SCL(GSCL)과 통합할 수 있고, VTS는 디바이스 SCL(DSCL)과 통합할 수 있다.<br>● 예시적인 가상화 서비스 절차들을 지원하기 위해 새로운 자원 구조들이 개발될 수 있다.   |
| oneM2M      | oneM2M은 M2M 서비스 계층을 타깃으로 하기 때문에, 예시적인 가상화 아키텍처 및 서비스 기능들은 잠재적으로는 그것의 범위 내에 있으며, 진보된 M2M 서비스 기능들로서 채택될 수 있다.  |
| 3GPP        | 3세대 파트너십 프로젝트(3GPP)는, 기지국 가상화 및 공유와 같은 네트워크 공유(참고로 본 명세서에 완전히 포함되는 3GPP TS 23.251: "Network sharing; Architecture and functional description"에서 논의된 바와 같음)에 대한 연구/작업 항목을 갖는다. 또한, 물리적 스마트 폰을 다수의 가상 폰으로 가상화하는 것이 가능하다. 결과적으로, 예시적인 가상화 아키텍처 및 서비스 기능들은 3GPP 제어 평면에 통합될 수 있다.<br>● 예시적인 가상화 절차들을 지원하기 위해 NAS(non-access stratum) 또는 단문 메시지 서비스(SMS) 또는 다른 접근법들에 기초한 새로운 시그널링 프로토콜들이 개발될 수 있다. |
| IETF/IRTF   | 예시적인 가상화 아키텍처 및 서비스 기능들은 (참고로 본 명세서에 완전히 포함되는 3GPP TS 23.251: "Network sharing; Architecture and functional description"에서 논의된 바와 같음) IETF COMAN(constrained management) 및 IRTF(Internet Research Task Force) SDNRG(software defined networking research group)에 적용될 수 있다. 예를 들어, 예시적인 가상화 아키텍처 및 서비스 기능들을 위해 새로운 관리 데이터 모델이 개발될 수 있다.   |
| OMA         | 가상화는 디바이스 관리(DM) 기능들의 소정 탑입일 수 있다. 예시적인 가상화 아키텍처 및 서비스 기능들은 OMA(Open Mobile Alliance), 특히 OMA 디바이스 관리(DM)에 적용될 수 있다.<br>● 예시적인 가상화 절차들을 지원하기 위해 새로운 관리 객체들이 개발될 수 있다.  |
| ONF         | 예시적인 가상화 아키텍처 및 서비스 기능들은, 현재 OpenFlow가 라우팅 테이블들의 동적 구성만을 지원하기 때문에, 가상화 기능들에 대한 지원을 부가함으로써, (참고로 본 명세서에 완전히 포함되는 3GPP TS 23.251: "Network sharing; Architecture and functional description"에서 논의된 바와 같음) ONF(Open Networking Foundation) OpenFlow 프로토콜을 확장할 수 있다.   |

[0022]

도 5에서, 일반 가상화 서비스 아키텍처(500)는 상이한 자원 도메인들에서 IoT 자원들이 어떻게 관리될 수 있는지를 도시한다. "자원 도메인"의 정의는 일부 실시예들에서 변할 수 있다. 예를 들어, IoT 서비스 제공자의 요소들 및 IoT 시스템들 모두는 단일의 자원 도메인으로서 간주될 수 있거나, 또는 IoT 서비스 제공자는 자원 위치, 자원 탑입, 지원 서비스 및/또는 애플리케이션 카테고리, 확장성 등과 같은 기준들에 기초하여 모든 IoT 자원을 다수의 도메인으로 분할할 수 있다.

[0023]

도 5에 도시된 예시적인 일반 가상화 서비스 아키텍처에서, 자원 도메인들(501 및 502) 각각은, 가상화 관리자 ("VTM" 및 "가상화 관리 엔티티"로 또한 지칭될 수 있음), 가상화 서버("VTS" 및 "가상화 서버 엔티티"로 또한 지칭될 수 있음), 가상화 클라이언트("VTC" 및 "가상화 클라이언트 엔티티"로 또한 지칭될 수 있음) 및 가상화 브로커("VTB" 및 "가상화 브로커 엔티티"로 또한 지칭될 수 있음)의 기능들을 설명하는 이하의 표 2에 열거된 요소들 중 하나 이상을 포함할 수 있다. 실시예에서, 각각의 요소는 논리 서비스 기능 및 가상화 서비스이다. 따라서, 각각의 가상화 요소는 도 1 내지 도 3에 도시된 바와 같은 다양한 IoT 엔티티들에 존재할 수 있다. 각각의 가상화 요소의 기능들은 본 명세서에서 보다 상세하게 설명된다.

표 2

[0024]

| 아키텍처 요소들  | 기능들   |
|---|---|
| 가상화 관리자(VTM)<br>(예를 들어, 도 5의 510, 511, 512 및 520) | <ul style="list-style-type: none"> <li>● 자원들, 가상화 정책들 및 가상화 콘텍스트를 관리함.</li> <li>● VTC와 VTS(또는 VTB) 사이의, VTB들 사이의 또는 VTS들 사이의 상호작용들을 용이하게 함.</li> <li>● 적절한 VTS를 식별하기 위해 가상화 알고리즘들을 수행함.</li> <li>● 가상화 정책 및 콘텍스트 정보에 의해 구동되는 동적 가상화 조정 또는 적응을 용이하게 함/트리거함.</li> </ul> |

|   |   |
|---|---|
| 가상화 서버(VTS)<br>(예를 들어, 도 5의 515, 521 및 522)       | <ul style="list-style-type: none"> <li>● 자원들을 VTM에 등록함.</li> <li>● VTC 또는 VTB에 의해 요청된 바와 같이 가상 자원들을 생성함.</li> <li>● 가상 자원 이동(virtual resource migration)과 같은 협력 가상화를 가능하게 하기 위해 다른 VTS와 협력함.</li> <li>● VTM/VTB/VTC, 가상화 콘텍스트 또는 가상화 정책에 의해 트리거되는 바와 같이 동적 가상화 조정 또는 적용을 실행함.</li> <li>● 가상 자원들의 성능을 모니터링하여, VTM에 보고함.</li> </ul> |
| 가상화 클라이언트<br>(VTC)<br>(예를 들어, 도 5의 516 및 527)     | <ul style="list-style-type: none"> <li>● VTM으로부터 적절한 VTS 또는 VTB를 발견함.</li> <li>● VTS 또는 VTB에 대해 가상화 요청들을 발행함.</li> </ul>  |
| 가상화 브로커(VTB)<br>(예를 들어, 도 5의 513, 514, 523 및 524) | <ul style="list-style-type: none"> <li>● VTC와 VTS 사이의 상호작용들을 도움.</li> <li>● VTC로부터의 가상화 요청들을 처리 및 변환함.</li> <li>● 적절한 VTS를 식별하기 위해 가상화 알고리즘들을 수행함.</li> <li>● 레거시 VTS 및 VTC와 예시적인 VTS 및 VTC를 상호연동함.</li> </ul>  |
| 레거시 VTS(예를 들어, 도 5의 525)                          | <ul style="list-style-type: none"> <li>● 통상의 또는 사유(proprietary) 가상화 프로토콜들만을 지원함.</li> </ul>   |
| 레거시 VTC(예를 들어, 도 5의 526)                          | <ul style="list-style-type: none"> <li>● 통상의 또는 사유 가상화 프로토콜들만을 지원함.</li> </ul>  |

[0025] 일반 가상화 서비스 아키텍처(500)에서, 가상화 요소들은 본 명세서에 설명된 가상화 기능들을 실현하기 위해 이하의 표 3에 열거된 예시적인 가상화 인터페이스들을 통해 서로 상호작용할 수 있다.

표 3

| 인터페이스들                                  | 설명  |
|---|---|
| VT1<br>(예를 들어, 도 5의 539, 565)           | <p>다음의 상호작용들을 지원하기 위한 VTS와 VTC 사이의 인터페이스:</p> <ul style="list-style-type: none"> <li>● 기본적으로, VTC는 이 인터페이스를 통하여 VTS에 대해 가상화 요청들을 발행하고 VTS로부터 응답을 수신한다.</li> <li>● 나중에, VTC는, 계류 중인 가상화 요청을 소거하거나 또는 생성된 가상 자원을 조정/검색/삭제하라는 요청들을 VTS에 대해 발행할 수 있다.</li> </ul>  |
| VT2<br>(예를 들어, 도 5의 554)                | <p>다음의 상호작용들을 지원하기 위한 VTC와 VTM 사이의 인터페이스:</p> <ul style="list-style-type: none"> <li>● VTC는 VTM으로부터 적절한 VTS 또는 VTB를 발견한다.</li> </ul>  |
| VT3<br>(예를 들어, 도 5의 552)                | <p>다음의 상호작용들을 지원하기 위한 VTS와 VTM 사이의 인터페이스:</p> <ul style="list-style-type: none"> <li>● VTS는 그것의 자원들을 VTM에 등록하고, VTM에 등록된 자원들을 업데이트할 수 있다.</li> <li>● VTS는 그 자신 및 그것의 자원들에 대한 콘텍스트 정보를 VTM에 보고한다.</li> <li>● VTS는 VTM에 유지된 가상화 정책들 및 가상화 콘텍스트 정보에 액세스한다.</li> <li>● VTM은 VTS에서의 새로운 가상화 정책들을 업데이트하고 시행한다.</li> </ul> |
| VT4<br>(예를 들어, 도 5의 531, 532 및 551)     | <p>다음의 상호작용들을 지원하기 위한 VTM과 다른 VTM 사이의 인터페이스:</p> <ul style="list-style-type: none"> <li>● VTM들은 VTS와 VTB 및 그들의 자원에 대한 정보 및 콘텍스트 정보를 교환한다.</li> <li>● VTM들은 VTC들로부터의 발견 요청들을 협력 서빙한다.</li> <li>● VTM들은 VTB들 또는 VTS들로부터 오프로딩되는 가상화 계산을 협력 서빙한다.</li> <li>● VTM은 가상화 정책들 및 가상화 콘텍스트를 협력 관리한다.</li> </ul>               |
| VT5<br>(예를 들어, 도 5의 538, 540, 562, 560) | <p>다음의 상호작용들을 지원하기 위한 VTC와 VTB 사이의 인터페이스:</p> <ul style="list-style-type: none"> <li>● VTC는 이 인터페이스를 통하여 VTB에 대해 가상화 요청들을 발행한다.</li> <li>● 나중에, VTC는 계류 중인 가상화 요청을 소거하거나 또는 생성된 가상 자원을 조정/검색/삭제하라는 요청들을 VTB에 대해 발행할 수 있다.</li> </ul>  |
| VT6<br>(예를 들어, 도 5의 537, 557, 561)      | <p>다음의 상호작용들을 지원하기 위한 VTS와 VTB 사이의 인터페이스:</p> <ul style="list-style-type: none"> <li>● VTB는 이 인터페이스를 통해 가상화 요청들을 VTS로 포워딩한다.</li> <li>● 나중에, VTB는 계류 중인 가상화 요청을 소거하거나 또는 생성된 가상 자원을 조정/검색/삭제하라는 요청들을 VTB에 대해 발행할 수 있다.</li> </ul>   |
| VT7<br>(예를 들어, 도 5의 536, 559)           | <p>다음의 상호작용들을 지원하기 위한 VTB와 다른 VTB 사이의 인터페이스:</p> <ul style="list-style-type: none"> <li>● 하나의 VTB는 다른 (하나 이상의) VTB들로 가상화 요청들을 포워딩할 수 있다.</li> <li>● 하나의 VTB는 다른 VTB들에 유지된 계류 중인 가상화 요청들을 소거할 수 있다.</li> </ul>   |

|  |  |
|--|--|
| VT8<br>(예를 들어, 도 5의 558)                 | 다음의 상호작용들을 지원하기 위한 VTS와 다른 VTS 사이의 인터페이스:<br>● VTS는 생성된 가상 자원을 하나의 VTS로부터 다른 VTS로 이동시킨다.<br>● VTS는 다른 VTS로부터 더 많은 자원들을 요청하거나 빌린다.  |
| VT9<br>(예를 들어, 도 5의 534, 535, 553 및 555) | 다음의 상호작용들을 지원하기 위한 VTB와 VTM 사이의 인터페이스:<br>● VTB는 VTM에 그 자신을 등록하고 그것의 콘텍스트 정보를 보고한다.<br>● VTB는 VTM에 유지된 가상화 정책들 및 가상화 콘텍스트 정보에 액세스한다.<br>● VTM은 VTB에서의 새로운 가상화 정책들을 업데이트하고 시행한다.<br>● VTB는 VTM에 가상화 계산을 오프로딩한다. |
| VT10<br>(예를 들어, 도 5의 563)                | 다음의 상호작용들을 지원하기 위한 레거시 VTS와 VTB 사이의 인터페이스:<br>● VTB는 VTC로부터의 가상화 요청을 레거시 VTS가 이해할 수 있는 레거시 가상화 요청으로 변환한다.  |
| VT11<br>(예를 들어, 도 5의 564)                | 다음의 상호작용들을 지원하기 위한 레거시 VTC와 VTB 사이의 인터페이스:<br>● VTB는 레거시 VTC로부터의 레거시 가상화 요청을 VTB 또는 VTS가 이해할 수 있는 포맷으로 변환한다.   |

[0027] 예시적인 실시예에서, VTM(예를 들어, 도 5의 510, 511, 512 및 520)은 자원 도메인 내의 IoT 자원들의 목록을 유지할 수 있다. VTM은 가상화 관련 정책 및 콘텍스트 정보를 또한 유지할 수 있다. VTM은 VTB, VTS 및/또는 다른 VTM에서의 가상화 정책을 업데이트하고 시행할 수 있다. 예시적인 실시예에서, 예를 들어 VTM들(510, 511 및 512)이 자원 도메인(501)에 존재하는 도 5에서 알 수 있는 바와 같이, 자원 도메인은 하나의 또는 하나보다 많은 VTM을 가질 수 있다. VTM은 VTC, VTB 또는 VTM 중 하나 이상과의 통신을 통해 단순한 자원 발견을 수행할 수 있으며, 그리고 또는 그 대신에 복잡한 가상화 알고리즘들을 이용하여, VTC, VTB 또는 VTM에 의해 요청된 바와 같은 적절한 자원들을 식별할 수 있다. VTM은 자원 또는 VTS, VTB 또는 VTC가 하나의 자원 도메인으로부터 다른 자원 도메인으로 이동하는 것을 도울 수 있다. 하나의 자원 도메인 내의 다수의 VTM(예를 들어, 도 5의 VTM들(510, 511 및 512))은 계층 또는 메시 구조를 형성하여, 자원 유지보수 및 발견 성능을 향상시킬 수 있다. VTM은 VTS들 사이의 또는 VTB들 사이의 협력을 또한 용이하게 할 수 있다.

[0028] 예시적인 실시예에서, VTS(예를 들어, 도 5의 515, 521 및 522)는 하나 이상의 물리 자원에 액세스할 수 있다. VTS는 VTC 또는 VTB로부터 가상화 요청을 수신할 수 있고, 이에 응답하여, 요청된 가상 자원을 생성할 수 있다. 가상 자원들은 VTM으로부터 수신된 하나 이상의 정책에 포함된 파라미터들 및 요건들에 따라 생성될 수 있다. VTS는 VTM에 그것의 자원들을 등록하고 업데이트할 수 있다. VTS들은 상호-VTS 협력 자원 가상화(cross-VTS collaborative resource virtualization) 및 관리를 가능하게 하기 위해서 서로 통신할 수 있다.

[0029] VTC들(예를 들어, 도 5의 516 및 527)은 가상 자원 생성을 요청하기 위해 가상화 요청들을 VTS 또는 VTB로 송신 할 수 있다. 이러한 요청을 송신하기 이전에, VTC는 이 요청을 송신할 적절한 VTS 및/또는 VTC를 결정하기 위해서 발견 요청을 VTM에 송신할 수 있다.

[0030] VTB(예를 들어, 도 5의 513, 514, 523 및 524)는 중간 노드로서 VTC와 VTS 사이의 상호작용들을 도울 수 있다. 예를 들어, VTB는 VTC로부터의 착신 가상화 요청들을 지능형으로 처리할 수 있다(예를 들어, VTB는 요청들을 VTS로 포워딩하기 이전에 이 요청들을 버퍼링 및/또는 병합할 수 있다). 다수의 VTB는 성능 및 확장성을 향상 시키기 위해 계층 또는 메시 브로커 구조를 형성할 수 있다. VTB는 레거시 VTS들 및 VTC들(예를 들어, 각각 도 5의 525 및 526)과 개시된 VTS들 및 VTC들 사이의 통신 및 상호운용성을 용이하게 할 수 있다.

[0031] 이하의 표 4는 수개의 예시적인 IoT 가상화 시나리오들, 및 VTC들, VTS들 및 VTB들의 예시적인 위치들을 열거한다. 개시된 가상화 인터페이스들을 지원할 수 있는 예시적인 프로토콜들이 또한 표 4에 제시된다. 예를 들어, 가상화 요청 메시지들 및 대응하는 응답들은 다른 것들 중에서 HTTP 및 CoAP와 같은 프로토콜들에 구속될 수 있다. HTTP 또는 CoAP와 같은 프로토콜들은 다양한 가상화 서비스 요청 및 응답 메시지들을 반송하기 위한 기저의 전송 프로토콜로서 이용될 수 있다. 가상화 요청 및 응답 메시지들은 HTTP 또는 CoAP 메시지의 페이로드 내에 캡슐화될 수 있다. 대안적으로 또는 부가적으로, 가상화 서비스 요청 또는 응답 내의 소정의 정보는 HTTP 및/또는 CoAP 헤더들 및/또는 옵션들 내의 필드들에 구속될 수 있다. 예를 들어, 실시예에서, 가상화 서비스 요청들 및 응답 프로토콜 프리미티브들은 HTTP 또는 CoAP 요청 및 응답 메시지들의 페이로드에서 반송될 수 있는 JSON(JavaScript Object Notation) 또는 XML(eXtensible Markup Language) 설명들로서 인코딩될 수 있다. 따라서, 가상화 기능들 및 서비스들은 가상화 서비스 프로토콜 JSON 및/또는 XML 프리미티브들을 인코딩 및/또는 디코딩할 수 있고, 이러한 가상화 서비스 프리미티브들을 교환하기 위한 기저의 전송으로서 HTTP 및/또는 CoAP를 이용할 수 있다. 다른 실시예들에서, 기존의 프로토콜들을 이용하는 것보다는, 예시적인 가상화 인터페

이스들 및 이러한 인터페이스들을 이용하여 전송된 메시지들은 새로운 가상화 서비스 프로토콜로서 구현될 수 있다.

표 4

| [0032]             | 시나리오들  | 설명  | 가상화 서비스 기능들  | 가상화 인터페이스 및 프로토콜 예 |
|--------------------|--|---|--|--------------------|
| 1. IoT 디바이스 가상화    | 3GPP MTC 디바이스는 통합된 다수의 센서(예를 들어, 온도, 연기, 습도 등)를 갖는다. IoT 네트워크 애플리케이션은 온도 센서를 갖는 가상 MTC 디바이스를 생성하라고 요청하기 위해 직접적으로 MTC 디바이스에 신호한다.   | VTC: IoT 네트워크 애플리케이션.<br>VTS: MTC 디바이스에서의 서비스 기능.<br>VTM: 3GPP 코어 네트워크에서의 서비스 기능.   | VT1: HTTP 프로토콜<br>VT2: HTTP 프로토콜<br>VT3: HTTP 프로토콜 |                    |
| 2. IoT 디바이스 가상화    | 3GPP MTC 디바이스는 통합된 다수의 센서(예를 들어, 온도, 연기, 습도 등)를 갖는다. IoT 네트워크 애플리케이션은 습도 센서를 갖는 가상 MTC 디바이스를 생성하라고 요청하기 위해 간접적으로 3GPP 코어 네트워크 엔티티들(예를 들어, MTC-IWF)을 통해 MTC 디바이스에 신호한다.   | VTC: IoT 네트워크 애플리케이션.<br>VTB: 3GPP 코어 네트워크 엔티티(예를 들어, MTC-IWF)에서의 서비스 기능.<br>VTS: MTC 디바이스에서의 서비스 기능.<br>VTM: 3GPP 코어 네트워크에서의 서비스 기능. | VT5: HTTP 프로토콜<br>VT6: HTTP 프로토콜<br>VT9: HTTP 프로토콜 |                    |
| 3. IoT 게이트웨이 가상화   | IoT 게이트웨이(예를 들어, 셀룰러 접속을 갖는 WiFi 액세스 포인트)는 길가를 따라 배치된다. 게이트웨이는 잠재적인 산사태를 모니터링하기 위해 센서 네트워크 및 이동 차량들 양쪽 모두에 대한 접속들을 제공한다. 클라우드에서의 IoT 서버는 2개의 가상 게이트웨이를 생성하기를 원하는데, 하나의 가상 게이트웨이는 트래픽 모니터링 및 지능형 수송 시스템(intelligent transport system)을 위한 것이며, 다른 가상 게이트웨이는 산사태 모니터링을 위한 것이다. | VTC: IoT 서버에서의 서비스 기능.<br>VTS: IoT 게이트웨이에서의 서비스 기능.<br>VTM: IoT 서버에서의 서비스 기능.   | VT1: HTTP 프로토콜<br>VT2: HTTP 프로토콜<br>VT3: HTTP 프로토콜 |                    |
| 4. IoT 센서 네트워크 가상화 | IoT 게이트웨이는 도시에 배치되는 대규모 센서 네트워크에 접속한다. 센서 네트워크는 온도, 공해, 조명, 트래픽 등과 같은 다양한 각 데이터를 제공한다. 클라우드에 배치된 IoT 서버는 다수의 상이한 가상 센서 네트워크를 요청 및 생성하기를 원한다. 각각의 가상 센서 네트워크는 공기 품질 모니터링, 일기 예보 등과 같은 상이한 애플리케이션들을 지원한다.  | VTC: IoT 서버에서의 서비스 기능.<br>VTB: IoT 게이트웨이에서의 서비스 기능.<br>VTS: IoT 센서 노드에서의 서비스 기능.<br>VTM: IoT 서버에서의 서비스 기능.                            | VT5: HTTP 프로토콜<br>VT6: CoAP 프로토콜<br>VT9: HTTP 프로토콜 |                    |
| 5. IoT 사물 가상화      | IoT 사물은, 물리 사물에 대한 액세스들이 IoT 서버에 생성될 가상 사물 상에 오프로딩될 수 있도록 그 자신을 IoT 서버에 가상화하기를 원한다. IoT 서버는 동일한 물리 IoT 사물에 대한 다수의 가상 사물을 생성할 수 있다. IoT 사물은 가능하게는 IoT 게이트웨이를 통해 그 자신을 인터넷에 접속하기 위한 통신 모듈을 가질 수 있다는 점에 유의한다.   | VTC: IoT 사물에서의 서비스 기능.<br>VTB(선택적임): IoT 게이트웨이에서의 서비스 기능.<br>VTS: IoT 서버에서의 서비스 기능.<br>VTM: IoT 서버에서의 서비스 기능.                         | VT5: CoAP 프로토콜<br>VT6: HTTP 프로토콜<br>VT9: HTTP 프로토콜 |                    |

[0033] 도 6은 예시적인 VTC-대-VTS 가상화 아키텍처(600)를 도시하며, 이 가상화 아키텍처는 VTS(630), VTC(640), VTM들(610 및 620)을 포함할 수 있다. 가상화 인터페이스 VT1(635)은 VTS(630)와 VTC(640) 사이의 통신을 가

능하게 할 수 있다. 가상화 인터페이스 VT2(645)는 VTC(640)와 VTM(620) 사이의 통신을 가능하게 할 수 있다. 가상화 인터페이스 VT3(625)은 VTS(630)와 VTM(620) 사이의 통신을 가능하게 할 수 있다. 가상화 인터페이스 VT4(615)는 VTM(610)과 VTM(620) 사이의 통신을 가능하게 할 수 있다. 위의 표 4에서 언급된 바와 같이, 수개의 IoT 가상화 시나리오가 고려되는데, 각각의 VTC, VTS 및 VTM의 일부 예시적인 위치들은 이 표에 제공되어 있다.

[0034] VTS(예를 들어, 도 6의 VTS(630))는 로직 서비스 기능을 갖는 시스템일 수 있다. VTS는 물리 자원(예를 들어, IoT 디바이스들, IoT 게이트웨이들, IoT 서버들)에 대한 액세스를 갖거나 제공하는 임의의 IoT 엔티티들 상에 존재할 수 있다. 일부 실시예들에서, VTS는 VTC로부터 가상화 요청들을 수신하고, 이에 응답하여 VTC를 위한 가상 자원들을 생성할 수 있다. 생성된 가상 자원은 물리 자원과 공존할 수 있다. 예를 들어, IoT 네트워크 애플리케이션(즉, VTC)은, VT1 인터페이스(예를 들어, 도 6의 VT1(635))를 통하여, IoT 디바이스를 다수의 가상 디바이스로 가상화하라는 가상화 요청을 IoT 디바이스(즉, VTS)에 대해 발행할 수 있다. VTS와 임의의 물리 자원들 사이의 내부 인터페이스는 VTS가 존재하는 IoT 엔티티에서 필요할 수 있다는 점에 유의한다.

[0035] VTS는 각종 동작들을 수행할 수 있다. 이것들 중에는 자원 등록이 있는데, 여기서 VTS는, VTC가 적절한 VTS를 발견할 수 있도록 또는 VTM이 그것의 VTS들을 요약하여 이들을 다른 VTM에 재등록할 수 있도록 VT3(예를 들어, 도 6의 VT3(625)) 인터페이스를 이용하여 그 자신 및 그것의 물리 자원들을 VTM에 등록한다. VTS는 다수의 VTM에 등록할 수 있다.

[0036] VTS에 의해 수행되는 다른 동작은 가상화 요청 관리일 수 있으며, 여기서 VTS는 VT1 인터페이스를 통해 수신된 VTC로부터의 가상화 요청들을 분석한다. VTS는 이러한 착신 가상화 요청들을 수락, 거부 또는 버퍼링할 수 있다. VTC로부터 가상화 요청을 수신한 이후의 그 VTC로부터의 요청 시에, VTS는 버퍼링된 및/또는 계류 중인 가상화 요청을 소거할 수 있다.

[0037] VTS에 의해 수행되는 또 다른 동작은 가상 자원 생성일 수 있으며, 여기서 VTS는 VTC의 요청 시에 가상 자원을 생성하고, VT1 인터페이스를 통해 요청 VTC로 응답을 회신한다. VTS는 생성된 가상 자원을 VT3 인터페이스를 통해 VTM에 보고할 수 있다.

[0038] VTS에 의해 수행되는 또 다른 동작은 가상 자원 관리일 수 있으며, 여기서 VTS는, 기존의 가상 자원을 소거, 조정 또는 검색하라는 또는 다수의 가상 자원을 결합하라는 추가 요청들을 VTC로부터 수신할 수 있다.

[0039] VTS에 의해 수행되는 다른 동작은 가상화 콘텍스트 관리일 수 있다. 가상화 콘텍스트 정보는 VTS에서 지원되는 가상화 타입(들), 가상화 가격책정 정보, 및 VTC의 요건, 성능 측정치, 대응하는 물리 자원, 수명 등과 같은 가상 자원에 대한 정보를 포함할 수 있지만, 이에 제한되지는 않는다. VTS는 각각의 가상 자원에 대해 이러한 콘텍스트 정보를 생성할 수 있다. VTS는, 가상 자원 성능이 VTC의 요건들을 충족시키는 것을 보장하기 위해 생성된 가상 자원을 모니터링할 수 있다. 생성된 가상 자원에 대하여 잠재적인 이슈가 존재하는 경우, VTS는 할당된 물리 자원들을 자동으로 증가시키고, 연관된 VTC에 통지를 송신하고, 그리고/또는 이러한 이슈를 VTM에 보고할 수 있다.

[0040] VTS에 의해 수행되는 또 다른 동작은 가상화 정책 관리일 수 있으며, 여기서 VTS는 VTM에 유지된 가상화 정책들에 액세스할 수 있다. VTS는 VTM으로부터 특정 가상화 정책들을 다운로드하고, 이들을 국부적으로 유지할 수 있다.

[0041] VTC(예를 들어, 도 6의 VTC(640))는 가상 자원들을 생성하라는 가상화 요청들을 VTS에 송신하는 로직 서비스 기능을 갖는 시스템일 수 있다. VTC는 사용자 애플리케이션일 수 있거나, 또는 다른 IoT 엔티티들에 존재할 수 있다. 예를 들어, IoT 네트워크 애플리케이션과 같은 VTC가 가상화 요청을 VTS와 같은 IoT 디바이스에 송신하여, IoT 디바이스가 그 자신을 다수의 가상 디바이스로 가상화하는 것을 요청할 수 있다.

[0042] VTC는 각종 동작들을 수행할 수 있다. 이것들 중에는 VTS 루업이 있는데, 여기서 VTC는, 원하는 가상 자원들을 제공가능할 수 있는 잠재적인 VTS들을 발견하기 위해 루업 요청을 인터페이스 VT2(예를 들어, 도 6의 VT2(645))를 통해 VTM에 송신한다. VTC는 VTM에 가입하고, 이용가능한 새로운 VTS들이 존재할 때, 이전에 이용가능한 VTS들이 더 이상 이용가능하지 않을 때, 그리고/또는 임의의 다른 VTS 스테이터스 또는 상태가 변경될 때 VTM으로부터 자동 통지들을 수신할 수 있다.

[0043] VTC에 의해 수행되는 다른 동작은 가상 자원 생성일 수 있으며, 여기서 VTC는 가상 자원의 생성을 요청하기 위해 가상화 요청을 VTS에 송신한다. VTC는 순차적으로 또는 동시에 다수의 상이한 가상화 요청을 VTS에 송신할 수 있다. 요청들의 동시 송신은 VTS가 다수의 요청을 함께 고려할 수 있게 하고, 다음에 더 많은 요청들이 허

용 및 만족되어, 증가된 가상화 성공률을 초래할 수 있다. 이에 반해, 각각의 가상화 요청이 순차적으로 그리고 독립적으로 처리되는 순차적인 가상화 요청 처리는 더 낮은 가상화 성공률을 초래할 수 있다. VTC는, VT1 인터페이스를 통해 VTS에 소거를 송신하는 것에 의해, 이전에 송신되어 VTS에서 여전히 계류 중인 가상화 요청을 소거할 수 있다.

[0044] VTC에 의해 수행될 수 있는 다른 동작은 가상 자원 관리일 수 있으며, 여기서 VTC는, VT1 인터페이스를 통하여, VTS에서 다수의 가상 자원을 결합하라는 그리고/또는 기존의 가상 자원을 소거, 조정 또는 검색하라는 부가적인 요청들을 발행할 수 있다.

[0045] VTM(예를 들어, 도 6의 VTM(620))은 VTC와 VTS 사이의 상호작용들을 용이하게 하는 로직 서비스 기능을 갖는 시스템일 수 있다. 그것은 IoT 서버에 그리고/또는 클라우드에 존재할 수 있다. 선택적으로, VTM은 VTC들로부터 가려져 있는 VTS의 일부일 수 있다.

[0046] VTM은 각종 동작들을 수행할 수 있다. 이것들 중에는 물리/가상 자원 관리가 있는데, 여기서 VTM은 VTC 또는 VTS에 의해 질의될 수 있는 이용가능한 VTS들 및 그들의 물리 자원들의 목록을 유지할 수 있다. VTM은, 다수의 VTM이 더 확장가능한 접근법으로 이용가능한 VTS들 및 물리 자원들을 협력 추적할 수 있도록 VT4 인터페이스(예를 들어, 도 6의 VT4(615))를 통해 다른 VTM과 상호작용할 수 있다. 선택적으로, VTM은, VTC에 액세스가능하지 않을 수 있는 관리 목적들을 위한 가상 자원들과 물리 자원들 사이의 매핑 관계를 유지할 수 있다.

[0047] VTM에 의해 수행될 수 있는 다른 동작은 가상화 정책 관리이며, 여기서 VTM은 가상화 정책들을 동적으로 조정하고, 무효화(void)하고, 그리고/또는 VT3 인터페이스를 통해 VTS에 푸시할 수 있다. VTM은 동적으로 VT4 인터페이스를 통해 다른 VTM으로부터 가상화 정책들을 풀링(pull)하거나 푸시할 수 있다.

[0048] VTM에 의해 수행될 수 있는 다른 동작은 가상화 콘텍스트 관리이며, 여기서 VTM은 VTS 및/또는 VTC에 의해 액세스 및/또는 조작될 수 있는 물리 자원들 및 가상 자원들에 대한 가상화 콘텍스트 정보를 유지할 수 있다.

[0049] VTM에 의해 수행될 수 있는 다른 동작은 가상화 과금 관리이며, 여기서 VTM은 VTC들로부터의 루업 이력 정보를 유지하고, VTS에 의해 보고된 바와 같은 가상화 요청 처리 결과들을 유지한다. 이 정보에 기초하여, VTM은 가상화 과금 레코드들을 발생시킬 수 있다.

[0050] VT1 인터페이스(예를 들어, 도 6의 VT1(635))는 VTS와 VTC 사이의 상호작용들을 핸들링하는 시스템일 수 있다. 이러한 상호작용들은, VTS에서 가상 자원들을 생성, 업데이트, 조정 및/또는 제거하라고 VTC가 요청하는 것을 포함할 수 있다. 예를 들어, VTC는 VTS에 의해 이전에 생성된 다수의 가상 자원이 결합되는 것을 요청할 수 있다. 이러한 상호작용들은, VTS에서 계류 중인 가상화 요청을 소거하라고 VTC가 요청하는 것을 또한 포함할 수 있다. 이러한 상호작용들은, 가상 자원들에서의 변경들 및 모니터링된 성능에 대한 통지들을 VTS가 VTC로 송신하는 것을 더 포함할 수 있다. 이러한 상호작용들은, VTS가 가상 자원들의 성능 결과들과 같은 통지들을 VTC에 송신할 수 있도록 VTS 및 VTC가 VT1을 통한 로직 접속을 확립 및 유지하는 것을 또한 포함할 수 있다.

[0051] VT2 인터페이스(예를 들어, 도 6의 VT2(645))는 VTC와 VTM 사이의 상호작용들을 용이하게 하는 시스템일 수 있다. 이러한 상호작용들은, 원하는 VTS들을 찾도록 VTM에 요청하는 루업 요청을 VTC가 VTM에 송신하는 것, 및 VTM이 VTC에 응답을 회신하는 것을 포함할 수 있다. 이러한 상호작용들은, VTC가 VTM에 유지된 가상화 정책들 및 가상화 콘텍스트 정보에 액세스하는 것을 또한 포함할 수 있다. 이러한 상호작용들은, VTC가 자원 가입을 VTM에 송신하고, 가상화 콘텍스트 정보, 가상화 정책들 및/또는 VTS들에서의 변경들에 대한 자동 통지를 VTM으로부터 수신하는 것을 더 포함할 수 있다.

[0052] VT3 인터페이스(예를 들어, 도 6의 VT3(625))는 VTS와 VTM 사이의 상호작용들을 핸들링하는 시스템일 수 있다. 상호작용들은 VTS가 그 자신을 VTM에 등록하는 것을 포함할 수 있다. 등록 중에, VTS는 가격책정 정보 및 다른 가상화 콘텍스트 정보를 표시할 수 있다. 상호작용들은, VTS가 그것의 물리 자원 및 생성된 가상 자원들을 VTM에 보고하는 것을 더 포함할 수 있다. 이러한 상호작용들은, VTS가 가상화 요청 프로세스 결과들을 VTM에 보고하는 것, 및 VTS가 악성 VTC들을 VTM에 보고하는 것을 더 포함할 수 있다. VTM은 악성으로서 식별된 VTC들로부터의 "VTS 루업" 메시지들을 차단할 수 있다.

[0053] VT4 인터페이스(예를 들어, 도 6의 VT4(615))는 2개의 VTM 사이의 상호작용들을 핸들링하는 시스템일 수 있다. 이러한 상호작용들은, VTM이 VTS 등록을 다른 VTM에 포워딩 및/또는 종합(aggregating)하는 것, VTM이 다른 VTM으로부터 원하는 VTS를 루업하는 것, 및 VTM이 다른 VTM에서의 가상화 정책들에 액세스하고 조정하는 것을 포함할 수 있다.

- [0054] 도 7은 VTS(또는 다른 실시예들에서는 VTC)가 본 명세서에 개시된 기본적인 가상화 서비스 아키텍처에 기초하여 다수의 VTC(또는 다른 실시예들에서는 하나 이상의 VTS)와 인터페이스할 수 있는 시스템(700)을 도시한다. 하나의 VTC-다수의 VTS 실시예(예를 들어, 도 7의 VTC(740))에서, VTS(예를 들어, 도 7의 VTS(720))는 VTC에 의한 요청을 이행하기에 충분한 자원들을 제공가능하지 않을 수 있다. 이 경우, VTC는 다수의 VTS를 선택하고, 그들 각각으로 별개의 가상화 요청들을 송신할 수 있다. 그러나, VTC는 기본적인 VTC-대-VTS 가상화 서비스 아키텍처에서 얼마나 많은 그리고 어느 VTS들이 선택되어야 하는지를 결정할 수 있다. 하나의 VTS-다수의 VTC 실시예에서, VTS(예를 들어, 도 7의 VTS(710))는 다수의 VTC에 서비스를 제공할 수 있다. VTS는 VTS가 충분한 물리 자원들을 갖는 경우에 다수의 VTC에 대한 가상 자원들을 생성하고 제공할 수 있다.
- [0055] 도 8은 브로커 기반 가상화 서비스 아키텍처의 실시예를 나타내는 예시적인 시스템(800)을 도시하며, 여기서 VTB는 로직 서비스 기능 및 하나 이상의 가상화 인터페이스를 갖는 시스템이다. 위의 표 4는 수개의 IoT 가상화 실시예를 열거하며, 이러한 실시예들에서의 VTB들의 위치들을 상세화한다. VTB(예를 들어, 도 8의 VTB(840) 및 VTB(850))는 투명한 방식으로 또는 투명하지 않은 방식으로 VTS와 VTC 사이의 상호작용들을 조정하기 위해 VTC 및 VTS 양쪽 모두의 기능들 및 동작들을 수행할 수 있다. 예를 들어, VTB는 레거시 VTS에 VTC를 접속할 수 있다. VTB는 다수의 VTS를 조정할 수 있고, 다른 VTB와 또한 인터페이스할 수 있다.
- [0056] VTB는 각종 동작들을 수행할 수 있다. 이것들 중에는, VTC 및 VTS가 VTM으로부터 잠재적인 VTB를 발견할 수 있도록 VTB가 그 자신을 VTM에 등록하는 것이다. 또한, VTB는 VTC로부터 가상화 요청들을 수신하고, 이러한 요청들을 VTS에 포워딩할 수 있다. VTB는 VTC를 대신하여 하나 이상의 적절한 VTS를 검색하고 선택할 수 있다. VTB는 단일의 가상화 요청을 다수의 가상화 요청으로 분할할 수 있다. 또한, VTB는 다수의 가상화 요청을 단일의 가상화 요청으로 종합할 수 있다.
- [0057] VTB가 수행할 수 있는 다른 동작들은, 차별화되며 우선화된 방식으로 착신 요청들을 버퍼링하고 이러한 요청들을 VTS에 포워딩하는 것을 포함한다. VTB는 VTS를 대신하여 VTC들을 서빙할 수 있다. VTB는 하나의 VTS로부터 다른 VTS로의 가상 자원 이동을 조정할 수 있다. VTB는 VTC로부터 가상화 요청들을 수신하고, 이러한 요청들을 다른 VTB에 포워딩할 수 있다. VTB는 적절한 VTS 및/또는 VTB를 발견하고 선택하기 위해 VTM과 인터페이스할 수 있다. VTB는 동적 정책 조정 및 시행에 있어서 일 실시예에서는 사용자에 대한 가상화 콘텍스트 정보 및 가상화 정책들에 액세스하기 위해 VTM과 인터페이스할 수 있다. VTB는 상이한 타입의 VTS들과 VTC들의 상호연동을 용이하게 할 수 있다. 예를 들어, 도 8에 도시된 바와 같이, VTB(850)는 레거시 가상화 인터페이스 VT10(814)을 통해 레거시 VTS(830)와 인터페이스한다. VTB(850)는 레거시 VTS(830)와 VTC(840) 사이에 변환을 수행할 수 있다.
- [0058] 레거시 VTS(예를 들어, 레거시 VTS(830))는 자원 가상화를 지원하기 위해 종래의 로직 서비스 기능을 갖는 시스템일 수 있다. 그것은 직접적으로 VTC들 또는 다른 VTS들과 통신가능하지 않을 수 있다. 대신에, 그것은 예를 들어 레거시 인터페이스 VT10(814)을 경유하여 VTB(850)를 통해 VTS(820)와 통신할 수 있다.
- [0059] VT5 인터페이스(예를 들어, 도 8의 VT5(815))는 VTC와 VTB 사이의 상호작용들을 핸들링하는 시스템일 수 있다. VT5 인터페이스는 VT1 인터페이스와 유사한 기능들을 지원할 수 있다.
- [0060] VT6 인터페이스(예를 들어, 도 8의 VT6(812) 및 VT6(817))는 VTB와 VTS 사이의 상호작용들을 핸들링하는 시스템일 수 있다. VT6 인터페이스는 VT1 인터페이스와 유사한 기능들을 또한 지원할 수 있지만, (본 명세서에서 보다 상세하게 논의된) VTB-제어 다중-VTS 조정에 관련되는 부가적인 기능들을 또한 지원할 수 있다.
- [0061] VT7 인터페이스(예를 들어, 도 8의 VT7(813))는 2개의 VTB 사이의 상호작용들을 핸들링하는 시스템일 수 있다. 이러한 상호작용들은, 하나의 VTB가 가상화 요청들을 다른 VTB에 포워딩하는 것, 및 포워딩하는 VTB에 가상화 응답을 송신하는 것을 포함할 수 있다.
- [0062] VT8 인터페이스(예를 들어, 도 8의 VT8(811))는 2개의 VTS 사이의 상호작용들을 핸들링하는 시스템일 수 있다. 이러한 상호작용들은 (본 명세서에서 보다 상세하게 논의된) VTB-제어 다중-VTS 조정에서 이용되는 바와 같이 직접적인 VTS-대-VTS 상호작용들을 포함할 수 있다. VT8 인터페이스는 하나의 VTS로부터 다른 VTS로의 가상 자원 이동을 또한 지원할 수 있다. 일부 실시예들에서, VTC에는 이동된 가상 자원이 통지되지 않는다.
- [0063] VT9 인터페이스(예를 들어, 도 8의 VT9(818))는 VTB와 VTM 사이의 상호작용들을 핸들링하는 시스템일 수 있다. VT9 인터페이스는 VT2 인터페이스 및 VT3 인터페이스와 유사한 기능들을 지원할 수 있다.
- [0064] VT10 인터페이스(예를 들어, 도 8의 VT10(814))는 VTB와 레거시 VTS 사이의 상호작용들을 핸들링하는 시스템일

수 있다. 또한, VT10 인터페이스는 레거시 또는 사유 가상화 인터페이스일 수 있다.

[0065] VT11 인터페이스(예를 들어, 도 8의 VT11(816))는 VTB와 레거시 VTC 사이의 상호작용들을 핸들링하는 시스템일 수 있다. 또한, VT11 인터페이스는 레거시 또는 사유 가상화 인터페이스일 수 있다.

[0066] 도 9는 가상화 서비스(910)와 다른 IoT 서비스들(920) 사이의 예시적인 상호작용들(930)을 도시한다. 예시적인 가상화 요소들 또는 기능들(예를 들어, VTC(911), VTS(912), VTB(913) 및 VTM(914))은 IoT 서비스들로서 구현될 수 있으며, 가상화 서비스들로 지정될 수 있다. IoT 서비스 플랫폼은 다른 것들 중에서 이동성 관리 서비스(921), 이벤트 관리 서비스(922), ID(identity) 관리 서비스(923), 발견 서비스(924), 정책 관리 서비스(925) 및 과금 서비스(926)를 포함할 수 있다. 가상화 서비스(910)는, 다음의 상호작용들을 이용함으로써 IoT 자원 가상화를 더 효율적으로 하기 위해서 IoT 서비스들과 인터페이스할 수 있다.

[0067] 가상화 서비스(910)는 새로운 가상화, 가상화 소거 또는 가상화 조정/적응을 트리거하기 위해 이동성 관리 서비스(921)로부터 트리거를 수신할 수 있다. 가상화 서비스(910)는 가상 자원 이동을 이행하는 것을 돋기 위해 이동성 관리 서비스(921)를 이용할 수 있다.

[0068] 가상화 서비스(910)는 이벤트 관리 서비스(922)에 제공될 수 있는 새로운 이벤트들을 발생 및/또는 생성할 수 있다. 가상화 서비스(910)는 새로운 가상화, 가상화 소거 또는 가상화 조정이나 적응을 트리거하기 위해 이벤트 관리 서비스(922)로부터 트리거를 수신할 수 있다. 가상화 서비스(910)는 이벤트 관리 서비스(922)에서의 이벤트들(예를 들어, 가상화 관련 이벤트들)을 조작할 수 있다(예를 들어, 이벤트들의 검색, 업데이트, 삭제, 결합).

[0069] 가상화 서비스(910)는 인증 및 다른 보안 목적들을 위한 ID 매핑과 같은 ID 관리 서비스(923)로부터의 서비스들을 요청할 수 있다. 가상화 서비스(910), 특히 VTS(911) 및 VTM(914)은 새로 생성된 가상 자원(예를 들어, 객체, 센서, 접속성, 네트워크 등)에 대해 새로운 ID 또는 식별자를 생성하고 저장할 수 있다. 가상화 서비스(910)는 ID 관리 서비스(923)에 저장된 가상화 관련 ID들 또는 식별자들에 액세스하고 조작할 수 있다.

[0070] 가상화 서비스(910)는, 이동성 관리 서비스(921), 이벤트 관리 서비스(922), ID 관리 서비스(923), 정책 관리 서비스(925) 및 과금 서비스(926)와 같은 다른 서비스들을 찾기 위해 발견 서비스(924)를 이용할 수 있다. 대안적으로 또는 부가적으로, 가상화 서비스(910)는 다른 가상화 서비스들을 발견하기 위해 발견 서비스(924)를 이용할 수 있다.

[0071] 가상화 서비스(910)는 정책 관리 서비스(925)로부터 가상화 관련 정책들을 획득할 수 있다. 이러한 정책들은 정책 관리 서비스(925)로부터 자동 송신들로서 수신되고/되거나, 필요할 때 가상화 서비스(910)에 의해 검색될 수 있다. 정책 관리 서비스(925)는 VTM(914), VTB(913) 및/또는 VTS(912)에서의 기존의 가상화 관련 정책들을 업데이트하거나 소거할 수 있다.

[0072] 가상화 서비스(910)는 가상화 요청들 및 가상 자원들의 레코드들을 유지할 수 있고, 과금 서비스(926)에 제공될 과금 레코드들을 발생시킬 수 있다. 과금 서비스(926)는 가상화 서비스(910)의 과금 레코드 발생 동작들을 구성 및 관리할 수 있다. 과금 서비스(926)는 가상화 서비스(910)에 의해 발생된 과금 레코드들에 액세스하고 풀링할 수 있다.

[0073] 실시예에서, 본 명세서에 설명된 바와 같은 상이한 가상화 인터페이스들을 통해 2개의 가상화 요소(예를 들어, VTC, VTS, VTB 및 VTM) 사이에 메시지들이 교환될 수 있다. 메시지들은 페이로드로서 그리고/또는 헤더들 또는 옵션 필드들 내에서 상이한 프로토콜들(예를 들어, HTTP 및 CoAP)로 포함될 수 있다. 위의 표 4는 HTTP 및 CoAP가 적용될 수 있는 인터페이스들의 수개의 예를 예시한다. 대안적으로, 가상화 요청 및 응답 메시지들을 전송하는데 새로운 가상화 서비스 프로토콜이 이용될 수 있다.

[0074] 도 10은 콘텍스트 인식 가상화 실시예들에서 이용될 수 있는 예시적인 신호 흐름(1000)을 도시한다. 이러한 실시예들에서, VTC(1010)는 VTB(1020)를 통해 VTS(1030)에서 가상 자원이 생성되는 것을 요청할 수 있다. 도 10에 도시된 예시적인 실시예는 가상화 효율을 향상시키기 위한 가상화 콘텍스트 정보일 수 있다.

[0075] 1011에서, VTS(1030)는 그 자신 및 그것의 물리 자원들을 등록하기 위해 "VTS 등록" 메시지를 VTM(1040)에 송신한다. 대안적으로, VTS(1030)는 그것이 VTM(1040)에 이미 등록된 경우에 "VTS 업데이트" 메시지를 VTM(1040)에 송신할 수 있다. 이 VTS 등록 메시지 또는 VTS 업데이트 메시지는 수개의 파라미터 중 임의의 것을 포함할 수 있다. 이것들 중에는, VTS(1030)가 제공할 수 있는 자원들에 대한 가격정책 정보(예를 들어, 과금 방안 및 레이트), 각각의 탑재의 자원이 지원할 수 있는 가상 자원들의 최대 개수, 각각의 탑재의 자원을 생성하기 위한

응답 시간, VTS(1030)가 가장 자원을 생성할 수 있는 각각의 탑의 자원의 입도(granularity), 각각의 탑의 자원의 볼륨, 각각의 탑의 자원의 물리적 위치, 각각의 탑의 자원의 수명, VTS(1030)가 관리하고/하거나 제공할 수 있는 각각의 탑의 자원 속성을 및 자원 탑들, 및 VTS(1030)의 물리적 위치를 나타내는 정보가 있다. 1012에서, 등록의 성공 또는 실패를 나타내는 응답이 VTS(1030)로 회신될 수 있다.

[0076] 1013에서, VTC(1010)는 제1 가상화 요청을 VTB(1020)에 송신할 수 있다. 이 메시지는 VTC(1010)에 대한 콘텍스트 정보, 예컨대 그것의 위치, 그것의 이동 속도, 그것의 능력 등을 포함할 수 있다. 이 메시지는 하나 이상의 자원 요청 항목들("RRI들")을 포함할 수 있다. 각각의 RRI는 다음의 콘텍스트 정보, 예컨대 요청된 자원의 탑, 요청된 자원의 볼륨, 요청된 자원에 대한 적정 가격, 요청된 자원의 예상 위치, 요청된 자원의 예상 응답 시간, 요청된 자원의 예상 수명, 및 자원 요청 항목이 VTB(1020)에 의해 처리될 수 있는 허용가능한 지연을 포함할 수 있다.

[0077] 1014에서, VTB(1020)는 콘텍스트 인식 가상화 요청 처리를 수행할 수 있다. VTB(1020)는 콘텍스트 인식 접근법으로 1013에서 수신된 착신 요청 메시지를 분석하고 처리할 수 있다. 예를 들어, VTB(1020)는 VTC(1010)에 대한 콘텍스트 정보, 예컨대 VTC(1010)의 위치 또는 적정 가격에 기초하여 요청(1013)을 거부할 수 있다. 다른 예에서, VTB(1020)는 각각의 RRI에 대해 포함된 "허용가능한 지연" 정보에 기초하여 요청(1013)을 버페링하거나 즉시 처리할 수 있다. VTB(1020)가 잠시 동안 요청(1013)을 거부하거나 버페링하기로 결정하는 경우, 도 10과 관련하여 설명된 나머지 메시지들 및 액션들은 수행되지 않을 수 있다.

[0078] 1015에서, VTB(1020)는 VTS 루업 메시지를 VTM(1040)에 송신한다. 이 메시지는 메시지(1013)에서 요청된 자원들을 잠재적으로 제공할 수 있는 VTS 후보들의 목록을 발견하는데 이용될 수 있다. VTS 루업 메시지는 콘텍스트 정보, 예컨대 메시지(1013)에 포함된 콘텍스트 정보 중 임의의 것 또는 그 전부, 메시지(1013)가 다수의 RRI를 포함하였는지 여부, 및 ID 및 위치와 같은 VTB(1020)에 대한 콘텍스트 정보를 포함할 수 있다. 메시지(1013)가 다수의 RRI를 포함하는 경우, VTB(1020)는 일부 RRI를 버페링하고, 따라서 다른 버페링되지 않은 RRI에 대한 콘텍스트 정보만을 포함할 수 있다. 대안적으로, VTB(1020)는 이전에 그 자신을 VTM(1040)에 등록하였을 수 있고, VTM(1040)은 VTB(1020)의 콘텍스트 정보를 유지할 수 있다. 그 경우, VTB(1020)의 콘텍스트 정보가 변경될 때까지 VTS 루업 메시지에서 그것의 콘텍스트 정보를 폐기백할 필요가 없다. 1016에서, VTM(1040)은 VTS 후보들의 목록을 VTB(1020)에 리턴한다. 등록 메시지(1011)에 열거된 바와 같은 각각의 VTS에 대한 콘텍스트 정보가 이 메시지에서 폐기백될 수 있다.

[0079] 1017에서, VTB(1020)는 콘텍스트 인식 가상화 전략 및 VTS 선택을 수행할 수 있다. 이 동작에서, VTB(1020)는, 요청(1013)에 그리고 VTM(1040)으로부터 리턴된 응답(1016)에 포함된 콘텍스트 정보에 기초하여, 가상화 전략(예를 들어, 분할-및-병합(split-and-merge))을 결정하고, 적절한 VTS들을 선택할 수 있다. 예를 들어, VTB(1020)는 요청(1013)을 다수의 가상화 요청 메시지로 분할할 수 있다.

[0080] 1018에서, VTB(1020)는 선택된 VTS(1030)에 제2 가상화 요청을 송신한다. 이 메시지는 VTS(1030)가 특정의 요구된 가상 자원들을 생성하는 것을 요청한다. VTB(1020)는 요청(1013)을 VTS(1030)가 이해할 수 있는 상이한 포맷의 요청(1018)으로 변환할 수 있다. 부가적으로, 요청(1018)은 요청(1013)에 포함된 것보다 더 적은 RRI를 포함할 수 있다. 요청(1013)에서와 같이, 요청(1018)도 또한 콘텍스트 정보를 포함할 수 있다.

[0081] 1019에서, VTS(1030)는 요청(1018)에 포함된 콘텍스트 정보에 기초하여 요청(1018)을 수락할지 또는 거부할지를 결정하고, 응답을 VTB(1020)에 송신한다. VTS(1030)가 요청(1018)을 수락하기로 결정하는 경우, 그것은 대응하는 물리 자원(들)을 할당하고 생성된 가상 자원과 오리지널 물리 자원 사이에 매핑 관계를 확립함으로써 요청된 가상 자원을 생성할 것이다. VTS(1030)가 요청(1018)의 요건들을 충족시키기에 충분한 물리 자원들을 갖고 있지 않은 경우, VTS(1030)는 다른 VTS와의 직접적인 상호작용들을 통해 그 VTS로부터 자원들을 빌릴 수 있다.

[0082] 1021에서, VTB(1020)는 응답(1019)을 응답(1021)으로 변환하여 그것을 VTC(1010)에 포워딩한다. VTS(1030)가 요청(1018)을 거부하는 경우, VTB(1020)는 다른 VTS를 재선택하고, 요청을 새로 선택된 VTS에 재송신할 수 있다. VTS(1030)가 요청(1018)을 거부하는 경우, VTS(1030)는 다른 VTS를 VTB(1020)에 추천할 수 있거나, VTB(1020)를 대신하여 다른 VTS에 직접적으로 접촉할 수 있다.

[0083] 1022에서, VTS(1030)는, 남아있는 물리 자원들, 과금 레이트에 대한 변경들 및 임의의 다른 변경들과 같은 그것의 콘텍스트 정보를 업데이트하기 위해 콘텍스트 업데이트 메시지를 VTM(1040)에 송신한다. 또한, VTS(1030)는 생성된 가상 자원에 대한 과금 레코드를 발생시키고, 이 과금 레코드를 VTM(1040)에 보고할 수 있다. 1023에서, VTB(1020)는, 그것의 남아있는 물리 자원들, 과금 레이트에 대한 변경들 및 임의의 다른 변경들을 업

데이트하기 위해 콘텍스트 업데이트 메시지를 VTM(1040)에 송신한다.

- [0084] 협력 가상화 실시예들에서, VTC는 제1 VTB를 통해 특정 가상 자원들을 생성할 수 있지만, 요청된 가상 자원들은 단일의 VTS에 의해 제공가능하지 않을 수 있고, 따라서 다수의 VTS에 의해 제공되어야 한다. 이러한 상황을 다루기 위한 예시적인 가상화 분할-및-병합 프로세스가 도 11의 신호 흐름에 도시되어 있다. 동일한 VTB를 통해 가상 자원들을 요청하는 다수의 VTC가 존재하고, 요청된 가상 자원들이 단일의 VTS에서 잠재적으로 생성될 수 있는 경우, 요청들은 병합될 수 있다. 도 12에 도시된 바와 같은 예시적인 가상화 병합-및-분할(merge-and-split) 프로세스는 이러한 상황을 다루는데 이용될 수 있다.
- [0085] 도 11은 개시된 가상화 분할-및-병합 프로세스를 이용하는 실시예들에서 이용될 수 있는 예시적인 신호 흐름(1100)을 도시한다. 1111에서, VTC(1110)는 가상화 요청을 VTB(1120)에 송신한다. 1112에서, VTB(1120)는 요청(1111)을 분석하고, 단일의 VTS가 이 요청을 만족시킬 수 없다는 것을 알게 된다. VTB(1120)는 이 프로세스 동안 VTM과 접촉할 수 있다는 점에 유의한다. 또한 1112에서, VTB(1120)는 요청(1111)을 다수의 요청(즉, 가상화 요청 A 및 가상화 요청 B)으로 분할하고, 2개의 적절한 VTS들인 VTS(1130) 및 VTS(1140)를 선택한다. VTB(1120)는 요청(1111)을 2개보다 많은 요청으로 분할하고, 2개보다 많은 VTS를 선택할 수 있다는 점에 유의한다. 실시예에서, VTB(1120)는 VTS 후보들의 목록을 얻기 위해 VTM과 함께 본 명세서에 제시된 바와 같은 VTS 루업 절차를 이용하였을 수 있다.
- [0086] 1113에서, VTB(1120)는 가상화 요청 A를 VTS(1130)에 송신하고, 1114에서 VTS(1130)로부터 응답을 수신한다. 1115에서, VTB(1120)는 가상화 요청 B를 VTS(1140)에 송신하고, 1116에서 VTS(1140)로부터 응답을 수신한다. 1117에서, VTB(1120)는 응답 A 및 B를 병합하여 종합된 응답을 발생시키고, 1118에서 이 종합된 응답을 VTC(1110)에 송신한다.
- [0087] 도 12는 개시된 가상화 병합-및-분할 프로세스를 이용하는 실시예들에서 이용될 수 있는 예시적인 신호 흐름(1200)을 도시한다. 1211에서, VTC(1210)는 가상화 요청 A를 VTB(1230)에 송신하고, 1212에서, VTC(1220)는 가상화 요청 B를 VTB(1230)에 송신한다. 1213에서, VTB(1230)는 가상화 요청 A 및 가상화 요청 B를 베퍼링하고, 단일의 병합된 요청으로 병합한다. 1214에서, 병합된 요청은 VTS(1240)에 송신된다. VTB(1230)는 VTS 후보들의 목록을 얻기 위해 VTM에 VTS 루업을 송신하였을 수 있다. 1215에서, VTS(1240)는 VTB(1230)에 응답을 회신한다. 1216에서, VTB(1230)는 이 응답을 가상화 요청 A 및 가상화 요청 B에 각각 대응하는 응답 A 및 응답 B로 분할한다. 다음에, 1218에서 응답 A가 VTC(1210)에 송신되고, 1217에서 응답 B가 VTC(1220)에 송신된다.
- [0088] 실시예에서, 하나의 VTM 내에서의 가상 자원 이동을 위한 프로세스가 구현될 수 있다. VTB 또는 VTC는 제1 VTS로부터 다른 VTS로 가상 자원들을 이동시킬 필요가 있을 수 있다. 이러한 모든 요소를 관리하는 단일의 VTM이 존재할 수 있다. VTB 또는 VTC가 이전에 제1 VTS를 통해 가상 자원을 생성한 이후에, 로드밸런싱, 이동성, 에너지 소비 감소, 정책 기반 가상화 적응 및/또는 조정, 콘텍스트 기반 가상화 적응 및/또는 조정을 비롯한 각종 인자들 중 임의의 것으로 인해 이동이 필요할 수 있다. 따라서, VTB 또는 VTC는 다른 VTS를 선택할 수 있고, 제1 VTS로부터 새로 선택된 VTS로 기존의 가상 자원을 이동시킬 수 있다.
- [0089] 도 13은 하나의 VTM 내에서의 가상 자원 이동을 위한 개시된 절차를 이용하는 실시예들에서 이용될 수 있는 예시적인 신호 흐름(1300)을 도시한다. 1311에서, 자원들을 이동시킬 필요성이 검출되는 것으로 인해, VTC/VTB(1310)는 이 예에서 새로운 VTS인 VTS(1330)를 선택할 수 있다. 이것을 행하기 위해서, VTC/VTB(1310)는 본 명세서에 설명된 바와 같이 VTS 후보들의 목록을 얻기 위해 VTM(1340)과 같은 VTM에 VTS 루업 메시지를 송신하였을 수 있다. 자원 이동은, 가상화 콘텍스트 정보 및/또는 가상화 정책들의 변경들로 인해, VTM(예를 들어, VTM(1340)), VTB(예를 들어, VTC/VTB(1310)), 또는 현재 가상 자원을 호스팅하고 있는 VTS(예를 들어, VTS(1320))에 의해 트리거될 수 있다는 점에 유의한다.
- [0090] 1312에서, VTC/VTB(1310)는 이동 요청을 새로 선택된 VTS(1330)에 송신할 수 있다. 이 메시지는 요청된 자원에 대한 콘텍스트 정보 및 VTS(1310)에 대한 콘텍스트 정보를 포함할 수 있다. 1313에서, VTS(1330)는 VTC/VTB(1310)에 응답을 회신한다. VTS(1330)가 이동 요청을 거부한 경우에는, VTC/VTB(1310)는 다른 VTS를 선택하고, 다시 시도할 수 있다. VTS(1330)가 요청을 수락한 경우에는, VTS(1330)는 이전의 VTS(1320)로부터 콘텍스트 전송을 수신하도록 준비될 것이다.
- [0091] 1314에서, VTC/VTB(1310)는 "이동 통지"를 VTS(1320)에 송신할 수 있다. 이 메시지는 이동을 준비하는데 이용될 수 있는 새로운 VTS(1330)에 대한 콘텍스트 정보를 포함할 수 있다. 1315에서, VTS(1320)는 콘텍스트 전송

메시지를 새로운 VTS(1330)에 송신한다. 이 메시지는 이동될 가상 자원과 연관된 콘텍스트 정보의 일부 또는 전부를 포함할 수 있다. 예를 들어, 가상 자원이 스토리지인 경우, 저장된 콘텐츠는 VTS(1320)로부터 새로운 VTS(1330)로 전송될 수 있도록 표시될 수 있다. 1316에서, VTS(1330)는 콘텍스트 전송이 성공적이었는지를 나타내는 응답을 VTS(1320)에 송신한다. 전송이 성공적인 경우, 응답(1316)의 수신 시에, VTS(1320)는 가상 자원 및 관련 콘텍스트 정보를 소거하거나 또는 다른 방식으로 삭제 또는 제거할 수 있다.

[0092] 1317에서, VTS(1320)는 이동 통지(1314)에 응답하는 응답을 VTC/VTB(1310)에 송신한다. 1318에서, VTS(1330)는, 예를 들어 이동으로 인한 VTS(1330)에서의 자원 변경들을 업데이트하기 위해 VTM(1340)과 "콘텍스트 업데이트"를 수행한다. 1319에서, VTS(1320)도, 예를 들어 이동으로 인한 VTS(1320)에서의 자원 변경들을 업데이트하기 위해 VTM(1340)과 "콘텍스트 업데이트"를 또한 수행한다. 최종적으로, 1321에서, VTC/VTB(1310)는, 예를 들어 VTS들과의 VTC/VTB(1310)의 관계를 업데이트하기 위해 VTM(1340)과 콘텍스트 업데이트를 수행한다.

[0093] 실시예에서, 상이한 VTM들에 걸친 가상 자원 이동을 위한 프로세스가 구현될 수 있다. VTB 또는 VTC는 제1 VTS에서 이전에 생성된 자원들을 다른 VTS로 이동시킬 필요가 있을 수 있다. 제1 VTM은 VTB 또는 VTC를 관리할 수 있는 한편, 제2 VTM은 제2 VTB 및 VTS를 관리할 수 있다. 실시예에 따르면, 가상 자원은 하나의 VTM 도메인으로부터 다른 VTM 도메인으로 이동될 수 있다.

[0094] 도 14는 하나의 VTM 도메인으로부터 다른 것으로의 가상 자원 이동을 위한 개시된 절차들을 이용하는 실시예들에서 이용될 수 있는 예시적인 신호 흐름(1400)을 도시한다. 1411에서, VTB(1410)는 VTM(1460)의 도메인에 있는 적절한 VTB를 찾기 위해 VTS 루업 메시지를 VTM(1450)에 송신한다. 1412에서, VTM(1450)은 VTM(1460)의 도메인에 있는 적절한 VTB를 찾기 위해 VTM(1460)에 VTS 루업을 송신한다. 1413에서, VTM(1460)은 결정된 VTB(들)에 대한 정보, 예컨대 어드레스들을 포함할 수 있는 응답을 VTM(1450)에 송신한다. 1414에서, VTM(1450)은 그 응답을 VTB(1410)에 송신한다.

[0095] 1415에서, VTB(1410)는 VTB 등록 메시지를 VTM(1460)에 송신하는데, 일부 실시예들에서는 이 메시지에 VTS(1450)에 대한 정보가 포함된다. 1416에서, VTM(1460)은 VTB(1410)에 응답을 회신한다. VTB(1410)가 더 조기의 교환들로부터 VTM(1460)의 도메인에 있는 VTB 및 VTS에 대한 정보를 결정하지 않은 경우, 1417에서, 그 것은 VTS 루업 메시지를 VTM(1460)에 송신하고, 1418에서, VTB 및 VTS 정보를 갖는 응답을 수신할 수 있다.

[0096] 1419에서, VTB(1410)는 이동 요청을 VTS(1440)에 송신한다. 이 메시지는, 이전의 단계들에서 VTM(1460)이 VTB를 선택하는 경우에 VTB(1430)를 통해 레일레이될 수 있다. VTS(1420)에 대한 정보가 이 메시지에 포함될 수 있다. 1421에서, VTS(1440)는, VTB(1430)를 통해 또한 레일레이될 수 있는 응답을 VTB(1410)에 송신할 수 있다. 1422에서, VTB(1430) 및 VTS(1440)에 대한 정보를 포함하는 이동 통지가 VTS(1420)에 송신될 수 있다. 이에 응답하여, VTS(1420)는, 그 엔티티가 알려져 있는 경우에 VTB(1430)를 통해, 콘텍스트 전송 메시지(1423)를 VTS(1440)에 송신할 수 있다. 1424에서, 그 엔티티가 알려져 있는 경우에 VTB(1430)를 통해, 응답이 VTS(1440)에 의해 송신될 수 있다. 1425에서, 응답은 VTS(1420)에 의해 VTB(1410)에 레일레이된다.

[0097] 1426에서, VTS(1440)는 이동된 가상 자원 및 다른 가상화 콘텍스트 정보를 VTM(1460)에 알리기 위해 콘텍스트 업데이트를 VTM(1460)에 송신한다. 1427에서, VTS(1420)는 이동된 가상 자원 및 다른 가상화 콘텍스트 정보를 VTM(1450)에 알리기 위해 콘텍스트 업데이트를 VTM(1450)에 송신한다. 1428에서, VTB(1410)는 이동된 가상 자원 및 다른 가상화 콘텍스트 정보를 VTM(1450)에 알리기 위해 콘텍스트 업데이트를 VTM(1450)에 송신한다. 그리고 최종적으로, 1429에서, VTB(1430)는 이동된 가상 자원 및 다른 가상화 콘텍스트 정보를 VTM(1460)에 알리기 위해 콘텍스트 업데이트를 VTM(1460)에 송신한다.

[0098] 클라우드 기반 가상화 실시예들에서, VTC는 제1 VTB를 통해 특정 VTS에서 가상 자원들을 생성하라고 요청할 수 있다. 그러나, 제1 VTB는 계산 집약적 가상화 알고리즘들을 구동할 제한된 능력을 가질 수 있고, 적절한 VTS를 결정가능하지 않을 수 있다. 이러한 실시예에서, 제1 VTB는 가상화 계산을 VTM에 오프로딩할 수 있다.

[0099] 도 15는 가상화 계산을 오프로딩하기 위한 개시된 절차들을 이용하는 실시예들에서 이용될 수 있는 예시적인 신호 흐름(1500)을 도시한다. 1511에서, VTC(1510)는 가상화 요청을 VTB(1520)에 송신할 수 있다. 그러나, VTB(1520)는 높은 계산 오버헤드를 일반적으로 도입하는 가상화 알고리즘들로 인해 적절한 VTS들을 결정하는데 필요한 가상화 알고리즘들을 구동가능하지 않을 수 있고, VTB(1520)는 일부 VTS들에 대한 제한된 정보만을 가질 수 있고, 이는 전역적 최적 솔루션을 획득하기 어렵게 할 수 있거나, 또는 특정의 미리 구성된 정책들은 VTB(1520)가 특정 타입의 가상 자원들에 대해 가상화 알고리즘을 구동하는 것을 방지할 수 있다. 대신에, VTB(1520)는 가상화 계산 태스크들을 VTM(1540)에 오프로딩할 수 있는데, 이는 VTS들에 대한 더 완전한 정보 및

더 강력한 계산 능력을 갖는 클라우드에 배치될 수 있다. 따라서, 1512에서, VTB(1520)는 오프로딩 가상화 계산 메시지를 VTM(1540)에 송신할 수 있다. 이 메시지는 요청(1511)에 포함된 콘텍스트 정보의 일부 또는 전부를 포함할 수 있다. 요청(1511)에 포함된 일부 RRI들은 VTM(1540)에 송신되는 것보다는 VTB(1520)에 버퍼링될 수 있다. 일부 실시예들에서, VTC(1510)는, 예를 들어 VTB(1520) 또는 VTM(1540)으로부터의 이전의 통지들로부터 VTB(1520)가 계산 집약적 가상화 알고리즘들을 수행할 수 없다고 학습하는 경우에, 가상화 요청을 VTM(1540)에 직접 송신할 수 있다.

[0100] VTM(1540)은 가상화 계산을 수행하고, VTB(1520)에 대해 적절한 VTS들을 결정한다. 1513에서, 그것은 선택된 VTS들의 목록을 VTB(1520)에 리턴한다. 1514에서, VTB(1520)는 VTM(1540)에 의해 제공된 목록으로부터 선정되는 선택된 VTS인 VTS(1530)에 가상화 요청을 송신한다. VTS(1530)는 요청을 수락하지 여부를 결정하고, 1515에서 응답을 리턴한다. VTS(1530)가 요청을 수락하기로 결정하는 경우, 그것은 대응하는 물리 자원들을 할당하고 생성된 가상 자원과 오리지널 물리 자원 사이에 매핑 관계를 확립함으로써 요청된 가상 자원들을 생성할 것이다. VTS(1530)가 요청을 거부하는 경우, VTB(1520)는 단계들(1512 및 1513)을 반복하고 요청을 다른 VTS에 송신함으로써 다른 VTS를 획득할 수 있다. 1516에서, VTB(1520)는, 프로세스에서 임의의 필요한 변환을 수행하여, VTS(1530)로부터의 응답을 VTC(1510)에 포워딩한다.

[0101] 도 16은 가상 자원 제공자(VRP: virtual resource provider)의 예시적인 실시예를 도시한다. 도 16의 시스템(1600)에서, IoT 자원 가상화를 관리하며 IoT VRP들을 용이하게 하도록 가상화 서비스들이 레버리징될 수 있다. 시스템(1600)은, 물리 자원들(1612, 1622 및 1632)을 각각 갖는 IoT 자원 제공자들("RESP들")(1610, 1620 및 1630)을 포함한다. 시스템(1600)은, 사용자들 및/또는 애플리케이션들에게 가상 자원들을 제공할 수 있으며 가상화 서비스들을 호스팅하지만 물리 자원들을 갖지 않는 IoT VRP들(1640 및 1650)을 또한 포함한다. VRP들(1640 및 1650)은 사용자들 또는 애플리케이션들의 요구들에 기초하여 정적으로 또는 동적으로 IoT RESP들(1610, 1620 및 1630)로부터 또는 서로로부터 또는 다른 IoT VRP들 및 RESP들로부터 자원들을 요청할 수 있다. RESP의 예는, 움직이는 자동차들로부터 클라우드로 데이터 트래픽을 레일레이하거나 접속하기 위해 고속도로들을 따라 많은 액세스 포인트를 배치하는 네트워크 제공자일 수 있다. VRP의 예는 자동차 제조자 또는 지능형 수송 서비스 제공자일 수 있다. 이러한 엔티티들은 물리적 액세스 포인트들을 소유할 필요는 없지만, 대신에 실시간 트래픽 통지 및 자동차 상태 모니터링과 같은 지능형 수송 서비스들을 자동차 운전자들에게 제공하기 위해 네트워크 제공자로부터 가상 액세스 포인트들을 임대할 필요가 있다. 이러한 실시예들에서, 자동차 제조자 또는 지능형 수송 서비스 제공자는 가상 자원 제공자이고, 네트워크 제공자는 물리 자원 제공자이고, 자동차 운전자들은 사용자들이고, 액세스 포인트 제조자는 물리 자원 제조자이다. 이들 각각은 본 명세서에 개시된 예시적인 가상화 기능성들, 인터페이스들 및 절차들을 지원할 수 있다.

[0102] 도 16에 도시된 바와 같이, IoT VRP(1640)는 IoT VRP(1650)로부터 가상 자원들을 요청할 수 있고, 그 반대도 이루어질 수 있다. IoT VRP(1640)는 또한 RESP들(1610, 1620 및 1630)과 같은 IoT RESP들로부터 자원들을 요청할 수 있다. IoT RESP의 예는, 환경들을 모니터링하기 위해 인근에 대규모 센서 네트워크 인프라스트럭처를 배치하는 회사일 수 있다. IoT VRP를 포함하여 센서 네트워크 인프라스트럭처는 사용자들에게 임대될 수 있다. 다른 IoT RESP 예는, 고속도로를 따라 다수의 액세스 포인트 또는 게이트웨이를 배치하는 회사일 수 있다. 이러한 게이트웨이들 및 액세스 포인트들은 VRP들 및 사용자들에 의해 임대될 수 있다.

[0103] 사용자들 및 애플리케이션들은, 사용자들 및 애플리케이션들 관점으로 투명하게 제공될 수 있는 자원들을 IoT VRP들 또는 IoT RESP들로부터 요청할 수 있다. IoT 사용자들 및 애플리케이션들은, 요청된 자원들이 IoT VRP로부터 승인되는지 또는 IoT RESP로부터 승인되는지를 알지 못할 수도 있고/있거나 알 필요가 없을 수도 있다.

[0104] IoT RESP(1610) 및 레거시 VTS(1611)와 같이 IoT RESP가 레거시 VTS만을 갖는 경우에, 이러한 RESP는 VT10 인터페이스를 통해 IoT VRP에 있는 VTB에 접속할 수 있는데, 예컨대 레거시 VTS(1611)는 VT10(1671)을 이용하여 IoT VRP(1640)에 있는 VTB(1641)에 접속되어 있다. 레거시 VTS(1621) 및 VTB(1623)를 호스팅하는 IoT RESP(1620)와 같이 IoT RESP가 VTB를 호스팅하는 경우, RESP는 VT7 인터페이스 및 VT9 인터페이스를 통해 IoT VRP에 있는 VTB들 및 VTM들에 각각 접속할 수 있다. 예를 들어, VTB(1623)는 VT7(1672)을 통해 VTB(1641)에 접속되며, VT9(1673) 및 VT9(1674)를 통해 VTM(1642) 및 VTM(1651)에 각각 접속된다. IoT RESP(1630)가 VTS(1634)를 호스팅하는 것과 같이 IoT RESP가 VTS를 호스팅하는 경우, IoT RESP는 VT6 인터페이스 및 VT3 인터페이스를 통해 IoT VRP에 있는 VTB 및 VTM에 각각 접속할 수 있다. 예를 들어, VTS(1634)는 VT3(1675)을 통해 VTM(1651)에 접속되며, VT6(1676)을 통해 VTB(1652)에 접속된다. 이 도면에서 알 수 있는 바와 같이, 시스템(1600) 도체에 있는 다른 엔티티들은 본 명세서에 제시된 인터페이스들 및 수단들을 이용하여 접속된다.

- [0105] 도 17은 기지국 가상화 관리의 예를 도시한다. 시스템(1700)은 셀룰러 코어 네트워크(1710), 기지국(1720), 및 사용자들과 애플리케이션들(1730)을 포함할 수 있다. 가상화 기능들인 VTM(1713), VTC(1711) 및 VTB(1712)는 독립형 코어 네트워크 엔티티들로서 셀룰러 코어 네트워크(1710)에 존재할 수 있거나, 또는 MTC-IWF, HSS 또는 MME와 같은 기존의 엔티티들에 통합될 수 있다. VTS(1721)는 기지국(1720)에 존재할 수 있다. 셀룰러 코어 네트워크(1710)에 있는 VTC(1711)는 인터페이스 VT1(1741)을 통해 기지국(1720)에 있는 가상 기지국들(1722, 1723 등)의 생성 및 관리를 트리거할 수 있다. 제3자 사용자들과 애플리케이션들의 VTC(1731)도 인터페이스 VT5(1744)를 경유하여 셀룰러 코어 네트워크(1710)에 존재하는 VTB(1712)를 통해 기지국(1720)에 있는 가상 기지국들(1722, 1723 등)의 생성 및 관리를 또한 트리거할 수 있고, 그에 따라 가상 기지국 관리는 셀룰러 코어 네트워크(1710)의 제어 하에 있는 것으로 된다.
- [0106] 도 18a는 IoT 가상화 서비스들을 위한 시스템들 및 방법들의 하나 이상의 개시된 실시예가 구현될 수 있는 예시적인 M2M 또는 IoT 통신 시스템(10)의 도면이다. 일반적으로, M2M 기술들은 IoT를 위한 블록들의 구축을 제공하고, 임의의 M2M 디바이스, 게이트웨이 또는 서비스 플랫폼은 IoT 서비스 계층뿐만 아니라 IoT의 컴포넌트 등 일 수 있다.
- [0107] 도 18a에 도시된 바와 같이, M2M/IoT 통신 시스템(10)은 통신 네트워크(12)를 포함한다. 통신 네트워크(12)는 고정 네트워크 또는 무선 네트워크(예를 들어, WLAN, 셀룰러 등) 또는 이종 네트워크들의 네트워크일 수 있다. 예를 들어, 통신 네트워크(12)는 음성, 데이터, 비디오, 메시징, 방송 등과 같은 콘텐츠를 다수의 사용자에게 제공하는 다중 액세스 네트워크들로 이루어질 수 있다. 예를 들어, 통신 네트워크(12)는 CDMA(code division multiple access), TDMA(time division multiple access), FDMA(frequency division multiple access), OFDMA(orthogonal FDMA), SC-FDMA(single-carrier FDMA) 등과 같은 하나 이상의 채널 액세스 방법을 이용할 수 있다. 또한, 통신 네트워크(12)는, 예를 들어, 코어 네트워크, 인터넷, 센서 네트워크, 산업용 제어 네트워크, 개인 영역 네트워크, 융합된 개인 네트워크(fused personal network), 위성 네트워크, 홈 네트워크 또는 기업 네트워크와 같은 다른 네트워크들을 포함할 수 있다.
- [0108] 도 18a에 도시된 바와 같이, M2M/IoT 통신 시스템(10)은 M2M 게이트웨이 디바이스(14) 및 M2M 단말 디바이스들(18)을 포함할 수 있다. 원하는 대로 임의의 개수의 M2M 게이트웨이 디바이스들(14) 및 M2M 단말 디바이스들(18)이 M2M/IoT 통신 시스템(10)에 포함될 수 있다는 것이 인식될 것이다. M2M 게이트웨이 디바이스들(14) 및 M2M 단말 디바이스들(18) 각각은 통신 네트워크(12) 또는 직통 라디오 링크(direct radio link)를 통해 신호들을 송신 및 수신하도록 구성될 수 있다. M2M 게이트웨이 디바이스(14)는 고정 네트워크 M2M 디바이스들(예를 들어, PLC)뿐만 아니라 무선 M2M 디바이스들(예를 들어, 셀룰러 및 네트워크)이 통신 네트워크(12)와 같은 운영자 네트워크들을 통해 또는 직통 라디오 링크를 통해 통신하는 것을 허용한다. 예를 들어, M2M 디바이스들(18)은 통신 네트워크(12) 또는 직통 라디오 링크를 통해 데이터를 수집하여 이 데이터를 M2M 애플리케이션(20) 또는 M2M 디바이스들(18)에 송신할 수 있다. M2M 디바이스들(18)은 M2M 애플리케이션(20) 또는 M2M 디바이스(18)로부터 데이터를 또한 수신할 수 있다. 또한, 데이터 및 신호들은 후술되는 바와 같이 M2M 서비스 플랫폼(22)을 통해 M2M 애플리케이션(20)에 송신되며 이 애플리케이션으로부터 수신될 수 있다. M2M 디바이스들(18) 및 게이트웨이들(14)은, 예를 들어, 셀룰러, WLAN, WPAN(예를 들어, Zigbee, 6LoWPAN, 블루투스), 직통 라디오 링크 및 유선을 비롯한 다양한 네트워크들을 통해 통신할 수 있다. 개시된 가상화 관리자들(VTM들), 가상화 서버들(VTS들), 가상화 클라이언트들(VTC들) 및 가상화 브로커들(VTB들)을 포함하여 본 명세서에 설명된 엔티티들 중 임의의 것은 M2M 디바이스들(18), 게이트웨이들(14) 및 서비스 플랫폼(22)과 같은 디바이스들 또는 엔티티들에서 전체적으로 또는 부분적으로 구현될 수 있다. 이러한 모든 실시예는 본 개시물의 범위 내에 있는 것으로서 고려된다.
- [0109] 도시된 M2M 서비스 플랫폼(22)은 M2M 애플리케이션(20), M2M 게이트웨이 디바이스들(14), M2M 단말 디바이스들(18) 및 통신 네트워크(12)에 대해 서비스들을 제공한다. M2M 서비스 플랫폼(22)은 원하는 대로 임의의 개수의 M2M 애플리케이션들, M2M 게이트웨이 디바이스들(14), M2M 단말 디바이스들(18) 및 통신 네트워크들(12)과 통신할 수 있다는 것이 이해될 것이다. M2M 서비스 플랫폼(22)은 하나 이상의 서버, 컴퓨터 등에 의해 구현될 수 있다. M2M 서비스 플랫폼(22)은 M2M 단말 디바이스들(18) 및 M2M 게이트웨이 디바이스들(14)의 모니터링 및 관리와 같은 서비스들을 제공한다. 또한, M2M 서비스 플랫폼(22)은 데이터를 수집하고, 이 데이터를 변환하여, 상이한 타입의 M2M 애플리케이션들(20)과 호환가능하게 할 수 있다. M2M 서비스 플랫폼(22)의 기능들은 셀룰러 코어 네트워크에서, 클라우드 등에서 각종 방식들로, 예를 들어 웹 서버로서 구현될 수 있다.
- [0110] 또한 도 18b를 참조하면, M2M 서비스 플랫폼은, 다양한 애플리케이션들 및 버티컬들(verticals)이 레버리징될 수 있는 서비스 전달 능력들의 코어 세트를 제공하는 서비스 계층(26)(예를 들어, 본 명세서에 설명된 바와 같

은 네트워크 서비스 능력 계층(NSCL))을 통상적으로 구현한다. 이러한 서비스 능력들은 M2M 애플리케이션들(20)이 디바이스들과 상호작용하고, 데이터 수집, 데이터 분석, 디바이스 관리, 보안, 빌링, 서비스/디바이스 발견 등과 같은 기능들을 수행할 수 있게 한다. 본질적으로, 이러한 서비스 능력들은 애플리케이션들에서 이러한 기능성들을 구현하는 부담을 없애고, 따라서 애플리케이션 개발을 단순화하며, 마켓까지의 비용 및 시간을 감소시킨다. 또한, 서비스 계층(26)은, 서비스 계층(26)이 제공하는 서비스들과 관련하여 다양한 네트워크들(12)을 통해 M2M 애플리케이션들(20)이 통신할 수 있게 한다.

[0111] 일부 실시예들에서, M2M 애플리케이션들(20)은, IoT 가상화 서비스들을 위해 가시된 시스템들 및 방법들을 이용할 수 있는 디바이스들을 포함하는 하나 이상의 퍼어-투-퍼어(peer-to-peer) 네트워크의 생성을 위한 기초를 형성하는 원하는 애플리케이션들을 포함할 수 있다. M2M 애플리케이션들(20)은, 수송, 건강 및 건강관리(wellness), 커넥티드 홈(connected home), 에너지 관리, 자산 추적, 및 보안 및 감시와 같지만 이에 제한되지는 않는 다양한 산업들에서의 애플리케이션들을 포함할 수 있다. 위에서 언급된 바와 같이, 디바이스들, 게이트웨이들 및 시스템의 다른 서버들에 걸쳐 구동되는 M2M 서비스 계층은, 예를 들어 데이터 수집, 디바이스 관리, 보안, 빌링, 위치 추적/지오펜싱(geofencing), 디바이스/서비스 발견, 및 레거시 시스템 통합과 같은 기능들을 지원하고, 서비스들로서 이러한 기능들을 M2M 애플리케이션들(20)에 제공한다. 설명된 서비스 계층 및 객체들과 상호작용하는 애플리케이션들은 M2M 애플리케이션들(20)의 것들과 같은 애플리케이션들일 수 있다.

[0112] 도 18c는 예를 들어 M2M 단말 디바이스(18) 또는 M2M 게이트웨이 디바이스(14)와 같은 예시적인 M2M 디바이스(30)의 시스템 도면이다. 도 18c에 도시된 바와 같이, M2M 디바이스(30)는 프로세서(32), 송수신기(34), 송/수신 요소(36), 스피커/마이크로폰(38), 키패드(40), 디스플레이/터치패드/인디케이터들(예를 들어, 하나 이상의 발광 다이오드(LED))(42), 비착탈식 메모리(44), 착탈식 메모리(46), 전원(48), GPS(global positioning system) 칩셋(50), 및 다른 주변 장치들(52)을 포함할 수 있다. M2M 디바이스(30)는 실시예에 부합하게 유지하면서 전술한 요소들의 임의의 서브조합을 포함할 수 있다는 것이 인식될 것이다. 이 디바이스는 IoT 가상화 서비스들을 위해 개시된 시스템들 및 방법들을 이용하는 디바이스일 수 있다.

[0113] 프로세서(32)는 범용 프로세서, 특수 목적 프로세서, 통상의 프로세서, 디지털 신호 프로세서(DSP), 복수의 마이크로프로세서, DSP 코어와 연관된 하나 이상의 마이크로프로세서, 제어기, 마이크로컨트롤러, 하나 이상의 주문형 집적 회로(ASIC), 하나 이상의 필드 프로그램가능 게이트 어레이(FPGA) 회로, 임의의 다른 타입 및 개수의 집적 회로(IC), 상태 머신 등일 수 있다. 프로세서(32)는 신호 코딩, 데이터 처리, 전력 제어, 입/출력 처리, 및/또는 M2M 디바이스(30)가 무선 환경에서 동작할 수 있게 하는 임의의 다른 기능성을 수행할 수 있다. 프로세서(32)는 송/수신 요소(36)에 연결될 수 있는 송수신기(34)에 연결될 수 있다. 도 18c는 프로세서(32) 및 송수신기(34)를 별개의 컴포넌트들로서 도시하고 있지만, 프로세서(32) 및 송수신기(34)는 전자 패키지 또는 칩에서 함께 통합될 수 있다는 것이 인식될 것이다. 프로세서(32)는 애플리케이션 계층 프로그램들(예를 들어, 브라우저들) 및/또는 라디오 액세스 계층(RAN: radio access-layer) 프로그램들 및/또는 통신들을 수행할 수 있다. 프로세서(32)는, 예를 들어, 액세스 계층 및/또는 애플리케이션 계층에서와 같이, 인증, 보안 키 합의(security key agreement) 및/또는 암호화 동작들과 같은 보안 동작들을 수행할 수 있다.

[0114] 송/수신 요소(36)는 M2M 서비스 플랫폼(9)으로 신호들을 송신하고/하거나, 이러한 서비스 플랫폼으로부터 신호들을 수신하도록 구성될 수 있다. 예를 들어, 실시예에서, 송/수신 요소(36)는 RF 신호들을 송신 및/또는 수신하도록 구성된 안테나일 수 있다. 송/수신 요소(36)는 다양한 네트워크들 및 무선 인터페이스들(air interfaces), 예컨대 WLAN, WPAN, 셀룰러 등을 지원할 수 있다. 실시예에서, 송/수신 요소(36)는, 예를 들어, IR, UV 또는 가시 광 신호들을 송신 및/또는 수신하도록 구성된 방출기/검출기일 수 있다. 다른 실시예에서, 송/수신 요소(36)는 RF 및 광 신호 양쪽 모두를 송신 및 수신하도록 구성될 수 있다. 송/수신 요소(36)는 무선 또는 유선 신호들의 임의의 조합을 송신 및/또는 수신하도록 구성될 수 있다는 것이 인식될 것이다.

[0115] 또한, 송/수신 요소(36)가 도 18c에 단일 요소로서 도시되어 있지만, M2M 디바이스(30)는 임의의 개수의 송/수신 요소(36)를 포함할 수 있다. 보다 구체적으로는, M2M 디바이스(30)는 MIMO 기술을 이용할 수 있다. 따라서, 실시예에서, M2M 디바이스(30)는 무선 신호들을 송신 및 수신하기 위한 2개 이상의 송/수신 요소(36)(예를 들어, 다수의 안테나)를 포함할 수 있다.

[0116] 송수신기(34)는 송/수신 요소(36)에 의해 송신되는 신호들을 변조하고, 송/수신 요소(36)에 의해 수신되는 신호들을 복조하도록 구성될 수 있다. 위에서 언급된 바와 같이, M2M 디바이스(30)는 다중 모드 능력을 가질 수 있다. 따라서, 송수신기(34)는, M2M 디바이스(30)가 예를 들어 UTRA 및 IEEE 802.11과 같은 다수의 RAT를 통해 통신할 수 있게 하기 위한 다수의 송수신기를 포함할 수 있다.

- [0117] 프로세서(32)는, 비착탈식 메모리(44) 및/또는 착탈식 메모리(46)와 같은 임의의 타입의 적합한 메모리로부터의 정보에 액세스하고, 이러한 메모리에 데이터를 저장할 수 있다. 비착탈식 메모리(44)는 랜덤 액세스 메모리(RAM), 판독 전용 메모리(ROM), 하드 디스크, 또는 임의의 다른 타입의 메모리 스토리지 디바이스를 포함할 수 있다. 착탈식 메모리(46)는 SIM(subscriber identity module) 카드, 메모리 스틱, SD(secure digital) 메모리 카드 등을 포함할 수 있다. 다른 실시예들에서, 프로세서(32)는, M2M 디바이스(30) 상에, 예컨대 서버나 홈 컴퓨터 상에 물리적으로 위치되지는 않은 메모리로부터의 정보에 액세스하고, 이러한 메모리에 데이터를 저장할 수 있다. 프로세서(32)는, 다양한 상태들 및 파라미터들, 예컨대 발생 및/또는 수신되는 콘텍스트 업데이트들에 응답, 송신 및 발생되는 가상화 요청들, 및 본 명세서에 제시된 실시예들 중 일부 실시예들에 설명된 것들과 같은 임의의 다른 가상화 관련 상태들 및 파라미터들에 응답하여, 디스플레이 또는 인디케이터들(42) 상의 조명 패턴들, 이미지들 또는 컬러들을 제어하도록 구성될 수 있다.
- [0118] 프로세서(32)는 전원(48)으로부터 전력을 수신할 수 있고, M2M 디바이스(30)에서의 다른 컴포넌트들에 전력을 분배하고/하거나 이러한 전력을 제어하도록 구성될 수 있다. 전원(48)은 M2M 디바이스(30)에 전력을 공급하기 위한 임의의 적합한 디바이스일 수 있다. 예를 들어, 전원(48)은 하나 이상의 건전지 배터리(예를 들어, 니켈-카드뮴(NiCd), 니켈-아연(NiZn), NiMH(nickel metal hydrate), 리튬-이온(Li-ion) 등), 태양 전지, 연료 전지 등을 포함할 수 있다.
- [0119] 또한, 프로세서(32)는, M2M 디바이스(30)의 현재 위치에 관한 위치 정보(예를 들어, 경도와 위도)를 제공하도록 구성될 수 있는 GPS 칩셋(50)에 연결될 수 있다. M2M 디바이스(30)는 실시예에 부합하게 유지하면서 임의의 적합한 위치 결정 방법에 의해 위치 정보를 취득할 수 있다는 것이 인식될 것이다.
- [0120] 프로세서(32)는, 부가적인 특징, 기능성, 및/또는 유선 또는 무선 접속성을 제공하는 하나 이상의 소프트웨어 및/또는 하드웨어 모듈을 포함할 수 있는 다른 주변 장치들(52)에 또한 연결될 수 있다. 예를 들어, 주변 장치들(52)은 가속도계, 전자 나침반, 위성 송수신기, 센서, (사진 또는 비디오를 위한) 디지털 카메라, USB(universal serial bus) 포트, 진동 디바이스, 텔레비전 송수신기, 핸즈프리 헤드셋, 블루투스® 모듈, FM(frequency modulated) 라디오 유닛, 디지털 뮤직 플레이어, 미디어 플레이어, 비디오 게임 플레이어 모듈, 인터넷 브라우저 등을 포함할 수 있다.
- [0121] 도 18d는 예를 들어 도 18a 및 도 18b의 M2M 서비스 플랫폼(22)이 구현될 수 있는 예시적인 컴퓨팅 시스템(90)의 블록도이다. 컴퓨팅 시스템(90)은 컴퓨터 또는 서버를 포함할 수 있고, 소프트웨어가 어디에 또는 어느 수단에 의해 저장되거나 액세스되든지, 주로 이러한 소프트웨어의 형태로 이루어질 수 있는 컴퓨터 판독가능 명령어들에 의해 제어될 수 있다. 이러한 컴퓨터 판독가능 명령어들은 중앙 처리 유닛(CPU)(91) 내에서 실행되어, 컴퓨팅 시스템(90)이 작동하게 할 수 있다. 많은 알려진 워크스테이션, 서버 및 개인용 컴퓨터에서, 중앙 처리 유닛(91)은 마이크로프로세서라고 지칭되는 단일 칩 CPU에 의해 구현된다. 다른 머신들에서, 중앙 처리 유닛(91)은 다수의 프로세서를 포함할 수 있다. 코프로세서(81)는 메인 CPU(91)와는 별개의 선택적인 프로세서이며, 부가적인 기능들을 수행하거나 CPU(91)를 돋는다. CPU(91) 및/또는 코프로세서(81)는 IoT 가상화 서비스들을 위해 개시된 시스템들 및 방법들에 관련되는 데이터를 수신하고, 발생시키며 처리할 수 있다.
- [0122] 동작 시에, CPU(91)는 명령어들을 폐치, 디코드 및 실행하고, 컴퓨터의 메인 데이터 전송 경로인 시스템 버스(80)를 통해 다른 자원들로/로부터 정보를 전송한다. 이러한 시스템 버스는 컴퓨팅 시스템(90)에서의 컴포넌트들을 접속하고, 데이터 교환을 위한 매체를 정의한다. 통상적으로, 시스템 버스(80)는 데이터를 송신하기 위한 데이터 라인들, 어드레스들을 송신하기 위한 어드레스 라인들, 및 인터럽트들을 송신하고 시스템 버스를 동작시키기 위한 제어 라인들을 포함한다. 이러한 시스템 버스(80)의 예는 PCI(Peripheral Component Interconnect) 버스이다.
- [0123] 시스템 버스(80)에 연결된 메모리 디바이스들은 랜덤 액세스 메모리(RAM)(82) 및 판독 전용 메모리(ROM)(93)를 포함한다. 이러한 메모리들은 정보가 저장 및 검색되는 것을 허용하는 회로를 포함한다. ROM(93)은 일반적으로 쉽게 수정될 수 없는 저장된 데이터를 포함한다. RAM(82)에 저장된 데이터는 CPU(91) 또는 다른 하드웨어 디바이스들에 의해 판독되거나 변경될 수 있다. RAM(82) 및/또는 ROM(93)에 대한 액세스는 메모리 제어기(92)에 의해 제어될 수 있다. 메모리 제어기(92)는, 명령어들이 실행될 때 가상 어드레스들을 물리 어드레스들로 변환하는 어드레스 변환 기능을 제공할 수 있다. 메모리 제어기(92)는, 시스템 내의 프로세스들을 격리시키며 시스템 프로세스들을 사용자 프로세스들로부터 격리시키는 메모리 보호 기능을 또한 제공할 수 있다. 따라서, 제1 모드에서 구동되는 프로그램은 그 자신의 프로세스의 가상 어드레스 공간에 의해 매핑된 메모리에만 액세스 할 수 있고; 프로세스들 사이에 공유하는 메모리가 셋업되지 않았다면 다른 프로세스의 가상 어드레스 공간 내

의 메모리에는 액세스할 수 없다.

[0124] 또한, 컴퓨팅 시스템(90)은, CPU(91)로부터 주변 장치들, 예컨대 프린터(94), 키보드(84), 마우스(95) 및 디스크 드라이브(85)로 명령어들을 통신하는 것을 담당하는 주변 장치 제어기(83)를 포함할 수 있다.

[0125] 디스플레이 제어기(96)에 의해 제어되는 디스플레이(86)는 컴퓨팅 시스템(90)에 의해 발생된 비주얼 출력을 디스플레이하는데 사용된다. 이러한 비주얼 출력은 텍스트, 그래픽스, 애니메이티드 그래픽스(animated graphics) 및 비디오를 포함할 수 있다. 디스플레이(86)는 CRT 기반 비디오 디스플레이, LCD 기반 평면 패널 디스플레이, 가스 플라즈마 기반 평면 패널 디스플레이, 또는 터치 패널로 구현될 수 있다. 디스플레이 제어기(96)는 디스플레이(86)에 송신되는 비디오 신호를 발생시키는데 요구되는 전자 컴포넌트들을 포함한다.

[0126] 또한, 컴퓨팅 시스템(90)은 도 18a 및 도 18b의 네트워크(12)와 같은 외부 통신 네트워크에 컴퓨팅 시스템(90)을 접속하는데 이용될 수 있는 네트워크 어댑터(97)를 포함할 수 있다. 실시예에서, 네트워크 어댑터(97)는 IoT 가상화 서비스들을 위해 개시된 시스템들 및 방법들에 관련되는 데이터를 수신 및 송신할 수 있다.

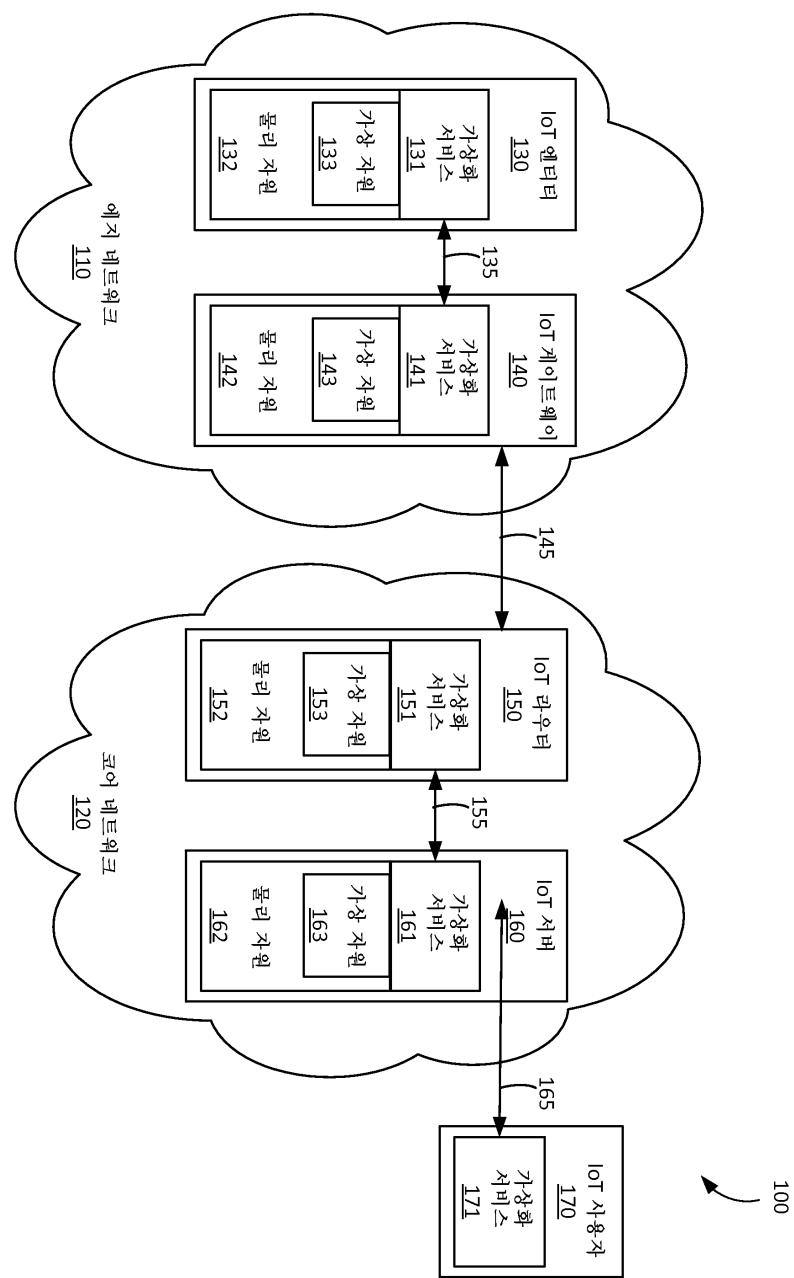
[0127] 본 명세서에 설명된 시스템들, 방법들 및 프로세스들 중 임의의 것 또는 이를 전부는 물리 디바이스 또는 장치로서 구현된 컴퓨터 관독가능 스토리지 매체 상에 저장된 컴퓨터 실행가능 명령어들(즉, 프로그램 코드)의 형태로 구현될 수 있다고 이해된다. 이러한 명령어들은, 컴퓨터, 서버, M2M 단말 디바이스, M2M 케이트웨이 디바이스 등과 같은 머신에 의해 또는 머신 내에 구성된 프로세서에 의해 실행될 때, 본 명세서에 설명된 시스템들, 방법들 및 프로세스들을 실시, 수행 및/또는 구현한다. 구체적으로는, 전술한 단계들, 동작들 또는 기능들 중 임의의 것은 이러한 컴퓨터 실행가능 명령어들의 형태로 구현될 수 있다. 컴퓨터 관독가능 스토리지 매체는 정보의 스토리지를 위해 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성의, 착탈식 및 비착탈식 매체를 포함하지만, 이러한 컴퓨터 관독가능 스토리지 매체는 신호들을 포함하지는 않는다. 컴퓨터 관독가능 스토리지 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 다른 메모리 기술, CDROM, DVD(digital versatile disks) 또는 다른 광학 디스크 스토리지, 자기 카세트, 자기 테이프, 자기 디스크 스토리지 또는 다른 자기 스토리지 디바이스, 또는 원하는 정보를 저장하는데 이용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 물리 매체를 포함하지만, 이에 제한되지는 않는다.

[0128] 도면들에 도시된 바와 같이, 본 개시물의 대상의 바람직한 실시예들을 설명하는데 있어서, 명료성을 위해 특정 용어가 이용된다. 그러나, 청구 대상은 이와 같이 선택된 특정 용어에 제한되는 것으로 의도되지는 않으며, 각각의 특정 요소는 유사한 목적을 달성하기 위해 유사한 방식으로 동작하는 모든 기술적 등가물을 포함한다고 이해되어야 한다.

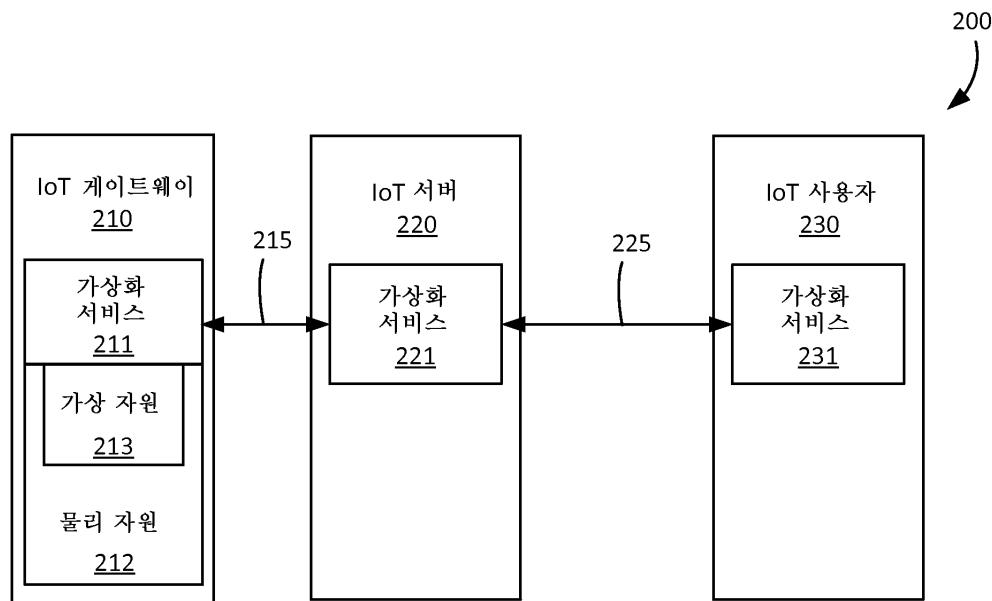
[0129] 이러한 서면 설명은 최상의 모드를 비롯하여 본 발명을 개시하기 위해, 그리고 임의의 디바이스들 또는 시스템들을 제조하고 이용하는 것 및 임의의 통합된 방법들을 수행하는 것을 비롯하여 본 발명을 실시하는 것을 관련 기술분야의 통상의 기술자가 가능하게 하기 위해 예들을 이용한다. 본 발명의 특허가능한 범위는 청구항들에 의해 정의되며, 관련 기술분야의 통상의 기술자에게 떠오르는 다른 예들을 포함할 수 있다. 이러한 다른 예들은, 그것들이 청구항들의 기재와 상이하지 않은 구조적 요소들을 갖는 경우에 또는 그것들이 청구항들의 기재와 미미한(insubstantial) 차이를 갖는 등가의 구조적 요소들을 포함하는 경우에, 청구항들의 범위 내에 있는 것으로 의도된다.

도면

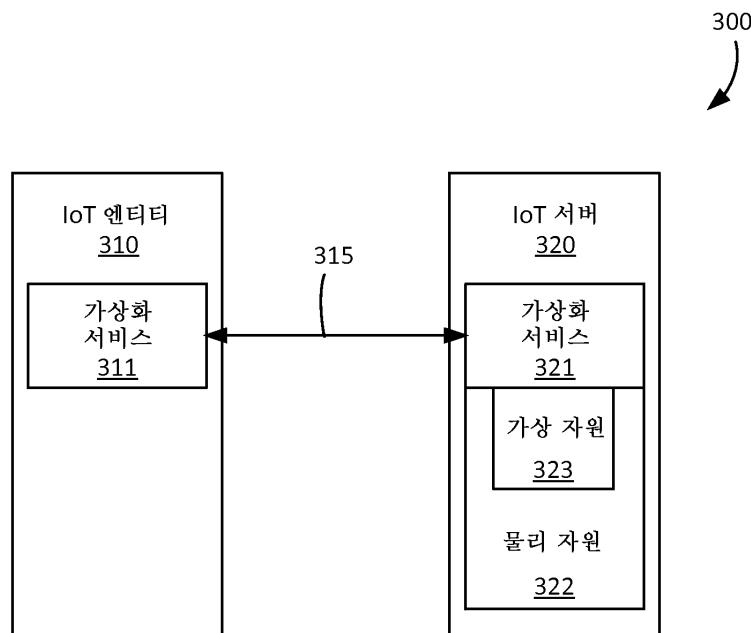
도면1



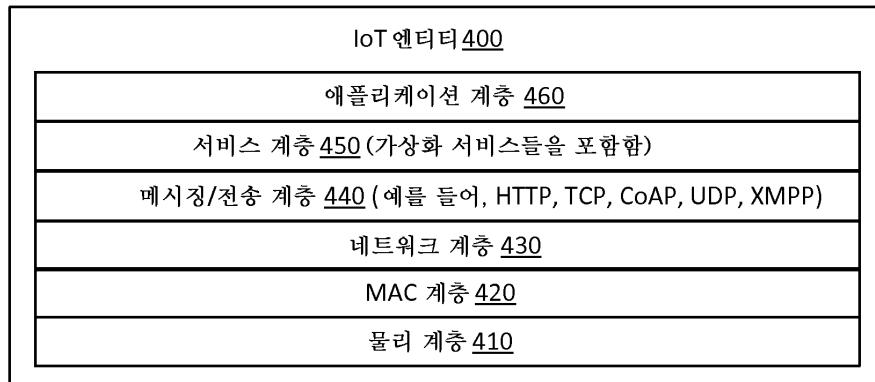
## 도면2



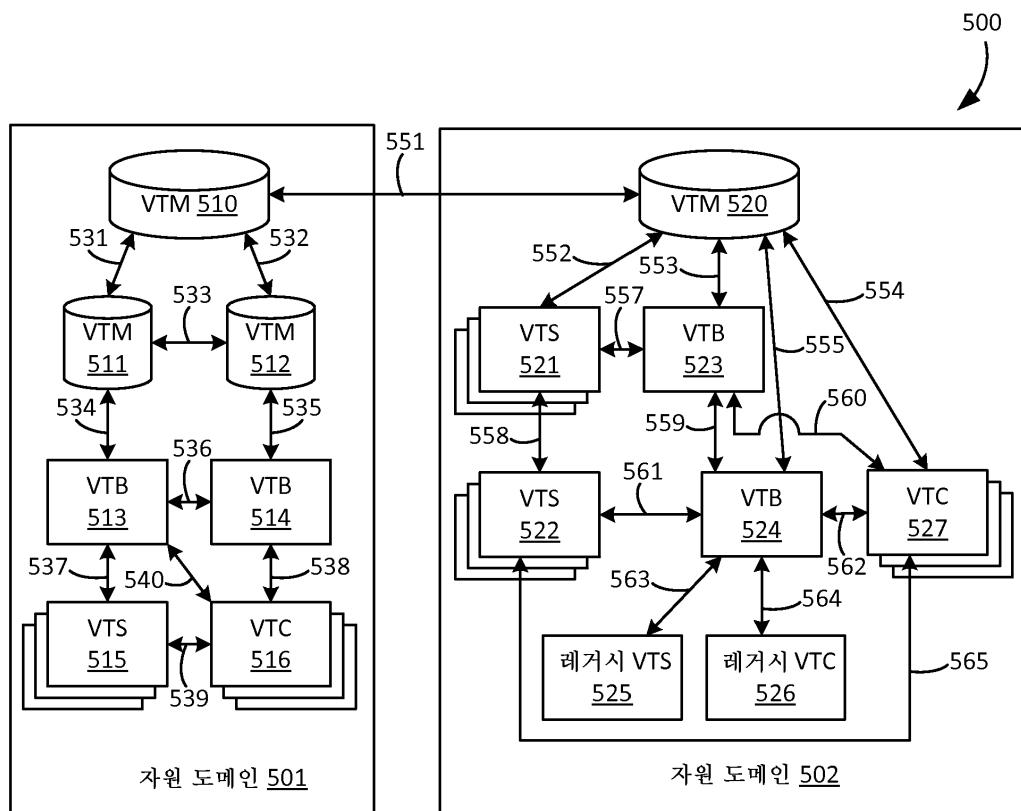
## 도면3



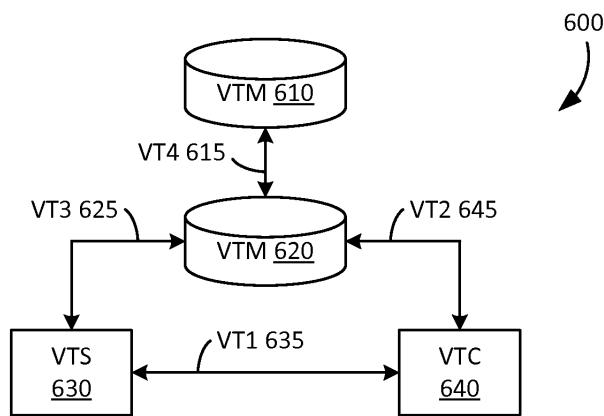
## 도면4



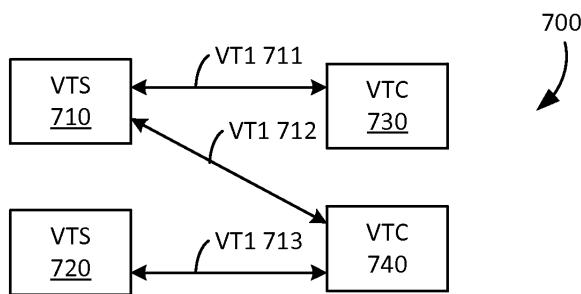
## 도면5



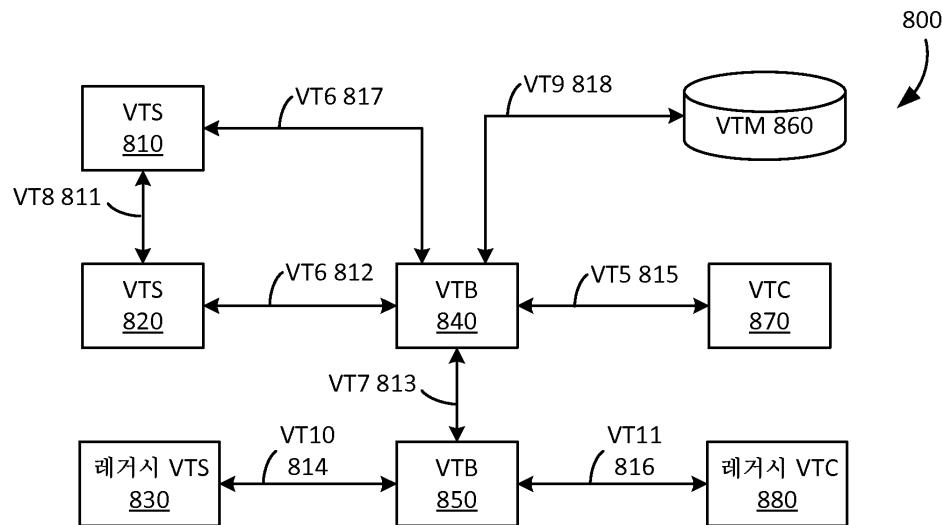
## 도면6



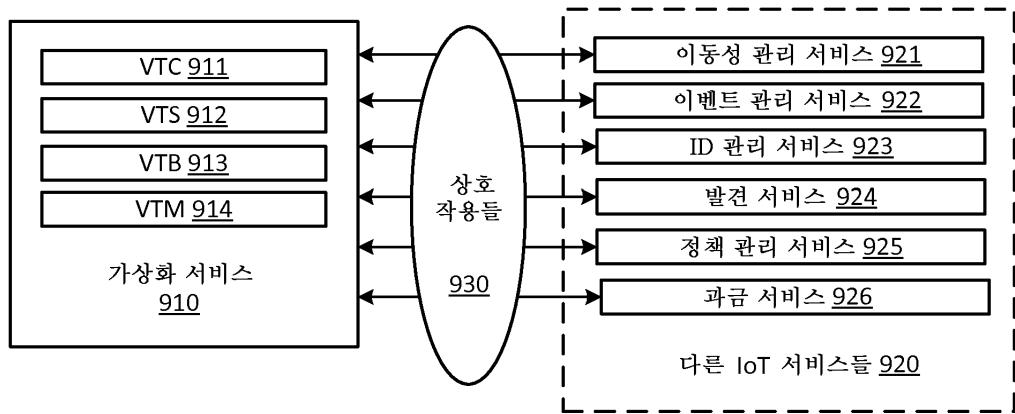
## 도면7



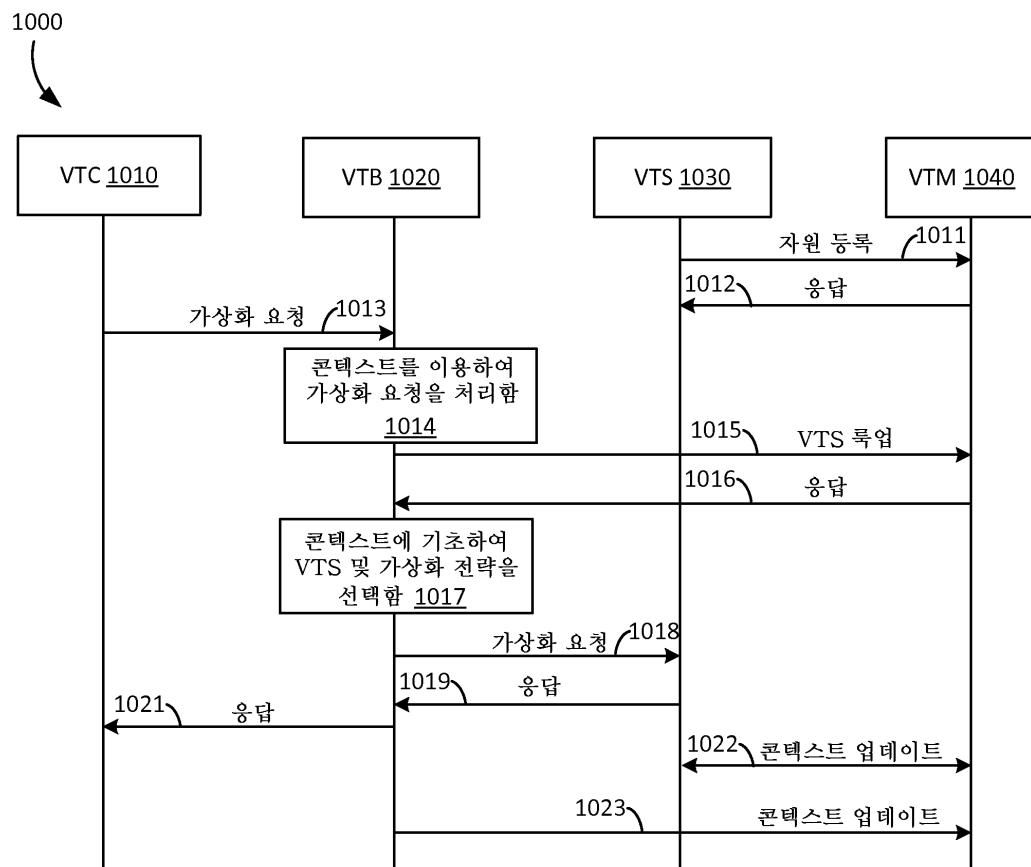
## 도면8



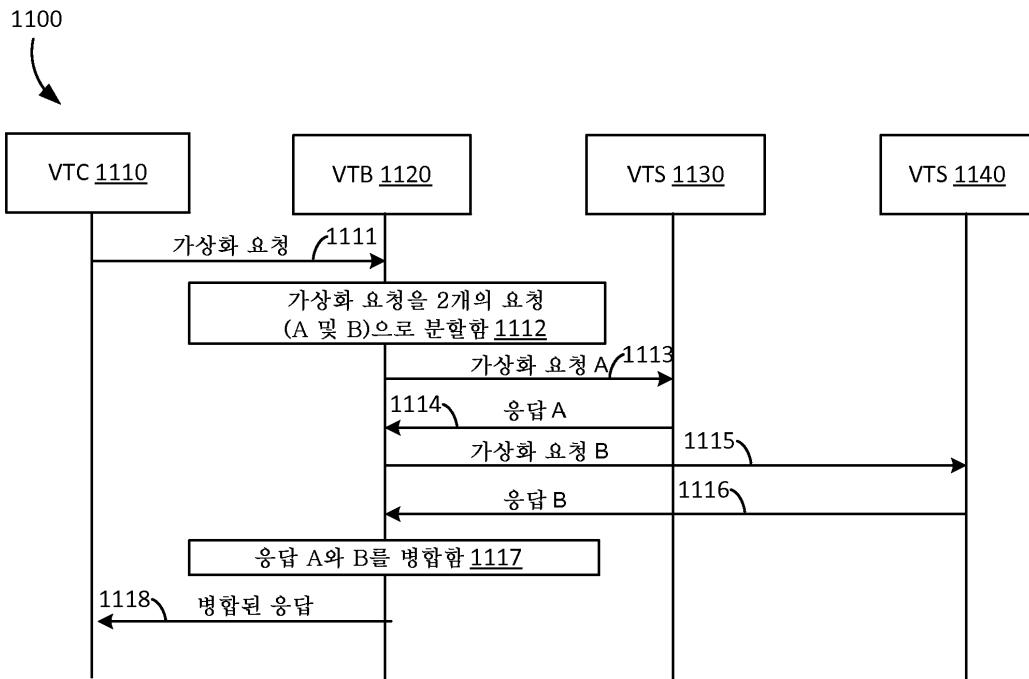
## 도면9



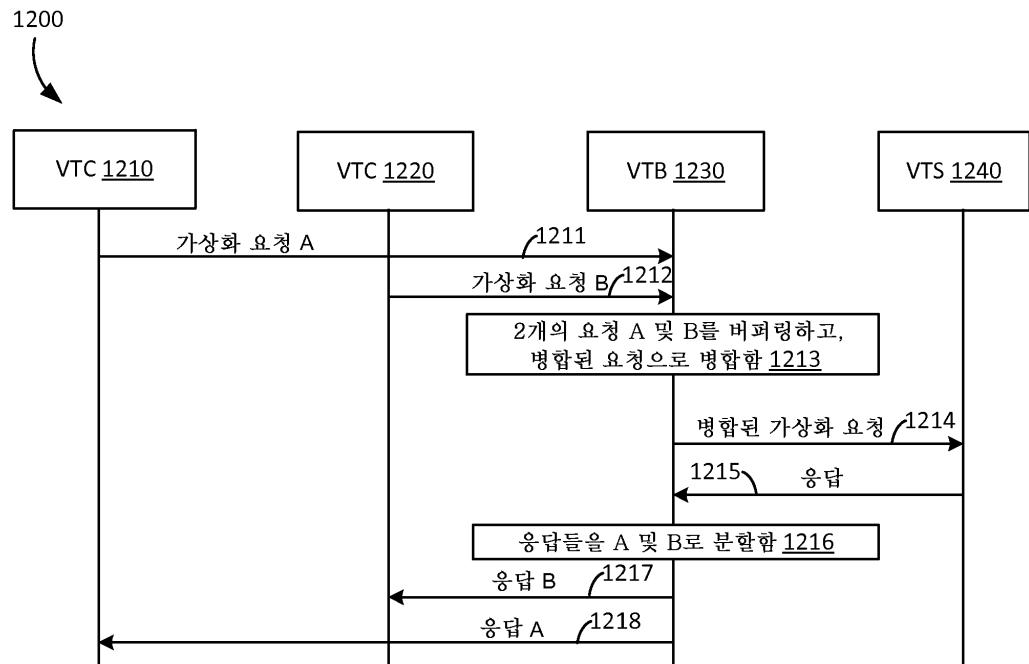
## 도면10



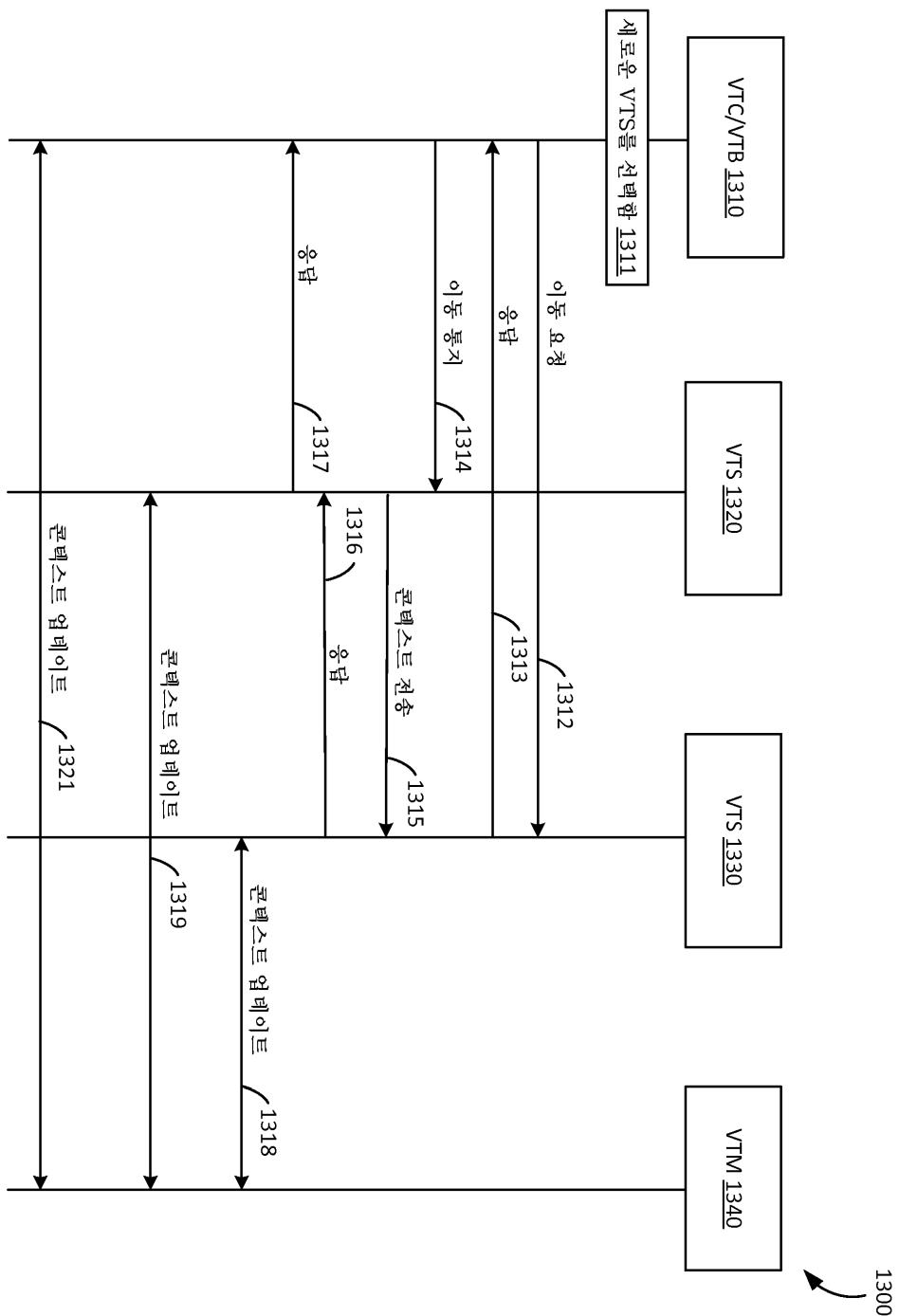
## 도면11



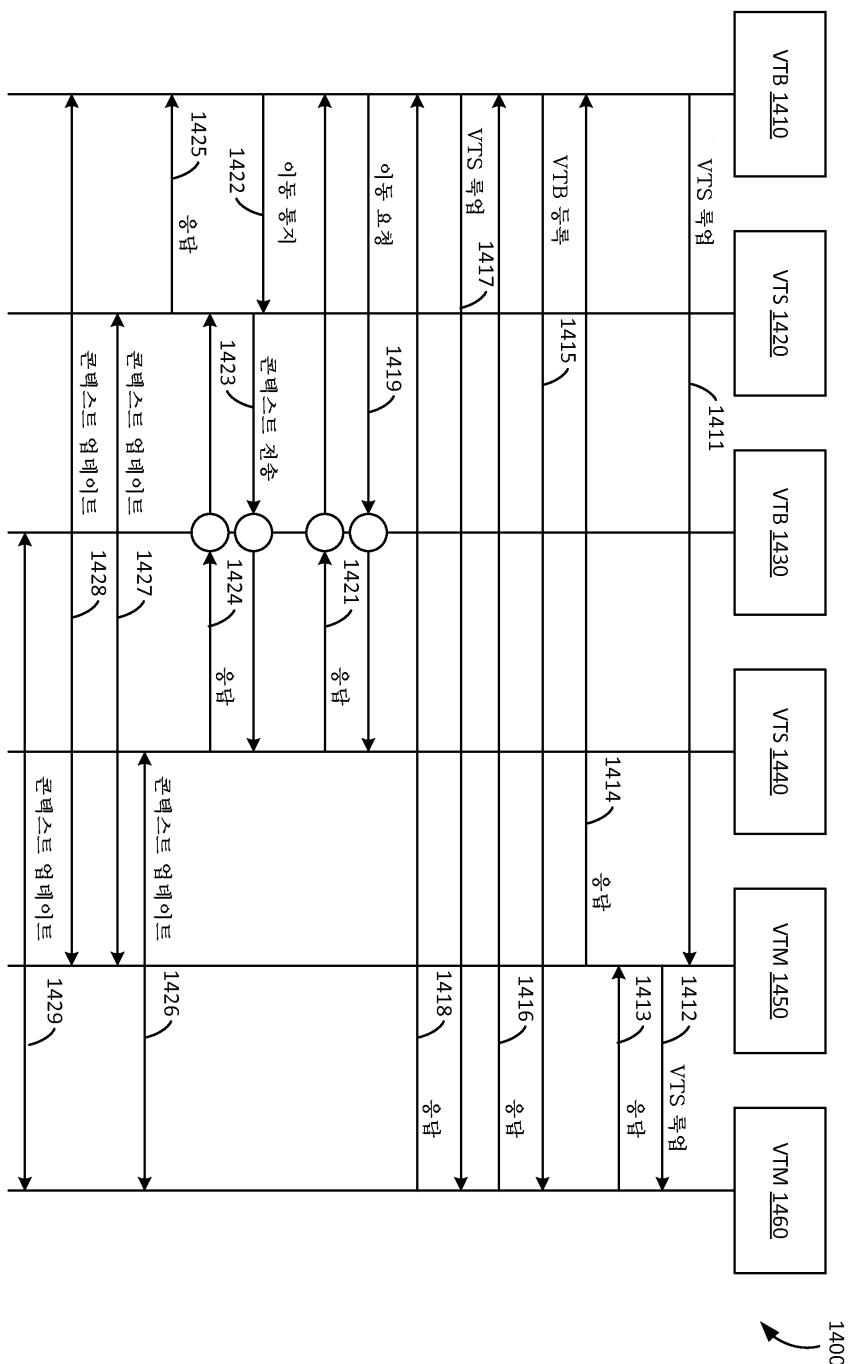
## 도면12



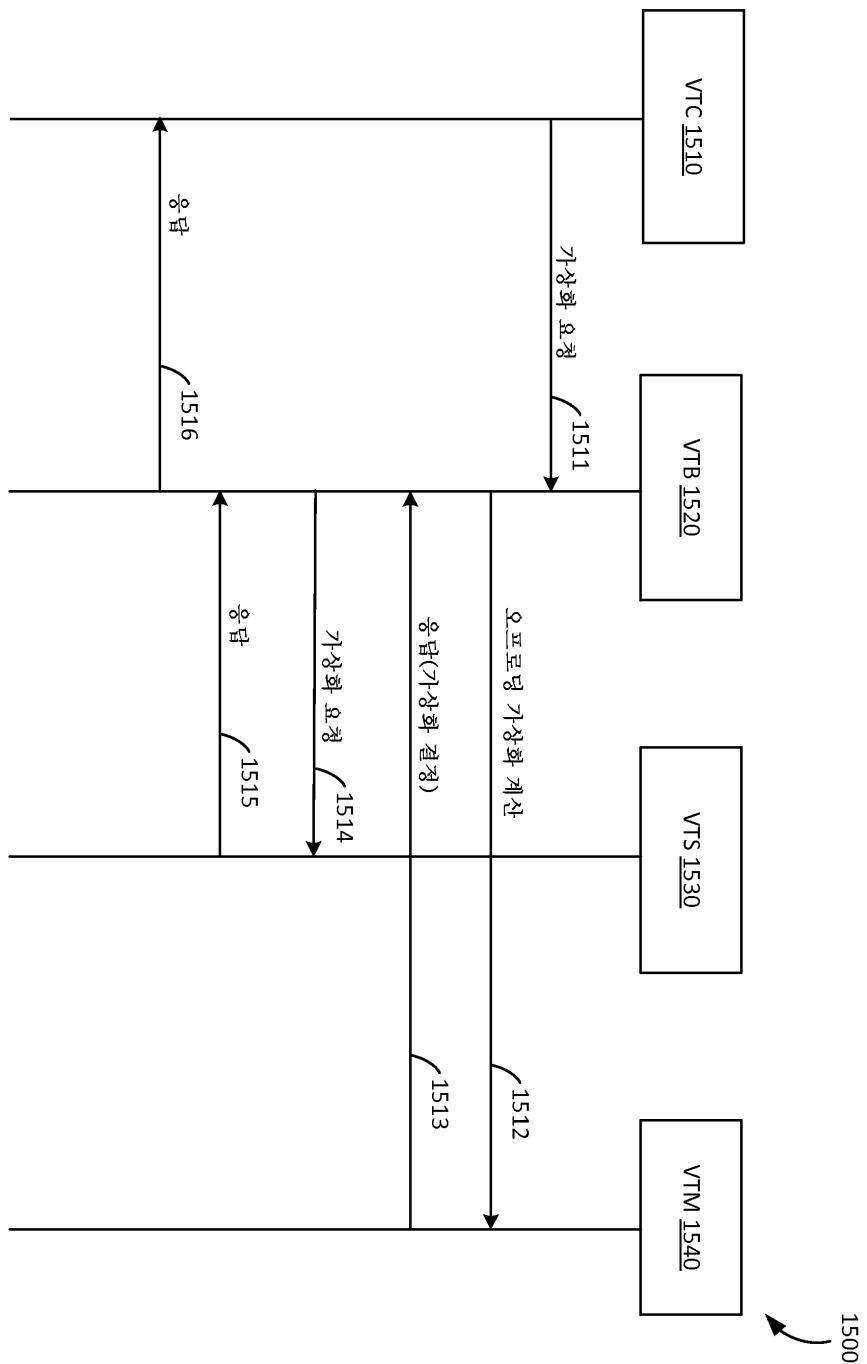
도면13



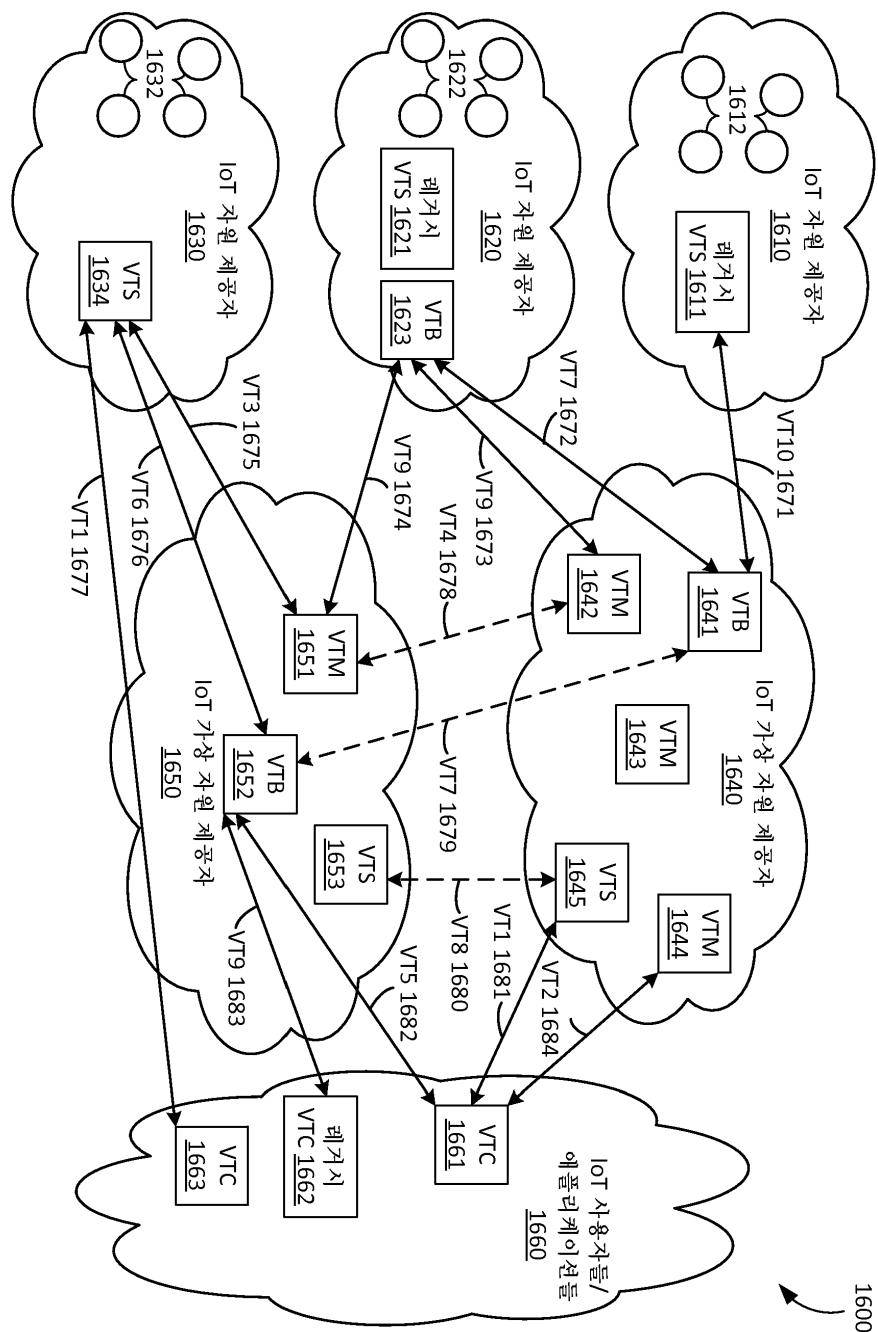
도면14



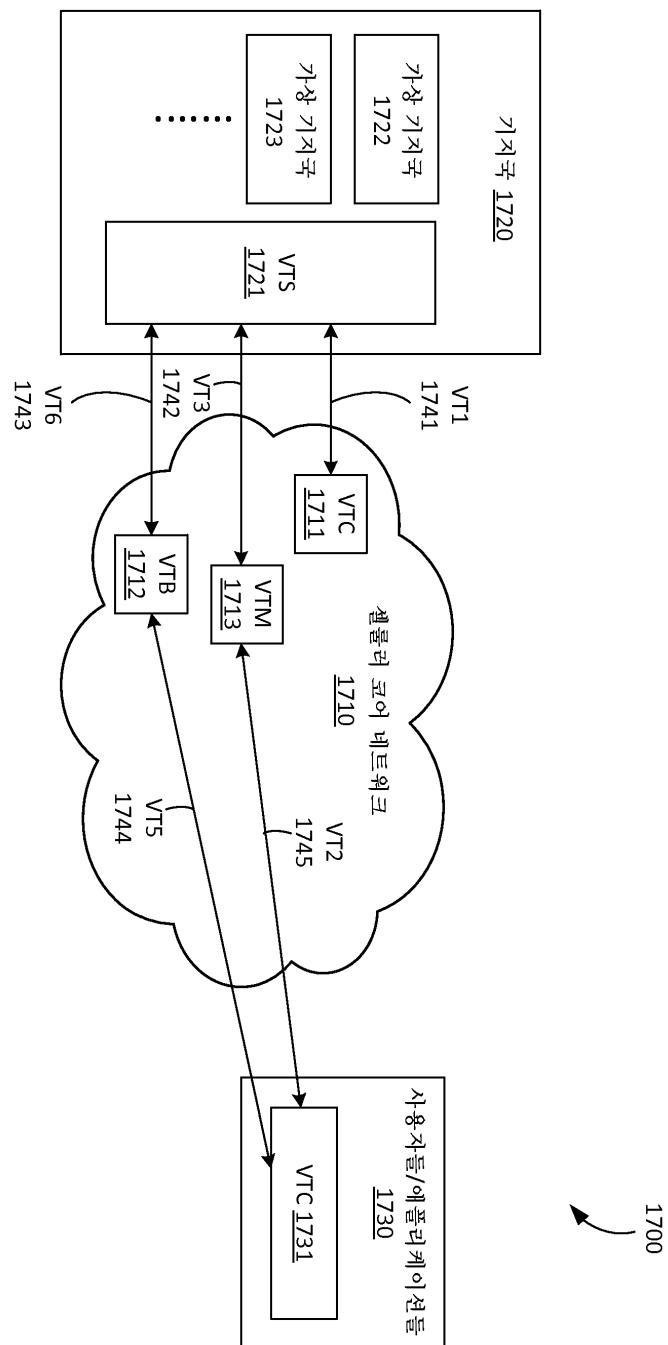
도면15



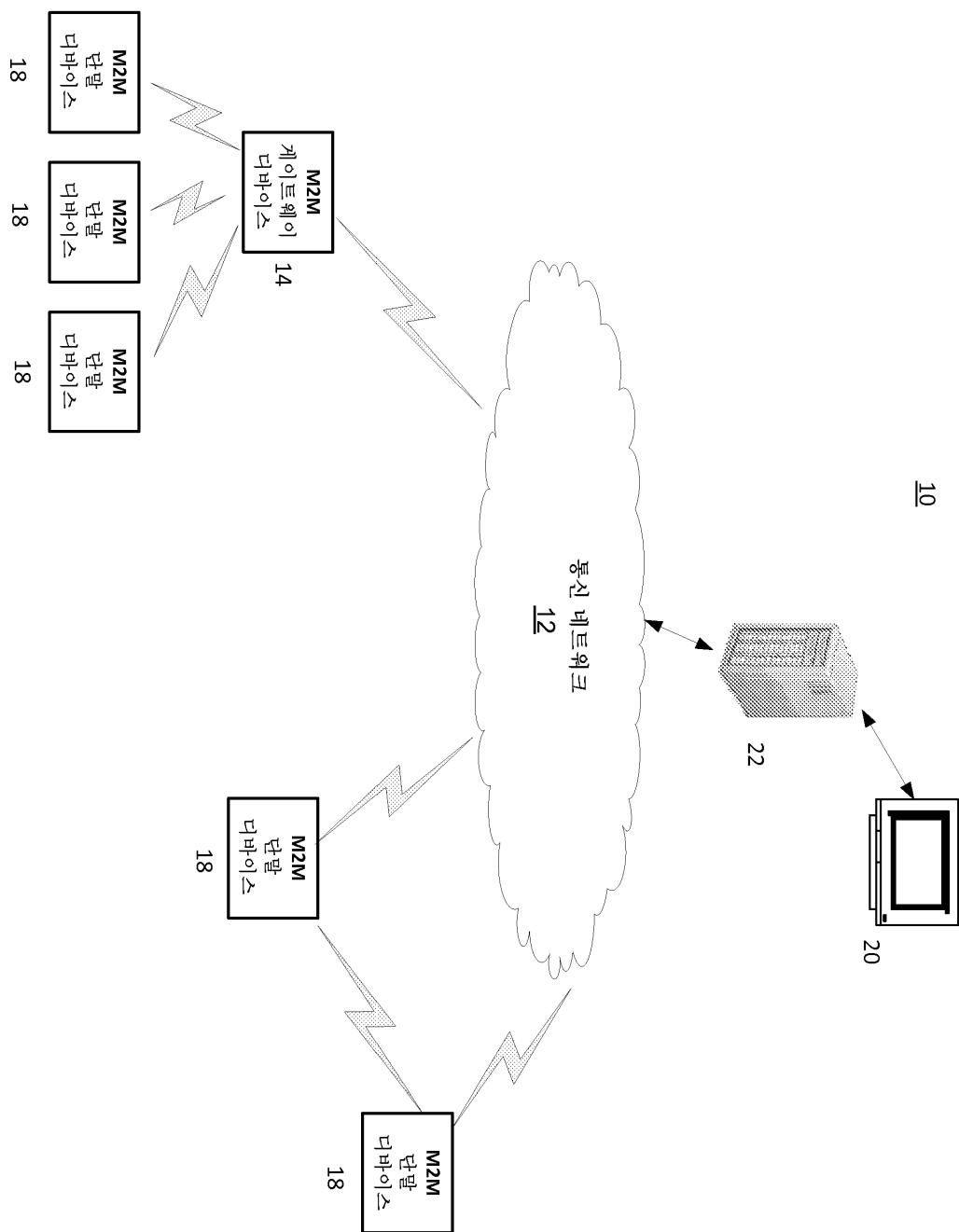
도면16



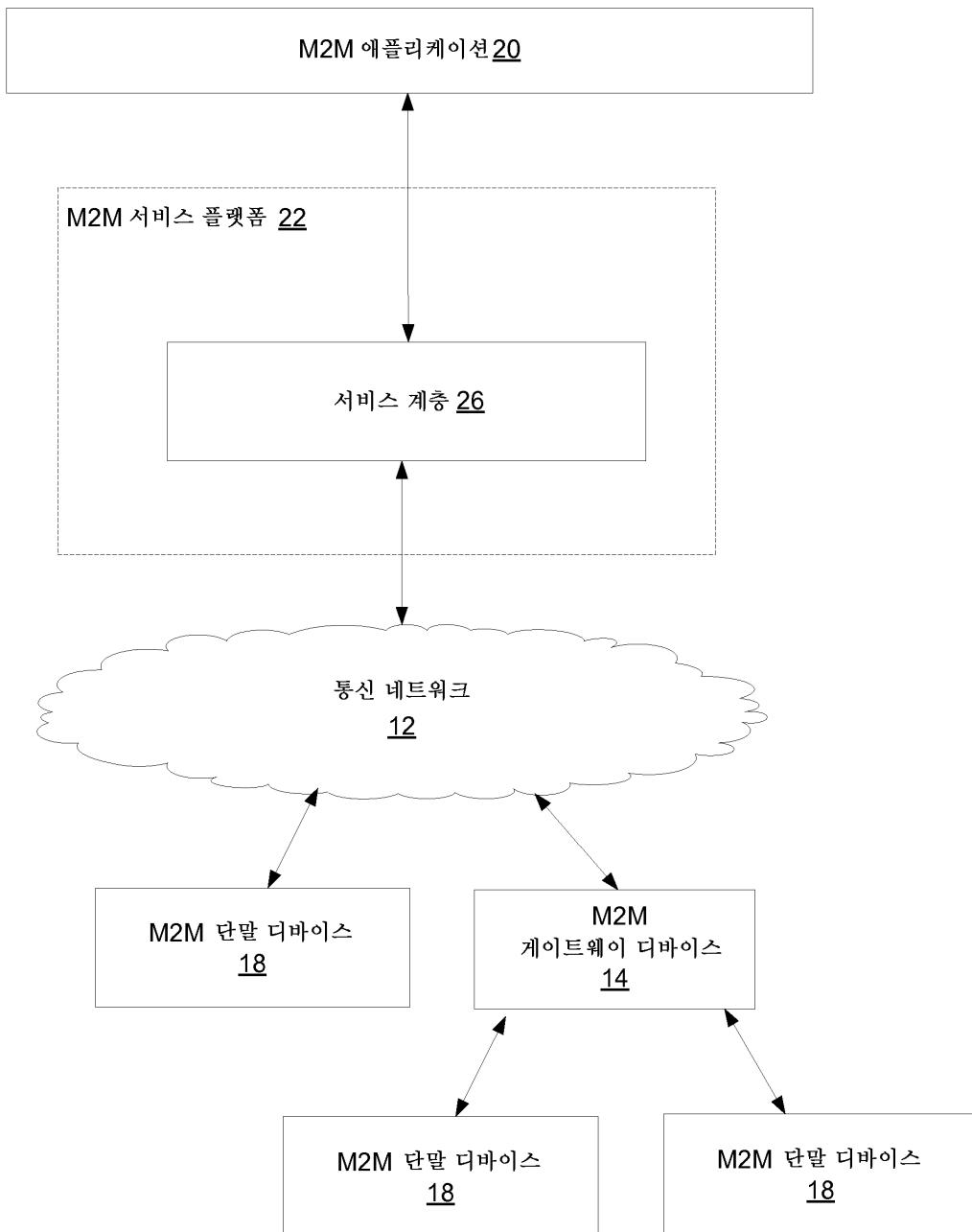
도면17



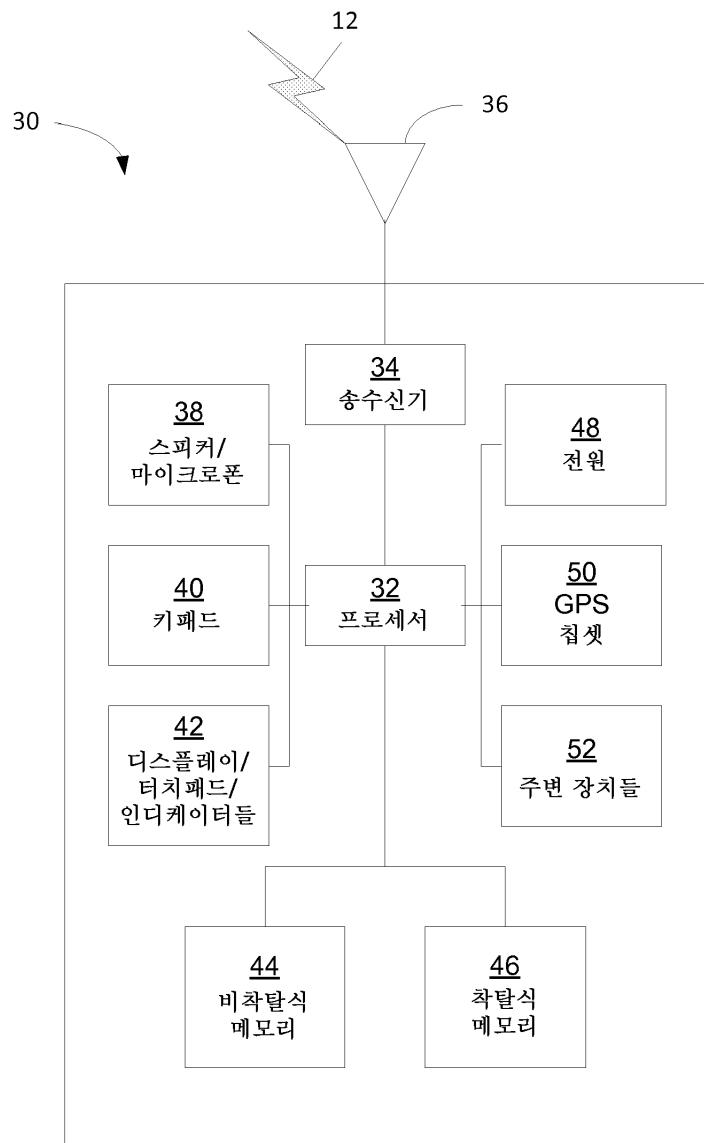
도면18a



## 도면18b



## 도면18c



## 도면 18d

