



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2012-0099239  
(43) 공개일자 2012년09월07일

(51) 국제특허분류(Int. Cl.)  
G06F 13/16 (2006.01) G06F 13/38 (2006.01)  
(21) 출원번호 10-2012-7013558  
(22) 출원일자(국제) 2010년10월26일  
심사청구일자 없음  
(85) 번역문제출일자 2012년05월25일  
(86) 국제출원번호 PCT/US2010/054172  
(87) 국제공개번호 WO 2011/056612  
국제공개일자 2011년05월12일  
(30) 우선권주장  
61/255,002 2009년10월26일 미국(US)

(71) 출원인  
웨어러블, 인코포레이티드  
미국 일리노이주 60062 노쓰브룩 찰스 드라이브  
3825  
(72) 발명자  
마스텐브룩, 브라이언, 이.  
미국 일리노이주 60193 샤움버그 브라이튼 코트  
1015  
클라프맨, 매튜, 에이치.  
미국 일리노이주 60062 노쓰브룩 찰스 드라이브  
3825  
(74) 대리인  
특허법인 동원

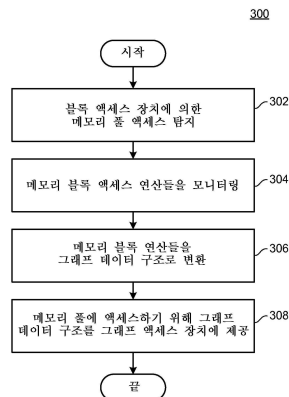
전체 청구항 수 : 총 21 항

(54) 발명의 명칭 블록 액세스 장치와 그래프 액세스 장치 사이에 공유된 메모리 풀에 대한 동시 액세스

(57) 요약

그래프 액세스 장치 및 블록 액세스 장치는 두 장치들 사이에 공유된 메모리 풀에 동시에 액세스할 수 있다. 메모리 풀은 단일 논리 메모리로서 액세스된 하나 이상의 메모리 어레이들을 포함할 수 있다. 블록 액세스 장치는 메모리 풀에 메모리 블록들의 플랫 어레이(flat array)로서 액세스하고, 그래프 액세스 장치는 메모리 풀에 계층적 파일 시스템으로서 액세스한다. 블록 액세스 장치가 메모리 풀에 액세스하는 동안에, 그것에 의해 수행된 하나 이상의 메모리 블록 액세스 연산들을 모니터링함으로써 동시 액세스가 달성된다. 블록 액세스 연산들은 메모리 풀을 계층적 파일 시스템에 맵핑하는 복수의 포인터들을 포함하는 그래프 데이터 구조로 변환된다. 프로세서는 메모리 풀에 대한 액세스를 조절하고, 그래프 액세스 장치를 그래프 데이터 구조에 따라 블록 액세스 장치와 동시에 메모리 풀에 액세스시키도록 구성된다.

대표도 - 도10



## 특허청구의 범위

### 청구항 1

메모리 풀에 액세스하면서 블록 액세스 장치에 의해 수행된 하나 이상의 메모리 블록 액세스 연산들을 모니터링하는 단계;

상기 블록 액세스 연산들을, 상기 메모리 풀을 계층적 파일 시스템에 맵핑하는 복수의 포인터들을 포함하는, 그래프 데이터 구조로 변환시키는 단계; 및

상기 그래프 액세스 장치가, 상기 그래프 데이터 구조에 따라, 상기 블록 액세스 장치와 동시에 상기 메모리 풀에 액세스하는 단계를 포함하여 구성되는,

그래프 액세스 장치와 블록 액세스 장치가 두 장치들 사이에 공유된 메모리 풀에 동시에 액세스하는 방법.

### 청구항 2

제1항에 있어서,

상기 블록 액세스 장치가 상기 메모리 풀의 제1 메모리 블록에 쓰기(write)를 시도하는 단계;

상기 그래프 액세스 장치가 현재 상기 제1 메모리 블록에 액세스하고 있는 지를 결정하는 단계;

상기 그래프 액세스 장치가 상기 제1 메모리 블록에 액세스하고 있는 경우에, 상기 블록 액세스 장치가 상기 제1 메모리 블록 대신에 제2 메모리 블록에 쓰기를 수행하는 단계;

상기 그래프 데이터 구조를 분해하여, 상기 블록 액세스 장치에 의해 수행된 쓰기의 적어도 하나의 메모리 위치(memory location)를 표시하는 단계를 더 포함하여 구성되는, 방법.

### 청구항 3

제2항에 있어서, 상기 그래프 데이터 구조를 분해하는 단계가, 상기 그래프 데이터 구조의 포인터를 상기 제1 메모리 블록을 표시하는 것에서 상기 제2 메모리 블록을 표시하는 것으로 바꾸는 것을 포함하는, 방법.

### 청구항 4

제2항에 있어서, 상기 그래프 데이터 구조를 분해하는 단계가, 상기 그래프 액세스 장치가 상기 제1 메모리 블록에 대한 그것의 액세스를 완료한 후에, 상기 제2 메모리 블록의 콘텐츠를 상기 제1 메모리 블록에 복사하는 것을 포함하는, 방법.

### 청구항 5

제1항에 있어서, 상기 메모리 풀이 단일 논리 메모리(single logical memory)인, 방법.

### 청구항 6

제1항에 있어서, 상기 메모리 풀이 주변 장치에 포함되는, 방법.

### 청구항 7

제1항에 있어서, 상기 메모리 풀이 복수의 개별적인 네트워크화 장치들 사이에 분산되는, 방법.

### 청구항 8

제1항에 있어서, 상기 그래프 액세스 장치가 무선 링크를 통해 상기 메모리 풀에 액세스하도록 구성된 무선 클라이언트를 포함하는, 방법.

### 청구항 9

제1항에 있어서, 상기 블록 액세스 장치가 USB 연결을 통해 상기 메모리 풀에 액세스하도록 구성된 USB 호스트를 포함하는, 방법.

#### 청구항 10

제1항에 있어서, 상기 블록 액세스 장치가, 상기 메모리 블록 액세스 연산들 동안에 상기 메모리 풀에 액세스 하기 위해, 각기 상기 메모리 풀에 위치된, 이산 시작 위치 메모리 어드레스 및 이산 끝 위치 메모리 어드레스를 제공하는, 방법.

#### 청구항 11

제1항에 있어서, 상기 모니터링 단계가:

상기 블록 메모리 액세스 연산들이 상기 그래프 데이터 구조를 저장하는 상기 메모리 풀의 메모리 위치들에 쓰기를 수행하고 있는지를 결정하는 단계; 및

상기 블록 메모리 액세스 연산들이, 데이터 콘텐츠를 저장하는 상기 메모리 풀의 메모리 위치들에, 쓰기를 수행하고 있는지를 결정하는 단계를 포함하는, 방법.

#### 청구항 12

제11항에 있어서, 상기 변환시키는 단계가, 상기 블록 메모리 액세스 연산들 동안에 상기 블록 액세스 장치에 의해 쓰기가 수행된 상기 메모리 위치들을 토대로 상기 그래프 데이터 구조를 업데이트하는 것을 포함하는, 방법.

#### 청구항 13

제1항에 있어서, 상기 변환시키는 단계가, 상기 메모리 풀의 미리 정해진 메모리 어드레스 위치들에 액세스하는 블록 메모리 액세스 연산들의 횟수(count)를 토대로 상기 그래프 데이터 구조를 업데이트하는 것을 포함하는, 방법.

#### 청구항 14

그래프 액세스 장치와 블록 액세스 장치 사이에 공유된 메모리 풀;

상기 메모리 풀에 액세스 하면서, 상기 블록 액세스 장치에 의해 수행된 하나 이상의 메모리 블록 액세스 연산들을 모니터링하기 위한 수단;

상기 블록 액세스 연산들을, 상기 메모리 풀을 계층적 파일 시스템에 맵핑하는 복수의 포인터들을 포함하는 그래프 데이터 구조로 변환시키기 위한 수단; 및

그래프 액세스 장치를, 상기 그래프 데이터 구조에 따라, 상기 블록 액세스 장치와 동시에 상기 메모리 풀에 액세스시키기 위한 수단을 포함하여 구성되는, 장치.

#### 청구항 15

그래프 액세스 장치와 블록 액세스 장치 사이에 공유된 메모리 풀;

상기 블록 액세스 장치와 통신하도록 구성된 제1 통신 인터페이스;

상기 그래프 액세스 장치와 통신하도록 구성된 제2 통신 인터페이스;

상기 메모리, 그리고 상기 제1 및 제2 통신 인터페이스들에 작동가능하게 결합된 것으로서,

상기 메모리에 액세스하면서, 상기 블록 액세스 장치에 의해 수행된 하나 이상의 메모리 블록 액세스 연산들을 모니터링하고,

상기 블록 액세스 연산들을, 상기 메모리를 계층적 파일 시스템에 맵핑하는 복수의 포인터들을 포함하는 그래프 데이터 구조로 변환시키고, 그리고

상기 블록 액세스 장치가 또한 상기 메모리에 액세스하는 동안에, 상기 그래프 액세스 장치가 상기 그래프 데이터 구조에 따라 상기 메모리에 동시에 액세스하도록 구성된 프로세서;를 포함하여 구성되는, 주변 장치.

#### 청구항 16

제15항에 있어서, 상기 메모리가, 상기 주변 장치와 적어도 하나의 다른 별개의 장치 사이에 분산된 메모리

풀에 포함되는, 주변 장치.

#### 청구항 17

제15항에 있어서, 상기 제2 통신 인터페이스가 복수의 그래프 액세스 장치들과 통신하도록 구성되고, 상기 메모리가 상기 블록 액세스 장치와 복수의 그래프 액세스 장치들 사이에 공유되는, 주변 장치.

#### 청구항 18

제15항에 있어서, 상기 프로세서가 상기 메모리에 저장된 콘텐츠를 암호화하도록 구성된, 주변 장치.

#### 청구항 19

제15항에 있어서, 상기 프로세서가 상기 메모리에 저장된 콘텐츠를 트랜스코딩하도록(transcode) 구성된, 주변 장치.

#### 청구항 20

제15항에 있어서, 상기 프로세서가, 상기 블록 액세스 장치를 인증하고, 정식으로 인증된 블록 액세스 장치에만 상기 메모리에 대한 액세스를 승인하도록 구성된, 주변 장치.

#### 청구항 21

제15항에 있어서, 상기 프로세서가, 상기 그래프 액세스 장치를 인증하고, 정식으로 인증된 그래프 액세스 장치에만 상기 메모리에 대한 액세스를 승인하도록 구성된, 주변 장치.

### 명세서

### 기술 분야

[0001] 35 U.S.C. § 119 하의 우선권 주장

[0002] 본 특허출원은, 2009년 10월 26일자로 출원되고, 발명의 명칭이 "하나 이상의 전자 장치 및/또는 데이터 네트워크에 저장, 통신, 전산 및 서비스를 추가하기 위한 방법, 시스템, 장치 및 제조 물품(Method, Systems, Apparatuses, and Articles of Manufacture for Adding Storage, Communications, Computation, and Services to One or More Electronic Devices and/or Data Networks)"이며, 본 명세서에 완전히 기술된 것처럼 그 전체가 명백히 본 명세서의 참고문헌을 이루는, 미국 예비특허출원 제61/255,002호에 대한 우선권을 주장하는 출원이다.

[0003] 기술 분야

[0004] 본 발명은 일반적으로 전자 데이터 시스템들, 그리고 특히, 데이터 저장을 제공하는 전산 시스템들 및/또는 통신 시스템들에 관한 것이다.

### 배경 기술

[0005] 컴퓨터 산업은 주변 장치들을 위한 몇 가지 연결성 표준들(connectivity standards)을 가지고 있으며, 최근 몇 년간 가장 대중적인 것은 범용 직렬 버스(universal serial bus: USB)이다. USB 및 그 밖의 연결성 표준들은 컴퓨터를 넘어서 수많은 전자 장치들 사이의 연결을 가능하게 한다. 가장 널리 사용되는 USB 주변 장치들 중 일부는 데이터 저장과 무선 연결성을 제공하는 장치들이었다. 구체적인 용도에 특유한 연결성 프로토콜들(connectivity protocols) 및 협약들(conventions)을 정의해온 몇 가지 일반적인 USB "유형들"이 있다. 이러한 정의(definitions)는 이 프로토콜들을 따르는 USB 장치들로 하여금 호스트(host)에 특별한 소프트웨어를 추가할 필요 없이 호스트 시스템들과 상호운용성을 가지게 한다. 예를 들어, 일반적인 USB 연결성 프로토콜은, USB 호스트들로 하여금 그 규격을 따르는 여하한 USB 주변 장치에 대해/그로부터 데이터 쓰기(write)/읽기(read)를 수행할 수 있게 하는 USB 대용량 저장 규격(USB Mass Storage specification)이다.

[0006] 다기능 주변 장치들이 공지되어 있다. 예를 들어, 데이터 저장에 대한 암호화/암호해독과 같은, 보다 유용한 기능들을 할 수 있도록 중앙 처리 장치(central processing unit: CPU)를 포함하는 USB 주변 장치들이 있다. 또한, 예를 들어, 데이터 저장과 무선 연결성의 결합체들이 단일 주변 장치에 장착되었다. 이 다기능 주변 장치들이 USB 장치들인 경우에, 그들은 물리적으로 하나의 단일 주변 장치 내에 있음에도, 논리적으로 별개의

USB 장치들로서 동작하는 것이 일반적이다. 각 기능은, 호스트와 각 기능 사이의 상호운용성(interoperability)을 정의한, 그 각각의 USB 프로파일을 따르지만, 이 두 기능들은 서로 배타적으로 동작한다. 예를 들어, 데이터 저장은 USB 대용량 저장 프로토콜을 독립적으로 사용할 것인 반면, 무선 기능은 USB 무선 네트워킹 프로토콜을 사용할 것이다. 이러한 독립 기능들은 호스트를 위한 개별적인 소프트웨어 드라이버들을 필요로 하거나, 호스트에게 주변 장치의 여러 기능들에 액세스하기 위해 다양한 USB 프로토콜들을 지원하는 것을 필요로 할 수 있다.

[0007] 현존하는 USB 아키텍처들은 다음과 같은 유형들로 이루어진다. 도 1은 USB 주변 장치(14)에 연결된 USB 호스트(12)를 포함하는 전형적인 종래 기술 USB 시스템(10)을 보여준다. USB 저장 주변 장치(14)는 솔리드-스테이트 썸 드라이브(solid-state thumb drive), 회전 매체 하드-드라이브(rotating media hard-drive), CD, DVD 드라이브 또는 그 동등물일 수 있다. USB 저장 프로토콜을 작동시키기 위한 상태 머신(state machine)으로 이루어진 USB 인터페이스(16)가 있으며, 이것은 물리적 데이터 저장부(18)에 연결되어, 읽기(reading) 및/또는 쓰기(writing)를 위해 데이터의 "블럭들"(예를 들어, 어드레스로 불리낼 수 있는 512 블럭들)로 구성된 어레이의 데이터에 액세스한다. USB 호스트(12)는 파일들, 폴더들, 디렉토리들 및 저장 관리 어레이들[예를 들어, 파일 배치표들(file allocation tables)]과 같은 고차 레벨 데이터 구조들(higher level data structures)을 정의한다.

[0008] 도 2는 USB 주변 장치(22)가 USB 인터페이스(16)에 결합된 무선 또는 유선 통신 전자 기기(24)를 포함하는 것을 제외하고는, 도 1의 시스템(10)과 유사한 아키텍처(architecture)를 가진, 종래 기술 USB 통신 시스템(20)을 보여준다. 이 장치는 USB 표준에 부합할 수 있거나, 통신 전자 기기(24)에 적절히 인터페이스 연결하기 위해 호스트(12) 상에 특수한 소프트웨어 드라이버를 필요로 할 수 있다.

[0009] 도 3은 USB 주변 장치(32)가 단일 USB 주변 장치(32) 내에 데이터 저장부(18)와 무선 또는 유선 통신 전자 기기(24)의 두 가지 기능들을 결합시킨, 종래 기술 시스템(30)을 나타낸다. USB 프로토콜은 USB 호스트(12)가 두 개의 사실상 개별적인 USB 주변 장치들로 여기는 다양한 논리적 경로들을 가능케 한다. 공유 전원 공급 장치들(sharing power supplies) 및 USB 슬레이브 컨트롤러(USB slave controller)와 같은, 이러한 예에서 저장 및 무선 기능들에 의한 내부 주변 장치 구성요소들의 약간의 물리적 공유가 있지만, USB 주변 장치(32) 내에서 저장 및 무선 기능들 사이의 직접적인 통신은 없다. 그들은 논리적으로 상호 배타적인 장치들로 취급된다.

[0010] 도 4는 USB 주변 장치(42)가, USB 호스트(12)에 대해 트랜스페어런트한(transparent) USB 주변 장치(42)에 내장된 기능[이 경우에는, 암호화/암호해독(44)]을 포함하는 종래 기술 시스템(40)을 보여준다. USB 호스트(12)는 데이터 저장부(18)를 통상적인, 암호화되지 않은 데이터 저장 어레이로 여기지만, 예를 들어, 장치(42) 상의 지문 센서 또는 패스워드를 사용한 정식 인증이 옳지 않으면, 암호해독은 실패할 것이며, USB 호스트(12)는 데이터 저장부(18)에 액세스할 수 없을 것이다.

## 발명의 내용

### 해결하려는 과제

[0011] 발명의 요약

[0012] 호스트에 대해 업데이트할 필요 없이 주변 장치들에 기능성을 추가할 필요가 있다. 추가적인 호스트 소프트웨어에 대한 필요성을 감소시키는 것이, 저가 장치들의 경우에 특히, 바람직하다. 또한, 현재의 프로토콜들로 인한 단점들을 처리하기 위해 그리고/또는 사용자에게 복잡성을 가중시키지 않고서 기능성을 추가하기 위해 주변 장치들에 추가적인 기능성을 추가할 필요가 있다. 이러한 필요성 등이 본 명세서에 개시된 개선들에 의해 처리된다.

### 과제의 해결 수단

[0013] 본 발명의 한 측면에 의하면, 그래프 액세스 장치(graph access device)와 블록 액세스 장치(block access device)로 하여금 두 장치들 사이에 공유된 메모리 풀에 동시에 액세스하게 하는 방법이 제공된다. 그래프 액세스 장치는, 그래프, 예컨대, 계층적 파일 시스템 구조(hierarchical file system structure)로 구성된 데이터에 액세스하고, 그것에 의해 그래프가 데이터 구조를 사용하여 메모리 풀 내의 특정 어드레스들로 분해되는, 컴퓨터와 같은 전자 장치일 수 있다. 블록 액세스 장치는, 여하한 그래프 액세스가 먼저 이 장치 특유의 블록 메모리로 바뀌어진, 이산 시작 위치 어드레스들(discrete start location addresses) 및 끝 위치 어드레스들(end location addresses)을 사용하여 메모리에 액세스하는, 전자 장치이다. 메모리 풀은, 적어도

부분적으로, 주변 장치에 포함될 수 있다. 이 방법은, 메모리 풀에 액세스하면서 블록 액세스 장치에 의해 수행된 하나 이상의 메모리 블록 액세스 연산들(memory block access operations)을 모니터링하는 단계; 상기 블록 액세스 연산들을, 상기 메모리 풀을 계층적 파일 시스템(hierarchical file system)에 맵핑하는(mapping) 복수의 포인터들(pointers)을 포함하는 그래프 데이터 구조(graph data structure)로 변환시키는(translating) 단계; 및 상기 그래프 액세스 장치가, 상기 그래프 데이터 구조에 따라, 상기 블록 액세스 장치와 동시에 상기 메모리 풀에 액세스하는 단계를 포함한다.

[0014] 다른 측면에 의하면, 본 발명의 장치는 그래프 액세스 장치와 블록 액세스 장치 사이에 공유된 메모리 풀을 포함한다. 이 장치는 또한 메모리 풀에 액세스 하면서 블록 액세스 장치에 의해 수행된 하나 이상의 메모리 블록 액세스 연산들을 모니터링하기 위한 수단; 상기 블록 액세스 연산들을, 상기 메모리 풀을 계층적 파일 시스템에 맵핑하는 복수의 포인터들을 포함하는 그래프 데이터 구조로 변환시키기 위한 수단; 및 그래프 액세스 장치를, 상기 그래프 데이터 구조에 따라, 상기 블록 액세스 장치와 동시에 상기 메모리 풀에 액세스시키기 위한 수단을 포함한다.

[0015] 또 다른 측면에 의하면, 본 발명의 주변 장치는, 메모리, 제1 통신 인터페이스 및 제2 통신 인터페이스를 포함한다. 메모리는 그래프 액세스 장치와 블록 액세스 장치 사이에 공유된다. 제1 통신 인터페이스는 블록 액세스 장치와 통신하도록 구성되고, 제2 통신 인터페이스는 그래프 액세스 장치와 통신하도록 구성된다. 프로세서는 메모리에 액세스하면서 블록 액세스 장치에 의해 수행된 하나 이상의 메모리 블록 액세스 연산들을 모니터링하고; 블록 액세스 연산들을, 메모리를 계층적 파일 시스템에 맵핑하는 복수의 포인터들을 포함하는 그래프 데이터 구조로 변환시키고; 블록 액세스 장치가 또한 메모리에 액세스하는 동안, 그래프 액세스 장치를, 그래프 데이터 구조에 따라, 메모리에 동시에 액세스시키도록 구성된다.

[0016] 또 다른 측면에 의하면, 본 발명의 주변 장치는 외부 호스트 또는 클라이언트 장치와 통신하고, 호스트/클라이언트 장치가 블록 또는 그래프 액세스를 사용하여 통신할 필요가 있는지의 여부를 인텔리전트 결정(intelligent determination)한다. 그 다음에 주변 장치는 그에 따라 외부 장치와 통신한다.

[0017] 본 명세서에 기술된 개선들의 다른 측면들, 특징들, 장점들 및 변형들이 다음의 도면들과 상세한 설명의 검토로 이 분야의 통상의 지식을 가진 자에게 명백하거나 명백하게 될 것이다. 모든 이러한 추가적인 측면들, 특징들, 변형들 및 장점들이 이 상세한 설명에 포함되고, 첨부된 특허청구범위에 의해 보호되는 것으로 의도된다.

### 도면의 간단한 설명

[0018] 도면들은 오직 설명만을 위한 것이며, 첨부된 특허청구범위의 한계를 구획하지 않는 것으로 이해되어야 한다. 또한, 도면의 구성요소들은 반드시 일정한 비율로 작성된 것은 아니다. 도면들에서, 동일한 도면 부호들은 도면 전체에 걸쳐 같은 요소들을 나타낸다.

도 1-4는 종래 기술의 주변 장치 시스템들을 도시한다.

도 5는 메모리에 대한 동시 액세스를 가능케 하는 다기능 주변 장치를 포함하는 예시적 시스템의 특정 구성요소들을 도시한 블록 다이어그램이다.

도 6은 도 5의 다기능 주변 장치에 포함된 메모리에 저장된 예시적 콘텐츠의 컨셉 다이어그램(conceptual diagram)이다.

도 7은 둘 이상의 통신 인터페이스들을 포함하는 대안적인, 예시적 다기능 주변 장치 아키텍처의 특정 구성요소들을 도시한 블록 다이어그램이다.

도 8은 도 5에 도시된 다기능 주변 장치의 구체적인, 예시적 구현의 특정 구성요소들을 도시한 블록 다이어그램이다.

도 9는 다기능 주변 장치의 예시적인 소프트웨어 아키텍처를 도시한 다이어그램이다.

도 10은 그래프 액세스 장치와 블록 액세스 장치로 하여금 공유 메모리 풀(shared memory pool)에 동시에 액세스하게 하기 위한 예시적인 방법을 도시한 플로우차트이다.

도 11은 그래프 데이터 구조를 블록 메모리 어레이에 맵핑하는 하나의 예를 도시한 컨셉 다이어그램이다.

도 12는 블록 액세스 장치 및 그래프 액세스 장치를 위한 동시 메모리 액세스를 가능케 하기 위해 새도우 메



모리 기술(shadow memory technique)을 사용하는 하나의 예를 도시한 컨셉 다이어그램이다.

도 13은 메모리 풀에 대한 동시 액세스를 가능케 하는 예시적인 새도우 메모리 방법을 도시한 플로우차트이다.

도 14는 클라우드 컴퓨팅(cloud computing)을 지원하기 위해 복수의 장치들에 걸쳐 분산된(distributed) 메모리 풀을 도시한 컨셉 다이어그램이다.

도 15는 메모리 풀을 특정 사용자에게 한정된 영역들로 파티션하는(partitioning) 하나의 예를 도시한 컨셉 다이어그램이다.

도 16은 축소된 메모리 공간을 사용하는 TCP/IP 패킷들을 전송하는 예시적인 방법을 도시한 플로우차트이다.

도 17은 주변 장치를 포함하는 지불결제 프로세싱 시스템(payment processing system)을 도시한 컨셉 다이어그램이다.

## 발명을 실시하기 위한 구체적인 내용

### [0019] 발명의 상세한 설명

[0020] 도면들을 참조하고 구체화한 다음의 상세한 설명은, 특허청구된 것의 하나 이상의 특정 실시예들을 설명하고 예시한다. 본 발명을 한정하기 위해서가 아닌 본 발명을 예시하고 가르치기 위해 제공된 이 실시예들은 이 분야의 통상의 지식을 가진 자들이 특허청구범위에 의해 정의되는 본 발명을 실시할 수 있을 만큼 충분히 도시되고 설명되어 있다. 따라서, 본 발명을 불명료하게 만드는 것을 피하기 위해 적절한 경우에, 상세한 설명은 이 분야의 통상의 지식을 가진 자들에게 공지된 특정 정보를 생략할 수 있다.

[0021] "예시적(인)"이라는 용어는, 본 명세서 전반에 걸쳐 "하나의 예, 경우 또는 사례의 역할을 하는"이라는 의미로 사용된다. "예시적"이라고 본 명세서에 기술된 여하한 실시예 또는 특징은, 반드시 다른 실시예들 또는 특징들 보다 바람직하거나 유리한 것으로 해석되어야 하는 것은 아니다.

[0022] 본 명세서에 개시된 것은, 호스트 시스템에 대한 업데이트 필요성을 최소화하거나 제거하면서 현존 시스템들의 유용성을 개선하고 그리고/또는 사용자에게 복잡성을 가중시키지 않고서 새로운 특징들을 제공하는 주변 장치용 신규한 솔루션들(solutions)이다.

[0023] 이러한 솔루션들은, 주변 장치들, 예컨대, USB 대용량 저장 프로토콜과 같은 데이터 저장을 위해 광범위하게 사용되는 프로토콜들을 이용하는 USB 주변 장치들을 위한 새로운 방법들 및 아키텍처들로 구성되고, 추가적인 호스트 소프트웨어 또는 다른 USB 통신 프로토콜들을 필요로 하지 않고서 추가적인 기능성을 추가한다. 이러한 추가적인 기능성은, 추가적인 통신 인터페이스들, 보안, 서버, 동기화 및 인터넷과 같은 더 큰 네트워크들에 연결하는 다른 서비스들을 포함할 수 있으나 이에 한정되지 않는다.

[0024] 본 명세서에 기술된 예시적인 아키텍처들과 방법들이 USB를 사용하지만, 이러한 방법들과 아키텍처들은 USB에 한정되지 않으며, 그 대신에 다른 버스들(buses)과 통신 프로토콜들이, 예를 들어, eSATA, iSCSI, 또는 그 동등물이 사용될 수 있다.

[0025] 추가적인 장점은, 본 명세서에 개시된 주변 장치 아키텍처들 및 방법들을 이용함으로써 가능해진 새로운 서비스들을 제공하는 것이다. 예를 들어, 사용자에게 복잡성을 가중시키지 않고서 사용자들을 위한 서비스들을 제공하기 위해 더 넓은 광역 네트워크들을 통합한 신규한 시스템 아키텍처들이 또한 개시되어 있다.

### [0026] I. 주변 장치 아키텍처

[0027] 도 5는, 블록 액세스 장치(52)와 하나 이상의 그래프 액세스 장치들(56, 58, 60)에 의한 메모리(66)에 대한 동시 액세스를 가능케 하는 다기능 주변 장치(54)를 포함하는 예시적 시스템(50)의 특정 구성요소들을 도시한 블록 다이어그램이다. 시스템(50)은 블록 액세스 장치(52), 주변 장치(54) 및 그래프 액세스 장치들(56, 58, 60)을 포함한다. 블록 액세스 장치(52)는 유선(wireline 또는 wired) 연결을 통해 주변 장치(54)와 통신하고, 그래프 액세스 장치들(56-60)은 하나 이상의 무선 링크들을 통해 주변 장치(54)와 통신한다.

[0028] 각각의 그래프 액세스 장치(56-60)는, 여하한 전자 장치, 예를 들어, 계층적 파일 시스템 구조와 같은, 그래프 프로 구성된 데이터에 액세스하는, 컴퓨터, PDA(personal digital assistant), 셀룰러 폰 비디오 게임 콘솔, 또는 그 동등물일 수 있으며, 그 그래프는 그래프 데이터 구조를 사용하여 메모리(66) 내의 특정 어드레스들로 분해된다. 그래프 데이터 구조는 그래프(들)과 메모리 풀(들) 사이의 한 세트의 맵핑들을 정의한다. 메모

리 풀은 하나의 단일, 논리 메모리 어레이로서 액세스된 하나 이상의 메모리 어레이들이다.

- [0029] 더욱 구체적으로, 그래프 데이터 구조는 메모리(66)를 계층적 파일 시스템에 맵핑하는 복수의 포인터들을 포함한다. 본 명세서에 기술된 주변 장치들(54, 70, 100)과 함께 사용될 수 있는 그래프 데이터 구조들의 예들은, 파일 저장 데이터베이스, 예컨대, 파일 배치표(FAT) 또는 그 변형들(vFAT, FAT32, FAT16, exFAT, NTFS, NFS 및 그 동등물)을 포함할 수 있다.
- [0030] 메모리(66)는 하나의 메모리 풀이거나, 다른 물리적 메모리들 및/또는 장치들을 포함하는 더 큰 메모리 풀의 일부일 수 있다. 데이터 구조 및 데이터 콘텐츠는 동일한 메모리 풀에 존재할 수 있는데, 예를 들어, 포인터들을 포함하는 파일 배치표 (FAT) 파일 시스템과 데이터가 공통의 메모리 풀에 함께 저장된다. 그와 달리, 그래프 데이터 구조가 별개의 메모리에 저장될 수 있다. 경우에 따라, 그래프 액세스 장치(56-60)는 Wi-Fi(와이-파이) 클라이언트이다. 다른 경우에, 그래프 액세스 장치(56-60)는 USB 호스트일 수 있다.
- [0031] 도 6은, 그래프 데이터 구조(152)와 데이터 콘텐츠(154)를 공유 메모리(66)에 함께 저장한 메모리(66)의 예시적인 부분(150)을 도시한 상위-레벨(high-level) 컨셉 다이어그램이다. 도시된 예에서, 그래프 데이터 구조(152)는 메모리 위치들(0-m)에 존재하고, 데이터(154)는 어드레스들(m+1 내지 n)에 위치한 메모리 블록들에 존재한다. 데이터(154)와 그래프 데이터 구조(152)는, 이웃한 메모리 위치들에 있는 것으로 도시되어 있지만, 이웃한 메모리 블록들에 위치될 필요가 없으며, 메모리(66) 또는 다른 메모리들에 분산될 수 있다.
- [0032] 도 5로 되돌아가서, 블록 액세스 장치(52)는 여하한 전자 장치, 예를 들어, 컴퓨터, PDA, 셀룰러 폰, 비디오 게임 콘솔, 또는 그래프 데이터 구조를 사용하지 않고, 하위-레벨 어드레싱 스킴(low-level addressing scheme), 즉, 이산 시작 위치 어드레스들과 끝 위치 어드레스들을 사용하여 메모리(66)에 액세스하는 그 동등물일 수 있다. 경우에 따라, 블록 액세스 장치(52)는 USB 대용량 저장 프로토콜을 주변 장치(54)에 액세스하도록 작동시키는 USB 호스트이다.
- [0033] 주변 장치(54)는, 프로세서(62), 유선 인터페이스(64), 메모리(66) 및 무선 인터페이스(68)를 포함한다. 프로세서(62)는 여하한 적합한 디지털 프로세서, 예컨대, 소프트웨어/펌웨어 명령어들을 실행하는 마이크로프로세서 또는 DSP, 주문형 집적 회로(application specific integrated circuit: ASIC), 하드웨어에서 실행된 상태 머신, 또는 본 명세서에 개시된 기능들을 수행하기 위한 그 동등물일 수 있다. 메모리(66)는 SD(secure digital) 카드, SDHC 카드, SDXC 카드, 하드 드라이브, 또는 솔리드 스테이트 메모리와 같은, 주변 장치(54)와 탈착가능 메모리를 포함하는, 여하한 적합한 메모리 장치일 수 있다. 블록 액세스 장치(52)와 주변 장치(54) 사이의 유선 통신은 USB인 것이 바람직하고, 무선 통신은 Wi-Fi일 수 있다. 따라서, 유선 인터페이스(64)는 상업적으로 구입가능한 USB 인터페이스 모듈일 수 있으며, 무선 인터페이스(68)는 상업적으로 구입가능한 Wi-Fi 인터페이스 모듈일 수 있다.
- [0034] 주변 장치(54)는 하나 보다 많은 기능(function)을 포함하는데, 이러한 기능들 중의 하나는 블록 액세스 장치(52) 또는 그래프 액세스 장치들(56-60) 중의 어느 하나가 액세스 가능한 메모리(66) 내의 데이터 저장 어레이일 수 있다. 도시된 예에서, 다른 기능은 무선 통신 인터페이스(68)이다. 데이터 저장 어레이는 주변 장치(54) 상의 공유 메모리로 취급된다. 그것은 주변 장치(54)에 의해 지원받는 기능들 및/또는 외부 장치들 사이에 공유된다. 프로세서(62)는 각 기능을 메모리(66)에 맵핑하고, 블록 액세스 및 그래프 액세스 장치들(52, 56-60)에 의한 메모리(66)에 대한 액세스들을 제어한다. 공유 메모리(66)를 사용하면, 주변 장치(54), 예컨대, 무선 인터페이스(68) 및 데이터 저장 어레이에 의해 제공된 기능들은 외부 블록 액세스 장치(52) 및 그래프 액세스 장치들(56-60) 사이에 데이터를 공유하거나 통신하기 위해 서로 직접 통신할 필요가 없다. 대신에, 기능들은 메모리(66)를 통해 서로 간접적으로 통신한다.
- [0035] 프로세서(62)는 블록 액세스 장치(52) 및 그래프 액세스 장치들(56-60)에 의한 메모리(66)에 대한 동시 액세스를 허용하도록 구성된다. 이것은 메모리(66)에 액세스하면서 블록 액세스 장치(52)에 의해 수행된 하나 이상의 메모리 블록 액세스 연산들을 모니터링하는 프로세서(62)에 의해 달성된다. 아래에 더 상세히 설명될 바와 같이, 프로세서(62)는 블록 액세스 연산들을, 메모리(66)를 계층적 파일 시스템에 맵핑하는 복수의 포인터들을 포함하는 그래프 데이터 구조로 변환시킨다. 이것은 어떤 그래프 액세스 장치(56-66)이든 메모리(66)에 블록 액세스 장치(52)와 동시에 액세스하게 한다. 블록 액세스 장치가 또한 메모리(66)에 액세스하는 동안, 하나 이상의 그래프 액세스 장치들(56-66)이 그래프 데이터 구조에 따라 메모리(66)에 액세스할 수 있다.
- [0036] 본 명세서에 사용된, "동시(적)" 및 "동시에"라는 용어는, 그들의 일반적인 의미들을 가질 수 있으나, "일견하여(apparently) 동시적인" 및 "일견하여 동시에"라는 의미, 즉, 주변 장치(54, 70 또는 100) 내에서 실제의 메모리 액세스들이 순차적으로 일어날 수 있지만, 블록 액세스 장치(52) 및 적어도 하나의 그래프 액세스 장



치들(56-60)에서의 프로세스들 그리고/또는 사용자들에게는 메모리 액세스들이 동시에 일어나는 것으로 보인다. 이는 의미를 또한 가진다. 주변 장치 메모리에 대한 동시 액세스를 제공하기 위한 두 가지 실제적인 방법들은, 동시적인 것처럼 보이는 속도의 다중화 액세스이거나, 주변 장치 내에 멀티-포트 메모리를 포함함에 의한 진정한 동시 액세스일 수 있다.

[0037] 블록 액세스 장치(52)가 도 5에 유선 인터페이스(64)를 통해 주변 장치에 연결된 것으로 도시되어 있지만, 그와 달리 블록 액세스 장치가 무선 연결 또는 무선 및 유선 연결들의 결합을 통해 연결될 수 있다. 마찬가지로, 그래프 액세스 장치들(56-60)도 무선 링크, 또는 무선 및 유선 연결들의 결합 대신에, 그와 달리 유선 연결을 사용하여 주변 장치(54)에 연결될 수 있다. 또한, 수행 중인 블록 액세스가 없고 그래프 액세스만이 존재하는 상황들이 있다.

[0038] 이 주변 장치(54)는, 도 7에 도시된 바와 같은, 더 일반적인 주변 장치 아키텍처로 확장될 수 있다. 도 7은 둘 보다 많은 통신 인터페이스들(72, 74, 76, 78)을 포함하는 대안적인, 예시적 다기능 주변 장치(70)의 특정 구성요소들을 도시한 블록 다이어그램이다. 주변 장치(70)는 또한 프로세서(62), 메모리(66), 그리고 배터리, 예컨대, 재충전가능 배터리 또는 다른 적합한 전력 저장 요소(power storage component), 예컨대, 슈퍼-캐패시터(super-capacitor)일 수 있는 전원(power source)(80)을 포함한다. 각각의 통신 인터페이스들(72-78)은 본 명세서에 개시된 여하한 것들과 같은 유선 또는 무선 인터페이스일 수 있다. 각각의 통신 인터페이스들(72-78)은 주변 장치(70) 상의 상이한 기능이고, 프로세서(62)는 인터페이스들(72-78)을 공유 메모리(66)에 맵핑한다. 통신 인터페이스들(72-78)은 프로세서(62)에 의해 관리되는 공유 메모리(66)를 통해 서로 통신할 수 있고 그리고/또는 데이터를 전할 수 있다. 블록 액세스 장치들 또는 그래프 액세스 장치들은 하나 이상의 통신 인터페이스들(72-78)을 통해 주변 장치(70)에 연결될 수 있고, 본 명세서에 개시된 방법들을 사용하여 메모리(66)에 동시에 액세스할 수 있다.

[0039] 도 8은, 도 5에 도시된 다기능 주변 장치(54)의 구체적인, 예시적 구현의 특정 구성요소들을 도시한 블록 다이어그램이다. 도 8의 주변 장치(100)는, 중앙 처리 장치(central processing unit: CPU)(102) 또는 프로세서, 메모리(104), USB 인터페이스(106), 무선 인터페이스(108), 전원 관리 모듈(power management module)(110), 전원(112), 하나 이상의 센서들(113), RAM(random access memory)(114), ROM(read only memory)(116) 및 사용자 인터페이스(118)를 포함한다. 사용자 인터페이스(118)는 스위치(120) 및 시각적 표시기, 예컨대, LED(112)를 포함한다.

[0040] CPU(102)는 여하한 적합한 디지털 프로세서, 예컨대, 소프트웨어/펌웨어 명령어들을 실행하는 마이크로프로세서 또는 DSP, 주문형 집적 회로(ASIC), 하드웨어에서 실행되는 상태 머신, 또는 본 명세서에 개시된 기능들을 실행하기 위한 그 동등물일 수 있다. CPU(102)는 프로그래밍 명령어들 및 데이터를 저장하기 위한 RAM(114) 및 ROM(116)과 같은 온-보드 메모리(on-board memory)를 포함하는, 기성(off-the-shelf) 마이크로프로세서인 것이 바람직하다.

[0041] ROM(116)은 주변 장치를 위한 부트(boot) 또는 스타트-업 코드(start-up code)를 저장할 수 있다. CPU(102)는 시동하고 그것의 코드를 작동시키기 위해 ROM(116)을 사용한다. 이러한 스타트-업 또는 부트, 코드를 위한 개별적인 ROM(116)에 대한 대안으로서, 부트 코드를 위한 메모리(104)의 일부를 사용하는 것이 또한 가능하다. RAM(114)은 CPU(102)에 의해 사용된 프로그래밍 명령어들과 데이터를 저장할 수 있다.

[0042] 메모리(104)는 여하한 적합한 메모리 장치일 수 있으며, 바람직하게는 탈착가능 SD 카드이다. SD 카드와 함께, 주변 장치(100)는 카드를 수용하기 위한 SD 포트를 포함할 수 있다. 메모리(104)는 메모리 풀이거나, 다른 물리적 메모리들 및/또는 장치들을 포함하는 더 큰 메모리 풀의 일부일 수 있다. 다른 경우에, 메모리는, 플래시 메모리와 같은, 탈착가능하지 않은 내장형 솔리드-스테이트 메모리일 수 있다.

[0043] USB 인터페이스(106)는 CPU(102)에 인터페이스 연결하는 디지털 버스를 구비한 상업적으로 구입가능한 Wi-Fi 인터페이스 모듈인 것이 바람직하며, 무선 인터페이스(108)는 CPU(102)에 인터페이스 연결하는 디지털 버스를 구비한 상업적으로 구입가능한 Wi-Fi 인터페이스 모듈인 것이 바람직하다.

[0044] 주변 장치 무선 인터페이스(108)는 표준 웹 프로토콜들(standard web protocols)을 통해 제공되는 Wi-Fi (802.11) 인터페이스일 수 있다. 주변 장치(100)는 IP 스택 및 HTTP 서버를 포함한다. 그것은 또한 그것의 존재를 무선 네트워크 상의 다른 장치들에게 알리기 위해 링크-로컬 멀티캐스트 DNS("ZeroConf" 또는 "Bonjour"로도 알려져 있음)를 사용한다. 주변 장치(100)에 대한 액세스는 Wi-Fi 암호화 (WEP 또는 WPA)의 사용을 통해, MAC 어드레스 필터링을 통해, 그리고/또는 응용 계층 보안(application layer security) (SSL 및 HTTP 인증)을 통해 제어될 수 있다.

- [0045] 주변 장치(100)는 연결된 복수의 무선 클라이언트들 사이의 파일들에 대한 액세스를 조정할(coordinate) 수 있다. 예를 들어, 복수의 연결된 클라이언트들 상의 매체(비디오 또는 오디오)에 액세스할 때, 주변 장치(100)는 모든 클라이언트들로 하여금 재생을 위해 동기화된 시작(synchronized start)을 요청하게 할 수 있다. 이러한 조정은 주변 장치(100)로부터 클라이언트들에게 보내진 프로그램을 통해 수행될 수 있고, 클라이언트(예를 들어, 리치 AJAX 클라이언트 어플리케이션) 상에서, 또는 표준 매체 스트리밍 프로토콜들을 통해 이루어질 수 있다.
- [0046] 전원(112)은 배터리, 예컨대, 재충전가능 배터리 또는 다른 적합한 전력 저장 요소, 예컨대, 주변 장치(100)를 USB 포트에 꽂을 때 재충전될 수 있는, 슈퍼-캐패시터일 수 있다. 주변 장치(100)에 추가된 전원(112)으로, 주변 장치(100)는 호스트, 예컨대, 호스트 USB 포트에서 주변 장치(100)를 뽑은 후에 기능할 수 있다. 이것이 일어날 때, 전원(112)은 주변 장치에 전력을 공급할 수 있다. 따라서, 주변 장치(100)는 USB 호스트에 연결된 동안이든 USB 호스트에 연결되지 않을 때든 독립형 장치로서 기능할 수 있다. 또한, 호스트가 연결되어 있더라도, 주변 장치(112)를 위한 충분한 전력을 기능에 제공할 수 없을 때, 전원(112)이 주변 장치(100)에 전력을 공급할 수 있다.
- [0047] 전원 관리 모듈(110)은 일반적으로 CPU(102)를 통해, CPU(102), USB 인터페이스(102), 전원(112) 및 사용자 인터페이스(118)에 인터페이스 연결한다. 모듈(110)은 사용자 인터페이스(118)의 LED(122)에 의해 표시되는 주변 장치 전력 상태 표시기들(재충전 상태, 낮은 전력 상태 등), 슬립 모드들(sleep modes) 및 USB 인터페이스(106)를 통한 재충전을 포함하는, 전원(112)을 관리하기 위한 기성 소프트웨어 및 하드웨어 구성요소들을 포함한다.
- [0048] 센서들(113)은 CPU(102)에 결합된 적합한 수의 또는 유형의 컨텍스트 센서들(context sensors)을 포함할 수 있다. 예를 들어, 센서들(113)은 실시간 시계, 상업적으로 구입가능한 GPS 로케이터 모듈, 가속도계, 생체인식 센서, 온도 센서, 또는 그 동등물을 포함할 수 있다. 센서들(113)의 출력은 블록 액세스 장치들 및/또는 그래프 액세스 장치들(예를 들어, 호스트 및/또는 클라이언트들)에게 메모리(104)에 대한 액세스를 승인하거나 거절하기 위한, 본 명세서에 기술된, 방법들과 함께 CPU(102)에 의해 사용될 수 있다.
- [0049] USB 호스트(예를 들어, 블록 액세스 장치(52))는, 주변 장치(100)를 사용하여, USB 인터페이스(106)와 통신한다. 무선 인터페이스(108)의 무선 통신 기능은 메모리(104)에 맵핑되고, USB 호스트와 독립적으로 기능한다. CPU(102)는 메모리(104)에 USB 호스트의 데이터 저장 기능 및 무선 통신 기능을 적절하게 맵핑한다. 더 정교한 무선 서버 및/또는 관리 기능들을 제공하는 것과 같은 추가적인 기능들이 주변 장치(100)에서 메모리 맵핑될 수 있다. USB 대용량 저장 프로토콜을 사용하여 주변 장치(100)를 액세스하는 USB 호스트에 의해, 메모리 맵핑은 USB 호스트가, USB 주변 장치들을 사용하여 종래의 방식으로 이루어지는 것처럼, 주변 장치(100)를 논리적으로 별개의 무선 장치 및 저장 장치로서가 아니라 단일 USB 저장 장치로 여기게 한다. 주변 장치의 추가된 기능성은 따라서 호스트에 대해 트랜스페어런트할 수 있다. USB 호스트는 무선 기능성을 위한 추가적인 드라이버들을 필요로 하지 않는다. 그것이 필요로 하는 모든 것은 메모리(104) 상의 데이터 저장 어레이와의 통신이다. 이것은 USB 호스트들이 USB 저장 프로토콜들을 이용하여 호스트에 대한 추가적인 소프트웨어 요구 (software requirements) 없이 추가된 기능성을 수용하는 것을 가능케 한다.
- [0050] CPU(102)는 메모리(104)의 특정 부분들을 다양한 기능들 및 대용량 저장에 내부적으로 할당함으로써 메모리를 맵핑할 수 있다. 이러한 할당(allocations)은 CPU(102)에 의해 관리되며, 무선 인터페이스(108)를 통해 주변 장치(100)와 통신하는 호스트 및 다른 장치들에 대해 트랜스페어런트할 수 있다.
- [0051] 메모리 맵핑은 주변 장치(100)에 위치한 물리적 저장부, 또는 여하한 세분화단위의 레벨(granularity level)의 여하한 가상적 및 물리적 조합 저장부를 참조할 수 있으며, 그 조합 저장부는 예를 들어, 각 폴더 또는 파일이 주변 장치 이외의 어딘가에 존재하고, 또는 동일한 파일의 부분들이 상이한 메모리들에 존재하고, 또는 메모리에 대한 액세스가 통신 인터페이스들(106, 108)에 대한 기능 변화를 가져오는 것을 말한다.
- [0052] CPU(102)는 또한 본 명세서에 개시된 방법들을 사용하여 기능들, 블록 액세스 장치들 및 그래프 액세스 장치들에 의한 메모리(104)에 대한 동시 액세스를 허용하도록 구성될 수 있다.
- [0053] 다양한 기능들을 공용 (또는 복합 공용) 메모리 공간에 맵핑하는 주변 장치 아키텍처는, 주변 장치(100)로 하여금, USB 호스트에 대한 하나의 단일 논리 장치로서 나타나는, 무선 네트워크 인터페이스 및 데이터 저장 기능을 뛰어넘는, 다양한 기능들을 포함하게 할 수 있다. 예를 들어, 주변 장치(100)가 무선 인터페이스(108), 메모리(104)의 대용량 데이터 저장 및 다른 기능들을 포함할 수 있지만, USB 호스트는 주변 장치 기능들을 하나의 단일 파일 시스템으로 볼 수 있다. 따라서, USB 호스트는 파일들과 상호작용하는 것처럼 주변 장치(100)

상의 모든 기능들을 처리할 수 있다. 따라서 호스트는 소프트웨어 응용 계층 레벨(software application layer level)에서 주변 장치 기능들과 인터페이스 연결될 수 있다. 이것은 USB 호스트로 하여금 주변 장치(100)를 하나의 기능으로서 대우하게 하며, 호스트와 독립적으로, 주변 장치(100)에 의한 기능들에 대한 맵핑이 있다(USB 호스트에 의해 셋팅된 그 제2 기능을 제어하기 위해 메모리(104)에 저장된 직접적 또는 내재적 셋팅들이 있을 수 있기는 하다).

- [0054] 주변 장치(100)의 무선 인터페이스들(106, 108)이든 USB든 연결된 장치/호스트에 대해 하드웨어가 제공하는 다이렉트 인터페이스(direct interface)가 아니며; 그들은 결국 주변 장치 소프트웨어에 의해 제어되며, 따라서 어느 하나의 매체를 통해서 저장 및/또는 기능들이 어떻게 제공되는지를 그 소프트웨어가 결정하게 한다. 특히, 이것은 그 소프트웨어로 하여금:
- [0055] 1. (무선 인터페이스 상의 내부 ROM 및 SD 저장부로부터의 데이터의 결합과 같은) 저장의 하나의 논리 풀(logical pool of storage)을 만들기 위해 데이터의 다양한 소스들을 결합시키게 하고;
- [0056] 2. 인증 또는 다른 팩터들(factors)에 의해 데이터에 대한 액세스를 제한하거나 차단하게 하고; 그리고/또는;
- [0057] 3. 암호해독 전에 호스트 또는 클라이언트로부터의 키(key)를 필요로 하는, SD 카드와 같은, 메모리(104) 상의 데이터를 암호화하게 하고; 그리고/또는;
- [0058] 4. USB 또는 무선 인터페이스(106, 108)의 어느 하나를 통한 저장의 논리 풀에 대한 독립 데이터 전송을 가지는 추가적인 기능성을 제공하게 한다.
- [0059] 도 8의 예에서, Wi-Fi 트랜잭션들(transactions)이, 대용량 저장 장치로서 주변 장치(100)와 상호작용하는 USB 호스트와 독립적으로, 무선 인터페이스(108)를 통해 주변 장치(100)에 연결된 외부 서버와 주변 장치(100) 사이에 일어날 수 있으며, 예를 들어, 공유 메모리(104)에 파일들로서 저장된 것은 무엇이든지 Wi-Fi 인터페이스를 통해 외부 서버 또는 하나 이상의 다른 유사한 장치들과 CPU(102)에 의해 자동적으로 동기화될 수 있다.
- [0060] 주변 장치(100)는 또한 USB 호스트로 하여금 데이터 저장 및/또는 USB 데이터 저장 프로토콜들을 이용함으로써 제2 기능들의 양상들을 제어하게 하는 다양한 제어 방법들을 제공할 수 있다. 예를 들어, 전형적인 USB 대용량 저장 프로토콜들과 USB 호스트 소프트웨어는 파일 배치표(FAT) 또는 그 변형들(vFAT, FAT32, FAT16, exFAT, NTFS, NFS 및 그 동등물)과 같은 잘 알려진 파일 저장 데이터베이스를 사용한다. 제2 기능이, 예를 들어, Wi-Fi 무선 통신인 경우에, 데이터 드라이브의 볼륨 이름(volume name)은 호스트에서 Wi-Fi를 위한 SSID 이름에 직접 맵핑할 수 있다. 그러면 USB 호스트는 잘 알려진 볼륨 이름 변경 기능들(volume renaming functions)을 사용하여 SSID의 이름을 제어할 수 있다. 그와 달리, 각기 특수화된 파일 폴더 이름에 상응하는 다양한 SSID들이 존재할 수 있다. 이것은 사용자로 하여금 Wi-Fi 네트워크를 단순한 메뉴로서 선택할 때 다양한 SSID들을 검색하게 하고 적절한 그리고 허용된 액세스를 선택하게 한다. 다양한 사용자들/서버들이 아래에 더 상세히 논의된 바와 같이, SSID들을 통해 다양한 폴더들에 대해 상이한 보안 레벨들로 액세스할 수 있다.
- [0061] CPU(102)는 또한 트랜스코딩(transcoding) 및/또는 암호화/암호해독과 같은 추가적인 기능성을 제공하도록 구성될 수 있다. 트랜스코딩을 위해, CPU(102)는 예를 들어, 명령(command)을 내리는 호스트와 독립적으로, 비디오, 오디오, 또는 문서 파일과 같은 콘텐츠를 하나의 데이터 포맷으로부터 다른 데이터 포맷으로 자동으로 변환시킬 수 있다. CPU(102)는 또한 파일이 전송 중이든 전송된 후든, 메모리(104)의 저장부에 들어가는 데이터를 암호화하고 그리고/또는 암호해독할 수 있다.
- [0062] 주변 장치(100)의 기본 기능은 무선 및 유선 인터페이스들(106, 108)을 통한 저장부에 대한 액세스를 제공하는 것이다. 주변 장치(100)의 특정 구성은 USB 2.0 또는 SD/SDHC 저장부(메모리(104))에 대한 무선 액세스를 제공하기 위한 이러한 특징부들의 기본 구현을 포함한다. 암호화된 펌웨어 다운로드를 SD/SDHC 저장부(메모리(104))에 직접적으로 위치될 수 있으며, 그러면 이것은 내장된 ROM(116)을 업그레이드하기 위해 사용되거나, 그와 달리, 내장된 ROM(116)에 직접 다운로드될 수 있다. 다운로드는 USB 인터페이스 또는 Wi-Fi 인터페이스를 통해 일어날 수 있다. 복합 장치의 다른 구성요소는 표준 USB 2.0 SD 카드 리더 장치로서 제공된다.
- [0063] 주변 장치(100)는 USB를 통해 연결된 동안에, 호스트로부터 전력을 얻음으로써 그것의 내부 전원(112)을 충전한다. 주변 장치(100)는, USB를 통해 연결되지 않을 때, 주변 장치(100) 상의 푸쉬-버튼일 수 있는, 스위치(120)에 의해 전원이 켜지거나 꺼질 수 있다. 전원이 켜질 때, 주변 장치(100)는 [애드-호크(ad-hoc)/피어 투 피어(peer-to-peer) 모드, 인프라스트럭처(infrastructure) 모드, 및/또는 Wi-Fi 다이렉트(Wi-Fi Direct)와 같은 다양한 모드들을 사용하여] 802.11 무선 네트워크를 브로드캐스팅할(broadcast) 수 있다. 무선 네트워크



의 이름은 삽입된 SD 카드의 이름에 의해 결정될 수 있고[카드가 삽입되지 않으면, 디폴트 이름(default name)이 선택됨], 암호화 셋팅들은 카드 상의 파일에 의해 결정될 수 있다(카드가 삽입되지 않으면, 암호화되지 않음).

[0064] 무선 파일 액세스는 주변 장치(100)에 포함된 표준 HTTP 서버에 의해 제공될 수 있다. 주변 장치(100)는 그 자체에 사설, 링크-로컬 IP 어드레스를 할당하고, 연결된 클라이언트들로부터의 발견가능성(discoverability)을 지원하기 위해 링크-로컬 멀티캐스트 DNS 메시지들(link-local multicast DNS messages)에 응답한다. HTTP 서버는 연결된 iOS 장치들에게, 예를 들어, 주변 장치로부터의 파일들에 액세스하기 위해 사용되는, 아이폰들(iPhones)/아이패드들(iPads)에게 동적 AJAX 어플리케이션(dynamic AJAX application)을 보낼 수 있다. 다른 클라이언트들은 기본 파일 리스트를 수신한다.

[0065] 메모리(104)가 SD 또는 SDHC(secure digital high capacity) 카드인 주변 장치(100)의 구성에서, 주변 장치(100)는 전원이 켜질 때, SD/SDHC 카드의 삽입 또는 제거에 대해 스캔한다(scan). 카드가 주변 장치(100)에 삽입되거나 그로부터 제거될 때, 일련의 동작들이 그에 답하여 취해질 수 있다. 카드가 삽입될 때, 주변 장치(100)가 USB를 통해 연결되면, 그것은 호스트에게 카드가 삽입되었음을 통지할 수 있고, 이것은 호스트로 하여금 카드의 콘텐츠를 검토하고 변경할 수 있게 한다. 카드가 제거될 때, 주변 장치(100)는 마찬가지로 호스트에게 통지한다. 저장부로부터의 정보는 그 다음에 무선 액세스를 구성하기 위해 사용될 수 있다. 예를 들어, 카드의 볼륨 이름은 무선 인터페이스의 SSID를 설정하기 위해 사용될 수 있고, 또는 카드 상의 파일들은 암호화 키를 결정하기 위해 사용될 수 있다. 카드의 삽입 및 제거는 카드 상의 구성 데이터의 존재에 따라 무선 인터페이스(108)에 상태 변화를 일으킨다. 카드 상의 특정 파일들은 주변 장치(100)로 하여금 모든 카드들에, 또는 다른 결정적인 구성 정보를 가지지 않는 카드들에 적용하는 디폴트 파라미터들을 설정하게 한다.

[0066] 그와 다른 구성들에서, 직렬 ATA(SATA) 또는 SD(secure digital) 카드 인터페이스와 같은 유선 저장 인터페이스를 USB 인터페이스(106) 대신 사용하는 것이 가능하다. 또한, 3G/4G 셀룰러 모뎀, 블루투스, NFC, 적외선(infrared), 지그비 변종(ZigBee variant), 또는 그 동등물과 같은, Wi-Fi에 대한 대안적인 무선 인터페이스가 무선 인터페이스를 위해 사용될 수 있다. 다양한 무선 인터페이스들 (예를 들어, Wi-Fi 802.11에 더하여 블루투스)을 지원하는 것이 가능하다. 인터페이스들은 상이한 프로토콜들을 통해 동일한 서비스들을 제공할 수 있거나, 특유한 방식으로 결합될 수 있다. 블루투스 페어링 및 프록시미티(Bluetooth pairing and proximity)가 암호화된 저장부에 대한 액세스를 잠금해제하거나(unlock) 무선 클라이언트들을 인증하기 위한 키로서 사용될 수 있다. 주변 장치(100)는 블루투스(DUN)를 통해 인터넷 연결을 획득할 수 있고, 이것을 파일 서비스들을 위해 사용할 수 있다. 만약 그렇게 되면, 그것은 또한 Wi-Fi 또는 USB를 통해 다른 장치들과의 그 연결을 공유할 수 있다.

[0067] 그와 다른 구성들에서, 메모리(104)는, 하나 이상의 무선 또는 유선 인터페이스(106, 108)를 통해 통신하는 네트워크 서버와 같은, 주변 장치(100) 이외의 하나 이상의 장치들에 위치될 수 있다. 이러한 구성들에서, CPU(102)에 의해 수행된 메모리 맵핑은, 주변 장치(100) 상에 존재하는 물리적 메모리 저장 어레이가 존재하지 않는, 가상 메모리를 참조하고, 그러한 가상 메모리로의/가상 메모리로부터의 전송들은 주변 장치(100) 및 원격 저장 서버/유닛 사이의 통신들을 필요로 한다. 이 경우에, 주변 장치(102)는, 물리적 메모리 저장 어레이가 실제로 존재하지 않을 때, 주변 장치(102) 상에 물리적으로 위치된 것처럼, 다른 무선 또는 유선 인터페이스(106, 108)를 통해 다른 호스트/클라이언트들에게 데이터를 제공하도록 구성된다. 이러한 구성들에서, 호스트가 원격 메모리에 액세스 중일 때, 네트워크 지연시간 및 속도(latency and speed)와 같은 성능을 향상시키기 적합한 경우에, CPU(102)가 원격 저장 콘텐츠의 일부를 로컬 메모리 저장부(예컨대, RAM(114))에 캐시하는(cache) 것이 또한 가능하다. 이 경우에, 캐시된(cached) 데이터는 호스트에 대해 트랜스페어런트하고, 즉, 호스트가 로컬 메모리 저장 성능 편의를 획득하지만, 여전히 캐시된 데이터를 주변 장치(100) 상의 저장 어레이에 저장된 것으로 본다.

[0068] 도 9는 다기능 주변 장치(100)의 예시적인 소프트웨어 아키텍처(250)를 도시한 다이어그램이다. 주변 장치(100)의 소프트웨어 아키텍처(250)는 많은 계층적 구성요소들(layered components)로 구성된다.

[0069] 소프트웨어의 최하위 계층은 주변 장치 하드웨어에 대한 인터페이스 연결(interfacing)을 처리하고, USB 장치, SD 카드 인터페이스, Wi-Fi SDIO 인터페이스, LED 및 스위치들을 포함한 사용자 인터페이스, 클럭킹(clocking) 및 전원 제어 및 배터리 충전을 제어한다. 소프트웨어의 최하위 계층은, USB 하드웨어 버스 인터페이스(276), SD 하드웨어 버스 인터페이스(278), SD 메모리 드라이버들(270), SDIO(SD input/output)(272), 타이머 서비스들(274), 예를 들어, 온도 센서, 실시간 시계, GPS 위치 모듈 및 그 동등물에 대한 소프트웨어 인터페이스들을 포함하는, 센서 인터페이스(들)(280), 사용자 인터페이스(282), 타이머 서비스들(274), 배터리

리 관리(284) 및 Wi-Fi 드라이버들(268)을 포함한다. 이 계층은 주변 장치 소프트웨어의 나머지가 사용하는 하드웨어 추상화 계층(Hardware Abstraction Layer)을 형성한다.

[0070] 주변 장치 소프트웨어의 상위 계층들은 장치의 중요한 특징들을 정의한다. 이 계층들은: USB 대용량 저장 프로토콜 소프트웨어(254), 펌웨어 업데이트 소프트웨어(256), 그래프 액세스 서버들(예를 들어, mDNS, HTTP, FTP 또는 그 동등물)(258), Wi-Fi 구성 소프트웨어(260), TCP/IP 스택(262), 전원 관리 및 구성 소프트웨어(264), 그래프 데이터 구조, 예컨대, FAT(266) 및 컨텍스트 인식 상태 머신(context aware state machine)(252)을 포함한다.

[0071] 컨텍스트 인식 상태 머신(252)은 본 명세서에 개시된 방법들을 실행하기 위한 소프트웨어 코드를 포함할 수 있다. 상태 머신(252)은 또한 보안 및 인증, 암호화/암호해독, 트랜스코딩, 메모리 및 기능 액세스 제어, 통신 채널 제어 등과 같은, 본 명세서에 개시된 추가적인 서비스들을 제공하기 위한 소프트웨어 코드를 포함할 수 있다.

[0072] 펌웨어 업데이트 소프트웨어(256)는 SD 카드 인터페이스(278) 또는 USB 인터페이스(106)를 통해 주변 장치 소프트웨어의 인-시스템 업데이트(in-system update)를 위해 사용된다. USB 규격은 정적이지 않으며, 프로토콜에 대한 업데이트들과 확장들이 주기적으로 일어난다. 다른 업데이트들은, 무선 네트워킹을 위한 새로운 표준과 같은 새로운 소프트웨어 드라이버들로 호스트를 업데이트할 필요성을 제거함으로써 상호운용성을 향상시키기 위한 새로운 프로파일들을 포함할 수 있다.

[0073] USB 대용량 저장 프로토콜 소프트웨어(254)는 첨부 디스크로서 연결된 호스트에 USB 인터페이스(106)를 통해 메모리(104)의 콘텐츠 (그리고 아마도 다른 데이터)를 제공한다. 다른 경우에, 메모리(104)에 대한 액세스로 네트워크 주변 장치인 호스트에 연결되는 USB 네트워크 액세스와 같은 다른 USB 프로토콜들이 존재할 수 있다.

[0074] 전원 관리 및 구성 소프트웨어(264)는 배터리 충전 및 전력 절감(power saving) 모드들을 제어한다.

[0075] 소프트웨어는 주변 장치(100)를 Wi-Fi 액세스 포인트(AP: access point), 또는 그와 달리, Wi-Fi 클라이언트로서 기능하도록 구성할 수 있다. Wi-Fi AP로서 작동하는 경우에, 주변 장치(100)는 주변 장치(100)를 둘러싼 포터블 개인 영역 네트워크(personal area network: PAN)를 제공한다. 다른 경우에, 주변 장치(100)는 동시 AP 및 클라이언트 액세스를 사용하여 기능하도록 만들어질 수 있으며, 와이-파이 다이렉트 표준(Wi-Fi Direct standard)을 이용할 수 있다.

[0076] 몇몇 구성들에서, 주변 장치(100)는 그것의 Wi-Fi AP 모드로 작동할 때 기능성을 강화하기 위해 하나 이상의 Wi-Fi 클라이언트들에 "메이팅된(mated)" 수 있다. 주변 장치(100)는, 필요한 경우에, 각 클라이언트의 MAC 어드레스들과 특정 로그인 크리덴셜들(login credentials)을 저장함으로써 클라이언트들에 메이팅된다. 주변 장치(100)가 AP 모드에 있을 때, 그것은 다른 AP로 전송된, 메이팅된 클라이언트의 MAC 어드레스를 위한 Wi-Fi 채널을 스니핑한다(sniff). 메이팅된 클라이언트는, 다른 AP에 대한 연결을 시도할 때, 그것의 MAC 어드레스를 브로드캐스팅하게 된다. 주변 장치(100)가 브로드캐스팅된 클라이언트의 MAC 어드레스를 탐지하면, 그것은 클라이언트의 연결 요청에 답하여 브로드캐스팅된 다른 AP의 목적지 MAC 어드레스(destination MAC address)를 기다리고 탐지한다. 그 다음에 주변 장치(100)는, 다른 AP에 연결하는데 저장된 로그인 크리덴셜들이 필요한 경우에, 목적지 MAC 어드레스를 클라이언트의 해당 로그인 크리덴셜들과 비교한다. 그와 달리, 주변 장치(100)는 클라이언트의 연결 전송들로부터 로그인 크리덴셜들을 스니핑할 수 있다. 주변 장치(100)가 다른 AP의 범위 내에 있는 경우에, 그것은 Wi-Fi 클라이언트 모드로 스위칭할 수 있고, 메이팅된 장치의 크리덴셜들을 사용하여 Wi-Fi 클라이언트로서 다른 AP에 연결될 수 있다.

[0077] 다른 소프트웨어 아키텍처들이 동일한 결과를 달성하기 위해 실행될 수 있다. 또한, 주변 장치 소프트웨어/펌웨어를 실행하기 위한 공지된 소프트웨어 기술을 사용하여 축소, 치환, 추가 및/또는 재배열들이 이루어질 수 있다.

## [0078] II. 메모리 풀에 액세스하는 방법

[0079] 도 10은, 그래프 액세스 장치와 블록 액세스 장치를 공유 메모리 풀에 동시에 액세스시키기 위한 예시적인 방법을 도시한 플로우차트(300)이다. 이 방법은 본 명세서에 기술된 여하한 주변 장치들(54, 70, 100)에 의해 실행될 수 있다. 그래프 액세스는 그래프로 구성된 데이터에 액세스하는 것을 가리키며, 그것에 의해 서버 또는 프로세서가 상위 레벨 액세스 명령들을 수신하고, 그래프 데이터 구조와 같은 데이터 구조에 의해, 그래프를 메모리 풀 콘텐츠로 분해한다. 블록 액세스는 이산 시작 위치 어드레스 및 끝 위치 어드레스로 메모리 풀



에 액세스하는 것을 가리킨다.

- [0080] 단계(302)에서, 블록 액세스 장치에 의한 메모리 풀에 액세스하려는 시도가 탐지된다. 주변 장치가 메모리 풀에 액세스할 수 있는 퍼미션(permission)을 승인하면, 주변 장치 프로세서가 메모리 풀에 액세스하면서 블록 액세스 장치에 의해 수행된 메모리 블록 액세스 연산들을 계속 모니터링한다[단계(304)]. 메모리 블록 액세스 연산들은 블록 읽기 또는 블록 쓰기 연산들을 포함할 수 있다. 프로세서는, 블록 액세스 중에, 연산들의 유형(읽기 또는 쓰기), 연산들의 수 및 연산들의 메모리 어드레스 위치들을 추적한다.
- [0081] 다음에는, 단계(306)에서, 블록 액세스 연산들이 메모리 풀을 나타내는 그래프 데이터 구조로 프로세서에 의해 변환된다. 그래프 데이터 구조는 메모리 풀을 계층적 파일 시스템에 맵핑하는 하나 이상의 포인터들을 포함한다. 단순한 그래프 데이터 구조들의 예들이 도 11-12 및 도 14-15에 나타나 있다. 분해 프로세스(resolution process), 즉, 복합 버퍼링된 업데이트들(multiple buffered updates)을 그래프 데이터 구조로 분해하는 것이 변환 단계(translation step)에 사용될 수 있다. 이 분야에 공지된 N-원 병합 알고리즘들(N-way merge algorithms)이 이러한 분해에 사용될 수 있다.
- [0082] 블록 액세스들 때문에 분해에 대한 필요성이 대두된다. 그래프 액세스가 메모리 풀에서 일어날 때마다, 그래프 데이터 구조는 언제나 잘 구성된 상태(well-formed state)로 남아있다. 그러나, 블록 액세스가 일어날 때, 그래프 데이터 구조는 일시적으로 잘못 구성된 상태(ill-formed state)로 남아 있을 수 있다. 주변 장치 프로세서는 미래의 어느 시점에 그래프를 분해할 것이다.
- [0083] 분해하는/분해 프로세스를 시작하는 때와 관련하여 다음과 같은 다양한 세분화단위(granularity)의 레벨들이 있다:
- [0084] 1. 최소 세분화(Least granular): 모든 액세스(블록 및 그래프, 읽기 & 쓰기 둘다)가 끝날 때까지, 분해를 지연시키는 것으로서, 즉, USB 호스트가 장치를 이탈시키고, 모든 Wi-Fi 클라이언트들이 인터페이스 연결이 끊길 때까지 분해를 지연시키고, 그 다음에 분해한다.
- [0085] 2. 블록 액세스가 끝날 때까지 분해를 지연시키고, 이어서 분해가 끝날 때까지 그래프 액세스를 차단하는 방식으로 그래프 데이터 구조를 업데이트하며, 그리고 나서 그래프 액세스를 다시 시작하는 것으로서, 예를 들어, 분해가 끝날 때까지 Wi-Fi 통신을 저지한다.
- [0086] 3. 어떤 파일들이 액세스되고 있는 지를 식별하고, 동시에 액세스되지 않고 있는 파일들을 위해 [개별적인 버퍼(buffer)를 사용하지 않고] 파일들을 즉시 분해한다.
- [0087] 4. 파일들의 액세스되고 있는 부분들을 식별하고, 그러한 부분들을 즉시 분해한다. 어떤 부분들이 액세스될 것인지 예측하고 액세스 가능성이 감소될 때까지 그러한 부분들의 분해를 피하는 것이 또한 개선 부분이다.
- [0088] 5. 블록 액세스가 그래프 데이터 구조를 일관된 상태로 남겨두는 때를 식별하고, 다음에 분해 프로세스를 적용한다.
- [0089] 장치 프로세서는, 그래프를 이전의 블록 연산들(block operations)과 일관되게 만들기 위해 주변 장치의 내부 그래프 데이터 구조를 업데이트함으로써 그리고/또는 블록 메모리를 그래프 액세스와 일관되게 만들기 위해 메모리 블록들을 이동(moving)/복사(copying) 함으로써, 메모리를 선택적으로 분해할 수 있다. 이러한 분해 연산들 중의 어느 하나 또는 둘 모두가, 일어났을 수도 있는 업데이트들을 분해하기 위해 프로세서 또는 호스트 장치에 의한 블록 메모리의 완전하지 않은 재스캔(rescan) 보다는, 일어날 수 있는 그래프(그래프 데이터 구조)에 대한 점진적 업데이트들이다. 이러한 점진적 분해는 그래프 액세스 장치 또는 블록 액세스 장치 중 어느 하나에 의한 특정 세트의 블록 액세스들이 완료된 후에 일어날 것이다.
- [0090] 경우에 따라, 블록 액세스 연산들의 변환은 어드레스 A의 요청된 메모리 블록을 다른 것으로, 예를 들어, 어드레스 B로 리다이렉팅하는(redirecting) 프로세서를 필요로 할 수 있다. 이러한 접근법은 주변 장치의 내부 그래프 데이터 구조가 업데이트되고, 블록 액세스 장치가 그 자신의 내부 그래프 데이터 구조를 아직 업데이트하지 않았을 때 사용된다. 추가적으로, 프로세서는 블록 액세스 연산들이 완료되기를 기다린 다음에 블록 및 그래프 구조를 서로 일관되게 유지하기 위해 블록들을 다시 정리하고(re-order) 그리고/또는 내부 그래프 데이터 구조의 파트들을 재구성할(restructure) 수 있다.
- [0091] 단계(308)에서, 주변 장치 프로세서는 그래프 데이터 구조를 그래프 액세스 장치에 제공한다. 이것은 업데이트된 그래프 데이터 구조를 그래프 액세스 장치가 액세스 가능한 메모리 영역에 저장함으로써 수행될 수 있다. 그래프 액세스 장치는 그 다음에 그래프 데이터 구조에 따라 블록 액세스 장치와 동시에 메모리 풀에

액세스할 수 있다.

- [0092] 도 11은 그래프 데이터 구조(352)를 블록 메모리 어레이(350)에 맵핑하는 하나의 예를 도시한 컨셉 다이어그램이다. 메모리 어레이(350)는 국지적인 또는 분산된, 실제의 또는 가상의, 본 명세서에 기술된 여하한 메모리들 [예를 들어, 메모리들(66 또는 104)]에 포함될 수 있다. 도시된 예에서, 메모리 어레이(350)는 메모리 블록들 [또는 클러스터들(clusters)] B1 - Bn [여기서, n은 여하한 적합한 정수 값임]을 포함한다. 각 블록은 여하한 적합한 수의 바이트(bytes)를 포함할 수 있다.
- [0093] 블록 액세스 장치는 블록 어드레스를 프로세서(62 또는 102)에 제공함으로써 액세스하도록 허용된, 어레이(350)의 여하한 블록들에 액세스할 수 있으며, 이것은 그 다음에 블록 액세스 연산에 따라, 데이터를 어드레스 지정된 블록(addressed block)으로 쓰거나 어드레스 지정된 블록으로부터의 데이터를 읽는다.
- [0094] 그래프 데이터 구조(352)는 메모리 어레이(350)에 계층적 그래프 표시를 부과한다. 그래프 데이터 구조는 파일 저장 데이터베이스, 예컨대, FAT 또는 그 변형들(vFAT, FAT32, FAT16, exFAT, NTFS, NFS 및 그 동등물)을 포함할 수 있다. 도시된 예에서, 그래프 데이터 구조(352)는 루트 디렉토리(root directory), R, 파일들(F<sub>1</sub>, F<sub>2</sub>, F<sub>3</sub>) 및 디렉토리(D<sub>1</sub>)를 포함한다. 그래프 데이터 구조의 이러한 요소들은 각각 요소에 상응하는 메모리 블록들을 식별하는 포인터들을 포함한다. 예를 들어, 디렉토리(D<sub>1</sub>)는 블록(B5)에 상응하고; 파일(F<sub>2</sub>)은 블록들(B6, B9-B10)에 상응하며; 파일(F<sub>3</sub>)은 블록들(B105-B106)에 상응한다.
- [0095] 그래프 액세스 장치는 그래프 데이터 구조(352)를 사용하여 메모리 어레이(350)에 액세스한다. 그래프 액세스 장치는 데이터 구조(352)의 파일들 및/또는 디렉토리들을 읽고, 쓰거나 그렇지 않으면 조작(manipulating)으로써 이것을 수행한다. 프로세서(62 또는 102)는 그래프 데이터 구조(352)의 포인터들을 사용하여, 어레이(350)의 상응하는 메모리 블록들에 액세스한다.
- [0096] 도 13은 블록 액세스 장치 및 그래프 액세스 장치에게 일견하여 동시적인 메모리 액세스를 허용하기 위한 새도우 메모리 기술의 하나의 예를 도시한 컨셉 다이어그램이다. 새도우 메모리 기술은 메모리 어레이(400)에서 쓰기 수행되고 있는 어드레스(들)이 다른 장치, 예컨대, 그래프 액세스 장치에 의해 동시에 액세스될 때, 블록 액세스 장치로부터의 블록 메모리 쓰기를 필수적으로 버퍼링한다(buffer). 이 기술 및 메모리 어레이(400)는, 메모리 어레이(400)가 주변 장치 메모리(66 또는 104)에 존재할 수 있는, 본 명세서에 기술된 여하한 주변 장치들(54, 70, 100)에서 실행될 수 있다.
- [0097] 새도우 메모리 버퍼(shadow memory buffer)는 주변 장치의 동일한 메모리 풀 또는 별개의 메모리 풀에 있을 수 있다. 새도우 메모리 버퍼가 동일한 메모리 풀에 있는 경우에, 버퍼링된 데이터(buffered data)는 메모리 어레이(400)의 미사용 영역들에 저장된다. 프로세서(62 또는 100)는 버퍼를 위한 메모리 영역이 남겨져 있음을 보여주는 그래프 데이터 구조(402)를 업데이트하여야 하고, 또는 프로세서(62 또는 102)는 새도우 버퍼의 위치를 나타내는 그 자신의 사용-메모리 테이블(used-memory table)을 보유한다. 그래프 액세스 장치 또는 블록 액세스 장치 쓰기가 일시적으로 사용된 버퍼 메모리의 위치에 나타날 때, 프로세서(62 또는 102)는 메모리 어레이(400)의 미사용 부분에 대한 쓰기로 리다이렉팅한다.
- [0098] 도시된 예시적인 시나리오에서, 블록 액세스 장치는 메모리 블록들(B105-B106)에 대한 블록 쓰기(block write)를 시도한다. 프로세서(62 또는 102)는, 그래프 액세스 장치가 현재 그래프 데이터 구조(402)를 사용하여 파일(F<sub>3</sub>)인 블록들(B105-B106)에 액세스하고 있는 지를 탐지한다. 프로세서(62 또는 102)는 블록 쓰기로부터 하여금 대신에 블록들(B115-B116)에 대한 쓰기를 수행하게 하는데, 이것이 새도우 버퍼의 역할을 한다. 그래프 액세스 장치가 파일(F<sub>3</sub>)에 대한 액세스를 완료한 후에, 프로세서(62 또는 102)는, 블록들(B105-B106) 대신에 블록들(B115-B116)을 향하도록 파일(F<sub>3</sub>)을 위한 그래프 데이터 구조(402)의 포인터들을 업데이트함으로써 블록 액세스 쓰기를 분해하거나, 또는 그와 달리, 블록들(B115-B116)의 콘텐츠를 각각 블록들(B105-B106)로 전송한 다음에 블록들(B115-B116)을 미사용 블록들로서 풀어준다(release).
- [0099] 도 13은 메모리 풀에 대한 일견하여 동시적 액세스를 가능케 하기 위해 새도우 메모리를 사용하는 방법을 더 충분히 설명하기 위해 제공된 플로우차트(450)이다. 단계(452)에서, 블록 액세스 장치는 메모리 어레이(400)의 파일(F<sub>3</sub>)의 메모리 블록들(B105-B106)에 대한 쓰기 연산(write operation)을 시도한다. 쓰기 연산을 탐지하면, 프로세서(62 또는 102)는 그래프 장치가 현재 블록들(B105-B106)에 상응하는 파일(F<sub>3</sub>)에 액세스하고 있는지의 여부를 판단한다[단계(454)]. 액세스하고 있지 않은 경우에, 프로세서(62 또는 102)는 쓰기가 블록들

(B105-B106)에 수행되도록 허가한다[단계(456)].

- [0100] 그래프 액세스 장치가 현재 동일한 메모리 블록들(B105-B106)에 액세스하고 있지 않으면, 프로세서(62 또는 102)는 블록 액세스 장치로 하여금 대신에 미사용 메모리 블록들(B115-B116)에 대한 쓰기를 수행하게 한다[단계(458)].
- [0101] 단계(460)에서, 프로세서(62 또는 102)는 그래프 액세스 장치의 메모리 액세스 연산들을 모니터링하여 그래프 액세스 장치가 파일(F<sub>3</sub>)에 대한 액세스를 수행하고 있는지를 판단한다. 그래프 액세스 장치가 블록들에 대한 그것의 액세스 연산을 완료하였을 때, 프로세서(62 또는 102)는 파일(F<sub>3</sub>)의 블록들(B115-B116)의 콘텐츠를 포함하도록 그래프 데이터 구조를 분해한다. 도 12와 관련하여 위에서 논의된 바와 같이, 버퍼 콘텐츠(B115-B116)는: 블록들(B115-B116)의 콘텐츠가 블록들(B105-B106)에 복사되거나, 파일(F<sub>3</sub>)을 위한 그래프 데이터 구조(402)의 포인터들이 블록들(B105-B106) 대신에 버퍼(B115-B116)를 향하도록 업데이트되고, 블록 쓰기 콘텐츠(block written contents)가 동일한 메모리 위치들(B115-B116)에 남아있는, 그 두 가지 방식들 중의 하나로 그래프 데이터 구조(402)로 분해된다.
- [0102] 도 14는 클라우드 컴퓨팅을 지원하기 위해 복수의 장치들(502, 506)에 걸쳐 분산된 메모리 풀을 도시한 컨셉 다이어그램이다. 메모리 풀은 주변 장치(502)에 존재하는 메모리 풀 A(508)을 포함한다. 주변 장치(502)는 본 명세서에 기술된 여하한 주변 장치들(54, 70, 100)일 수 있다. 메모리 풀은 또한 네트워크화 장치(networked device)(506)에 존재하는, 메모리 풀 B(510)를 포함한다. 네트워크화 장치(510)는 본 명세서에 기술된 무선 네트워크들과 유선 네트워크들(예를 들어, USB)을 포함하는, 여하한 적합한 네트워크 연결을 통해 주변 장치(502)에 연결된다.
- [0103] 주변 장치(502)는 분산된 메모리 풀을 USB 호스트 또는 Wi-Fi 클라이언트들과 같은 다른 연결된 장치들에 [개별적인 장치들에 물리적으로 위치된 개별적인 메모리 풀들 A 및 B(508, 510)로 구성되지만] 하나의 단일, 논리 메모리 풀로서 제공하도록 구성된다. 주변 장치(502)는 메모리 풀 B(510)에 맵핑하는 제2 그래프 데이터 구조(511)에 그것을 링크하는 포인터들을 포함하는 그래프 데이터 구조(509)를 사용함으로써 이것을 수행하며, 그것에 의해 메모리 풀 A 및 B(508, 510) 모두에 하나의 단일 논리 메모리로서 논리적으로 맵핑한다. 제2 그래프 데이터 구조(511)는 도 14에 도시된 네트워크 장치(506)의 메모리 풀 B에, 또는 메모리 풀 B(510)를 참조하기 위해 주변 장치(502)가 액세스 가능한 그 밖의 다른 곳에 저장될 수 있다.
- [0104] 외부 데이터 서비스들이 또한 참조되고, 메모리 풀들로서 처리될 수 있다. 예를 들어, RSS 뉴스 애그리게이터 서비스(RSS news aggregator service)는 뉴스 스토리(news stories)를 가진 다른 서버들을 향하는 그래프 데이터 구조를 가진 외부 제3자 서버일 수 있다. 이것은 그래프 맵핑을 통해 주변 장치(54, 100)에 의해 액세스될 수 있는 메모리 풀 및 그래프 데이터 구조이다.
- [0105] 주변 장치(508)의 그래프 데이터 구조(509)는 다음 중의 어느 하나에 의해 업데이트될 수 있다:
- [0106] 풀(Pull): 물리적으로 별개의 메모리 풀에 있는 풀링 데이터(pulling data)를 트리거할 수 있는 블록 또는 그래프 액세스를 통한 리소스(resources)에 대한 액세스.
- [0107] 푸쉬(Push): (새로운 Wi-Fi 네트워크에 연결하고, 페이스북 업데이트 등과 같은 서비스로부터의 통지를 입수하는) 트리거(trigger)로 하여금, 주변 장치(508)에서 떨어져 위치된 메모리 풀을 위한 그래프 데이터 구조(509)의 일부 또는 모두를 업데이트하기 위해 액세스를 시작하게 하는, 그 밖의 어딘가에 위치된 메모리 풀에 대한 액세스.
- [0108] 도 15는 메모리 풀(550)을 특정 사용자에게 한정된 영역들로 파티션하는 하나의 예를 도시한 컨셉 다이어그램이다. 주변 장치(100)에서 실행하는 컨텍스트 인식 상태 머신(252)은 사용자들을 하나 이상의 그래프 데이터 구조들 내의 특정 파일들과 디렉토리들로 제한하는 보안 메커니즘을 포함할 수 있다. 사용자는 예를 들어, 패스워드, 또는 메모리 풀(550)의 영역들에 대한 액세스를 획득하기 위한 다른 수단에 의해 인증될 수 있다. 액세스를 승인하기 위한 퍼미션들은 저장될 수 있고, 그래프 데이터 구조들의 파트로서 구성될 수 있다.
- [0109] 도시된 예에서, 사용자들(A, C 및 D)에게는 그래프 데이터 구조(553)로 표시되는, 제1 메모리 풀 파티션(552)에 대한 액세스가 주어진다. 사용자 B 에게만 그래프 데이터 구조(557)로 표시되는, 제3 메모리 풀 파티션(556)에 대한 액세스가 주어진다. 모든 사용자들(A, B, C 및 D)은, 모든 사용자들에 대해 공통적인, 디렉토리(D)에 의해, 그래프 데이터 구조(555)로 표시되는, 제2 메모리 풀 파티션(554)에 대한 액세스를 가진다.
- [0110] 도 16은 축소된 메모리 공간을 사용하는 TCP/IP 패킷들을 전송하는 예시적인 방법을 도시한 플로우차트(600)

이다. 이 방법은 주변 장치(100)에서 작동하는 소프트웨어 [예를 들어, TCP/IP 스택(262) 및/또는 상태 머신(252)]에 의해 실행될 수 있다. 이 방법은 TCP/IP 프로토콜을 실행하기 위한 메모리 저장공간 요구사항들(memory storage requirements)을 축소시키는 장점이 있는데, 이것은 한정된 메모리 및 리소스들을 가진 장치들, 예컨대, 주변 장치에서 중요할 수 있다.

[0111] 단계(602)에서, 패킷은 TCP/IP를 사용하여 전송된다. 주변 장치(100)와 관련하여, TCP/IP 패킷은 무선 인터페이스(108), 또는 USB 인터페이스(106)를 통해 전송될 수 있다. 종래의 TCP/IP를 사용하면, 전송이 성공적이었는 것을 나타내는, 확인응답(acknowledgement: ACK)이 수령자 TCP/IP 노드(node)로부터 수신될 때까지 전송된 패킷의 카피(copy)가 임시 메모리에 저장되는 것이 일반적이다. 만약 ACK가 수신되지 않으면, 발송자는 임시 메모리로부터 복사된 패킷을 신속히 회수하여 그것을 재전송할 수 있다. 그러나, 복사된 패킷을 일시적으로 저장하는 이러한 종래의 TCP/IP 방법을 사용하는 경우의 단점은, 비교적 많은 양의 메모리를 소모할 수 있고, 이것은 메모리 공간이 제한되는 어플리케이션들에서 바람직하지 않을 수 있다.

[0112] 이러한 단점을 극복하기 위해서, 도 16의 방법은 전송된 패킷을 복사하지 않고, 포인터를 그것이 메모리 풀에서 존재하는 곳인 전송된 패킷에 저장한다[단계(604)]. 패킷을 전송한 후에, 종래의 TCP/IP 프로토콜 단계들을 사용하여, ACK가 수령자로부터 수신되는지를 확인하기 위해 체크한다[단계(606)]. ACK가 수신되면, 추가적인 패킷들을 보낼 필요가 있는지를 결정하기 위한 판단이 이루어진다[단계(608)]. 만약 그렇다면, 이 방법은 단계(602)로 되돌아가고, 그렇지 않으면 이 방법은 종결된다.

[0113] 만약 TCP/IP 프로토콜에 의해 규정된 대로 수령자로부터 ACK가 수신되지 않으면, 포인터가 메모리 풀로부터 패킷을 회수하기 위해 사용되고[단계(610)], 패킷이 재전송된다[단계(612)].

### [0114] III. 주변 장치 보안

[0115] 주변 장치(54, 70, 100)에 의해 지원되는 메모리 풀 부분들은 장치/파티션 레벨에서 또는 개별적인 파일 레벨에서 암호화될 수 있다. 저장부의 개별적인 부분들이 상이한 키들로 암호화될 때, 전체 저장 영역은 (마스터 키를 사용하여 암호화된) 확보된 영역에 저장된 개별적인 서브-키들과 함께, 마스터 키를 사용하여 액세스 가능할 수 있거나; 전체 저장 영역이 하나의 키로 암호화될 수 있다.

[0116] 암호화된 저장부를 잠금해제하는 것은 유선 또는 무선 인터페이스들을 통해 수행될 수 있다. 호스트를 인증하기 위해서, 주변 장치(54, 70, 100)는 호스트에 표준 대용량 저장 장치를 제공할 수 있고, 호스트로 하여금 파일 쓰기를 수행하거나 키의 역할을 하는 특정 이름으로 디렉토리를 생성시키도록 요청할 수 있거나; 그것은 호스트에서 작동하는 프로그램과 통신할 수 있다. 무선 인증이 무선 또는 유선 액세스를 위해 저장부를 잠금해제하기 위해 사용될 수 있다. 잠금해제 후의 액세스는 시간이 정해질 수 있으며, 사용자가 저장부를 다시 명시적으로 잠금 때까지 지속될 수 있고, 그리고/또는 현재의 무선 네트워크로부터의 인증[화이트리스트(whitelist)]이 끝날 때까지 또는 다른 무선 네트워크들이 탐지될 때까지[블랙리스트(blacklist)] 지속될 수 있다. 무선 인증을 위해 사용된 액세스 제어의 동일한 모드가 사용될 수 있다. 예를 들어, 사용자는 WPA2를 사용하여 무선 액세스에 대해 인증할 수 있고, WPA2는 그 다음에 설정된 기간 동안 저장부에 대한 유선 액세스를 잠금해제할 것이다. 다른 표준 인증 방법들[NFC(near field communication: 근거리 무선 통신) 페어링, RFID, 또는 블루투스]이 또한 저장부를 잠금해제하기 위해 사용될 수 있다.

[0117] 주변 장치 프로세서는 또한 블록 액세스 또는 그래프 액세스 연결성의 제공 여부를 결정할 수 있다. 이 경우에, 프로세서는 특정 파이프들(연결들)을 통해서든 또는 메모리 풀의 다양한 부분들에 대해서든 특정 액세스를 차단하는 방법을 결정하기 위해 상태 머신을 사용한다. 이를 수행하기 위해, 프로세서는 호스트로부터의 요청들을 살펴보고, 요청에 따라 제공할 액세스의 유형을 결정하거나, 알려진 우선순위 리스트(priority list)로부터 서비스 공급을 제공하고, 장치가 어떻게 응답하는지를 보거나, 이전에 저장된 셋팅을 토대로 한 연결을 허용하거나, 시간, 이전부터 존재하는 특정 연결들 또는 식별된 호스트들의 존재, 주변 장치 위치, 로컬 메모리 풀에 저장된 데이터의 유형, 또는 그 동등물과 같은, 결정된 컨텍스트 상태를 토대로 연결을 허용한다.

[0118] 주변 장치 프로세서가 연결성을 결정하기 위해 따를 수 있는 일련의 규정들이 있다. 이러한 규정들은, 주변 장치가 인식하고 있는, 연결된 클라이언트들 또는 사용가능한 클라이언트들 (Wi-Fi 네트워크 존재), 메모리 풀의 데이터, 미리 설정된 조건들, 센서 데이터 (시간, 온도, 광, 위치), 그래프 및/또는 데이터 비교 (인증 등)와 같은 다양한 컨텍스트들을 모두 참조할 수 있다.

[0119] 이러한 비교들은 지불결제 시스템 등을 가능케 한다.



- [0120] 다른 구성은, 블록 액세스가 아닌 그래프 액세스로 네트워크 장치인 클라이언트에 연결된 주변 장치이다(블록 액세스로 WebDAV 또는 USB 대용량 저장 장치가 아닌, 웹, 즉, 데이터에 대한 그래프 액세스를 팝 업(pop up)할 PC에 네트워크 장치로서 동작하는 것).
- [0121] 연결된 클라이언트로부터 수신된 데이터는, 블록 액세스를 통한 그래프 액세스를 통한, 또한 액세스 퍼미션 이벤트들(access permission events)을 일으킬 수 있다. 예를 들어, 파일은, 클라이언트 장치 연결성을 변경하기 위한 일련의 파라미터들, 예를 들어, Wi-Fi 대 3G/4G 데이터 연결에 대한 연결성을 온/오프하는 텍스트 파일을, 가지도록 쓰기 수행될 수 있다. 다른 예는, 특정 외부 네트워크 서비스 및/또는 메모리 풀로부터의 RSS 피드들(feeds)을 캡처하기(capture) 위해 그래프의 파일 디렉토리를 네이밍(naming)하는 것이다.
- [0122] 프로세서가 특정 데이터 및 연결성 이벤트들(connectivity events)을 모두 기초로 하여 결정하는 결합된 데이터 및 연결성 인텔리전스(combined data and connectivity intelligence)가 또한 있을 수 있다. 이것은 특정 시점에 주변 장치의 현재의 컨텍스트에 기여할 수 있다.
- [0123] IV. 그래프 액세스-대-블록 액세스 맵핑
- [0124] 주변 장치(54, 100)는, 무선 인터페이스(68, 108)를 통해 획득한 로컬 저장 및 원격 콘텐츠의 조합으로부터 최종적으로 유래된, 유선 인터페이스(64, 106)를 통한 저장을 제공할 수 있다. 무선 콘텐츠는 다양한 소스들로부터 얻을 수 있으며, 주변 장치 소프트웨어에 의해 로컬 저장부에 비동기적으로(asynchronously) 캐시될 수 있다.
- [0125] 예시적인 정보 소스들은 다음을 포함한다:
- [0126] 1. 로컬 네트워크 상의 파일 공유;
- [0127] 2. 팟캐스트(podcast), 포토스트림(photostream), 또는 정보의 다른 컬렉션을 포함하여 구성되는 RSS 피드; 및/또는
- [0128] 3. 온라인 네트워크, 예컨대, 유튜브(YouTube)로부터의 비디오들의 컬렉션.
- [0129] 정보의 이러한 소스들은 각각 유선 인터페이스(64, 106)를 통해 액세스 가능한 개별적인 디렉토리에 맵핑될 수 있다. 다양한 소스들이, 장치 레벨(복수의 부착 저장 장치들의 제공)에서 또는 파일 시스템 레벨(다수의 정보 소스들로부터 가상의 파일 시스템 및 FAT 테이블의 생성)에서 함께 결합될 수 있다.
- [0130] 특별한 사용자에게 볼륨/파티션, 파일, 및/또는 폴더 레벨 액세스를 가능케 하는 능력이 또한 있다. "Bob"이라는 라벨을 붙인 폴더는, "Bob"이 동일한 네트워크 상에 있을 때 그리고/또는 원격 서버를 통해 액세스할 때, 사용자 "Bob"이 이 폴더에 대한 퍼미션을 가지도록 하는 액세스와 관련될 수 있다. 다른 것과 페어링하고, 데이터의 일부 또는 모두가, (그들이 대리로서 하든 및/또는 원격 저장으로서 하든) 원격 서버들을 통해서든, 또는 예를 들어, Wi-Fi 사이에 직접 동기화하든, 장치들 사이에 동기화되도록 항상 시도하는, 하나 보다 많은 주변 장치를 가지는 것이 또한 특유한 기능이다.
- [0131] 무선 정보에 액세스할 때, 그러한 정보의 일부 또는 모두가 적절하게 주변 장치(54, 70, 100) 상의 메모리(66, 104)에 위치될 수 있거나, 그러한 정보가 전혀 메모리에 위치되지 않을 수 있다. 예를 들어, 파일 공유는 데이터의 캐싱(caching)을 필요로 하지 않을 수 있으며; 여하한 파일 요청들이 즉시 무선 파일 공유 요청들(wireless file share requests)로 변환될 것이다. 캐싱이 추가적인 성능을 위해 추가될 수 있다.
- [0132] 주변 장치(54, 70, 100) 상의 폴더는 SS 피드들과 같은 다른 원격 콘텐츠를 위한 다운로드 영역으로서 지정될 수 있으며, 콘텐츠는 이 폴더에 이용가능하게 됨에 따라 장치에 의해 자동적으로 다운로드될 것이다.
- [0133] 무선 상태 변화가 또한 다운로드가 개시되게 할 것이다. 예를 들어, 설정된 시각에 실행된(remembered) 무선 네트워크가 이용가능하게 되면, 주변 장치(54, 70, 100)가 업데이트된 정보를 탐색하고 다운로드할 수 있다. 다른 예는, 주변 장치(54, 70, 100)가 카스테레오의 USB 포트에 플러그 연결된 경우에, 차가 소유자의 소유차도로 돌아갈 때, 그것이 홈 무선 네트워크에 연결되어 최신 팟캐스트들을 다운로드하거나, 파일들을 소유자의 컴퓨터와 동기화시킬 것이다.
- [0134] 무선 상태 변화도 역시 저장부에 대한 액세스를 제어할 수 있다. 예를 들어, 특정 무선 네트워크에 대한 연결은 특정 파일들 또는 폴더들을 검토(view) 및 변경에 사용가능하게 만들 수 있다. 주변 장치(54, 70, 100)는 또한 저장부의 특정 디렉토리 또는 영역에 대한 쓰기가 항상 가능하지만, 그 영역으로부터 파일들을 읽는 것은 특정 액세스 포인트의 범위 내에 있을 때, 또는 특정 액세스 포인트와 연관된 때를 제외하고는 가능하지



않도록 구성될 수 있다. 무선 네트워크들의 존재는 또한 파일들의 삭제를 일으킬 수 있다["원격 삭제(remote wipe)"].

[0135] 주변 장치(54, 70, 100)가, 최신의 프로세싱이 이루어짐에 따라, 네트워크 SSID들 및/또는 MAC 어드레스들을 탐지하고 기록하는 것이 또한 가능하다. 이것은 시간이 지남에 따라 장치(54, 70, 100)가 위치한 곳을 추적하는 것을 도울 수 있으며, GPS 또는 다른 삼각 측량법(triangulation)/허용가능 액세스를 위한 위치 기능과 함께 사용될 수 있거나 다른 기능들을 트리거할 수 있다.

[0136] 장치(54, 70, 100)가 USB를 통해 호스트에 연결되고, 주변 장치가 USB 네트워크 프로파일을 에뮬레이트하는(emulate) 구성이 또한 가능하다. 호스트는 주변 장치(54, 70, 100)를 네트워크 연결로 보고, 브라우저(browser) 또는 유사 어플리케이션을 시작한 다음에, 서버에 액세스하는 것처럼 데이터를 스트리밍할(stream) 것이다. 그러면 주변 장치(54, 70, 100)는, 그 콘텐츠가 주변 장치 상에 있든 그리고/또는 원격 서버로부터 모아지든, 콘텐츠를 브라우저에 대화식으로(interactively) 공급하는 서버의 역할을 할 수 있다. 이것은 콘텐츠가 원격 서버로부터 인터넷을 통해 전달되는 것과 유사한 방식으로 안전한 방식으로 콘텐츠 전달을 가능하게 할 수 있다.

[0137] 특정 파일이 (이름, 콘텐츠, 또는 다른 요소들에 의해 발견된) 저장부에 존재할 때, 장치가 호스트에 제공된 액세스의 모드를 변화시킬 수 있다. 특별한 이름을 가진 CD 또는 DVD 이미지가 저장부에서 발견되면, 주변 장치(54, 70, 100)가 첨부 USB CD 또는 DVD 리더인 호스트에 드라이브에 삽입된 가상의 광 디스크의 콘텐츠를 결정하는 파일을 제공할 수 있다. 이러한 특별한 기능은 실제의 광 드라이브(optical drive)가 없는 컴퓨터들에 운영 체제들을 설치하기 쉽게 만든다.

[0138] 저장부의 일부가 주변 장치 기능성을 위해 확보될 수 있다. 이 영역은 장치 또는 파티션 레벨에서 확보될 수 있거나, 장치 그 자체에 원하는 크기의 파일을 할당함으로써 확보될 수 있다. 파일이 변경되거나 주변 장치(54, 70, 100)로부터 삭제되는 경우에, 그것의 콘텐츠는 사용자가 나중에 데이터를 복구하도록 확보된 영역에 복사될 수 있다.

#### [0139] V. 클라우드 서비스(Cloud Services)

[0140] 유선 인터페이스와 무선 인터페이스의 조합이 주변 장치 소유자에게 추가적인 서비스들을 제공하기 위해 사용될 수 있다. 무선 인터페이스가 인터넷에 연결될 때, 유선 또는 무선 인터페이스들을 통해 저장부에 취해진 다양한 동작들이 인터넷으로부터의 저장으로 변환될 수 있거나 보완될 수 있다.

[0141] 주변 장치(54, 70, 100)는 온라인 백업 기능성을 제공할 수 있다. 저장부의 어떤 부분들이 사용자에게 의해 변화되었는지를 모니터링하고, 그러한 부분들을 확보된 메모리 영역에 보존된 테이블에 "dirty"로 표시함으로써, 주변 장치(54, 70, 100)는 저장부의 어떤 부분들이 인터넷을 통해 서버에 백업 되어야 하는지를 안다. 전송된 데이터의 콘텐츠는 서버로 전송하기 전에 주변 장치(54, 70, 100)에 의해 암호화될 수 있다. 장치(54, 70, 100)를 잃어버리면, 장치(54, 70, 100)의 저장부의 와이프(wipe)를 개시하라는 명령(command)이 인터넷으로부터 장치(54, 70, 100)로 보내질 수 있고, 새로운 주변 장치가 잃어버린 장치로부터의 데이터의 콘텐츠로 프로그래밍될 수 있다. 마찬가지로, 온라인 백업이 변경되면, 그것은 이 과정을 역순으로 적용하여 주변 장치가 온라인 백업에서의 변화들에 부합되도록 확실히 할 수 있다.

[0142] 주변 장치(54, 70, 100)는 통신 인터페이스를 통한 파일들의 조작에 따라 동작할 수 있고, 동일하거나 다른 통신 인터페이스를 통해 인터넷 저장부에 대한 동작들로 이들을 변환시킬 수 있다. 예를 들어, 장치(54, 70, 100) 상의 특정 폴더는, 정책(policy)을 따라 특정 사용자에게 주어진 액세스로, 파일 공유 영역으로 지정될 수 있다. 파일을 이 폴더로 이동시키거나 복사하는 것은, 원격 저장부에 대한 무선 인터페이스를 사용하여 그 파일의 전송을 개시할 것이며, 그렇게 되면, 다른 사용자들이 액세스 가능할 것이다. 하나의 폴더가 다른 사용자로부터의 파일들을 수신하기 위한 드롭박스(dropbox)로 지정될 수 있으며, 이 경우에, 주변 장치(54, 70, 100)는 원격 저장부로부터의 정보를 질의하고(query) 불러올(fetch) 수 있으며, 장치(54, 70, 100) 상의 폴더에 이러한 정보를 위치시킬 수 있다. 장치(54, 70, 100) 상의 모든 파일들이 특유한 개별적인 키들로 암호화되면, 파일은 암호화된 형태로 전송될 수 있고, 이러한 파일을 잠금해제할 키는 보안 수단을 통해 다른 사용자와 공유될 수 있다.

[0143] 인터넷 서비스는 추가적인 파일들을 주변 장치 저장부에 추가시키는 결과를 가져오는 파일 조작들에 따라 동작을 취할 수 있거나, 다른 동작들을 취할 수 있다. 예를 들어, 특정 디렉토리에 파일을 위치시키는 것은, 그 파일이 미리 정해진 이메일 어드레스로 이메일 첨부물로서 보내지게 할 수 있다. 다른 디렉토리에 비디오 파

일을 위치시키는 것은, 인터넷 서비스 상에서 일어나는 포맷 변환(format conversion)을 일으킬 수 있으며, 변환된 파일은 그 다음에 장치(54, 70, 100) 상의 디렉토리에 다운로드된다.

- [0144] 다른 구성에서, 주변 장치(54, 70, 100)는 주변 장치(54, 70, 100) 상에 실제로 존재하는 것보다 더 큰 데이터 어레이를 호스트에 제공한다. 이 경우에, 파일 저장부는 일반적으로 호스트 쪽에 나타나지만, 데이터 저장부는 실제로 네트워크화 장치 상의 다른 위치에 있으며, 주변 장치에 의해 캐시된다.
- [0145] 주변 장치(54, 70, 100)는 또한 캐시인 주변 장치에 의해 지원되는 메모리 풀을 사용하여, 한명 이상의 사용자들을 위한 클라우드 캐싱(cloud caching)을 가능케 하도록 구성될 수 있다. 이를 수행하기 위해, 주변 장치는 하나 이상의 온-라인 클라우드 서버들 및 사용자의 장치(예를 들어, 호스트 또는 클라이언트) 사이의 네트워크 통신 경로에 포함된다.
- [0146] 클라우드 서버에 대한 연결이 정지되는(go down) 경우에, 사용자 장치는 주변 장치(54, 70, 100)에 여전히 연결되어 있다. 주변 장치(54, 70, 100)는 클라우드 서버(들)로부터 전송된 데이터를 캐시한다. 그러면 데이터 저장 주변 장치의 역할을 하는 주변 장치(54, 70, 100)는, 나중에 액세스가 다시 유효하게 될 때 클라우드 서버(들)에 대해 동기화한다.
- [0147] 그와 달리, 클라우드 서버들에 대한 대역폭(bandwidth)을 사용하는 너무나 많은 사용자들이 있는 경우에, 주변 장치(54, 70, 100)는, 순차적으로 클라우드 서버들에 액세스하여 사용자들에 걸쳐 대역폭 시간을 분산시키면서, 각 사용자를 위해 캐시할 수 있고/예비-인출할(pre-fetch) 수 있다.
- [0148] 주변 장치(54, 70, 100)는 또한 AP에 장착된 구글 기어스(Google Gears) 또는 썬-클라이언트 네트워크 캐시(thin-client network cache)의 역할을 하도록 구성될 수 있다. 이러한 구성에서, 주변 장치(54, 70, 100)는, 네트워크 연결성이 유효하게 될 때, 한 명 이상의 사용자들에 대해 네트워크 리소스들에 대한 대체시스템(fallback)이 존재하지 않는 것으로 가장한다(simulate). 이러한 구성의 목적은 밀집 네트워크들(congested networks)로 응답 시간을 개선하고 그리고/또는 간헐적 네트워크 연결성(intermittent network connectivity)으로 네트워크/클라우드 기능성에 대한 액세스를 제공하는 것이다. 이 기능을 사용한 하나의 예로서, 네트워크/인터넷 액세스가 존재하지 않으면서, Wi-Fi AP 모드에 있을 때, 사용자는 주변 장치(54, 70, 100)로 이메일을 보낼 수 있다. 인터넷 액세스가 가능하게 될 때, 주변 장치(54, 70, 100)는 클라이언트가 되고, 예를 들어 구글 기어스를 사용하여, 인터넷을 통해 연결하고 이메일을 보낸다.
- [0149] 주변 장치(54, 70, 100)는 또한 그것의 로컬 메모리에 저장된 광고들 또는 스니핑(sniffing) 및/또는 위치 정보[스카이훅(Skyhook), GPS, 또는 그 동등물]를 기반으로 네트워크 서버로부터의 광고들을 삽입하도록 구성될 수 있다. 주변 장치(54, 70, 100)가 동시 AP 및 클라이언트 모드를 둘다 가질 때, 그것은 광고들을 제거할 수 있고, 그 광고들을 그 로컬 메모리 저장부로부터의 광고들로 대체할 수 있다. 이 광고들은 클릭하면 찾아낼 수 있고, 데이터는 주변 장치의 그래프 데이터 구조내에 저장될 수 있다.
- [0150] 또한, 주변 장치(54, 70, 100)가 AP 모드에 있을 때, 그것은 몇몇 SSID들을 제공할 수 있다. 예를 들어, 그 하나는 "비메오(Vimeo)"일 수 있다. 만약 선택되어 AP가 아이폰을 감지하면, 아이폰 유튜브 플레이어를 사용하도록 비메오가 유튜브로 위장될 수 있다.
- [0151] VI. 지불결제 프로세싱 시스템
- [0152] 도 17은, 본 명세서에 기술된 주변 장치들(54, 70, 100) 중의 하나일 수 있는 주변 장치(702)를 포함하는, 지불결제 프로세싱 시스템(700)을 도시한 컨셉 다이어그램이다. 지불결제 프로세싱 시스템(700)은, 주변 장치(702), 스마트폰 또는 PDA(704), 스토어 또는 머천트 무선(store or merchant wireless) 액세스 포인트(AP), 예컨대, Wi-Fi AP, 판매-시점-관리(point-of-sale: POS) 터미널(708) 및 지불결제 프로세싱 서버(710)를 포함한다. 주변 장치(702)는 스마트폰(704) 그리고 Wi-Fi 연결들과 같은 무선 채널들을 사용하는 스토어 AP(706)와 통신한다. 주변 장치(702)는 또한 POS 터미널(708)과 통신하기 위해 Wi-Fi와 같은 무선 통신들, 또는 그와 달리, 본 명세서에 개시된 것들과 같은 유선 연결을 사용할 수 있다. POS 터미널(708)은 데이터 네트워크를 통해 서버(710)와 통신한다. 서버(710) 및 스토어 AP(706)는 인터넷(712)에 네트워크 접속된다.
- [0153] 시스템(700)은 스마트폰(704) 및 주변 장치(702)를 가진 고객으로부터 Wi-Fi 연결들을 통한 지불결제 프로세싱을 허용하는 것이 바람직하다. 시스템(700)은 최소한 두 가지 유형의 지불결제 거래들(payment transaction)을 지원할 수 있다. 제1 거래 유형에서, 시스템(700)은 다음과 같은 과정을 사용한다:
- [0154] 1. 주변 장치(702)가 스마트폰(704)을 위한 Wi-Fi 채널들을 스캔하고, 그것을 찾아내면 그것과 안전하게 페어

링한다.

[0155] 2. 상점 계산원이 POS 터미널(708)에 거래 금액이 나오게 한다.

[0156] 3. POS 터미널(708)이 Wi-Fi 연결을 통해 총 거래금을 주변 장치(702)로 보내고, 이것이 다시 그 총 거래금을 Wi-Fi를 통해 스마트폰(704)으로 보낸다.

[0157] 4. 고객이 스마트폰(704)을 통해 거래를 승인한다.

[0158] 5. 주변 장치(702)가 POS 터미널(708)로 고객에 의해 거래가 승인되었다는 표시를 전송한다.

[0159] 6. 그러면 주변 장치(702)가 스마트폰(704)으로부터 연결해제된다.

[0160] 제2 유형의 거래로, 시스템은 다음과 같은 과정을 사용한다:

[0161] 1. 스마트폰(704)이 유효 범위를 가진 주변 장치(702)에 의해 Wi-Fi 네트워크를 통해 전송된 MAC 어드레스를 식별하는 어플리케이션을 작동시킨다.

[0162] 2. valu MAC 어드레스를 확인하면, 스마트폰(704)이 주변 장치(702)에 대한 Wi-Fi 연결을 개시한다.

[0163] 3. 주변 장치(702)는 고객의 계좌에 관한 정보, 예컨대, 로열티, 쿠폰들, 또는 그 동등물을 얻기 위해 POS 터미널(708) 또는 스토어 AP(706)와 통신한다. 통신은 무선 채널 B를 통해 또는 서버(710) 및/또는 인터넷(712)을 통한 POS 터미널(708)을 거쳐 일어날 수 있다.

[0164] 4. 본 과정은 그 다음에 상기 제1 거래 유형의 3-6 단계를 실행한다.

[0165] 슈퍼마켓의 복합 계산대와 같은, 복수-레인(multi-lane) 지불결제 시스템들에서, 클라이언트, 예컨대, 스마트폰(704) 및 서버, 예컨대, 주변 장치(702) 사이의 페어링에 대한 혼동이 있을 수 있다. 복수 경로 문제를 해결하기 위한 다양한 방법들이 있다. 그러한 방법의 하나는 RF 삼각 측량법을 사용하는 것이다. 다른 방법은 한 레인(lane)에서 다른 레인으로의 통신을 막기 위해 주변 장치(702)로부터 방출된 무선 신호의 범위를 다양하게 하는 것이다. 이것을 달성하기 위해, 주변 장치(702)는 다음과 같은 과정을 실행하도록 구성될 수 있다:

[0166] 1. 주변 장치(702)는, 스마트폰(704)이 근거리 범위 내에 있는 경우에, 통신이 시작될 때까지, 그 전송의 Wi-Fi 범위를 몇 인치로 제한한다(도 17의 채널 A). 주변 장치는 그것의 Wi-Fi 전송 전력을 감소시킴으로써 이것을 수행한다. 스마트폰(704)은 고객이 주변 장치(702)와 스마트폰(704)을 가까이 뮴으로써 일정 범위 내에 있게 된다.

[0167] 2. 일단 스마트폰(704)과의 Wi-Fi 통신이 이루어지면, 주변 장치(702)의 전송 범위가 증가되어 고객이 폰을 정상적으로 잡을 수 있게 할 수 있다. 일단 주변 장치(702)와 스마트폰(704) 사이에 Wi-Fi 세션(session)이 확립되면, 레인들 사이의 간섭은 일어나지 않을 것이다.

[0168] 3. 주변 장치(702)가 원격 WAN/인터넷 연결들을 위해 상점의 Wi-Fi AP(706)와 통신할 필요가 있는 경우에, 그 범위가 고객의 스마트폰(704)에 대한 짧은 범위로부터 상점의 Wi-Fi AP(706)에 대한 긴 범위로 멀티플렉싱된 방식으로 변화될 수 있다(도 17의 채널 B). 주변 장치는 또한 필요에 따라 클라이언트 모드와 액세스 포인트 모드 사이에서 변화할 수 있다.

[0169] 4. 일단 지불결제 프로세스가 완료되면, 프로세스가 1 단계로 되돌아간다.

[0170] 유틸리티(utility)를 개선시키기 위해 주변 장치에 추가적인 기능들이 추가될 수 있다. 이러한 기능들은, 보안, 매체 재생의 동기화, 복수의 파일들을 단일 파일로 결합[예를 들어, TV 상의 사진공유(photosharing)], 비디오 및/또는 오디오 파일들, 문서들, 예컨대, .doc 내지 .pdf와 같은 파일들의 자동 트랜스코딩 및 그 동등물을 포함할 수 있다.

[0171] 본 발명은 주변 장치로 하여금 USB 주변 장치 및 독립형(stand alone) 장치로서의 새로운 기능성을 둘다 제공하는 새로운, 강화된 기능성을 창출하게 하는 새로운 방법들과 아키텍처들을 제공한다. 아키텍처의 특정 구성들은 종래의 USB 대용량 저장 장치와 동일한 기능을 가지는 것을 수반하지만, USB 포트를 통해 직접 인터페이스 연결되는 것이 아니라 주변 장치에 포함된 데이터 저장부를 수반하는 공유 메모리 스킴(share memory scheme)을 통해 인터페이스 연결되는 추가적인 기능성을 추가함으로써 신규한 것이다. 추가적인 기능성은 무선 모뎀, 예컨대, Wi-Fi, 3G/4G 셀룰러 모뎀, 블루투스, NFC, 적외선, 또는 유선 통신, 예컨대, USB, 이더넷(Ethernet) 또는 그 동등물과 같은 통신들일 수 있다. 다른 방법에서, 프로세서와 같은 복합 상태 머신(complex state machine)이 USB 포트와 함께 또는 그것과 독립적으로 기능성 연산들을 제공한다. 이러한 기능

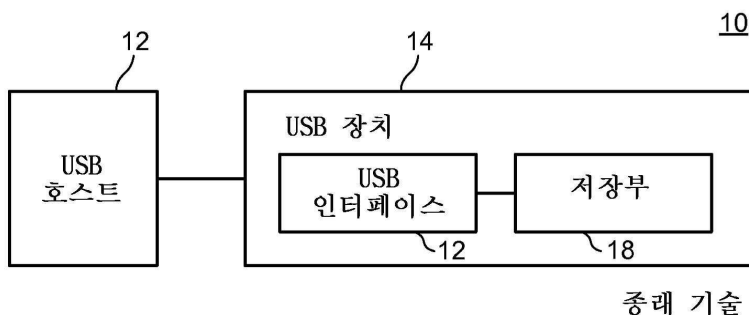
성 연산들은, USB 포트 및 다른 기능들 (예컨대, 통신 기능들, 데이터 저장부에 대한 파일 또는 데이터베이스 액세스) 사이의 메모리 액세스의 하위-레벨 중재(low-level arbitration), 또는 예컨대, 웹 서버, 데이터 동기화 엔진(data synchronization engine), 암호화/암호해독, 인증 및 다른 복합 상태 기능들과 같은 고차 레벨 기능들을 포함한다.

[0172] 본 명세서에 기술된 시스템들, 장치들 및 그들 각각의 구성요소들의 기능성, 그리고 방법 단계들 및 블록들은 프로세서에 의해 실행되는 하드웨어, 소프트웨어 또는 펌웨어, 또는 그들의 여하한 적절한 조합으로 실행될 수 있다. 소프트웨어/펌웨어는 하나 이상의 디지털 회로들 또는 프로세서들, 예컨대, 마이크로프로세서들, DSP들, 내장형 컨트롤러들, 또는 지적 재산(intellectual property: IP) 코어들에 의해 실행가능한 명령어 세트들[예를 들어, 코드 세그먼트들(code segments)]을 가진 하나 이상의 프로그램들일 수 있다. 소프트웨어/펌웨어에서 실행되는 경우에, 명령어들 또는 코드가 하나 이상의 컴퓨터-판독가능 매체에 저장될 수 있다. 컴퓨터-판독가능 매체는, 한 곳에서 다른 곳으로 컴퓨터 프로그램의 전송을 촉진하는 여하한 매체를 비롯하여, 컴퓨터 저장 매체 및 통신 매체를 둘다 포함한다. 저장 매체는 컴퓨터에 의해 액세스될 수 있는 여하한 사용가능한 매체일 수 있다. 하나의 예로서, 그리고 제한 없이, 이러한 컴퓨터-판독가능 매체는 RAM, ROM, EEPROM, NAND/NOR 플래시, CD-ROM, 또는 다른 솔리드-스테이트 저장부, 광 디스크 저장부, 자기 디스크 저장부 또는 다른 자기 저장 장치들, 또는 명령어들 또는 데이터 구조들의 형태로 원하는 프로그램 코드를 수반하거나 저장하기 위해 사용될 수 있고, 컴퓨터에 의해 액세스될 수 있는, 여하한 다른 매체를 포함하여 이루어질 수 있다. 본 발명에 사용된 디스크(disk) 및 디스크(disc)는, 콤팩트 디스크(compact disc: CD), 레이저 디스크(laser disc), 광 디스크(optical disc), 디지털 다기능 디스크(digital versatile disc: DVD), 플로피 디스크(floppy disk) 및 블루-레이 디스크(blue-ray disc)를 포함하며, 여기서 디스크들(disks)은 일반적으로 데이터를 자기적으로 재생하는 반면, 디스크들(disks)은 데이터를 레이저로 광학적으로 재생한다. 상술한 것들의 조합들이 또한 컴퓨터-판독가능 매체의 범위 내에 포함되어야 한다.

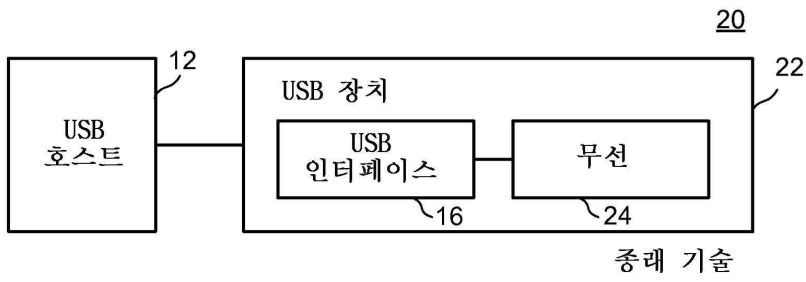
[0173] 상술한 방법들, 디바이스들, 시스템들 및 장치들의 다른 실시예들 및 변형들이 본 발명의 가르침들을 고려하여 이 분야의 통상의 지식을 가진 자들의 머리에 쉽게 떠오를 것이다. 따라서, 전술한 설명은 본 발명을 설명하는 것이며, 본 발명을 제한하지 않는다. 본 발명은, 상기 명세서 및 첨부된 도면들과 함께 검토할 때, 모든 이러한 다른 실시예들 및 변형들을 커버하는, 다음의 특허청구범위에 의해서만 제한되는 것이다. 따라서, 본 발명의 범위는, 상기 상세한 설명으로 한정되지 않아야 하며, 그 대신에 그 동등물들의 완전한 범위와 함께 첨부된 특허청구범위와 관련하여 결정되어야 한다.

## 도면

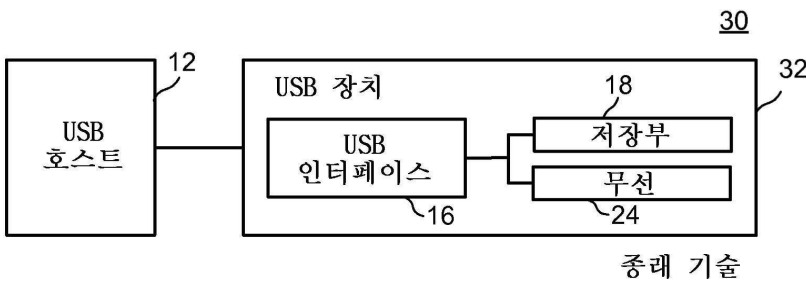
### 도면1



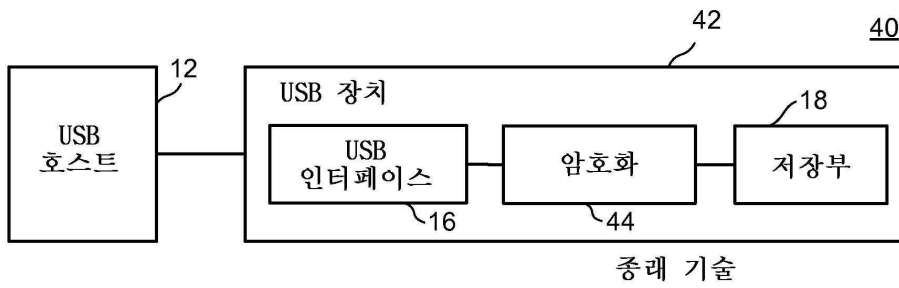
도면2



도면3

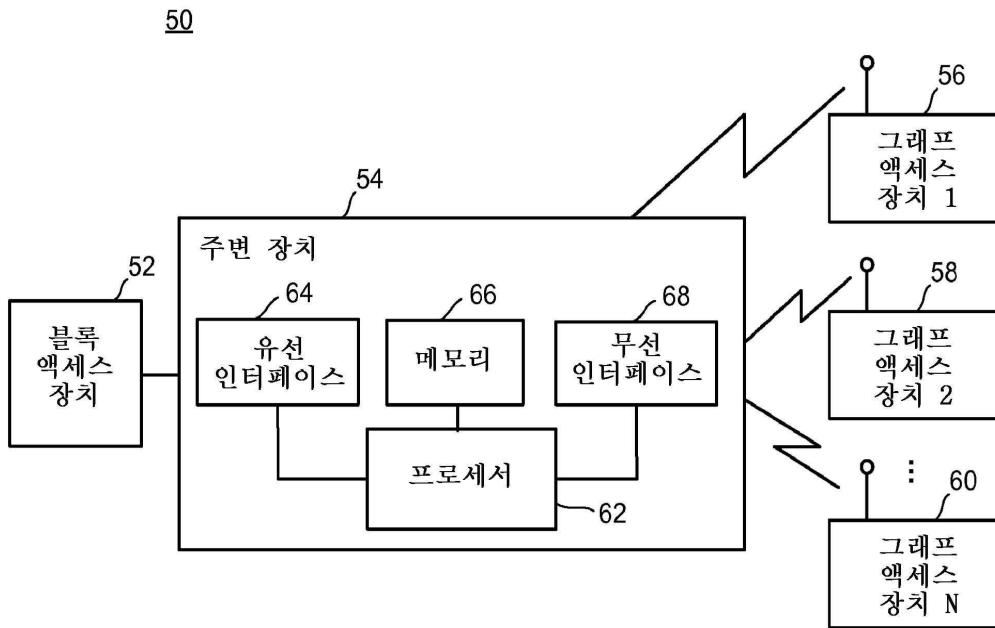


도면4

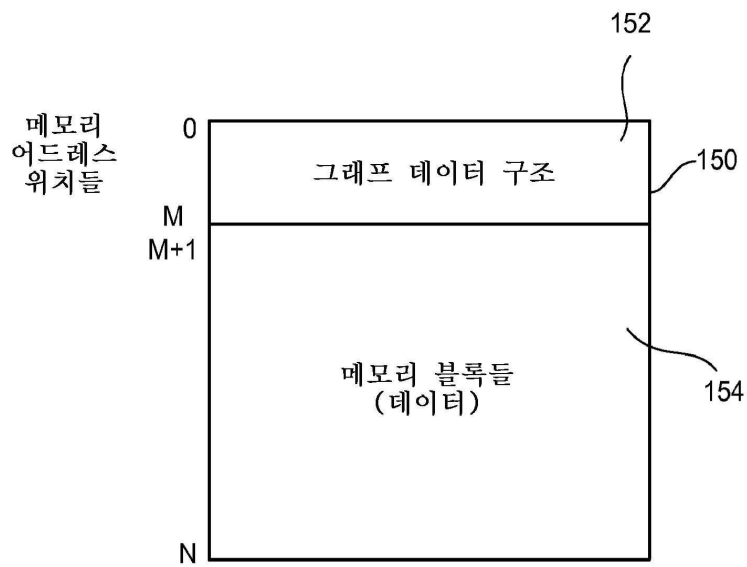




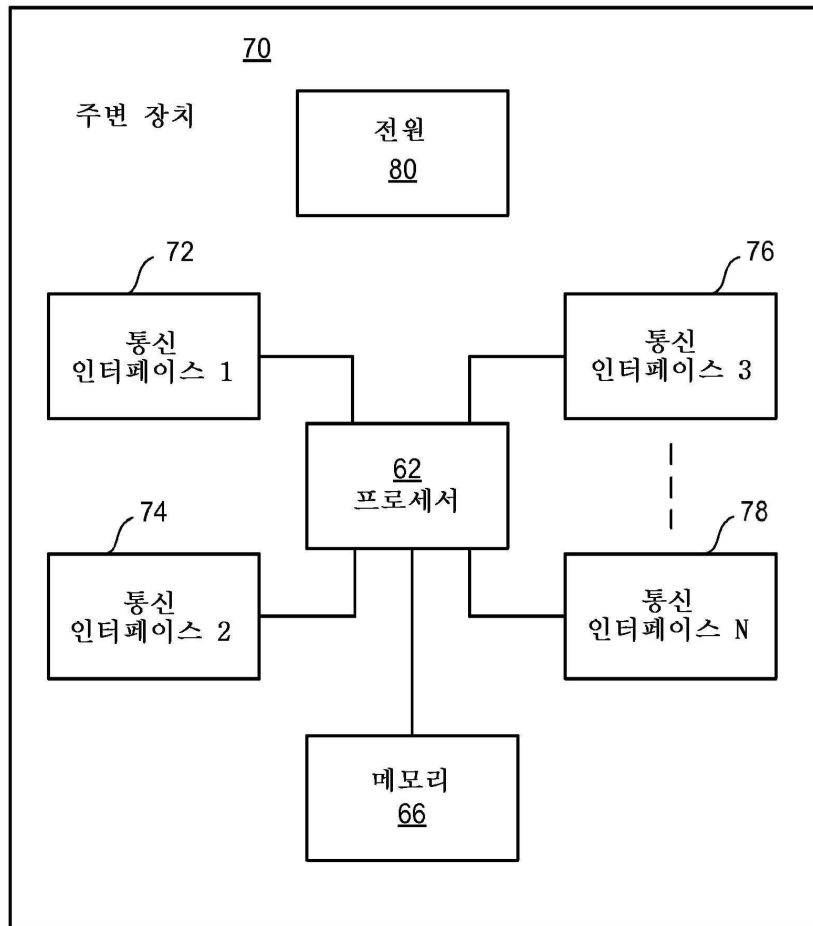
도면5



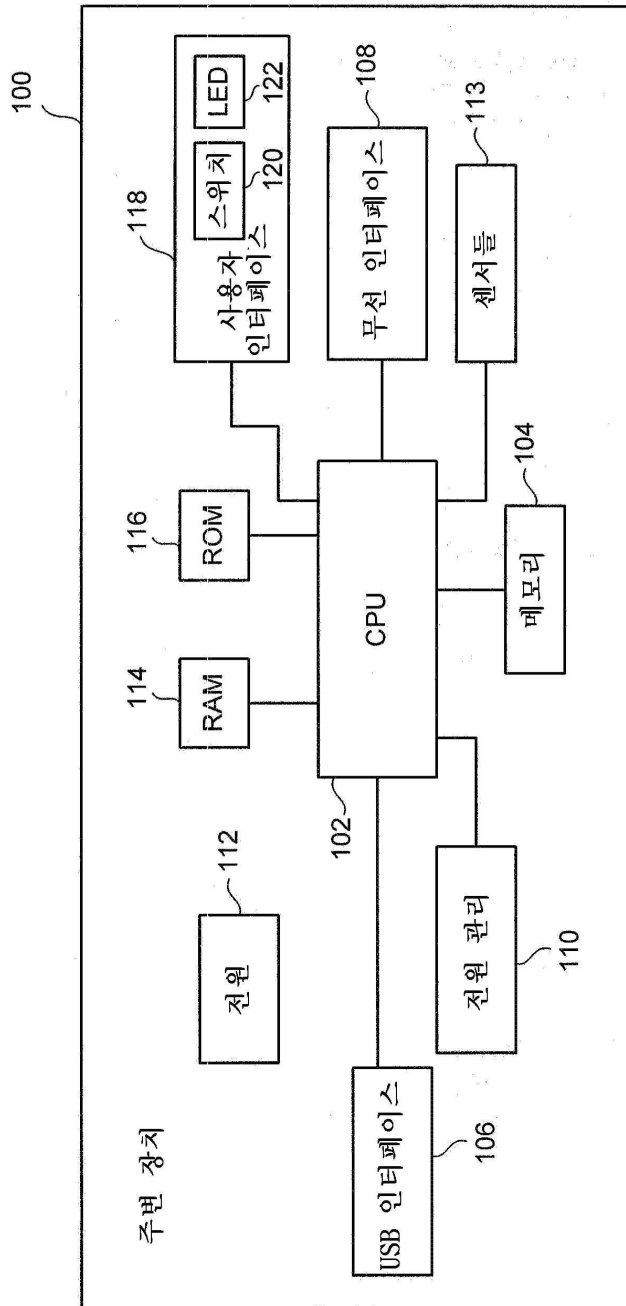
도면6



도면7



도면8

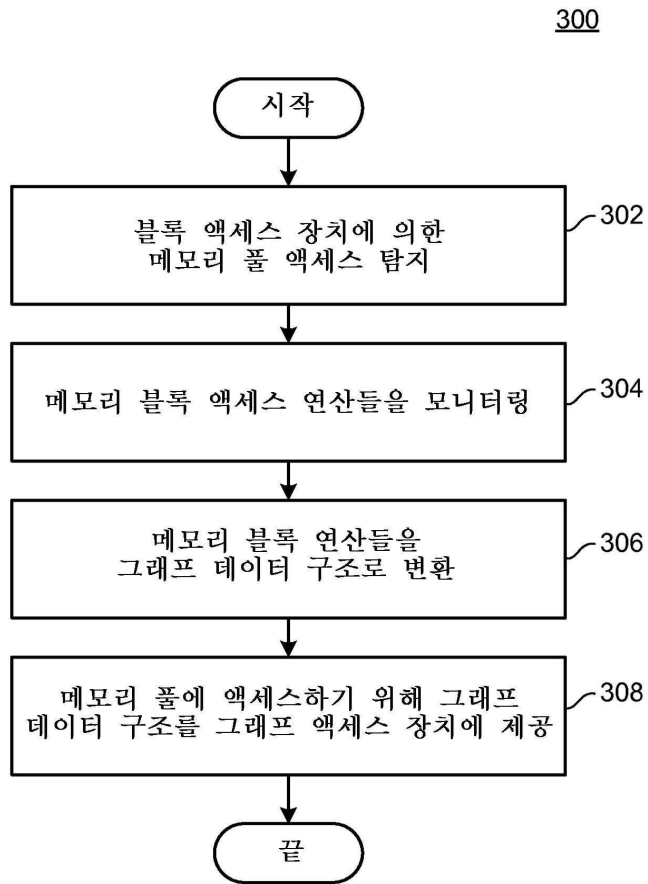


도면9

250

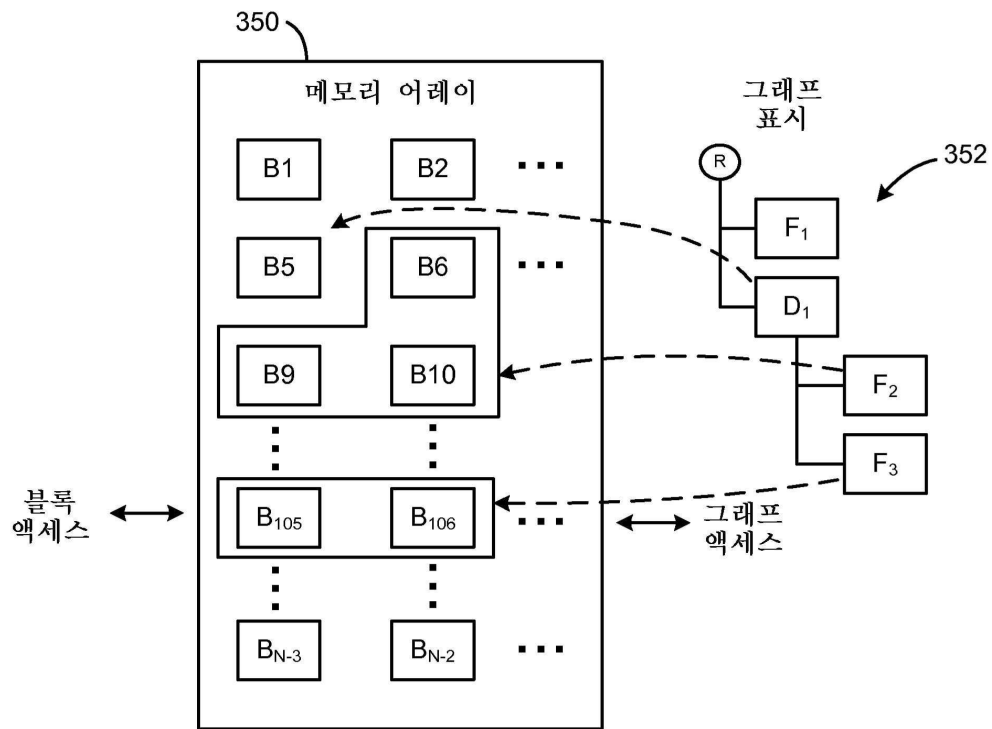
252					컨테스트 인식 상태 머신 (보안, 암호화/암호해독, 액세스 제어, 통신 채널 제어 등)				
254	USB 대용량 저장	256 펌웨어 업데이트	258 그래프 액세스 서버들 (mDNS, HTTP, FTP 등)	260	262	264	전원 관리 및 구성	282 사용자 인터페이스 (LED, 스위치)	284 배터리 관리
				Wifi 구성	TCP/IP				
		266 FAT	268 Wifi 드라이버들						
		270 SD 메모리	272 SDIO	274 타이머 서비스들					
		278 SD 하드웨어 버스		280 센서 (RTC, 온도, 위치 등)					
276	USB 하드웨어 버스								

도면10

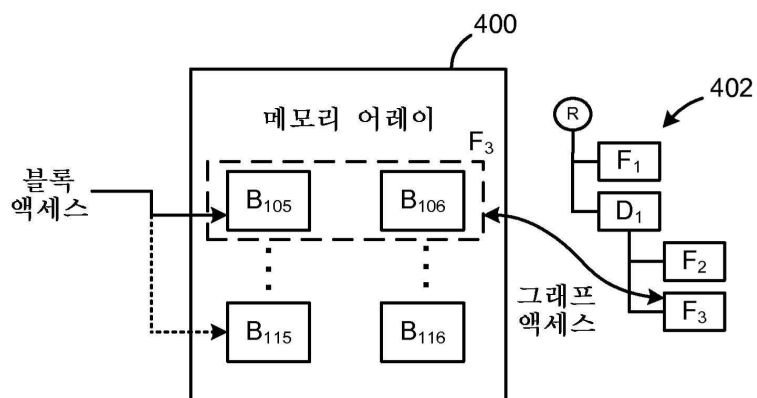




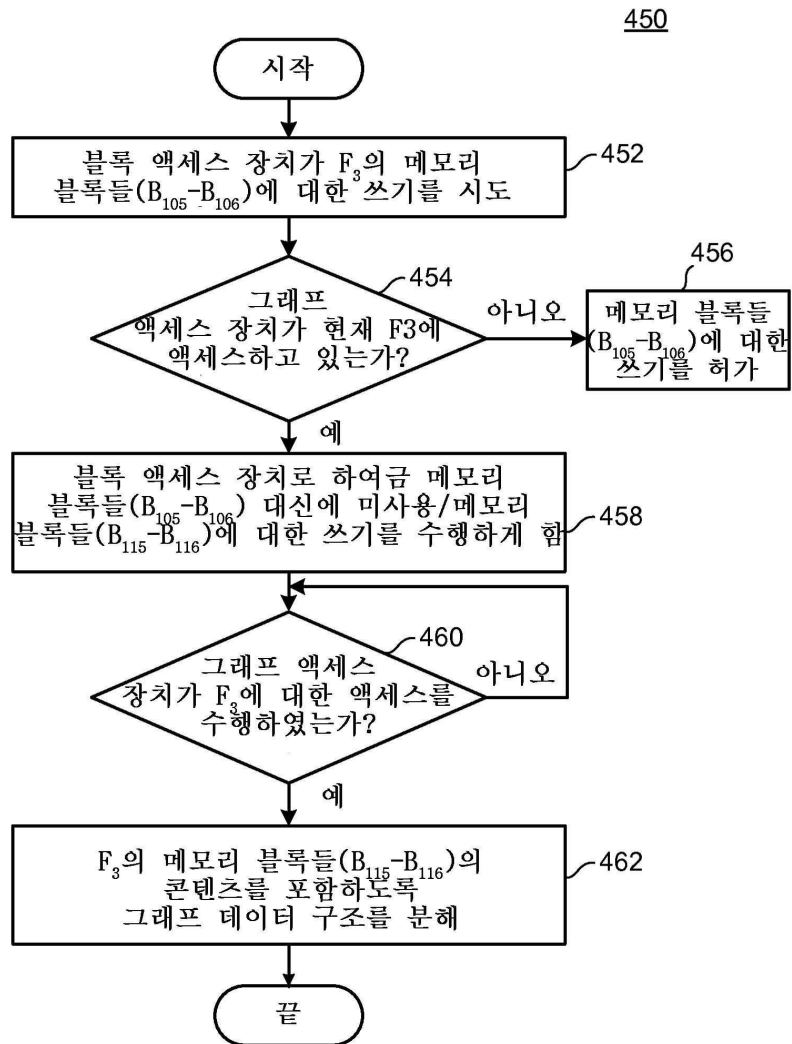
도면11



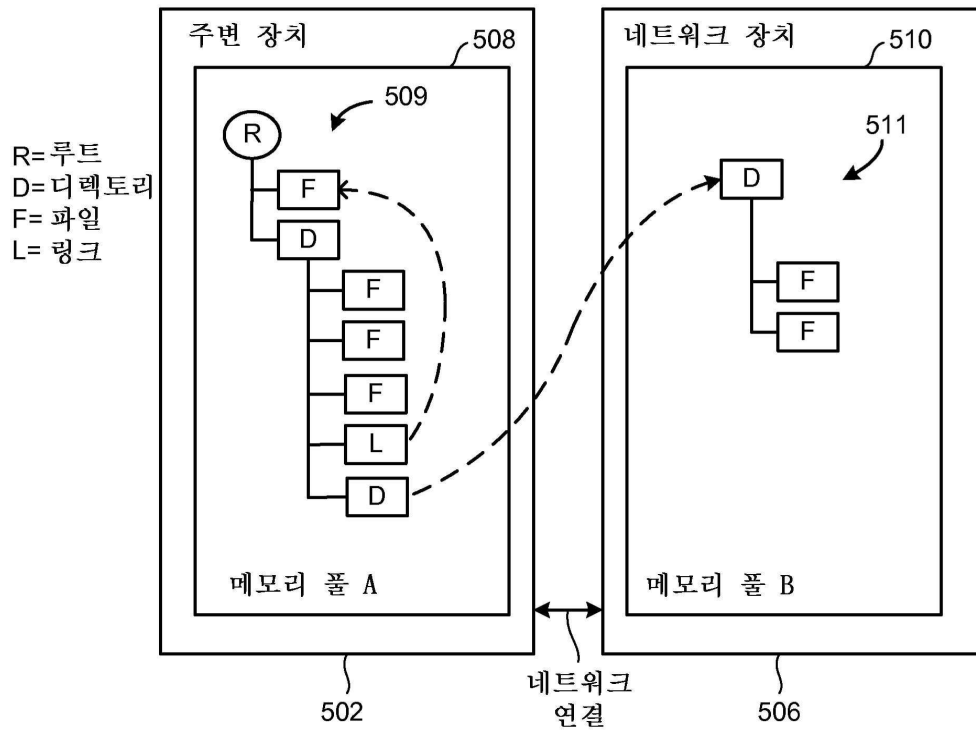
도면12



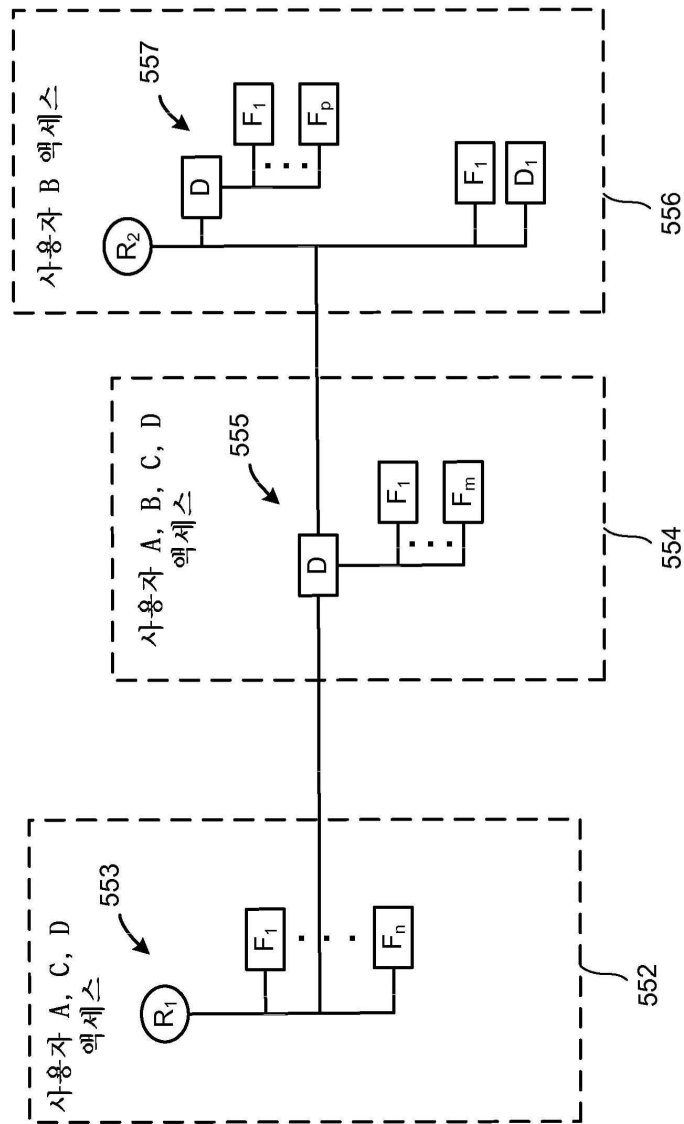
도면13



도면14



도면15

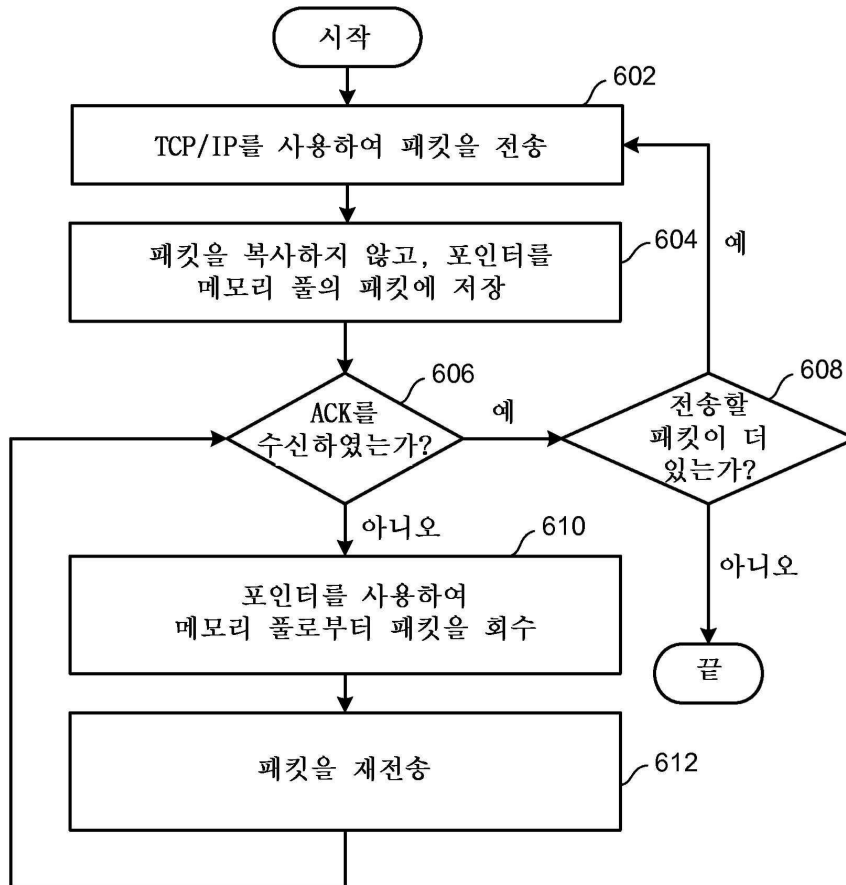


550



도면16

600



도면17

