



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2009-0034322  
(43) 공개일자 2009년04월07일

(51) Int. Cl.

H04L 12/42 (2006.01) H04L 29/02 (2006.01)

(21) 출원번호 10-2008-7031853

(22) 출원일자 2008년12월29일

심사청구일자 없음

번역문제출일자 2008년12월29일

(86) 국제출원번호 PCT/US2007/008036

국제출원일자 2007년03월29일

(87) 국제공개번호 WO 2008/005086

국제공개일자 2008년01월10일

(30) 우선권주장

11/428,133 2006년06월30일 미국(US)

(71) 출원인

마이크로소프트 코포레이션

미국 워싱턴주 (우편번호 : 98052) 레드몬드 원  
마이크로소프트 웨이

(72) 발명자

카키바야, 고타라 크리스나, 알.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로  
소프트 웨이

쑤, 루

미국 98052-6399 워싱턴주 레드몬드 원 마이크로  
소프트 웨이

하샤, 리차드, 엘.

미국 98052-6399 워싱턴주 레드몬드 원 마이크로  
소프트 웨이

(74) 대리인

양영준, 백만기

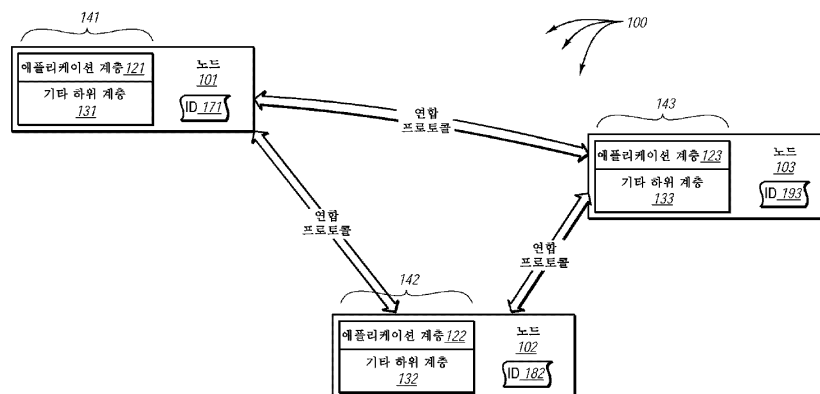
전체 청구항 수 : 총 39 항

(54) 랑데뷰 연합 내에서의 근접시간 통신을 위한 방법, 컴퓨터 판독가능 저장 매체 및 컴퓨터 프로그램 제품

### (57) 요약

본 발명은 랑데뷰 연합 내에서 근접시간 통신을 용이하게 해주는 방법, 시스템, 및 컴퓨터 프로그램 제품으로 확장된다. 노드는 방계 링 및 방계 링으로의 대응하는 진입 노드를 포함하는 방계 링 세트 엔트리 테이블을 유지한다. 노드는 링들로 이루어진 트리 내의 링들의 구성에 관해 서로를 갱신하기 위해 방계 링 세트 엔트리 상태를 교환할 수 있다. 노드는 노드의 방계 링인 링들의 진입 노드를 식별하기 위해 방계 링 세트 엔트리 테이블은 물론 다른 노드들을 참조할 수 있다. 메시지가 방계 링 내의 진입 노드로 전송될 수 있다. 메시지는 타겟 근접 링 내의 진입 노드가 메시지를 해석하여 표시된 목적지 노드에 가장 가까운 노드 ID를 갖는 타겟 근접 링 내의 노드로 보내야 한다는 표시를 포함할 수 있다.

### 대표도



## 특허청구의 범위

### 청구항 1

컴퓨터 시스템에서, 링들로 이루어진 트리(1900)에서 근접지간 통신(inter-proximity communication)을 전송하는 방법으로서,

노드(1311)가 상기 노드(1311)의 지정된 방계 링(21)으로 메시지(1903)를 전송할 것인지를 판정하는 단계,

상기 노드(1311)가 상기 노드(1311)에 대한 방계 링 세트 엔트리를 저장하도록 구성된 방계 링 세트 엔트리 테이블(1904)에 액세스하는 단계 - 각각의 방계 링 세트 엔트리는 상기 노드(1311)의 방계 링(21) 및 상기 노드(1311)의 상기 방계 링(21)으로의 대응하는 적어도 하나의 진입 노드(6521)를 나타내도록 구성되어 있음 -,

상기 노드의 방계 링 세트 엔트리 테이블(1904)로부터 상기 지정된 방계 링(21)에 대한 적어도 하나의 방계 링 세트 엔트리를 식별하는 단계 - 상기 적어도 하나의 방계 링 세트 엔트리 각각은 상기 지정된 방계 링(21)의 적어도 하나의 진입 노드(6521)를 나타냄 -, 및

상기 메시지(1903)를 적어도 하나의 표시된 진입 노드(6521)로 전송하는 단계를 포함하는, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

### 청구항 2

제1항에 있어서, 노드가 상기 노드의 지정된 방계 링으로 메시지를 전송할 것인지를 판정하는 단계는,

상기 노드와 관련된 애플리케이션으로부터 표시를 수신하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

### 청구항 3

제1항에 있어서, 노드가 상기 노드의 지정된 방계 링으로 메시지를 전송할 것인지를 판정하는 단계는,

상기 링들로 이루어진 트리에서의 다른 노드로부터 표시를 수신하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

### 청구항 4

제1항에 있어서, 노드가 지정된 방계 링으로 메시지를 전송할 것인지를 판정하는 단계는,

상기 노드가 하나 이상의 방계 링/진입 노드 항목에 액세스하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

### 청구항 5

제1항에 있어서, 상기 노드의 방계 링 세트 엔트리 테이블로부터 상기 지정된 방계 링에 대한 적어도 하나의 방계 링 세트 엔트리를 식별하는 단계는,

상기 지정된 방계 링 및 상기 지정된 방계 링으로의 복수의 진입 노드를 식별해주는 방계 링/진입 노드 항목을 식별하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

### 청구항 6

제5항에 있어서, 상기 복수의 진입 노드를 적절한 진입 노드의 서브셋으로 해석하는 단계를 더 포함하는, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

### 청구항 7

제5항에 있어서, 상기 복수의 진입 노드를 단 하나의 적절한 진입 노드로 해석하는 단계를 더 포함하는, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

### 청구항 8

컴퓨터 시스템에서, 링들로 이루어진 트리(1900)에서 근접시간 통신을 전송하는 방법으로서,

발신측 노드가 메시지(1998)를 링들로 이루어진 트리(1900) 내에서 타겟 근접 링(1221) 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 하는지를 노드(1311)가 판정하는 단계,

상기 타겟 근접 링(1221) 및 상기 타겟 근접 링(1221)의 선조 링(221) 중 적어도 하나의 링의 멤버 노드인 것으로 알려진 하나 이상의 진입 노드(41221)를 식별하는 단계, 및

상기 메시지(1998)를 식별된 진입 노드(41221)로 전송하는 단계 - 상기 메시지(1998)는 상기 식별된 진입 노드(41221)가 상기 메시지(1998)를 해석하여 상기 타겟 근접 링(1221) 내의 표시된 목적지 노드에 가장 가까운 노드 ID를 갖는 노드로 보내야 함을 나타냄 - 을 포함하는, 링들로 이루어진 트리에서 근접시간 통신을 전송하는 방법.

#### 청구항 9

제8항에 있어서, 발신측 노드가 메시지를 링들로 이루어진 트리 내에서 타겟 근접 링 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 하는지를 노드가 판정하는 단계는,

발신측 노드가 메시지를 링들로 이루어진 트리 내에서 타겟 근접 링 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 한다는 표시에 상기 노드가 액세스하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접시간 통신을 전송하는 방법.

#### 청구항 10

제9항에 있어서, 발신측 노드가 메시지를 링들로 이루어진 트리 내에서 타겟 근접 링 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 한다는 표시에 상기 노드가 액세스하는 단계는,

송신측 노드가 메시지를 상기 송신측 노드의 방계 링인 타겟 근접 링 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 한다는 표시를 상기 노드가 수신하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접시간 통신을 전송하는 방법.

#### 청구항 11

제9항에 있어서, 발신측 노드가 메시지를 링들로 이루어진 트리 내에서 타겟 근접 링 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 한다는 표시에 상기 노드가 액세스하는 단계는,

송신측 노드가 메시지를 상기 송신측 노드의 방계 링의 서브-링인 타겟 근접 링 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 한다는 표시를 상기 노드가 수신하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접시간 통신을 전송하는 방법.

#### 청구항 12

제9항에 있어서, 발신측 노드가 메시지를 상기 노드의 방계 링인 타겟 근접 링 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 한다는 표시에 액세스하는 단계는,

상기 노드에 관련된 애플리케이션으로부터 표시를 수신하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접시간 통신을 전송하는 방법.

#### 청구항 13

제9항에 있어서, 발신측 노드가 메시지를 상기 노드의 방계 링인 타겟 근접 링 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 한다는 표시에 액세스하는 단계는,

다른 메시지로 표시를 수신하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접시간 통신을 전송하는 방법.

#### 청구항 14

제13항에 있어서, 다른 메시지로 표시를 수신하는 단계는,

링들로 이루어진 트리에서 다른 노드로부터의 메시지로 표시를 수신하는 단계를 포함하는 것인, 링들로 이루어

진 트리에서 근접지간 통신을 전송하는 방법.

#### 청구항 15

제9항에 있어서, 발신측 노드가 메시지를 링들로 이루어진 트리 내에서 타겟 근접 링 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 한다는 표시에 노드가 액세스하는 단계는,

발신측 노드가 메시지를 상기 발신측 노드의 방계 링 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 한다는 표시를 수신하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

#### 청구항 16

제9항에 있어서, 발신측 노드가 메시지를 링들로 이루어진 트리 내에서 타겟 근접 링 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 한다는 표시에 액세스하는 단계는,

발신측 노드가 메시지를 상기 발신측 노드의 방계 링의 서브-링 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 한다는 표시를 수신하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

#### 청구항 17

제8항에 있어서, 발신측 노드가 메시지를 링들로 이루어진 트리 내에서 타겟 근접 링 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 하는지를 노드가 판정하는 단계는,

상기 발신측 노드가 메시지를 링들로 이루어진 트리 내에서 타겟 근접 링 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 하는지를 상기 발신측 노드가 판정하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

#### 청구항 18

제8항에 있어서, 상기 타겟 근접 링 및 상기 타겟 근접 링의 선조 링 중 적어도 하나의 링의 멤버 노드인 것으로 알려진 하나 이상의 진입 노드를 식별하는 단계는,

상기 타겟 근접 링 내의 임의의 노드를 식별하기 위해 상기 노드에 있는 로컬 지식을 참조하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

#### 청구항 19

제8항에 있어서, 상기 타겟 근접 링 및 상기 타겟 근접 링의 선조 링 중 적어도 하나의 링의 멤버 노드인 것으로 알려진 하나 이상의 진입 노드를 식별하는 단계는,

상기 타겟 근접 링의 멤버인 진입 노드를 식별하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

#### 청구항 20

제8항에 있어서, 상기 타겟 근접 링 및 상기 타겟 근접 링의 선조 링 중 적어도 하나의 링의 멤버 노드인 것으로 알려진 하나 이상의 진입 노드를 식별하는 단계는,

상기 타겟 근접 링의 선조 링의 멤버인 진입 노드를 식별하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

#### 청구항 21

제20항에 있어서, 상기 타겟 근접 링의 선조 링의 멤버인 진입 노드를 식별하는 단계는,

상기 타겟 근접 링의 가장 가까운 선조 링인 상기 타겟 근접 링의 선조 링의 멤버인 진입 노드를 식별하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

#### 청구항 22

제8항에 있어서, 상기 타겟 근접 링 및 상기 타겟 근접 링의 선조 링 중 적어도 하나의 링의 멤버 노드인 것으로 알려진 하나 이상의 진입 노드를 식별하는 단계는,

상기 노드가 상기 타겟 근접 링의 선조 링인 송신측 노드의 방계 링 세트 엔트리 테이블 내의 노드를 식별하기 위해 방계 링 세트 엔트리 테이블을 참조하는 단계, 및

상기 메시지를 상기 선조 링 내의 상기 식별된 노드로 전송하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

#### 청구항 23

제22항에 있어서, 상기 선조 링 내의 상기 노드가 상기 타겟 근접 링 내의 임의의 노드를 식별하기 위해 로컬 지식을 참조하는 단계를 더 포함하는, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

#### 청구항 24

제22항에 있어서, 상기 선조 링 내의 노드가 상기 타겟 근접 링의 가장 가까운 선조 링에 있는 상기 식별된 노드의 방계 링 세트 내의 추가의 노드를 식별하기 위해 방계 링 세트 엔트리 테이블을 참조하는 단계, 및

상기 메시지를 상기 추가의 노드로 전송하는 단계를 더 포함하는, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

#### 청구항 25

제8항에 있어서, 상기 타겟 근접 링 및 상기 타겟 근접 링의 선조 링 중 적어도 하나의 링의 멤버 노드인 것으로 알려진 하나 이상의 진입 노드를 식별하는 단계는,

상기 노드가 상기 타겟 근접 링의 멤버 노드인 것으로 알려진 진입 노드를 식별하기 위해 진입 노드 디렉토리 메카니즘을 이용하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

#### 청구항 26

제25항에 있어서, 상기 송신측 노드가 상기 타겟 근접 링의 멤버 노드인 것으로 알려진 진입 노드를 식별하기 위해 진입 노드 디렉토리 메카니즘을 이용하는 단계는,

진입 노드 탐색 요청 메시지를 랑데뷰 지점으로 라우팅하는 단계, 및

상기 탐색 요청에 대응하는 진입 노드 탐색 응답을 수신하는 단계 - 상기 진입 노드 탐색 응답은 복수의 진입 노드를 포함함 - 를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

#### 청구항 27

제26항에 있어서, 상기 탐색 요청에 대응하는 진입 노드 탐색 응답을 수신하는 단계는,

상기 랑데뷰 지점으로부터 진입 노드 탐색 응답을 수신하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

#### 청구항 28

제26항에 있어서, 상기 탐색 요청에 대응하는 진입 노드 탐색 응답을 수신하는 단계는,

진입 노드 디렉토리 메카니즘에 등록되어 있는 하나 이상의 진입 노드를 포함하는 진입 노드 탐색 응답을 수신하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

#### 청구항 29

제28항에 있어서, 진입 노드 디렉토리 메카니즘에 등록되어 있는 하나 이상의 진입 노드를 포함하는 진입 노드 탐색 응답을 수신하는 단계는,

상기 랑데뷰 지점에 등록되어 있는 하나 이상의 진입 노드를 포함하는 진입 노드 탐색 응답을 수신하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

### 청구항 30

제28항에 있어서, 진입 노드 디렉토리 메카니즘에 등록되어 있는 하나 이상의 진입 노드를 포함하는 진입 노드 탐색 응답을 수신하는 단계는,

상기 랑데뷰 지점에 그 자신을 등록한 진입 노드를 포함하는 진입 노드 탐색 응답을 수신하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

### 청구항 31

제28항에 있어서, 진입 노드 디렉토리 메카니즘에 등록되어 있는 하나 이상의 진입 노드를 포함하는 진입 노드 탐색 응답을 수신하는 단계는,

다른 당사자에 의해 상기 진입 노드 디렉토리 메카니즘에 등록된 진입 노드를 포함하는 진입 노드 탐색 응답을 수신하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

### 청구항 32

제31항에 있어서, 다른 당사자에 의해 상기 진입 노드 디렉토리 메카니즘에 등록된 진입 노드를 포함하는 진입 노드 탐색 응답을 수신하는 단계는,

다른 랑데뷰 지점에 의해 상기 진입 노드 디렉토리 메카니즘에 등록된 진입 노드를 포함하는 진입 노드 탐색 응답을 수신하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

### 청구항 33

제26항에 있어서, 진입 노드에 대한 진입 노드 등록 요청을 수신하는 단계, 및

상기 진입 노드를 상기 랑데뷰 지점에 등록하는 단계를 더 포함하는, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법.

### 청구항 34

컴퓨터 시스템에서, 링들로 이루어진 트리(1900)에서 근접지간 통신(inter-proximity communication)을 전송하는 방법을 구현하기 위해 사용하는 컴퓨터 프로그램 제품으로서,

상기 컴퓨터 프로그램 제품은 프로세스에 의해 실행될 때 노드로 하여금 상기 방법을 수행하게 하는 컴퓨터 실행가능 명령어를 저장하고 있는 하나 이상의 컴퓨터 판독가능 저장 매체를 포함하며,

상기 방법은,

노드(1311)가 상기 노드(1311)의 지정된 방계 링(21)으로 메시지(1903)를 전송할 것인지를 판정하는 단계,

상기 노드에 대한 방계 링 세트 엔트리를 저장하도록 구성된 방계 링 세트 엔트리 테이블(1904)에 액세스하는 단계 - 각각의 방계 링 세트 엔트리는 상기 노드(1311)의 방계 링(21) 및 상기 노드(1311)의 상기 방계 링(21)으로의 대응하는 적어도 하나의 진입 노드(6521)를 나타내도록 구성되어 있음 -,

상기 노드의 방계 링 세트 엔트리 테이블(1904)로부터 상기 지정된 방계 링(21)에 대한 적어도 하나의 방계 링 세트 엔트리를 식별하는 단계 - 상기 적어도 하나의 방계 링 세트 엔트리 각각은 상기 지정된 방계 링(21)의 적어도 하나의 진입 노드(6521)를 나타냄 -, 및

상기 메시지(1903)를 적어도 하나의 표시된 진입 노드(6521)로 전송하는 단계를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법을 구현하기 위해 사용하는 컴퓨터 프로그램 제품.

### 청구항 35

제34항에 있어서, 상기 컴퓨터 판독가능 저장 매체가 시스템 메모리를 포함하는 것인, 링들로 이루어진 트리에서 근접지간 통신을 전송하는 방법을 구현하기 위해 사용하는 컴퓨터 프로그램 제품.

### 청구항 36

제34항에 있어서, 상기 컴퓨터 판독가능 저장 매체가 자기 디스크를 포함하는 것인, 링들로 이루어진 트리에서

근접시간 통신을 전송하는 방법을 구현하기 위해 사용하는 컴퓨터 프로그램 제품.

### 청구항 37

컴퓨터 시스템에서, 링들로 이루어진 트리(1900)에서 근접시간 통신을 전송하는 방법을 구현하기 위해 사용하는 컴퓨터 프로그램 제품으로서,

상기 컴퓨터 프로그램 제품은 프로세스에 의해 실행될 때 노드로 하여금 상기 방법을 수행하게 하는 컴퓨터 실행가능 명령어를 저장하고 있는 하나 이상의 컴퓨터 판독가능 저장 매체를 포함하며,

상기 방법은,

발신측 노드가 메시지(1998)를 링들로 이루어진 트리(1900) 내에서 타겟 근접 링(1221) 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 라우팅하려고 하는지를 판정하는 단계,

상기 타겟 근접 링(1221) 및 상기 타겟 근접 링(1221)의 선조 링(221) 중 적어도 하나의 링의 멤버 노드인 것으로 알려진 하나 이상의 진입 노드(41221)를 식별하는 단계, 및

상기 메시지(1998)를 식별된 진입 노드(41221)로 전송하는 단계 - 상기 메시지(1998)는 상기 식별된 진입 노드(41221)가 상기 메시지(1998)를 해석하여 상기 타겟 근접 링(1221) 내의 표시된 목적지 노드에 가장 가까운 노드 ID를 갖는 노드로 보내야 함을 나타냄 - 를 포함하는 것인, 링들로 이루어진 트리에서 근접시간 통신을 전송하는 방법을 구현하기 위해 사용하는 컴퓨터 프로그램 제품.

### 청구항 38

제37항에 있어서, 상기 컴퓨터 판독가능 저장 매체가 시스템 메모리를 포함하는 것인, 링들로 이루어진 트리에서 근접시간 통신을 전송하는 방법을 구현하기 위해 사용하는 컴퓨터 프로그램 제품.

### 청구항 39

제37항에 있어서, 상기 컴퓨터 판독가능 저장 매체가 자기 디스크를 포함하는 것인, 링들로 이루어진 트리에서 근접시간 통신을 전송하는 방법을 구현하기 위해 사용하는 컴퓨터 프로그램 제품.

## 명세서

### 배경 기술

- <1> 컴퓨터 시스템 및 관련 기술이 사회의 많은 측면들에 영향을 주고 있다. 실제로, 컴퓨터 시스템의 정보 처리 능력은 우리가 살아가고 일하는 방식을 변모시켰다. 이제는 컴퓨터 시스템이 컴퓨터 시스템의 등장 이전에 수작업으로 수행되었던 수많은 작업들(예를 들어, 워드 프로세싱, 스케줄링, 및 데이터베이스 관리)을 수행하는 것이 보통이다. 보다 최근에는, 컴퓨터 시스템들이 서로 또한 기타 전자 장치들에 결합되어 유선 및 무선 컴퓨터 네트워크를 형성하였으며, 이 컴퓨터 네트워크를 통해 컴퓨터 시스템들 및 기타 전자 장치들이 전자 데이터를 전송할 수 있다. 그 결과, 컴퓨터 시스템에서 수행되는 많은 작업들(예를 들어, 음성 통신, 전자 메일에 액세스하는 것, 가전 제품들을 제어하는 것, 웹 브라우징, 및 문서 인쇄)은 유선 및/또는 무선 컴퓨터 네트워크를 통한 다수의 컴퓨터 시스템들 및/또는 기타 전자 장치들 간의 전자 메시지의 교환을 포함한다.
- <2> 그렇지만, 네트워크 자원을 이용하여 컴퓨터화된 작업을 수행하기 위해, 컴퓨터 시스템은 네트워크 자원을 식별하고 그에 액세스하는 어떤 방법을 가지고 있어야만 한다. 그에 따라, 자원들은 통상적으로 자원들을 일의적으로 식별해줌과 동시에 자원들을 서로 구별하는 데 사용될 수 있는 고유의 식별자, 예를 들어, 네트워크 주소를 할당받는다. 따라서, 자원을 이용하고자 하는 컴퓨터 시스템은 자원에 대응하는 네트워크 주소를 사용하여 자원에 접속할 수 있다. 그렇지만, 컴퓨터 시스템이 네트워크 자원의 네트워크 주소를 미리 알고 있지 않은 경우, 네트워크 자원에 액세스하는 일이 어려울 수 있다. 예를 들어, 컴퓨터 시스템(또는 다른 네트워크화된 컴퓨터 시스템)이 네트워크 프린터의 네트워크 주소를 알고 있지 않은 한, 컴퓨터 시스템은 네트워크 프린터에서 문서를 인쇄할 수 없다.
- <3> 그에 따라, 컴퓨터 시스템이 이전에 모르고 있던 자원들을 식별(하여 액세스)하기 위한 다양한 메카니즘들(예를 들어, DNS(Domain Name System), AD(Active Directory), DFS(Distributed File System))이 개발되었다. 그렇지만, 서로 다른 컴퓨터 네트워크를 통해 액세스가능한 자원들(예를 들어, 장치들 및 서비스들)이 많고 다양하

기 때문에, 개발자들은 종종 각종의 서로 다른 자원 식별 및 액세스 메카니즘을 구현하는 애플리케이션들을 개발해야만 한다. 각각의 서로 다른 메카니즘이 서로 다른 코딩 요건을 가질 수 있으며, 애플리케이션에 필요한 모든 기능을 개발자에게 제공하는 것은 아니다.

<4> 예를 들어, DNS가 분산 관리 구조를 갖지만(즉, 중앙집중식 관리가 요구되지는 않지만), DNS는 충분히 동적이지 않고, 자기-구성적(self-organizing)이지 않으며, 약한 데이터 및 질의 모델을 지원하고, 일정한 일련의 루트(root)를 갖는다. 반면에, AD는 충분히 동적이지만 중앙집중식 관리를 필요로 한다. 게다가, 서로 다른 메카니즘의 측면들이 서로 호환되지 않을 수도 있다. 예를 들어, DNS를 사용하여 식별된 자원이 DFS 라우팅 프로토콜과 호환되지 않을지도 모른다. 따라서, 개발자는 어쩔 수 없이 가장 적당한 메카니즘을 선택하고 다른 메카니즘들의 이점을 포기해야만 할지도 모른다.

<5> 자원을 식별하는 메카니즘은 피어-투-피어 네트워크에서 특히 문제가 될 수 있다. DNS는 호스트 이름을 키(key)로 갖고 IP 주소를 값(value)으로 갖는 탐색 서비스(lookup service)를 제공하며, 이 탐색 서비스는 탐색 요청을 구현하기 위해 일련의 특별한 루트 서버(root server)에 의존한다. 게다가, DNS는 클라이언트들이 이름 서버 계층구조(name server hierarchy)를 순회할 수 있게 해주기 위한 정보(NS 레코드)의 관리를 필요로 한다. 따라서, 자원이 네트워크 상에서 식별될 수 있기 이전에, 자원이 DNS에 등록되어야만 한다. 정보의 입력에 의존하여 노드들이 빈번히 네트워크에 접속되고 그로부터 분리되는 대규모 네트워크가 항상 가능한 것은 아니다. 그에 부가하여, DNS는 호스트 또는 서비스를 찾아내는 일을 전담하고 있으며, 일반적으로 다른 유형의 자원에는 적용가능하지 않다.

<6> 그에 따라, 이들 단점을 해소하기 위해 자원 식별 및 액세스를 위한 다른 메카니즘이 개발되었다. 다수의 메카니즘은 DNS보다 더 확장성이 있는 분산 탐색 프로토콜(distributed lookup protocol)을 포함한다. 이들 메카니즘은 다양한 노드 배열 및 라우팅 알고리즘을 사용하여, 요청들을 대응하는 자원으로 라우팅하고 탐색을 위한 정보를 저장한다.

<7> 이들 메카니즘 중 적어도 하나는 네트워크 내의 각각의 노드에서 로컬 다중-레벨 이웃 맵(local multi-level neighbor map)을 이용하여 메시지들을 목적지 노드로 라우팅한다. 이렇게 함으로써 본질적으로 각각의 노드가 대응하는 노드 트리(그 노드의 이웃 맵에 있는 노드들의 트리)의 "루트 노드"인 아키텍처가 얻어진다. 메시지는 한 숫자씩(예를 들어, \*\*\*6 => \*\*46 => \*346 => 2346, 여기서 \*는 와일드카드를 나타냄) 점진적으로 목적지 ID로 라우팅된다. 이들 유형의 메카니즘의 라우팅 효율은  $O(\log N)$  라우팅 홉이고, 노드가  $O(\log N)$  크기의 라우팅 테이블을 유지할 것을 요구한다.

<8> 이들 메카니즘 중 적어도 하나의 다른 메카니즘은 노드들에 숫자들의 선형 링에서 가져온 고유 ID를 할당한다. 노드들은 (ID 값에 따른) 직계 후손 노드에 대한 포인터 및 ID 값이 값  $ID + 2^L$ 의 가장 가까운 후손인 노드에 대한 포인터를 포함하는 라우팅 테이블을 유지한다. 이들 유형의 메카니즘의 라우팅 효율도 역시  $O(\log N)$  라우팅 홉이고, 노드가  $O(\log N)$  크기의 라우팅 테이블을 유지할 것을 요구한다.

<9> 적어도 하나의 다른 메카니즘은  $O(\log N^{1/d})$  라우팅 홉을 필요로 하며, 노드가  $O(D)$  크기의 라우팅 테이블을 유지할 것을 요구한다. 따라서, 모든 이러한 메카니즘들의 라우팅 효율은, 적어도 부분적으로, 시스템에서의 노드의 수에 의존한다.

<10> 게다가, (이들 메카니즘 중 적어도 어떤 메카니즘의 경우) ID가 링을 따라 균일하게 분포될 수 있기 때문에, 링 상의 노드들 간의 라우팅이 얼마간 비효율적일 가능성이 항상 있다. 예를 들어, 라우팅 홉이 광대한 지리적 거리를 지나가거나, 비용이 많이 드는 링크를 통과하거나, 안전하지 않은 도메인을 통과하거나 기타 등등을 할 수 있다. 그에 부가하여, 메시지 라우팅이 다수의 홉을 필요로 하는 경우, 이러한 일들이 여러번 일어날 가능성이 있다. 불행히도, 이들 메카니즘은 노드들의 서로에 대한 (물리적 또는 기타) 근접성을 고려하지 않는다. 예를 들어, 링 상의 노드 분포에 따라, 뉴욕에서 보스턴으로 메시지를 라우팅하는 것은 그 메시지를 뉴욕에서 런던으로, 아틀란타로, 도쿄로, 그리고 나서 보스턴으로 라우팅하는 것을 수반할 수 있다.

<11> 그에 따라, 적어도 하나의 다른 보다 최근의 메카니즘은 근접성을 하나의 스칼라 근접성 메트릭(예를 들어, IP 라우팅 홉 또는 지리적 거리)으로 정의함으로써 근접성을 고려하고 있다. 이들 메카니즘은 라우팅 테이블 엔트리의 근접성-기반 선택의 개념을 사용하고 있다. 각각의 라우팅 테이블 엔트리에 대한 많은 "적당한" 노드 후보가 있기 때문에, 이들 메카니즘은 후보 노드들 중에서 가장 가까운 노드를 선택하려고 한다. 이들 메카니즘은 각각의 노드가 주어진 IP 주소를 갖는 노드에서 그 자신까지의 "거리"를 측정할 수 있는 기능을 제공할 수 있다. 더 멀리 떨어져 있는 노드로 라우팅하기 전에 목적지 쪽으로 진행하기 위해 보다 근접해 있는 노드들 간



에 메시지들이 라우팅된다. 따라서, 얼마간의 자원이 절감될 수 있고 라우팅이 더 효율적이다.

- <12> 불행히도, 이들 기존의 메카니즘은 통상적으로, 그 중에서도 특히, 노드들 간의 대칭 관계(즉, 제1 노드가 제2 노드를 그의 파트너로서 생각하는 경우, 제2 노드도 제1 노드를 파트너로 생각함), 링 상에서 양 방향으로(시계 방향으로 또한 반시계 방향으로) 메시지를 라우팅하는 것, 복수의 근접성 메트릭에 기초하여 노드들의 연결 리스트를 분할하는 것, 및 복수의 근접성 메트릭에 기초하여 메시지를 라우팅하는 것을 제공하지 않는다. 이들 결점은, 예를 들어, 네트워크의 모든 노드들로 데이터를 브로드캐스트할 때와 같이, 네트워크의 노드들 간의 동적이고 분산된 효율적인 데이터 전송을 제한할 수 있다.

### 발명의 상세한 설명

- <13> 본 발명은 랑데뷰 연합(rendezvous federation) 내에서의 근접지간 통신(inter-proximity communication)을 용이하게 해주는 방법, 시스템, 및 컴퓨터 프로그램 제품으로 확장된다. 어떤 실시예들에서, 노드의 방계 링 세트 엔트리 테이블(collateral ring set entry table)이 유지된다. 노드는 노드에 대한 방계 링 세트 엔트리를 저장하도록 구성된 방계 링 세트 엔트리 테이블에 액세스한다. 각각의 방계 링 세트 엔트리는 노드의 방계 링 및 노드의 방계 링에 적어도 하나의 대응하는 진입 노드를 나타내도록 구성되어 있다. 링들의 트리의 구성에 관한 정보를 유지하는 이용가능한 자원으로부터의 방계 링 세트 엔트리 테이블 정보가 탐색된다. 탐색된 방계 링 세트 엔트리 테이블 정보에 기초하여 적절한 방계 링 세트 엔트리 상태를 사용하여 방계 링 세트 엔트리 테이블이 갱신된다. 이 적절한 방계 링 세트 엔트리 상태는 노드의 방계 링 및 노드의 방계 링으로의 적어도 하나의 대응하는 진입 노드를 포함한다.
- <14> 다른 실시예들에서, 근접지간 통신은 링들로 이루어진 트리를 따라 전송된다. 근접지간 통신의 일 실시예에서, 노드가 노드의 지정된 방계 링으로 메시지를 전송하는지가 판정된다. 노드는 노드에 대한 방계 링 세트 엔트리를 저장하도록 구성된 방계 링 세트 엔트리 테이블에 액세스한다. 각각의 방계 링 세트 엔트리는 노드의 방계 링 및 노드의 방계 링으로의 대응하는 적어도 하나의 진입 노드를 나타내도록 구성되어 있다. 지정된 방계 링에 대한 적어도 하나의 방계 링 세트 엔트리는 노드의 방계 링 세트 엔트리 테이블로부터 식별된다. 적어도 하나의 방계 링 세트 엔트리 각각은 지정된 방계 링의 적어도 하나의 진입 노드를 나타낸다. 메시지가 적어도 하나의 표시된 진입 노드로 전송된다.
- <15> 근접지간 통신의 다른 실시예에서, 발신측 노드가 링들의 트리 내에서 타겟 근접 링(target proximity ring) 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 메시지를 라우팅하려고 하는지가 판정된다. 타겟 근접 링 및 타겟 근접 링의 선조 링(ancestor ring) 중 적어도 하나의 링의 멤버 노드(member node)인 것으로 알려져 있는 하나 이상의 진입 노드가 식별된다. 메시지가 식별된 진입 노드로 전송된다. 메시지는 식별된 진입 노드가 메시지를 분석할 것임을 타겟 근접 링 내의 표시된 목적지 노드에 가장 가까운 노드 ID를 갖는 노드에 알려준다.
- <16> 이 요약은 이하에서 상세한 설명에 더 기술되는 선택된 개념들을 간략화된 형태로 소개하기 위해 제공된 것이다. 이 요약은 청구된 발명 대상의 주요 특징들 또는 필수적인 특징들을 확인하기 위한 것이 아니며, 청구된 발명 대상의 범위를 정하는 데 보조 수단으로 사용되기 위한 것도 아니다.
- <17> 부가의 특징들 및 이점들이 이하의 설명에 기술될 것이며, 부분적으로 이 설명으로부터 명백하게 되거나, 본 명세서의 개시 내용을 실시하면 알 수 있다. 본 발명의 특징들 및 이점들은 첨부된 청구항들에 상세히 지적된 수단 및 조합에 의해 실현되고 달성될 수 있다. 본 발명의 특징들은 이하의 설명으로부터 더 명백하게 되거나 이후에 기술된 바와 같이 본 발명을 실시함으로써 알 수 있다.

### 실시예

- <48> 상기한 이점들 및 특징들과 기타 이점들 및 특징들이 달성될 수 있는 방식을 설명하기 위해, 이상에서 간략히 기술한 발명 대상에 대한 보다 상세한 설명이 첨부 도면에 도시된 구체적인 실시예들을 참조하여 행해질 것이다. 이들 도면이 단지 대표적인 실시예만을 나타내고 있고 따라서 범위를 제한하는 것으로 생각되어서는 안된다는 것을 염두에 두면서, 첨부 도면을 사용하여 실시예들이 더 구체적이고 상세하게 기술되고 설명될 것이다.
- <49> 본 발명은 랑데뷰 연합(rendezvous federation) 내에서의 근접지간 통신(inter-proximity communication)을 용이하게 해주는 방법, 시스템, 및 컴퓨터 프로그램 제품으로 확장된다. 어떤 실시예들에서, 노드의 방계 링 세트 엔트리 테이블(collateral ring set entry table)이 유지된다. 노드는 노드에 대한 방계 링 세트 엔트리를 저장하도록 구성된 방계 링 세트 엔트리 테이블에 액세스한다. 각각의 방계 링 세트 엔트리는 노드의 방계 링

및 노드의 방계 링에의 적어도 하나의 대응하는 진입 노드를 나타내도록 구성되어 있다. 링들의 트리의 구성에 관한 정보를 유지하는 이용가능한 자원으로부터의 방계 링 세트 엔트리 테이블 정보가 탐색된다. 탐색된 방계 링 세트 엔트리 테이블 정보에 기초하여 적절한 방계 링 세트 엔트리 상태를 사용하여 방계 링 세트 엔트리 테이블이 갱신된다. 이 적절한 방계 링 세트 엔트리 상태는 노드의 방계 링 및 노드의 방계 링으로의 적어도 하나의 대응하는 진입 노드를 포함한다.

<50> 다른 실시예들에서, 근접시간 통신은 링들의 트리를 따라 전송된다. 근접시간 통신의 일 실시예에서, 노드가 노드의 지정된 방계 링으로 메시지를 전송하는지가 판정된다. 노드는 노드에 대한 방계 링 세트 엔트리를 저장하도록 구성된 방계 링 세트 엔트리 테이블에 액세스한다. 각각의 방계 링 세트 엔트리는 노드의 방계 링 및 노드의 방계 링으로의 대응하는 적어도 하나의 진입 노드를 나타내도록 구성되어 있다. 지정된 방계 링에 대한 적어도 하나의 방계 링 세트 엔트리는 노드의 방계 링 세트 엔트리 테이블로부터 식별된다. 적어도 하나의 방계 링 세트 엔트리 각각은 지정된 방계 링의 적어도 하나의 진입 노드를 나타낸다. 메시지가 적어도 하나의 표시된 진입 노드로 전송된다.

<51> 근접시간 통신의 다른 실시예에서, 발신측 노드가 링들의 트리 내에서 타겟 근접 링(target proximity ring) 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 메시지를 라우팅하려고 하는지가 판정된다. 타겟 근접 링 및 타겟 근접 링의 선조 링(ancestor ring) 중 적어도 하나의 링의 멤버 노드(member node)인 것으로 알려져 있는 하나 이상의 진입 노드가 식별된다. 메시지가 식별된 진입 노드로 전송된다. 메시지는 식별된 진입 노드가 메시지를 분석할 것임을 타겟 근접 링 내의 표시된 목적지 노드에 가장 가까운 노드 ID를 갖는 노드에 알려준다.

<52> 본 발명의 범위 내에 있는 실시예는 컴퓨터 실행가능 명령어 또는 데이터 구조를 담고 있거나 저장하기 위한 컴퓨터 판독가능 매체를 포함한다. 이러한 컴퓨터 판독가능 매체는 범용 또는 전용 컴퓨터 시스템에 의해 액세스 가능한 임의의 이용가능 매체일 수 있다. 예를 들어, 이러한 컴퓨터 판독가능 매체는 RAM, ROM, EPROM, CD-ROM 또는 기타 광 디스크 저장 장치, 자기 디스크 저장 장치 또는 기타 자기 저장 장치, 또는 컴퓨터 실행가능 명령어, 컴퓨터 판독가능 명령어 또는 데이터 구조의 형태로 된 원하는 프로그램 코드 수단을 담고 있거나 저장하는데 사용될 수 있고 범용 또는 전용 컴퓨터 시스템에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있지만, 이것들로 한정되는 것은 아니다.

<53> 이 설명 및 이하의 청구 범위에서, "네트워크"는 컴퓨터 시스템들 및/또는 모듈들(예를 들어, 하드웨어 및/또는 소프트웨어 모듈) 사이에서의 전자 데이터의 전송을 가능하게 하는 (어쩌면 속도가 다른) 하나 이상의 데이터 링크로서 정의된다. 정보가 네트워크 또는 다른 통신 접속(유선, 무선, 또는 유선이나 무선의 조합)을 통해 컴퓨터 시스템에 전송 또는 제공되는 경우, 그 접속은 컴퓨터 판독가능 매체로 보아도 무방하다. 따라서, 이러한 접속은 모두 컴퓨터 판독가능 매체라고 해도 무방하다. 상기한 것들의 조합도 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다. 컴퓨터 실행가능 명령어는, 예를 들어, 범용 컴퓨터 시스템 또는 전용 컴퓨터 시스템으로 하여금 일정한 기능 또는 일군의 기능을 실행하게 하는 명령어 및 데이터를 포함한다. 컴퓨터 실행가능 명령어는, 예를 들어, 바이너리, 어셈블리어 등의 중간 형식 명령어, 또는 원시 코드조차 될 수 있다. 몇몇 실시예에서, 예를 들어, 전용 집적회로 또는 게이트 어레이(gate array) 등의 하드웨어 모듈은 본 발명의 원리를 실시하기 위해 최적화된다.

<54> 이 설명 및 첨부된 청구 범위에서, "노드"는 전자 데이터에 대한 동작을 수행하기 위해서 협동하는 하나 이상의 소프트웨어 모듈, 하나 이상의 하드웨어 모듈, 또는 이들의 조합으로서 정의된다. 예를 들어, 노드의 정의는 퍼스널 컴퓨터의 운영 체제 등의 소프트웨어 컴포넌트 뿐만 아니라, 퍼스널 컴퓨터의 하드웨어 컴포넌트도 포함한다. 모듈의 물리적 레이아웃은 중요하지 않다. 노드는 네트워크를 통해 결합된 하나 이상의 컴퓨터를 포함할 수가 있다. 이와 마찬가지로, 노드는 전자 데이터에 대한 동작을 수행하기 위해서 (메모리 및 프로세서 등의) 내부 모듈이 협동하는 (이동 전화 또는 개인 휴대 단말기 "PDA" 등의) 단일의 물리 장치를 포함할 수가 있다. 게다가, 노드는, 예를 들어, 전용의 집적회로를 포함한 라우터 등, 전용의 하드웨어를 포함할 수가 있다.

<55> 당업자라면, 퍼스널 컴퓨터, 랩톱 컴퓨터, 핸드헬드 장치, 멀티 프로세서 시스템, 마이크로프로세서-기반 또는 프로그램가능 가전 제품, 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 이동 전화, PDA, 페이지, 라우터, 게이트웨이, 브로커(broker), 프록시, 방화벽(fire wall), 리디렉터(redirector), 및 네트워크 주소 변환기 등을 비롯한 많은 유형의 노드 구성을 갖는 네트워크 컴퓨팅 환경에서 본 발명이 실시될 수 있다는 것을 잘 알 것이다. 본 발명은, 네트워크를 통해 (유선 데이터 링크, 무선 데이터 링크, 또는 유선 및 무선 데이터 링크의 조합에 의해) 연결된 로컬 및 원격 노드 둘다가 태스크를 실행하는 분산 시스템 환경에서도 실시될 수 있다. 분산 시스템 환경에서, 프로그램 모듈은 로컬 및 원격 메모리 저장 장치의 양쪽 모두에 배치될 수가 있다.

<56> **연합 아키텍처**

<57> 도 1은 연합 기반구조(100)를 나타낸 것이다. 연합 기반구조(100)는 다른 유형의 연합 파트너쉽(federating partnership)을 형성할 수 있는 노드(101, 102, 103)를 포함한다. 예를 들어, 노드(101, 102, 103)는 루트 노드를 이용하지 않고 피어(peer)로서 서로 연합할 수가 있다. 각각의 노드(101, 102, 103)는 각각 대응하는 ID(171, 182, 193)를 가진다.

<58> 일반적으로, 노드(101, 102, 103)는, 파트너쉽을 형성해 정보(예를 들어, 다른 노드와의 상호작용에 관계된 상태 정보)를 교환하기 위해서, 연합 프로토콜(federation protocol)을 이용할 수가 있다. 파트너쉽의 형성 및 정보의 교환은 자원에의 보다 효율적이고 신뢰성있는 액세스를 용이하게 해준다. 노드(101, 102, 103)의 사이에는, 기타 중간 노드(도시하지 않음)가 존재할 수 있다(예를 들어, 171 내지 193의 사이의 ID를 갖는 노드). 따라서, 예를 들어, 노드(101)와 노드(103)의 사이에서 라우팅되는 메시지는 하나 이상의 기타 중간 노드를 통과할 수 있다.

<59> 연합 기반구조(100) 내의 노드(기타 중간 노드를 포함함)는 대응하는 랑데뷰 프로토콜 스택(rendezvous protocol stack)을 포함할 수 있다. 예를 들어, 노드(101, 102, 103)는 각각 대응하는 랑데뷰 프로토콜 스택(141, 142, 143)을 포함한다. 각각의 프로토콜 스택(141, 142, 143)은 애플리케이션 계층(예를 들어, 애플리케이션 계층(121, 122, 123)), 및 기타 하위 계층(예를 들어, 대응하는 기타 하위 계층(131, 132, 133))을 포함한다. 랑데뷰 프로토콜 스택 내의 각각의 계층은 자원 요청을 대응하는 자원에 랑데뷰시키는 것에 관계된 다른 기능을 담당한다.

<60> 예를 들어, 기타 하위 계층은 채널 계층, 라우팅 계층, 및 기능 계층을 포함할 수가 있다. 일반적으로, 채널 계층은 1개의 종단점으로부터 다른 종단점으로(예를 들어, 노드(101)로부터 노드(103)으로), (예를 들어, WS-ReliableMessaging 및 SOAP(Simple Object Access Protocol)을 이용하여) 메시지를 신뢰성있게 전송하는 일을 맡고 있다. 채널 계층은 착신 및 발신하는 신뢰성있는 메시징 헤더를 처리하고 신뢰성있는 메시징 세션에 관계된 상태를 유지하는 일도 맡고 있다.

<61> 일반적으로, 라우팅 계층은 목적지 쪽으로의 그 다음 홉을 계산하는 일을 맡고 있다. 라우팅 계층은 착신 및 발신하는 어드레싱 및 라우팅 메시지 헤더를 처리하고 라우팅 상태를 유지하는 일도 맡고 있다. 일반적으로, 기능 계층은 가입(join) 및 탈퇴(depart) 요청, ping, 갱신, 및 기타 메시지 등, 랑데뷰 프로토콜 메시지를 발행 및 처리하는 일은 물론, 이들 메시지에 대한 응답을 생성하는 일을 맡고 있다. 기능 계층은 라우팅 계층로부터의 요청 메시지를 처리하고, 대응하는 응답 메시지가 있는 경우, 라우팅 계층을 이용하여 그 응답 메시지를 발신측 노드로 답신한다. 기능 계층은 요청 메시지를 개시하고, 라우팅 계층을 이용하여 요청 메시지를 전달한다.

<62> 일반적으로, 애플리케이션 계층은 기능 계층으로부터 전달된 비랑데뷰 프로토콜 관련 데이터(non-rendezvous protocol specific data)(즉, 애플리케이션 메시지를) 처리한다. 기능 계층은 애플리케이션 계층으로부터의 애플리케이션 데이터에 액세스할 수 있고 애플리케이션 데이터를 랑데뷰 프로토콜 메시지(예를 들어, ping 및 갱신)에서 취득하고 랑데뷰 프로토콜 메시지에 넣을 수 있다. 즉, 기능 계층은 애플리케이션 데이터가 랑데뷰 프로토콜 메시지에 피기백되도록 할 수가 있고, 랑데뷰 프로토콜 노드를 수신할 시에, 애플리케이션 데이터가 애플리케이션 계층으로 다시 전달되도록 할 수 있다. 몇몇 실시예에서, 애플리케이션 데이터는 자원 및 자원 이용관계(resource interest)를 식별하는 데 사용된다. 따라서, 애플리케이션 계층은 자원 및 자원 이용관계를 식별하기 위해 기타 하위 계층로부터 수신된 데이터 및 기타 하위 계층으로 전송되는 데이터를 처리하는 애플리케이션 관련 논리 및 상태를 포함할 수 있다.

<63> **연합 메카니즘**

<64> 노드는 여러가지 다른 메카니즘을 이용해 연합할 수가 있다. 제1 연합 메카니즘은 정보를 다른 모든 피어 노드(peer node)로 전달하는 피어 노드를 포함한다. 노드는, 연합 기반구조에 참가하려고 할 때, 예를 들어, WS-Discovery 등의 브로드캐스트/멀티캐스트 발견 프로토콜(broadcast/multicast discovery protocol)을 이용하여 그 존재를 공표하고, 브로드캐스트/멀티캐스트 탐색(broadcast/multicast find)을 발행하여 기타 노드를 검출한다. 그 다음에, 노드는 네트워크 상에 이미 존재하고 있는 기타 노드와 간단한 전달 파트너쉽(forwarding partnership)을 확립하고, 신규 참가 노드와의 새로운 파트너쉽을 받아들인다. 그 후, 노드는 모든 애플리케이션 관련 메시지를 그의 파트너 노드들 모두에게 단순히 전달한다.

<65> 제2 연합 메카니즘은 애플리케이션 관련 메시지를 그의 목적지로 가장 효율적으로 전송하는 피어 노드를 포함한

다. 신규 노드는, 연합 기반구조에 참가하려고 할 때, 예를 들어, WS-Discovery 등의 브로드캐스트/멀티캐스트 발견 프로토콜을 이용하여 그의 존재를 공표하고, 브로드캐스트/멀티캐스트 탐색을 발행하여 연합 기반구조의 일부인 기타 노드들을 검출한다. 다른 노드가 검출되면, 신규 노드는 그 다른 노드와 파트너쉽을 확립한다. 확립된 파트너쉽으로부터, 신규 노드는 연합 기반구조에 이미 참가하고 있는 기타 노드의 존재를 알게 된다. 이어서, 신규 노드는 이러한 새로 알게 된 노드와 파트너쉽을 확립하고 새로 착신하는 파트너쉽 요청을 받아들인다.

<66> 노드 참가/이탈(node arrival/departure) 및 특정의 애플리케이션 관련 메시지에 대한 관심의 등록 모두가 연합 기반구조를 통해 플러딩(flooding)되는 결과, 모든 노드가 기타 파트너 노드 및 애플리케이션 관련 메시지에 대한 관심의 등록에 대한 전체적 지식을 가진다. 이러한 전체적 지식을 이용하여, 어느 노드라도 애플리케이션 관련 메시지를 그 애플리케이션 관련 메시지에 관심을 표명한 노드에 직접 전송할 수 있다.

<67> 제3의 연합 메카니즘은 모든 애플리케이션 관련 메시지를 그의 목적지로 간접적으로 전달하는 피어 노드를 포함한다. 이 제3의 메카니즘에서, 노드는, 예를 들어, 128 비트 또는 160 비트의 식별자(ID) 등 ID를 할당받을 수 있다. 주어진 애플리케이션 관련 메시지에 대한 관심의 등록을 유지하는 책무를 지는 노드는 애플리케이션 관련 메시지의 목적지 ID(예를 들어, URI)를 이 128 비트 또는 160 비트의 ID 공간에 매핑(예를 들어, 해싱)함으로써 획득되는 ID에 가장 가까운 ID를 갖는 노드가 되도록 결정될 수 있다.

<68> 이 제3의 메카니즘에서, 노드 참가 및 이탈은 페브릭(fabric) 전체에 걸쳐 플러딩(flooding)된다. 한편, 특정의 애플리케이션 관련 메시지에 대한 관심의 등록은 이러한 등록 정보를 유지하는 책무를 지도록 결정된 노드에 전달된다. 확장성, 부하 분산, 고장 허용(fault-tolerance)을 위해, 특정의 애플리케이션 관련 메시지에 대한 관심의 등록을 수신하는 노드는 그의 이웃 집합(neighborhood set) 내에서 그 등록 정보를 신뢰성있게 플러딩할 수가 있다. 지정된 노드의 이웃 집합은 지정된 노드의 ID의 양측에서 미리 정해진 범위 내에 있는 ID를 갖는 노드들의 집합이 되도록 결정될 수가 있다.

<69> 제2 메카니즘과 유사하게, 신규 참가 노드는, 예를 들어, WS-Discovery 등의 브로드캐스트/멀티캐스트 발견 프로토콜을 이용하여 그의 존재를 공표하고, 로컬 브로드캐스트/멀티캐스트 탐색을 발행하여 이미 연합 기반구조의 일부인 노드를 검출한다. 신규 노드는 발견된 노드와 파트너쉽을 확립하고, 그 파트너쉽을 이용하여 연합 기반구조에 참가하고 있는 기타 신규 노드들의 존재에 관하여 알게 된다. 이어서, 신규 노드는 새로 발견된 노드와 추가의 파트너쉽을 확립하고, 새로 착신하는 파트너쉽 요청을 받아들인다. 신규 노드는 자기가 책임지고 있는 특정의 애플리케이션 계층 관련 자원에 대한 관심의 착신 등록을 그의 파트너로부터 받아들이고 그의 이웃 집합에 걸쳐 그 등록들을 플러딩할 수 있다. 따라서, 메시지는 일반적으로 (예를 들어, 신규 참가 노드와 파트너쉽을 확립한, 또는 파트너 노드가 알고 있는) 중간 라우팅 노드를 통해 그의 최종 목적지로 전달될 수가 있다.

<70> 착신 애플리케이션 관련 메시지를 수신한 것에 응답하여, 신규 노드는 메시지에 지정된 목적지에 대한 등록 정보를 유지하는 책임을 지고 있을 수 있는 파트너 노드로 메시지를 전달한다. 따라서, 이 제3 메카니즘을 사용하는 경우, 연합 기반구조 내의 모든 노드는 다른 노드들 모두에 대한 전체적 지식을 가지지만, 등록 정보는 노드들 간에 효율적으로 구분된다. 애플리케이션 관련 메시지는 이러한 애플리케이션 관련 메시지에 대한 관심의 등록 정보를 유지하는 책임을 맡고 있을 수 있는 파트너의 노드만을 통해 그의 최종 목적지로 전송된다. 따라서, 처리되는 메시지에 대한 관심의 등록 정보에 대한 전체적 지식을 갖는 파트너 노드로만 전달하는 것에 의해 간접 전송(indirection)이 달성된다. 이것은 모든 파트너 노드로 전달하는 것에 의해 간접 전송이 달성되는 제1 메카니즘과는 대조적이다.

<71> 제4 연합 메카니즘은 기타 피어 노드로 메시지를 라우팅하는 피어 노드를 포함한다. 이 제4 메카니즘은 노드 참가/이탈 및 특정의 애플리케이션 관련 메시지에 대한 관심의 등록 둘다가 모두 플러딩되는 대신에 라우팅된다는 점에서 적어도 제3 메카니즘과 다르다. 라우팅 프로토콜은 애플리케이션 관련 메시지와 그 애플리케이션 관련 메시지에 대한 관심을 표명하는 등록 메시지 간의 랑데뷰(rendezvous)를 보장하도록 설계된다.

<72> 도 2는 요청을 파트너에게로 간접적으로 라우팅하는 것을 용이하게 해주는 컴퓨터 아키텍처(200)의 일례를 나타낸 것이다. 컴퓨터 아키텍처(200)는 연합 기반구조에 참가하고 있는 잠재적으로 복수의 로컬 발견 범위(local discovery scope)에 걸쳐 분산되어 있는 서로 다른 유형의 컴퓨터 시스템 및 장치를 나타내고 있다.

<73> 워크스테이션(233)은 등록된 PnP 프로바이더 인스턴스(PnP provider instance)를 포함할 수 있다. 이 PnP 프로바이더 인스턴스의 존재를 파트너에 통지하기 위해, 워크스테이션(233)은 연합 기반구조를 통해 등록 요청



(201)을 라우팅한다. 등록 요청(201)은 최초로 랩톱(231)으로 전달되고, 이어서 랩톱(231)이 등록 요청(201)을 메시지 브로커(237)로 전달하며, 이어서 메시지 브로커(237)가 등록 요청(201)을 메시지 게이트웨이(241)로 전달한다. 메시지 게이트웨이(241)는 그의 데이터베이스에 등록 정보 등록 요청(201)을 저장하고 워크스테이션(233)으로 성공 메시지(204)를 답신한다.

<74> 그 후, 이번에는 실행 서비스의 프로바이더 인스턴스인 다른 등록된 프로바이더 인스턴스가 워크스테이션(233) 내에서 활성화된다. 이번에, 노드는 메시지 게이트웨이(241)가 등록을 담당하고 있다는 것을 알고 있고, 등록 요청(205)을 메시지 게이트웨이(241)로 직접 전달한다. 메시지 게이트웨이(241)는 그의 데이터베이스에 등록 정보 등록 요청(205)을 저장하고 워크스테이션(233)으로 성공 메시지(206)를 답신한다.

<75> 그 후, 프린터(236)(예를 들어, UPnP 프린터)의 전원이 켜지고 프린터(236)는 통보(announcement)(207)를 전송한다. 서버 234는 통보(207)를 검출하고, 등록 요청(208)을 메시지 브로커(237)로 라우팅한다. 메시지 브로커(237)는 등록 요청(208)을 메시지 게이트웨이(241)로 전달한다. 메시지 게이트웨이(241)는 그의 데이터베이스에 등록 정보 등록 요청(208)을 저장하고, 서버(234)에 성공 메시지(210)를 답신한다.

<76> 그 후, 퍼스널 컴퓨터(242)가 모든 장치를 발견하기 위해 탐색 요청(211)을 발행한다. 퍼스널 컴퓨터(242)는, 탐색 요청(211)을 어디로 전달해야할지를 모르기 때문에, 워크스테이션(243)을 통해 탐색 요청(211)을 라우팅한다. 등록 및 탐색 요청은 동일한 목적지로 라우팅되므로, 라우팅 프로토콜은 2개의 요청의 양대부를 본질적으로 보장하고, 그 결과, 워크스테이션(243)은 탐색 요청(211)을 메시지 게이트웨이(241)로 전달한다. 메시지 게이트웨이(241)는 유지하고 있는 등록 정보를 검색하고, 탐색 요청(211)을 워크스테이션(233) 및 서버(234)의 양쪽 모두로 전달한다. 워크스테이션(233) 및 서버(234)는 각각 응답 메시지(214 및 216)를 퍼스널 컴퓨터(242)로 전송한다.

<77> 이 제4 메카니즘은 요청에 지정된 등록에 대한 전체적 지식을 갖는 노드(메시지 게이트웨이(241))로 요청을 (플러딩하는 대신에) 라우팅하는 것에 의해 기능한다. 이 제4 메카니즘은, 이하에서 더 상세히 설명하는 바와 같이, N을 연합 기반구조에 참가하고 있는 노드의 수라고 할 때,  $O(\log N)$  호프로 라우팅이 달성될 수 있도록 본질적으로 보장한다. 이 제4 메카니즘은, 노드 파트너쉽 및 등록 정보 둘다를 효율적으로 구분하므로, 아주 대규모 모인 네트워크로, 심지어 인터넷으로도 확장된다.

<78> 다수의 연합 메카니즘에 대해 설명하였지만, 이 설명을 검토해 보면, 기타 연합 메카니즘도 가능하다는 것이 당업자에게는 명백하게 될 것이다.

#### <79> 연합 내에서의 노드들 간의 관계

<80> 따라서, 연합은 정보가 체계적이고 효율적으로 분산 및 배치될 수 있는 동적이면서 확장성이 있는 네트워크를 형성하기 위해 서로 협동하는 일련의 노드로 구성된다. 노드는 이항 관계(binary relation)를 이용해 정렬 리스트(sorted list)로서 연합에 참가하도록 구성되며, 이 이항 관계는 반사적(reflexive), 반대칭적(anti-symmetric), 추이적(transitive), 전체적(total)이며, 노드 ID의 영역에 걸쳐 정의된다. 정렬 리스트의 양쪽 단부는 결합되어 링을 형성한다. 따라서, 리스트 내의 각 노드는 (모듈로 산술(modulo arithmetic)을 이용한 결과) 그 자체를 정렬 리스트의 중앙에 있는 것으로 볼 수 있다. 게다가, 리스트는 어느 노드라도 리스트를 양방향으로 순회할 수 있도록 이중 연결되어 있다(doubly linked).

<81> 각각의 연합 노드는 0부터 어느 일정한 상한 사이에 있는 ID의 고정 집합으로부터 (예를 들어, 중복 검출을 갖는 난수 발생기에 의해) ID를 할당받을 수 있다. 따라서, 일정한 상한의 ID에 1을 가산하면, ID는 0이 된다(즉, 연결 리스트의 끝으로부터 연결 리스트의 처음으로 이동한다). 게다가, 노드 ID의 값 영역으로부터 노드 자신으로의 1:1 매핑 함수가 정의된다.

<82> 도 3은 연결 리스트(304) 및 대응하는 링(306)의 일례를 나타낸 것이다. 이러한 링이 주어진 경우, 이하의 함수가 정의될 수 있다.

<83> RouteNumerically(V, Msg): 노드 ID의 값 영역에 속하는 값 V 및 메시지 "Msg"가 주어진 경우, ID가 매핑 함수를 이용해 V에 매핑 될 수 있는 노드 X에 메시지를 배달한다.

<84> Neighborhood(X, S): Neighborhood는 노드 X의 양측에 있는 노드들의 집합이며, 그 요소수(cardinality)는 S이다.

<85> 연합 내의 모든 노드가 링의 전체적 지식을 갖는 경우, RouteNumerically(V, Msg)는 매핑 함수를 V에 적용하는 것에 의해 그 ID가 획득되는 노드 X에 Msg를 직접 전송하는 것에 의해 구현된다. 다른 대안으로서, 노드가 기

타 노드에 대한 한정적인(예를 들어, 바로 근처의 노드만의) 지식 밖에 갖지 않는 경우, RouteNumerically(V, Msg)는, 목적지 노드 X에 도착할 때까지, 링에 따라 연속하는 노드들에 메시지를 전달하는 것에 의해 구현된다.

- <86> 다른 대안으로서 (또한 유리하게는), 노드는 (전체적 지식을 가질 필요없이, 또는 바로 인접한 노드들 간의 라우팅을 구현할 필요 없이) 분산 이진 검색(distributed binary search)을 수행하는 데 충분한 링에 관한 지식을 저장할 수가 있다. 링 지식을 유지하는 것으로 각각의 노드가 받는 영향이 충분히 작지만 라우팅 호프의 수의 감소에 의한 라우팅 성능의 향상이 가능해지도록, 링에 관한 지식의 양이 구성될 수 있다.
- <87> 앞서 설명한 것처럼, ID는 자연수들로 이루어진 충분히 큰 유한 집합(그의 범위가 0 이상 어떤 일정 값 이하인 유한개의 수들의 집합에 걸쳐 있음을 의미함)에 걸쳐 정의된 " $<$ "(보다 작다) 관계를 이용해 할당될 수 있다. 따라서, 연합에 참가하는 모든 노드는 0 이상 어떤 적절히 선택된 상한 이하의 자연수를 할당받는다. 이 범위는 빈틈이 없어야 필요는 없고, 노드에 할당된 번호들 사이에 간극이 있어도 된다. 노드에 할당된 번호는 링 내에서의 그의 ID로서 사용된다. 매핑 함수는 2개의 노드 ID의 사이에 있는 수를 그 수에 수치적으로 가장 가까운 ID를 갖는 노드에 매핑함으로써 수 공간(number space) 안의 간극을 고려하고 있다.
- <88> 이 방법은 다수의 이점을 가진다. 각각의 노드에 균일하게 분포된 수(uniformly distributed number)를 할당함으로써, 링의 모든 세그먼트(segment)가 균일하게 채워질 가능성이 높아진다. 게다가, 후속자(successor), 선행자(predecessor), 및 이웃(neighborhood)의 계산이 모듈로 산술을 이용해 효율적으로 행해질 수 있다.
- <89> 몇몇 실시예에서, 연합 노드는 (예를 들어, 난수 발생이 이용되는 경우) 2개의 노드가 동일한 ID를 할당받을 가능성이 극히 낮을 정도로 큰 ID 공간 내로부터 ID를 할당받을 수 있다. 예를 들어, 노드는, b가 예를 들어 8 또는 16이고 n이 예를 들어 128 비트 또는 160 비트 상당 자리수라고 할 때, 0 내지  $b^n - 1$ 의 범위 내의 ID를 할당받을 수 있다. 따라서, 노드는, 예를 들어, 0 내지  $16^{40} - 1$ (즉, 대략  $1.461502E48$ )의 범위에 속하는 ID를 할당받을 수 있다. 0 내지  $16^{40} - 1$ 의 범위는, 예를 들어, 인터넷 상의 모든 노드에 고유의 ID를 할당하는 데 충분한 수의 ID를 제공한다.
- <90> 따라서, 연합 내의 각 노드는,
- <91> 0 내지  $b^n - 1$ 의 범위 내에 균일하게 분포된 수치값인 ID와,
- <92> 라우팅 테이블을 가질 수 있으며(모든 계산은 모듈로  $b^n$ 으로 행해짐), 이 라우팅 테이블은
- <93> 후속 노드(successor node)(s),
- <94> 선행 노드(predecessor node)(p),
- <95>  $s_j.s.id > (id + u/2)$ ,  $j \geq v/2 - 1$ 이고  $p_k.p.id < (id - u/2)$ 이며  $k \geq v/2 - 1$ 인 이웃 노드( $p_k, \dots, p_1, p, s, s_1, \dots, s_j$ ), 및
- <96>  $r_{\pm i} = \text{RouteNumerically}(id \pm b^i, \text{Msg})$ 인 라우팅 노드( $r_{-(n-1)}, \dots, r_{-1}, r_1, \dots, r_{n-1}$ )로 이루어져 있다. 여기서, b는 기수(number base)이고, n는 자리수로 나타낸 필드 크기이며, u는 이웃 범위이고, v는 이웃 크기이며, 계산은 모듈로  $b^n$ 으로 행해진다. 양호한 라우팅 효율 및 고장 허용을 위해, u 및 v의 값은, N를 연합에 물리적으로 참가하고 있는 노드의 총수라고 할 때,  $u=b$ 이고  $v \geq \max(\log_2(N), 4)$ 일 수 있다. N는, 예를 들어, ID가 균일하게 분포하고 있는 경우, 길이가 b 이상인 링 세그먼트(segment) 상에 존재하는 노드의 수로부터 추정될 수 있다. b 및 n의 전형적인 값은,  $b=8$  또는 16이고,  $n=128$  비트 또는 160 비트 상당 자리수이다.
- <97> 따라서, 라우팅 노드는 링 전체에 걸치는 대수 지표(logarithmic index)를 형성할 수 있다. 링 상에서의 노드의 위치에 따라, 예를 들어,  $id \pm b^i$  [단  $i=(1, 2, \dots, (n-1))$ ]를 원소로 하는 집합 내의 각각의 수의 위치에 기존 노드가 존재하는 경우, 정확한 대수 지표가 가능하다. 그러나, 집합 내의 각각의 수의 위치에 기존 노드가 존재하지 않는 경우도 있을 수 있다. 그러한 경우,  $id \pm b^i$ 에 가장 가까운 노드가 라우팅 노드로서 선택될 수 있다. 그 결과의 대수 지표는 정확하지 않고, 집합 내의 몇개의 수에서는 고유의 라우팅 노드가 없는 것조차 있을 수 있다.
- <98> 다시 도 3을 참조하면, 도 3은 정렬 리스트(304) 및 대응하는 링(306)의 형태로 된 연합 기반구조 내의 노드들

간의 이항 관계의 일례를 나타내고 있다. 정렬 리스트(304)의 ID 공간은 0 내지  $2^8-1$ (즉, 255)의 범위에 있다. 즉,  $b=2$ 이고  $n=8$ 이다. 따라서, 도 3에 나타난 노드에는 0 내지 255의 범위 내의 ID가 할당된다. 정렬 리스트(304)는 이항 관계를 이용하며, 이 이항 관계는 반사적, 반대칭적, 추이적, 전체적이며, 노드 ID의 영역에 걸쳐 정의된다. 정렬 리스트(304)의 양단은 결합되어, 링(306)을 형성한다. 이것에 의해, 도 3의 각 노드는 그 자신을 정렬 리스트(304)의 중앙에 있는 것으로 볼 수 있다. 정렬 리스트(304)는 어느 노드라도 정렬 리스트(304)를 양방향으로 순회할 수 있도록 이중 연결되어 있다. 정렬 리스트(304)(또는 링(306))를 순회하기 위한 계산은 모듈로  $2^8$ 로 수행된다. 따라서, 255(즉, 정렬 리스트(304)의 끝)+1 = 0(즉, 정렬 리스트(304)의 시작)이다.

<99> 라우팅 테이블은 ID 64의 후속 ID가 ID 72(ID 64로부터 시계 방향으로 바로 다음의 ID)인 것을 나타내고 있다. 후속 ID는, 예를 들어, (예를 들어 ID가 71인) 신규 노드가 참가하거나 또는 (예를 들어 ID 76인) 기존 노드가 연합 기반구조로부터 이탈하는 경우에 변할 수 있다. 이와 마찬가지로, 라우팅 테이블은 ID 64의 선행 ID가 ID 50(ID 64로부터 반시계 방향으로 바로 다음의 ID)인 것을 나타내고 있다. 선행 ID는, 예를 들어, (예를 들어 ID가 59인) 신규 노드가 참가하거나 또는 (예를 들어 ID 50인) 기존 노드가 연합 기반구조로부터 이탈하는 경우에 변할 수 있다.

<100> 라우팅 테이블은 또한 ID 64의 이웃 노드들의 집합이 ID 83, 76, 50, 및 46을 갖는 것을 나타내고 있다. 이웃 노드들의 집합은 ID 64의 지정된 범위(즉, 이웃 범위  $u$ ) 내에 있는 지정된 수의 노드(즉, 이웃 크기  $v$ )일 수 있다. 이웃 노드들의 집합을 식별하는 데, 예를 들어,  $V=4$  및  $U=10$  등 여러가지 서로 다른 이웃 크기 및 이웃 범위가 아마도 사용될 수 있다. 이웃 노드들의 집합은, 예를 들어, 노드가 연합 기반구조에 참가하거나 이탈할 경우, 또는 지정된 노드 수 또는 지정된 범위가 변경될 경우에 변할 수 있다.

<101> 라우팅 테이블은 또한 ID 64가 ID 200, 2, 30, 46, 50, 64, 64, 64, 64, 76, 83, 98, 135, 및 200을 갖는 노드들로 라우팅될 수 있는 것을 나타내고 있다. 이 리스트는  $id \pm 2^i$  (단,  $i=(1, 2, 3, 4, 5, 6, 7)$ )를 원소로 하는 집합 내의 각각의 수에 가장 가까운 노드를 식별하는 것에 의해 생성된다. 즉,  $b=2$ 이고  $n=8$ 이다. 예를 들어, ID 76를 갖는 노드는  $64+2^3$ , 즉 72에 가장 가까운 노드를 계산하는 것으로부터 식별될 수 있다.

<102> 노드는 메시지(예를 들어, 자원에서의 액세스의 요청)를 선행 노드, 후속 노드, 이웃 노드 집합 내의 임의의 노드, 또는 임의의 라우팅 노드로 직접 라우팅할 수 있다. 몇몇 실시예에서, 노드는 메시지를 라우팅 하기 위해 수치적 라우팅 함수(numeric routing function)를 구현한다. 따라서, ID가 수치적으로  $V$ 에 가장 가까운 연합 기반구조 내의 노드  $Y$ 로 Msg를 배달하고 노드  $Y$ 의 ID를 노드  $X$ 에 반환하기 위해, 노드  $X$ 에서 RouteNumerically( $V$ , Msg)가 실시될 수 있다. 예를 들어, ID 64를 갖는 노드는 ID 250를 갖는 노드로 메시지가 라우팅되도록 RouteNumerically(243, Msg)를 실시할 수 있다. 그러나, ID 250가 ID 64에 대한 라우팅 노드가 아니기 때문에, ID 64는 ID 2(243에 가장 가까운 라우팅 노드)로 메시지를 라우팅할 수 있다. 이번에는, ID 250를 갖는 노드로 메시지가 (직접 또는 추가의 중간 노드를 통해) 라우팅되도록 ID 2를 갖는 노드가 RouteNumerically(243, Msg)를 실시할 수 있다. 따라서, RouteNumerically 함수는 반복하여 호출되고, 각각의 호출은 메시지를 목적지에 더 가깝게 라우팅한다.

<103> 유리하게도, 본 발명의 다른 실시예들은 하나 이상의 근접성 카테고리(예를 들어, 지리적 경계, 라우팅 특성(예를 들어, IP 라우팅 홉), 관리 도메인, 조직 경계 등)의 복수의 근접성 기준에 기초하여 링을 링들로 이루어진 링 또는 링들로 이루어진 트리로 분할하는 것을 용이하게 해준다. 같은 유형의 근접성 기준을 이용하여 2회 이상 링을 분할할 수 있다는 것을 잘 알 것이다. 예를 들어, 대륙 근접성 기준과 국가 근접성 기준(둘다 지리적 경계의 근접성 카테고리임)에 기초하여 링이 분할될 수 있다.

<104> (난수 발생의 결과) ID 공간에 걸쳐서 ID가 균일하게 분포될 수 있기 때문에, 근접성 클래스(class)가 거의 같은 요소수(cardinality)를 가진다면, 순환 ID 공간(circular ID space)의 임의의 주어진 세그먼트(segment)가 서로 다른 근접성 클래스에 속하는 노드를 포함할 확률이 높다. 의미가 있는 통계적 거동을 얻는 데 충분한 수의 노드가 존재하는 경우, 이 확률은 한층 더 높아진다.

<105> 따라서, 임의의 주어진 노드의 이웃 노드는 일반적으로 근접성의 관점에 볼 때 잘 분산되어 있다. 공개된 애플리케이션 상태가 이웃 노드들 간에 복제될 수 있기 때문에, 공개된 정보도 역시 근접성의 관점에서 볼 때 잘 분산되어 있을 수 있다.

<106> 도 4는 근접 라우팅(proximal routing)을 용이하게 해주는 링들로 이루어진 링(400)을 나타내고 있다. 링(400

1)은 마스터 또는 루트 링이라고 볼 수 있으며, 각각의 링(402, 403, 및 404) 내의 모든 노드를 포함하고 있다. 각각의 링(402, 403, 404)은 지정된 근접성 기준에 기초하여 분할된 링(401)에 속하는 노드들의 부분 집합을 포함하고 있다. 예를 들어, 링(401)은 지리적 위치에 근거해 분할될 수가 있고, 이 경우 링(402)은 북아메리카에 있는 노드를 포함하고, 링(403)은 유럽에 있는 노드를 포함하며, 링(404)은 아시아에 있는 노드를 포함한다.

<107> 65,536( $2^{16}$ )개의 ID를 포함하는 수치 공간에서, ID 5345를 갖는 북아메리카의 노드로부터 ID 23345를 갖는 아시아의 노드로의 메시지의 라우팅은, 아시아의 노드의 이웃 노드(neighbor node)가 식별될 때까지, 메시지를 링(402) 내에서 라우팅하는 것을 포함할 수 있다. 그 후, 이웃 노드는 아시아의 노드로 메시지를 라우팅할 수 있다. 따라서, 북아메리카의 노드와 아시아의 노드 사이에서는, (복수의 홉이 아니라) 1회의 홉이 행해진다. 그에 따라, 라우팅이 자원 효율이 좋은 방식으로 수행된다.

<108> 도 5는 근접 라우팅을 용이하게 해주는 링들로 이루어진 근접 유도 분할 트리(proximity induced partition tree)(500)를 나타낸 것이다. 도시한 바와 같이, 링들로 이루어진 분할 트리(500)는 복수의 링을 포함하고 있다. 각각의 링은 정렬 연결 리스트(sorted linked list)의 일부분을 나타낸다. 각각의 링은 정렬 연결 리스트 내의 ID를 갖는 복수의 노드를 포함한다. 그러나, 잠재적인 노드가 많기 때문에(예를 들어, 분할 트리(500)의 ID 공간은  $b=16$ 이고  $n=40$ 일 수 있음), 명료함을 위해, 노드가 링 상에 명시적으로 나타내어져 있다.

<109> 분할 트리(500) 내에서, 루트 링(501)은 기준(571)(제1 관리 도메인 경계 기준)에 기초하여 서브-링(511, 512, 513, 514)을 포함하는 복수의 서브-링으로 분할된다. 예를 들어, DNS 이름의 각 성분은 근접성 기준이라고 볼 수 있으며, 이 성분들 간에는 오른쪽에서 왼쪽으로 읽었을 때의 DNS 이름 내에서의 출현 순서에 의해 유도된 부분 순서(partial order)가 존재한다. 따라서, 서브-링(511)은 기준(581)(제2 관리 도메인 경계 기준)에 기초하여 서브-링(521, 522, 523)을 포함하는 복수의 서브-링으로 더 분할될 수 있다.

<110> 서브-링(522)은 기준(572)(지리적 경계 기준)에 기초하여 서브-링(531, 532, 533)을 포함하는 복수의 서브-링으로 더 분할될 수 있다. 위치 기반 근접성 기준은 대륙, 국가, 우편 번호 등의 계열에 따라 부분적으로 순서화될 수 있다. 우편 번호는 그 자체가 계층적으로 구성되어 있으며, 이는 우편 번호가 근접성 기준의 부분 순서화된 서브-리스트를 더 유도할 수 있다고 볼 수 있음을 의미한다.

<111> 서브-링(531)은 기준(573)(제1 조직 경계 기준)에 기초하여 서브-링(541, 542, 543, 544)을 포함하는 복수의 서브-링으로 더 분할될 수 있다. 근접성 기준의 부분 순서화된 리스트는 주어진 기업이 어떻게 조직적으로 구조화되어 있는지, 즉, 부, 과, 제조 그룹 등의 계열에 따라 유도될 수가 있다. 따라서, 서브-링(543)은 기준(583)(제2 조직 경계 기준)에 기초하여 서브-링(551, 552)을 포함하는 복수의 서브-링으로 더 분할될 수 있다.

<112> 분할 트리(500) 내에서, 각각의 노드는 단일의 ID를 가지며, 루트로부터 시작되어 리프에 도달하는 대응하는 분할 경로를 따라 링에 참가한다. 예를 들어, 서브-링(552)에 참가하는 각각의 노드는 서브-링(543, 531, 522, 511), 및 루트(501)에도 참가한다. 목적지 노드(ID)로의 라우팅은, 이하와 같이, RouteProximally 함수를 실시하는 것에 의해 달성될 수 있다.

<113> RouteProximally(V, Msg, P): 노드 ID의 영역에 속하는 값 V 및 메시지 "Msg"가 주어진 경우, 근접성 기준 P에 의해 동등한 것으로 간주되는 노드들 중에서 V에 매핑될 수 있는 ID를 갖는 노드 Y로 메시지를 배달한다.

<114> 따라서, 라우팅은 주어진 링 내에서, 그 링 내에서의 라우팅에 의해 더 이상 전진할 수 없게 될 때까지, 목적지 노드에 점진적으로 가까워져 가는 것에 의해 달성할 수가 있지만, 전진할 수 없게 되는 것은 현재 노드와 그의 후속 또는 선행 노드 사이에 목적지 노드가 있다고 하는 조건으로부터 결정된다. 이 시점에서, 현재 노드는 자신이 참가하고 있는 그 다음으로 큰 링에 있는 그의 파트너 노드를 통해 라우팅을 시작한다. 루트 링을 향해 분할 경로를 따라 올라감으로써 목적지 노드에 점진적으로 가까워져 가는 이 프로세스는, RouteProximally 호출에서 최초로 지정된 요청된 근접성 컨텍스트 내에서 목적지 노드에 가장 가까운 노드에 도착할 때에 종료한다.

<115> 목적지 노드가 요청을 발신한 노드의 근접 이웃(proximal neighborhood)의 외부에 존재하기 때문에 그 이웃 내에서는 더 이상 전진할 수 없게 될 때까지, 라우팅 홉이 그 이웃 내에 남아 있을 수가 있다. 이 시점에서, 더 전진하기 위해 근접 이웃의 크기를 확대하기 위해서, 근접성 기준이 완화된다. 이 프로세스는 근접 이웃이 충분히 확대되어 목적지 노드(ID)를 포함하게 될 때까지 반복된다. 근접 이웃 기준의 연속적인 각각의 완화 이후에 행해지는 라우팅 홉은 근접 공간 내에서는 잠재적으로 보다 큰 점프가 될 수 있지만, 수치 공간 내에서는 그에 대응하여 이전의 호프와 비교해 보다 작은 점프를 한다. 따라서, 목적지에 도착하기 전에는 절대적으로 필요한 수의 이러한 (링 간의) 홉만이 행해진다.



- <116> 공개된 애플리케이션 데이터가 목적지 노드의 이웃 노드들 간에 복제될 때에는 분할 트리 아래에 복제되기 때문에, 탐색 메시지에 대해 몇개의 홉이 회피되는 경우가 있을 수 있다.
- <117> 근접 라우팅을 달성하기 위해, 각각의 연합 노드는 (단일 링에 대한 후속 노드 및 선행 노드와 마찬가지로) 멤버로서 참가하는 모든 링 내의 그의 후속 노드 및 선행 노드 - 근접 선행 노드, 근접 후속 노드, 및 근접 이웃 - 에 대한 참조를 유지한다. 라우팅이 효율적하도록 하기 위해, 노드는 (단일의 링에 대한 라우팅 노드와 마찬가지로) 링의 양쪽 절반에서의 지수함수적으로 증가하는 거리에 가장 가까운 기타 노드들을 라우팅 파트너로 하는 그러한 노드들의 참조도 유지할 수 있다. 몇몇 실시예에서, 한 쌍의 연속하는 후속 또는 선행 노드의 사이에 있는 라우팅 파트너 노드는 현재 노드와 후속 또는 선행 노드쌍 중에서 그에 수치적으로 가장 가까운 노드에 의해 각각 공유되는 동일한 최하위 링에 참가한다. 따라서, 목적지 노드로 향하는 라우팅 홉은 더 전진하기 위해서 절대로 필요한 때만 완화된 근접성 기준을 이용하는 것으로 천이한다(즉, 보다 상위의 링으로 천이한다). 그에 따라, 메시지가 대응하는 연합 노드와 효율적으로 라우팅될 수가 있다.
- <118> 몇몇 실시예에서, 노드는 동등한 기준 관계에 기초하여 메시지를 라우팅 하기 위해 근접 라우팅 함수(proximal routing function)를 실시한다. 따라서, 번호 V와 메시지 "Msg"가 주어졌을 때, 노드는 근접성 기준 P에 의해 동등한 것으로 간주되는 노드들 중에서 V에 매핑될 수 있는 ID를 갖는 노드 Y로 메시지를 배달하기 위해 RouteProximally(V, Msg, P)를 실시할 수 있다. 근접성 기준 P는 근접성 기준 P에 의해 근접성이 동등한 것(proximally equivalent)으로 간주되는 모든 노드에 공통의 선조(ancestor)인 분할 트리에서의 최하위 링을 식별한다. 근접성 기준 P는 루트 링으로부터 근접성 기준 P에 의해 식별된 링까지의 경로를 따라 발견되는 근접성 기준을 경로 분리 문자 "/"으로 분리하면서 연결하는 것에 의해 얻어지는 문자열로서 표현될 수 있다. 예를 들어, 서브-링(542)을 식별해주는 근접성 기준은 "Proximity : /.COM/Corp2/LocationA/Div2"로서 표현될 수 있다. 분할 트리(500) 내의 각각의 링에는, 예를 들어, SHA 기반 알고리즘을 이용해 그의 표현 문자열을 해싱함으로써 고유의 번호가 할당될 수 있다. 번호 0이 루트 링을 위해 예비되어 있는 경우, RouteNumerically(V, Msg)≡RouteProximally(V, Msg, 0)이라고 추론될 수 있다.
- <119> 예를 들어, 서브-링(544) 내의 노드는 (예를 들어, 서브-링(513) 내의 노드에) 보다 가까운 서브-링(531) 내의 노드를 식별하기 위해 RouteProximally를 실시할 수 있다. 이번에는, 서브-링(531)은 서브-링(522) 내에서 보다 가까운 노드를 식별하기 위해서 RouteProximally를 실시할 수 있다. 마찬가지로, 서브-링(522)은 서브-링(511) 내에서 보다 가까운 노드를 식별하기 위해 RouteProximally를 실시할 수 있다. 마찬가지로, 서브-링(511)은 서브-링(501)내에서 보다 가까운 노드를 식별하기 위해 RouteProximally를 실시할 수 있다. 따라서, RouteProximally 함수가 반복적으로 호출되고, 각각의 호출이 메시지를 목적지에 더 가깝게 라우팅하는 경우가 있을 수 있다.
- <120> 따라서, 근접성 기준이 고려되는 경우, 최종 목적지까지의 경로 상의 라우팅 홉은, 목적지 노드에 도착하거나 선택된 근접성 기준 하에서 더 이상 전진할 수 없게 될 때까지는, 요청을 발신하는 노드의 근접성 내에 남아 있을 수가 있지만, 수치 공간 내에서는 발신 노드와 목적지 노드 사이에 상당한 전진을 하며, 전진할 수 없게 된 시점에서, 목적지를 향해 새로운 전진을 실시하는 데 충분할 만큼만 근접성 기준이 완화된다. 예를 들어, 근접성 기준은 링(531)으로부터 링(522) 등에 이르기까지 메시지를 라우팅하는 데 충분할 만큼 완화될 수 있다.
- <121> 근접성에 대한 상기의 방법을 이용하면, 공개 정보를 주어진 링으로 한정하는 것이 가능하게 된다. 예를 들어, 조직은 조직 관련 정보가 (1) 조직의 신뢰 도메인 외부의 노드로의 이웃 복제(neighborhood replication)의 형태로 암묵적으로도 또한 (2) 이러한 정보의 탐색 요청에 대한 서비스를 제공하는 형태로 명시적으로도 조직의 신뢰 도메인 외부의 개체가 이용할 수 없게 보장하고자 할 수 있다. 제1 측면은 지정된 링 내에서 타겟 ID에 이웃하는 노드들 간에만 공개 정보를 복제하는 것에 의해 만족된다. 노드에 의해 발신되는 모든 메시지가 노드가 속하는 링을 루트 링에 향해 연속적으로 올라 가는 것에 의해 라우팅되기 때문에, 조직 내에서 발신된 모든 탐색 요청이 조직 내에 제한된 공개 정보를 찾아낼 수 있을 가능성이 높으며, 따라서 제2 측면을 암묵적으로 만족시킨다.
- <122> 또, 조직은 노드가 조직의 신뢰 도메인 외부의 노드와 자동적으로 연합하는 것을 좋아하지 않는다. 이러한 일은, 예를 들어, 방문 판매자가 고객의 건물에서 그의 랩톱 컴퓨터를 네트워크에 접속할 경우에 일어날 수 있다. 이상적으로는, 판매자가 소유하는 랩톱 컴퓨터는 자신의 홈 도메인 내에서 공개되어 있는 정보를 찾아내는 것, 및/또는 최하위의 선호되는 근접성 링에서 시작하여 자신의 홈 도메인 내의 노드와 연합하는 것을 바란다. 고객 도메인 내의 노드와 연합 하는 것은 일반적으로 허가되지 않는다. 이 시나리오를 지원하려면, 홈 도메인 내에서 씨드 노드(seed node)를 찾아낼 수 있어야만 한다. 이러한 씨드 노드는 홈 도메인 내에 공개되어 있는 정

보를 찾아내어 홈 연합(home federation)에 참가하고 공개 정보를 도메인 간에 선택적으로 가져오기(import) 및 내보내기(export)하기 위해 사용될 수 있다. 씨드 노드는 때때로 메시지 게이트웨이라고도 불린다.

<123> 기타 실시예에서, 개체는 루트 링 내의 씨드 노드에 대한 참조를 공개한다. 씨드 노드는 (표현 문자열을 해싱하는 것에 의해 얻을 수 있는 번호 등) 링과 연관된 (타겟 ID로서의) 고유 번호로 공개될 수 있다. 씨드 노드 정보는 게다가 루트 링 내의 대응하는 타겟 ID까지의 경로 상에 있는 여러가지 링 내의 노드들에 의해 주문형으로 캐싱될 수 있다. 이러한 주문형 캐싱(on demand caching)은 성능을 향상시키고 반정적인 정보(semi-static information)가 매우 빈번하게 검색되는 경우에 발생할 수 있는 핫스팟(hotspot)을 감소시킬 수 있다. 씨드 노드 정보는 DNS 등 기타 수단을 통해 획득될 수도 있다.

<124> 제한된 공개 정보에 고장 허용을 제공하기 위해, 각각의 노드는 자신이 참가하는 모든 링에서의 이웃 노드들의 집합을 유지할 수 있다. 상기와 같은 노드가 주어졌을 경우, 노드에 의해 유지되는 상태는 이하와 같이 요약될 수 있다.

<125> • 0 내지  $b^n - 1$ 의 범위 내에 균일하게 분포된 수치값인 ID.

<126> • 이하의 것들로 이루어진 라우팅 테이블(모든 계산이 모듈로  $b^n$ 으로 행해짐).

<127> o 노드가 참가하는 각각의 링, 즉 링 d에 대해,

<128> • 후속 노드(sd)

<129> • 선행 노드(pd)

<130> •  $s_{jd}.s_d.id > (id + u/2)$ ,  $j \geq v/2 - 1$ ,  $p_{kd}.p_d.id < (id - u/2)$ ,  $k \geq v/2 - 1$ 인 이웃 노드( $p_{kd}, \dots, p_{ld}, p_d, s_d, s_{ld}, \dots, s_{jd}$ )와

<131> o 적절한 경우,  $s_d \leq id + b^i \leq s_{d+1}$  또는  $p_{d+1} \leq id - b^i \leq p_d$ 이도록  $r_{\pm i} = \text{RouteProximally}(id \pm b^i, \text{updateMsg}, d)$ 인 라우팅 노드( $r_{-(n-1)}, \dots, r_{-1}, r_1, \dots, r_{n-1}$ )

<132> 여기서, b는 기수이고, n는 자리수로 나타낸 필드 크기이며, u는 이웃 범위이고, v는 이웃 크기이다.

<133> 링 "d" 내의 주어진 노드에 의해 유지되는 이웃 노드들의 부분 집합이 주어진 노드가 역시 참가하고 있는 자식 링(child ring) "d+1"에서 이웃 노드로서 다시 출현할 수 있다는 것에 유의해야 한다. 그 때문에, 주어진 노드가 참가하고 있는 D개의 링 모두에 걸쳐 주어진 노드에 의해 유지되는 이웃 노드의 총수의 상한이  $D \cdot \max(u, v)/2$ 로서 도출될 수 있다. 이것은 주어진 노드에 대한 1개의 참조만이 유지되고 최악 경우의 상한이 균형 트리(balanced tree)에 대한 것임을 고려하고 있다.

<134> 링이 복수의 대응하는 형제 서브-링(sibling sub-ring)으로 분할되는 경우, 예를 들어, 별칭 지정(aliasing)을 통해 지정된 노드가 동시에 2 이상의 복수의 대응하는 형제 서브-링에 참가하는 것이 허가되는 것에 유의한다. 별칭 지정은, 예를 들어, 서로 다른 서브-링으로부터의 서로 다른 상태를 지정된 노드와 연관시키기 위해 실시될 수가 있다. 따라서, 주어진 노드의 별칭(alias)이 동일한 ID를 갖지만, 각각의 별칭은 그들과 연관된 다른 상태를 가질 수 있다. 별칭 지정은 지정된 노드가 반드시 보다 특징적인 근접성 기준의 공통의 선조일 필요는 없는 다른 근접성 기준을 갖는 복수의 링에 참가하는 것을 가능하게 해준다. 즉, 지정된 노드는 근접성 트리의 복수의 가지에 참가할 수 있다.

<135> 예를 들어, 듀얼 NIC(유선 및 무선) 랩톱은 랩톱과 동일한 LAN 세그먼트(segment)를 공유하는 기타 무선 및 유선 노드의 양쪽 모두와 근접성면에서 동등하다고 볼 수 있다. 그러나, 이들 2개의 다른 근접성 기준은, 예를 들어, 조직의 멤버십에 기초한 기준 등, 다른 상위의 우선 근접성 기준(priority proximity criterion)을 적용한 후에만 적용가능한 서브-기준으로서 모델링될 수가 있다. 그 랩톱이 동일 조직에 속하기 때문에, 1) 유선에 있어서의 멤버십, 및 2) 무선 LAN 세그먼트(segment)에 있어서의 멤버십을 나타내는 2개의 서브-링 내의 별칭 지정된 노드는 랩톱이 속하는 조직을 나타내는 링 내의 단일의 노드에 병합된다. RouteProximally는 별칭 지정이 존재하는 경우에도 수정 없이 예상된 대로 기능한다는 것을 잘 알 것이다.

<136> 각각의 근접 링은 (잠재적으로 다른) 링 파라미터에 따라 구성될 수 있다. 링 파라미터는 이웃을 정의하기 위해 (예를 들어, 링 파라미터는 이웃 범위, 이웃 크기, ping 메시지 및 이탈 메시지 타이밍, 그리고 ping 및 이

탈 메시지의 배포 패턴을 나타낼 수 있음), (예를 들어, 앞서 설명한 상기의 제1 내지 제4의 연합 메카니즘 중에서, 또는 기타 연합 메카니즘 중에서) 특정의 연합 메카니즘을 나타내기 위해, 또는, 동일한 근접 링 내의 라우팅 파트너들 간의 통신 특성을 정의하기 위해 사용될 수 있다. 어떤 링 파라미터는 보다 범용적이어서, 복수의 서로 다른 연합 메카니즘에 적용될 수 있는 반면, 다른 링 파라미터는 보다 전용적이어서, 특정의 유형의 연합 메카니즘에만 적용된다.

<137> 보다 상위 레벨의 근접 링을 구성하는 데 사용되는 링 파라미터는, 몇몇 실시예에서는, 보다 하위의 근접 링에 의해 상속될 수가 있다. 예를 들어, 링(543)은 링(531)의 링 파라미터 중 몇개를 상속한다(차례로, 링(531)은 링(522)으로부터 상속하고, 이하 마찬가지이다). 따라서, 링(531)과 연관된 이웃 크기 및 이웃 범위는 링(541)과도 연관되어 있다.

<138> 그러나, 상속된 링 파라미터는 변경될 수 있고 및/또는 근접 링은 다른 링 파라미터에 따라 개별적으로 구성될 수 있다. 예를 들어, 링(511)이 다수의 노드를 포함하는 관리 도메인을 위한 것이고, 따라서 상기한 제4 연합 메카니즘이 링(511)에 더 적절한 경우가 있을 수 있다. 반면에, 링(521)이 비교적 소수의 노드를 갖는 소기업을 위한 것이고, 따라서 상기한 제2 연합 메카니즘이 링(521)에 더 적절한 경우가 있을 수 있다. 따라서, 링(521)과 연관된 링 파라미터는 링(511)과 연관된 링 파라미터와 다른 값으로 설정될 수 있다(또는 상속된 파라미터가 그러한 값으로 변경될 수 있다). 예를 들어, 특정의 유형의 연합 메카니즘을 나타내는 링 파라미터가 링(511)과 링(521) 간에 다를 수 있다. 마찬가지로, 이웃을 정의하는 파라미터도 링(511)과 링(521) 간에 다를 수 있다. 게다가, 링(521)은 상기한 제2 연합 메카니즘에 고유한 특정 파라미터에 따라 구성될 수가 있는 반면, 링(511)은 상기한 제4 연합 메카니즘에 고유한 특정 파라미터에 따라 구성될 수 있다.

<139> 따라서, 근접 링은 근접 링 내의 노드의 특성(예를 들어, 수, 포함된 자원 등)에 기초하여 유연하게 구성될 수 있다. 예를 들어, 관리자는 (예를 들어, 사용자 인터페이스를 통해) 구성 절차를 사용하여 근접 링의 링 파라미터를 선택할 수 있다. 구성 절차는, 예를 들어, 다른 방식으로 상속된 링 파라미터를 오버라이드(override)하는 등을 위해, 개개의 근접 링의 구성 뿐만 아니라, 근접 링들 간의 상속 관계의 구성도 용이하게 해줄 수 있다.

<140> 도 8은 연합 기반구조의 노드를 분할하는 방법(800)의 플로우차트의 일례를 나타낸 것이다. 방법(800)은 도 5의 분할 트리(500)의 링과 관련하여 설명된다. 방법(800)은 연합 기반구조 내의 노드에 할당된 노드 ID를 포함하는 정렬 연결 리스트에 액세스하는 동작(동작 801)을 포함한다. 예를 들어, 링(501)으로 표현되는 정렬 연결 리스트가 액세스될 수 있다. 정렬 연결 리스트의 노드 ID(링(501) 상에 나타난 노드)는 연합 기반구조(예를 들어, 연합 기반구조(100)) 내의 노드를 나타낼 수 있다.

<141> 방법(800)은 정렬 연결 리스트를 분할하기 위한 복수의 서로 다른 근접성 기준을 나타내는 근접성 카테고리에 액세스하는 동작(동작 802)을 포함한다. 예를 들어, 도메인 경계(561), 지리적 경계(562), 및 조직 경계(563)를 나타내는 근접성 기준이 액세스될 수 있다. 그러나, 신뢰 도메인 경계 등 기타 근접성 기준도 액세스된 근접성 기준에 표현될 수 있다. 근접성 카테고리는 사전에 작성된 부분적으로 순서화된 근접성 기준의 리스트를 포함할 수 있다. 링은 부분적으로 순서화된 근접성 기준의 리스트에 기초하여 분할될 수 있다.

<142> 방법(800)은 정렬 연결 리스트를 제1 근접성 기준에 기초하여 하나 이상의 제1 서브-리스트로 분할하는 동작(동작 803)을 포함하고, 하나 이상의 제1 서브-리스트의 각각은 정렬 연결 리스트에 속하는 노드 ID의 적어도 부분 집합을 포함한다. 예를 들어, 링(501)은 기준(571)에 기초하여 서브-링(511, 512, 513, 514)으로 분할될 수 있다. 서브-링(511, 512, 513, 514) 각각은 링(501)에 속하는 노드 ID의 다른 부분 집합을 포함할 수 있다.

<143> 방법(800)은 하나 이상의 제1 서브-리스트 중에서 선택된 제1 서브-리스트를 제2 근접성 기준에 기초하여 하나 이상의 제2 서브-리스트로 분할하는 동작(동작 804)을 포함하고, 하나 이상의 제2 서브-리스트 각각은 제1 서브-리스트에 포함되는 노드 ID의 적어도 부분 집합을 포함한다. 예를 들어, 서브-링(511)은 기준(581)에 기초하여 서브-링(521, 522, 523)으로 분할될 수 있다. 서브-링(521, 522, 523) 각각은 서브-링(511)에 속하는 노드 ID의 다른 부분 집합을 포함할 수 있다.

<144> 도 9는 노드의 라우팅 테이블을 채우는 방법(900)의 플로우차트의 일례를 나타낸 것이다. 방법(900)은 도 3의 정렬 연결 리스트(304) 및 링(306)과 관련하여 설명된다. 방법(900)은 선행 노드를 라우팅 테이블에 삽입하는 동작(동작 901)을 포함하고, 선행 노드는 정렬 연결 리스트의 제1 방향에서 현재 노드와 비교하여 현재 노드에 선행한다. 예를 들어, ID 50를 갖는 노드는 ID 64를 갖는 노드(현재 노드)의 선행 노드로서 라우팅 테이블에 삽입될 수 있다. 시계 방향(321)으로(정렬 연결 리스트(304)의 단부 A로부터 정렬 연결 리스트(304)의 단부 B

쪽으로) 이동하는 경우, ID 50를 갖는 노드는 ID 64를 갖는 노드에 선행한다. 선행 노드를 삽입하는 것에 의해, 현재 노드와 선행 노드 사이에 현재 노드가 선행 노드의 파트너이면서 선행 노드가 현재 노드의 파트너인 대칭적인 파트너십이 확립될 수 있다.

<145> 방법(900)은 후속 노드를 라우팅 테이블에 삽입하는 동작(동작 902)을 포함하고, 후속 노드는 정렬 연결 리스트의 제1 방향에서 현재 노드와 비교해 현재 노드에 후속한다. 예를 들어, ID 76를 갖는 노드가 ID 64를 갖는 노드(현재 노드)의 후속 노드로서 라우팅 테이블에 삽입될 수 있다. 반시계 방향(322)으로(정렬 연결 리스트(304)의 단부 A로부터 정렬 연결 리스트(304)의 단부 B 쪽으로) 이동하는 경우, ID 76를 갖는 노드는 ID 64를 갖는 노드에 후속한다. 후속 노드를 삽입하는 것에 의해, 현재 노드와 후속 노드 사이에 현재 노드가 후속 노드의 파트너이면서 후속 노드가 현재 노드의 파트너인 대칭적인 파트너십이 확립될 수 있다.

<146> 방법(900)은 적절한 이웃 노드를 라우팅 테이블에 삽입하는 동작(동작 903)을 포함하고, 이웃 노드는 제1 방향 및 이와 반대인 제2 방향 양쪽 모두에서 이웃 범위 및 이웃 크기에 기초하여 정렬 연결 리스트로부터 식별된다. 예를 들어, ID 83, 76, 50, 46을 갖는 노드는 ID 64를 갖는 노드(현재 노드)의 이웃 노드로서 라우팅 테이블에 삽입될 수 있다. 20의 이웃 범위 및 4의 이웃 크기에 기초하여, 시계 방향(321)에서 ID 83 및 76을 갖는 노드가 식별될 수 있고 반시계 방향(322)에서 (정렬 연결 리스트(304)의 단부 B로부터 정렬 연결 리스트(304)의 단부 A 쪽으로) ID 50 및 46을 갖는 노드가 식별될 수 있다. 몇몇 환경에서, 적절한 이웃 노드가 식별되지 않는 경우가 있을 수 있다. 이웃 노드를 삽입하는 것에 의해, 현재 노드와 이웃 노드의 사이에, 현재 노드가 이웃 노드의 파트너이면서 이웃 노드가 현재 노드의 파트너인 대칭적인 파트너십이 확립될 수 있다.

<147> 방법(900)은 적절한 라우팅 노드를 라우팅 테이블에 삽입하는 동작(동작 904)을 포함하고, 라우팅 노드는 제1 및 제2 방향 둘다에서 기수 및 연합 기반구조의 ID 공간의 필드 크기에 기초하여 정렬 연결 리스트로부터 식별되고, 라우팅 노드는 제1 및 제2 방향 둘다에서 정렬 연결 리스트의 대수 지표를 나타낸다. 예를 들어, ID 200, 2, 30, 46, 50, 64, 64, 64, 64, 64, 76, 83, 98, 135, 200을 갖는 노드는 ID 64를 갖는 노드의 라우팅 노드로서 라우팅 테이블에 삽입될 수 있다. 2의 기수 및 8의 필드 크기에 기초하여, 방향(321)에서 ID 64, 64, 76, 83, 98, 135, 200을 갖는 노드가 식별될 수 있고, 방향(322)에서 ID 64, 64, 50, 46, 30, 2, 200을 갖는 노드가 식별될 수 있다. 링(306)의 내부에 나타난 바와 같이, 라우팅 노드는 시계 방향(321) 및 반시계 방향(322) 양쪽 모두에서 정렬 연결 리스트(304)의 대수 지표를 나타낸다. 라우팅 노드를 삽입하는 것에 의해, 현재 노드와 라우팅 노드의 사이에, 현재 노드가 라우팅 노드의 파트너이면서 라우팅 노드가 현재 노드의 파트너인 대칭적인 파트너십이 확립될 수 있다.

<148> 도 7은 근접성 기준을 고려하는 노드 라우팅 테이블을 채우는 방법(700)의 플로우차트의 일례를 나타낸 것이다. 방법(700)은 도 5의 링과 관련하여 설명된다. 방법(700)은 현재 노드가 참가하는 계층적으로 분할된 각각의 라우팅 링에 대한 선행 노드를 라우팅 테이블에 삽입하는 동작(동작 701)을 포함한다. 각각의 선행 노드는 현재 노드가 참가하는 계층적으로 분할된 각각의 라우팅 링 내에서 제1(예를 들어, 시계) 방향으로 현재 노드에 선행한다. 계층적으로 분할된 라우팅 링은 대응하는 근접성 기준에 따라 분할되고, 양방향 연결 리스트(bi-directionally linked list)의 적어도 부분 집합(경우에 따라서는 양방향 연결 리스트 전체)을 포함한다. 예를 들어, 지정된 노드가 루트 링(501) 및 서브-링(511, 522, 523, 531, 542)에 참가하는 경우가 있을 수 있다. 따라서, 링(501) 및 서브-링(511, 522, 523, 531, 542) 각각의 내부에 속하는 지정된 노드에 대해 선행 노드가 선택된다.

<149> 방법(700)은 현재 노드가 참가하는 계층적으로 분할된 각각의 라우팅 링에 대한 후속 노드를 라우팅 테이블에 삽입하는 동작(동작 702)을 포함한다. 각각의 후속 노드는 현재 노드가 참가하는 계층적으로 분할된 각각의 라우팅 링 내에서 제1 방향으로 현재 노드에 후속한다. 예를 들어, 링(501) 및 서브-링(511, 522, 523, 531, 542) 각각의 내부에 속하는 지정된 노드에 대해 후속 노드가 선택된다.

<150> 방법(700)은 현재 노드가 참가하는 계층적으로 분할된 각각의 라우팅 링에 대한 적절한 이웃 노드를 라우팅 테이블에 삽입하는 동작(동작 703)을 포함한다. 이웃 노드는 제1(예를 들어, 시계) 방향 및 이와 반대인 제2(예를 들어, 반시계) 방향 양쪽 모두에서 이웃 범위 및 이웃 크기에 기초하여 현재 노드가 참가하는 계층적으로 분할된 라우팅 링으로부터 식별될 수 있다. 예를 들어, 링(501) 및 서브-링(511, 522, 523, 531, 542) 각각의 내부에 속하는 지정된 노드에 대해 이웃 노드가 식별될 수 있다.

<151> 방법(700)은 현재 노드가 참가하는 계층적으로 분할된 각각의 라우팅 링에 대한 적절한 라우팅 노드를 라우팅 테이블에 삽입하는 동작(동작 704)을 포함한다. 예를 들어, 링(501) 및 서브-링(511, 522, 523, 531, 542) 각각의 내부에 속하는 지정된 노드에 대해 라우팅 노드가 식별될 수 있다.



- <152> 몇몇 실시예에서, 리프 링(또는 별칭 지정을 이용하는 실시예에서는 복수의 리프 링)을 제외한, 노드 Y가 참가하는 각각의 근접 링 d에 대해 적절한 라우팅 노드가 삽입된다. 적절한 라우팅 노드는 이하의 식(들)에 기초하여 삽입될 수 있다.
- <153>  $Y.s_d.id < Y.id + b^i < Y.s_{d+1}.id$ 가 참인 경우, 링 d를 사용하거나,
- <154>  $Y.p_d.id < Y.id - b^i < Y.p_{d+1}.id$ 가 참인 경우, 링 d를 사용한다.
- <155> 상기 단계에서 링이 식별되지 않은 경우, 링 d로서 리드 링(lead ring)(예를 들어, 링(501))을 사용한다. 이제, 링 d는 노드 Y가 z에 가장 가까운 라우팅 파트너를 탐색해야 하는 근접 링이 된다.
- <156> 도 10은 메시지를 목적지 노드를 향해 라우팅하는 방법(1000)의 플로우차트의 일례를 나타낸 것이다. 방법(1000)은 도 3의 정렬 연결 리스트(304) 및 링(306)과 관련하여 설명된다. 방법(1000)은 수신 노드가 목적지를 나타내는 번호와 함께 메시지를 수신하는 동작(동작 1001)을 포함한다. 예를 들어, ID 64를 갖는 노드는 목적지가 212임을 나타내는 메시지를 수신할 수 있다.
- <157> 방법(1000)은 수신 노드가 대응하는 선행 노드보다 목적지로부터 수치적으로 더 멀리 있는지 및 대응하는 후속 노드보다 목적지로부터 수치적으로 더 멀리 있는지 중 적어도 하나를 판정하는 동작(동작 1002)을 포함한다. 예를 들어, 방향(322)에서, ID 64는 ID 50보다 목적지(212)로부터 더 멀리 있고, 방향(321)에서, ID 64는 ID 76보다 목적지(212)로부터 더 멀리 있다. 방법(1000)은 목적지가 수신 노드에 대응하는 이웃 노드 집합에 속하지 않는지를 판정하는 동작(동작 1003)을 포함한다. 예를 들어, ID 64를 갖는 노드는 목적지(212)가 83, 76, 50, 46으로 이루어진 이웃 집합에 속하지 않는 것으로 판정할 수 있다.
- <158> 방법(1000)은 수신 노드에 대응하는 라우팅 테이블에서 중간 노드를 식별하는 동작(동작 1004)을 포함하고, 중간 노드는 대응하는 라우팅 테이블 내의 다른 라우팅 노드보다 목적지에 수치적으로 더 가깝다. 예를 들어, ID 64를 갖는 노드는 ID 200를 갖는 라우팅 노드를 다른 라우팅 노드보다 목적지(212)에 수치적으로 더 가까운 것으로 식별할 수 있다. 방법(1000)은 중간 노드로 메시지를 전송하는 동작(동작 1005)을 포함한다. 예를 들어, ID 64를 갖는 노드는 ID 200를 갖는 노드로 메시지를 전송할 수 있다.
- <159> 도 11은 근접성 기준에 기초하여 메시지를 목적지 노드 쪽으로 라우팅하는 방법(1100)의 플로우차트의 일례를 나타낸 것이다. 방법(1100)은 도 4 및 도 5의 링과 관련하여 설명된다. 방법(1100)은 수신 노드가 목적지를 나타내는 번호 및 근접성 기준과 함께 메시지를 수신하는 동작(동작 1101)을 포함한다. 근접성 기준은 하나 이상의 노드 클래스를 정의한다. 수신 노드는 근접성 기준에 기초하여 하나 이상의 노드 클래스 중에서 선택된 현재 노드 클래스의 일부로서 메시지를 수신한다. 예를 들어, ID 172를 갖는 노드는 목적지가 201임을 나타내는 메시지와 목적지 노드가 링(401)에 의해 표현되는 클래스의 일부이어야 함을 나타내는 근접성 기준을 수신할 수 있다. ID 172를 갖는 노드는 링(404)의 일부로서 메시지를 수신할 수 있다.
- <160> 방법(1100)은, 선택된 노드 클래스 내의 노드들 중에서, 수신 노드가 대응하는 선행 노드보다 목적지로부터 수치적으로 더 멀리 있는지 및 대응하는 후속 노드보다 목적지로부터 수치적으로 더 멀리 있는지 중 적어도 하나를 판정하는 동작(동작 1102)을 포함한다. 예를 들어, 링(404) 내에서, ID 172를 갖는 노드는 시계 방향에서 ID 174를 갖는 노드보다 목적지(201)로부터 더 멀리 있고, 반시계 방향에서 ID 153을 갖는 노드보다 목적지(201)로부터 더 멀리 있다.
- <161> 방법(1100)은 근접성 기준에 의해 정의된 하나 이상의 노드 클래스 중 임의의 것에 대해 목적지가 수신 노드의 이웃 노드 집합에 속하지 않는지를 판정하는 동작(동작 1103)을 포함한다. 예를 들어, ID 172를 갖는 노드는 목적지(201)이 링(404) 또는 링(401) 내의 대응하는 이웃 집합에 속하지 않는 것으로 판정할 수 있다.
- <162> 방법(1100)은 수신 노드의 라우팅 테이블에서 중간 노드를 식별하는 동작(동작 1104)을 포함하고, 중간 노드는 라우팅 테이블 내의 다른 라우팅 노드보다 목적지에 수치적으로 더 가깝다. 예를 들어, ID 172를 갖는 노드는 ID 194를 갖는 노드를 링(404) 내의 다른 라우팅 노드보다 목적지(201)에 수치적으로 더 가까운 것으로서 식별할 수 있다. 방법(1100)은 중간 노드로 메시지를 전송하는 동작(동작 1105)을 포함한다. 예를 들어, ID 172를 갖는 노드는 ID 194를 갖는 노드로 수신된 메시지를 전송할 수 있다. ID 172를 갖는 노드는 앞서 정의된 부분적으로 순서화된 근접성 기준의 리스트에 따라 ID 194를 갖는 노드로 수신된 메시지를 전송할 수 있다.
- <163> 링(404) 내에서, 노드(194)는 가능한 한 목적지(201)에 가까울 수 있다. 따라서, 목적지 쪽으로의 추가의 라우팅이 그 다음 레그(next leg)에서 링(401) 내에서 행해질 수 있게 하는 데 충분한 만큼만 근접성이 완화될 수

있다. 즉, 링(404) 상에서는 목적지를 향해 더 이상 전진할 수 없기 때문에, 라우팅은 링(404)으로부터 링(401)으로 천이된다. 다른 대안으로서, ID 201를 갖는 노드가 링(401)에서 ID 194를 갖는 노드의 이웃 내에 있고, 그 결과, 더 이상 라우팅이 행해지지 않는 경우가 있을 수 있다. 따라서, 몇몇 실시예에서, 추가의 라우팅을 하기 위해서는, 그 다음 상위 링에 도달하도록 근접성 기준을 완화하는 것으로 충분하다.

<164> 그러나, 다른 실시예들에서, 추가의 라우팅이 행해질 수 있을 때까지 (또는 루트 링을 만날 때까지), 그 다음 상위 링으로의 천이를 일으키게 하는 근접성 기준의 점증적인 완화가 계속한다. 즉, 추가의 라우팅 전진이 행해질 수 있기 전에, 상위 링으로의 천이가 여러번 행해질 수 있다. 예를 들어, 이제 도 5를 참조하면, 링(531) 상에서 더 이상 라우팅 전진을 할 수 없는 경우, 링(511)으로 또는 심지어 루트 링(501)으로 천이하는 데 충분한 만큼 근접성 기준이 완화될 수 있다

#### <165> 노드 단계들

<166> 연합 기반구조에 참가하는 노드는 서로 다른 동작 단계들에서 동작할 수 있다. 노드에 대한 유효한 단계 값(phase value)은 순서화된 집합의 멤버인 것으로 정의될 수 있다. 예를 들어, {NodeId}.{InstanceIds}.{단계 값 [단계-상태 값: Inserting, Syncing, Routing, Operating].[Phase.Unknown Indication: 전송 시에 모르는 단계, 전송 시에 모르는 단계]}은 연합 기반구조 내의 주어진 노드의 단계-공간(phase-space)을 나타내는 한가지 가능한 순서화된 집합을 정의한다. 노드 인스턴스는 노드 단계-상태를 통해 Inserting에서 Syncing로, Routing로 또한 Operating으로 순서대로 천이(또는 전진)할 수 있다. 게다가, 몇몇 실시예들에서, 노드 인스턴스(node instance)는 이전의 노드 단계-상태로 다시 천이되지 않도록 구성될 수 있다. 몇몇 실시예들에서, 노드는 올라갈 때마다 그의 인스턴스 ID를 증가시킨다.

<167> 예를 들어, 노드 인스턴스는 Routing에서 다시 Syncing로 (또는 다시 Inserting으로), 이하 마찬가지로 천이하지 못하게 되어 있을 수 있다. 그에 따라, 몇몇 실시예들에서, 주어진 노드 인스턴스(예를 들어, (NodeId, InstanceId)에 의해 식별됨)가 특정의 노드 단계-상태(예를 들어, Operating)로 전진한 것을 알고 있는 경우, 이 주어진 노드 인스턴스가 이전의 노드 단계-상태로(예를 들어, Routing, Syncing, 또는 Inserting으로 다시) 복귀할 가능성이 없다는 것(몇몇 실시예에서는, 복귀하지 않는다는 것)도 알고 있다. 따라서, 특정의 노드 단계-상태 이전의 노드 단계에 있는 노드 인스턴스가 새로운 (전진된) 노드 인스턴스일 가능성이 많다.

<168> 몇몇 실시예들에서, 단계 정보 및 대응하는 인스턴스 Id(노드가 올라감에 따라 전진함)가 함께 전송된다. 따라서, 동일한 인스턴스에 대한 보다 하위의 노드 단계-상태가 보다 오래된 것으로 판정할 수 있다. 게다가, (임의의 단계-상태 값에 있는) 보다 새로운 노드 인스턴스를 알고 있는 경우, 보다 오래된 인스턴스에 관한 정보는 오래되어 쓸모 없는 것으로 간주된다.

<169> 때때로, 노드들은, 예를 들어, 처음으로 시동될 때, 정상 이탈을 통해, 또는 이상 종료(고장)의 결과로서, 서로와의 통신을 재부팅하거나 단절할 수 있다. 따라서, 임의의 노드 단계-상태에 있는 노드가 다른 노드들과의 통신을 재부팅하거나 단절할 가능성이 있다. 예를 들어, 고장으로 인해 Routing 단계-상태에 있는 노드가 재부팅하게 될 수 있다. 통신의 재부팅 또는 단절 동안에, 노드가 어떤 노드 단계-상태에 있는지를 판정할 방법이 없을 수 있다. 그에 따라, 노드가 재부팅되거나 노드와의 통신이 단절되는 경우, 노드의 단계-상태가 현재 알려져 있지 않음을 나타내기 위해 [Phase.Unknown Indication]이 설정될 수 있다. 그렇지만, 노드에 대한 이전에 표현된 및/또는 검출된 단계-상태가 유지될 수 있고 손실되지 않는다.

<170> [Phase.Unknown Indication]은 단계-상태 값이 전송될 때 단계-상태를 알고 있는지(예를 들어, phase.unknown이 설정되지 않은 단계 값) 또는 단계-상태가 이전에 표현된 단계-상태이고 단계-상태 값이 전송될 때 단계-상태를 모르고 있는지(예를 들어, phase.unknown이 설정되어 있는 단계 값)를 나타내는 데 사용될 수 있다. 따라서, 노드의 단계(그의 단계 값)는 단계-상태값 및 phase.unknown 표시 둘다를 사용하여 표현될 수 있다.

#### <171> 가입 프로토콜

<172> 때때로, 노드들은 기존의 연합에 가입 및 그로부터 이탈할 수 있다. 노드는 연합에 가입 및 이탈하기 위해 적절한 프로토콜을 실시할 수 있다. 예를 들어, 노드는 기존의 연합의 일부가 되기 위해 Join() 함수를 실시할 수 있다. Join() 함수를 실시하는 노드는, 최종적인 동작(operating) 단계-상태에 도달하기 전에, 3개의 순서화된 단계-상태, 즉 삽입(inserting) 단계-상태, 동기화(synchronizing) 단계-상태, 및 라우팅(routing) 단계-상태를 거쳐 천이할 수 있다. 다른 실시예들에서, 이들 특정의 순서화된 단계-상태들이 존재할지 않을 수 있는 반면, 다른 단계-상태들이 정의될 수 있다. 도 12a는 노드가 연합 기반구조 내에 멤버십을 확립하는 것의 일례를 나타낸 것이다. 도 12b는 연합 기반구조 내의 노드들이 메시지를 교환하는 것의 일례를 나타낸 것이다.

- <173> **삽입 단계:** 노드 Y5는 그의 노드 ID를 적어도 포함함과 동시에 연합에의 가입 동작을 나타내는 가입 메시지를 발행함으로써 이 단계-상태에 들어간다. 가입 메시지는 새로 가입하는 노드(노드 Y)에 의해 전송된 라우팅된 메시지일 수 있으며, 이 가입 메시지의 목적지 속성은 새로 가입하는 노드의 ID로 설정되어 있다. 이 단계-상태에서, 새로 가입하는 노드는 연합에서 그의 선행 노드와 후속 노드 사이에 삽입된다. 삽입 단계-상태는 이하의 알고리즘에 따라 실시될 수 있다(모든 계산이 모듈로  $b^n$ 으로 수행됨).
- <174> IP1. Y는 연합에 가입하고자 하는 가입 노드가 속해 있는 최하위 링의 일부로 이미 되어 있는 기존의 노드를 식별한다. 이것은 정적으로 구성되거나 DHCP 및/또는 DNS 및/또는 WS-Discovery 또는 (경우에 따라서는 공지되어 있는) 상수를 사용하여 동적으로 발견될 수 있다. 이 기존의 연합 노드를 E라고 하자.
- <175> IP2. Y는 노드 Y가 참가하는 모든 근접 링에서 Y.id에 수치적으로 가장 가까운 ID를 갖는 노드 X를 결정하기 위해 E.RouteNumerically(Y, joinMsg)를 호출한다. 이것은 다수의 노드로 가입 메시지를 라우팅하는 것을 포함할 수 있다.
- <176> IP3. 수치적 후속 노드(s) 및 선행 노드(p)를 결정한다. (유의할 점은 이하의 삽입을 행하는 데 필요한 데이터가 가입 메시지 및 그의 응답으로 전달될 수 있다는 것이다. 그 자체로서, 부가의 왕복(roundtrip)이 필요하지 않다.)
- <177> 사례 1:  $X.id > Y.id$
- <178>  $Y.s = X, Y.p = X.p, X.p.s = Y$ , 및  $X.p = Y$
- <179> 사례 2:  $X.id < Y.id$
- <180>  $Y.p = X, Y.s = X.s, X.s.p = Y$ , 및  $X.s = Y$
- <181> 가입 메시지에 응답하여, 노드 X(가입 메시지를 처리한 노드)는 가입 응답을 노드 Y로 반송할 수 있다. 가입 응답은 노드 Y에 대한 선행 노드(Y.p) 및 후속 노드(Y.s)를 알려줄 수 있다. 노드 Y는 가입 응답을 수신하고 그의 선행 노드 및 후속 노드를 알아내기 위해 가입 응답을 처리할 수 있다. 가입 응답을 처리한 후에, 노드 Y는 연합에서 약한 라우팅 참가자(weak routing participant)로 될 수 있다. 예를 들어, 노드 Y는 그에게로 전송된 메시지를 그의 후속 노드 또는 선행 노드로 단순히 전달하기만 할 수 있다. 따라서, 노드 Y가 연합 기반 구조에 삽입되지만, 라우팅 및 이웃 테이블이 채워지지 않는다. 이 시점에 도달하기 전에, 노드 Y는 노드 Y의 생존 단계(liveness phase)가 삽입 단계-상태에 있음을 나타내는 상태 메시지를 전송측 노드로 반환함으로써 자신에게 메시지를 전송하는 다른 노드들에게 자신에게로 전송된 메시지를 다른 노드를 통해 리디렉션하도록 요청한다.
- <182> 일반적으로, 때때로, 노드는 동기 요청 및 응답 메시지를 교환할 수 있다. 동기 요청 및 동기 응답 메시지는 송신자의 관점에서 볼 때 다른 노드들에 대한 생존 정보(예를 들어, 헤더)를 포함할 수 있다. 이웃에 있는 애플리케이션 계층들이 서로의 상태를 알도록 이웃 상태도 동기 요청 및 응답 메시지에 포함될 수 있다. 동기 요청 및 응답 메시지가 교환되는 때의 일례는 가입하는 노드의 동기화 단계-상태 동안이다. 그렇지만, 동기 요청 및 응답 메시지는 다른 동작 단계-상태 동안에도(예를 들어, 동작 단계-상태에 있는 동안) 교환될 수 있다.
- <183> 도 16은 메시지 모델 및 관련 처리 모델(1600)의 일례를 나타낸 것이다. 도 16에 나타낸 바와 같이, 노드는 동기 요청 메시지를 전송 및 수신할 수 있다. 예를 들어, 동기 요청 메시지(1601)는 함수 계층(1651)에서 새로 삽입된 노드(예를 들어, ID 144를 갖는 도 12b의 노드)로부터 수신될 수 있다. 애플리케이션 데이터(1602)(예를 들어, 이름공간 가입)는 동기 요청 메시지(1601)에 피기백될 수 있다. 함수 계층(1651)은 동기 요청 메시지에 포함된 애플리케이션 데이터를 애플리케이션 계층(1652)에 알려줄 수 있다. 예를 들어, 함수 계층(1651)은 애플리케이션 계층(1652)에 대한 이웃 상태 동기 이벤트(1603)(애플리케이션 데이터(1602)를 포함함)를 호출할 수 있다. 애플리케이션 데이터(1607)를 포함하는 동기 요청(1631)은 또한 처리 모델(1600)에서의 동기 요청(1601)의 처리와 유사하게 동기 요청(1631)을 처리하는 다른 노드로도 전송될 수 있다.
- <184> 어떤 함수 계층 이벤트(예를 들어, 동기 요청 메시지(1601), 동기 응답 메시지(1641), 또는 ping 메시지(1612))에 응답하여, 함수 계층(1651)은 애플리케이션 계층(1652)에 있는 이웃 상태 요청 함수(1604)를 호출할 수 있다. 이웃 상태 요청(1604)은 이웃에서 전달될 필요가 있는 상태를 획득하기 위한 애플리케이션 계층에 대한 요청이다. 이웃 상태 요청(1604)에 응답하여, 애플리케이션 계층(1652)은 선택적인 애플리케이션 데이터(1607)를 포함하는 이웃 상태(1606)를 함수 계층(1651)에 제공할 수 있다. 다른 대안으로서, 애플리케이션 계층(1652)은 어떤 애플리케이션 계층 이벤트에 응답하여 선택적인 애플리케이션 데이터(1607)를 포함하는 이웃 상

태(1606)를 전송할 수 있다. 상기한 것과 유사한 내부 메카니즘을 사용하여, 함수 계층(1651)은 애플리케이션 계층 이웃 상태를 전달하기 위해 선택적인 애플리케이션 데이터(1607)를 포함하는 동기 응답 메시지(1608)를 전송할 수 있다.

- <185> **동기화 단계:** 가입 응답 메시지를 처리한 후에, 노드 Y는 삽입 단계-상태에서 동기화(Syncing) 단계-상태로 천이한다. 동기화 단계-상태에서, 새로 삽입된 노드 Y는 이웃에 있는 노드들과 정보를 동기화한다. 일반적으로, 노드 Y는 삽입 단계-상태에서 식별된 적어도 그의 선행 및 후속 노드에 동기 메시지를 전송할 수 있다. 동기 메시지를 처리하는 이들 노드는 이들 처리 노드의 대응하는 이웃 및 라우팅 파트너 노드를 알려주는 동기 응답을 반환할 수 있다. 더 구체적인 예에서, 동기화 단계-상태가 이하의 알고리즘에 따라 구현될 수 있다(모든 계산은 모듈로  $b^n$ 으로 수행됨).
- <186> SP1. 노드 Y가 참가하는 각각의 근접 링에서 Neighborhood(Y.s) 및 Neighborhood(Y.p) 노드의 합집합으로부터 Neighborhood(Y)를 계산한다. 합집합 계산은 다음과 같이 행해질 수 있다.
- <187>  $s_j.s.id > (Y.id + u/2)$ ,  $j \geq v/2-1$ ,  $p_k.p.id < (Y.id - u/2)$ , 및  $k \geq v/2-1$ 인 ( $s_j, \dots, s_1, s, p, p_1, \dots, p_k$ )
- <188> SP2. 도 16을 간략히 참조하면, 선택적인 애플리케이션 관련 이웃 데이터(예를 들어, 애플리케이션 관련 데이터(1607))를 획득하기 위해 이웃 상태 요청(예를 들어, 이웃 상태 요청(1604))을 통해 Y의 로컬 애플리케이션 계층(예를 들어, 애플리케이션 계층(1652))에 질의를 한다.
- <189> SP3. Y의 관점에서 볼 때 각각의 근접 이웃 및 라우팅 파트너의 적어도 생존 상태 정보를 포함하는 적어도 근접 후속 및 선행 노드로 동기화 메시지를 전송한다. SP2를 통해 액세스되는 선택적인 애플리케이션 관련 이웃 데이터(예를 들어, 애플리케이션 데이터(1607))가 동기화 요청(1631)에 포함된다.
- <190> SP3. Y는 SP2에서 전송된 동기 메시지를 처리하는 노드들로부터 다시 동기 응답 메시지를 수신한다. 예를 들어, 노드 Y는 그의 계산된 이웃 내의 하나 이상의 노드와 동기화 메시지(요청/응답)를 교환할 수 있다. 노드 Y의 이웃 노드들 중 적어도 하나 및 경우에 따라서는 그 모두와 동기화 메시지가 교환된 후에, 계산된 이웃 노드들은 동기화된 데이터를 전달하기 위해 추가의 메시지를 교환할 수 있다. 동기화 메시지(요청 또는 응답)는, 예를 들어, 노드 이웃에 있는 타겟 노드와 그의 데이터를 사전 예방적으로 동기화시키기 위해 노드에 의해 전송되는 비라우팅되는 메시지일 수 있다.
- <191> SP4. SP3에서의 동기 응답 메시지(예를 들어, 동기 응답 메시지(1641))가 수신될 때, 이들 수신된 동기 응답 메시지에 존재하는 선택적인 애플리케이션 관련 이웃 데이터(예를 들어, 애플리케이션 데이터(1622))가 이웃 상태 동기 이벤트(1603)를 통해 Y의 애플리케이션 계층(1652)에 제공될 수 있다.
- <192> 동기화 단계-상태의 일부로서, 근접 후속 노드(예를 들어, Y.s) 및 선행 노드(Y.p)는 그의 라우팅 테이블을 새로 삽입된 노드(예를 들어, Y)와 교환한다. 동기 메시지를 수신하는 노드는 동기 응답을 전송하는 것으로 응답할 수 있다. 동기 응답은 응답측 노드의 관점에서 보는 것을 제외하고는 동기화 메시지와 유사하게 데이터를 전달한다. 동기 메시지 및 동기 응답 둘다는 애플리케이션 데이터를 전달(또는 피기백)할 수 있다. 따라서, 애플리케이션 데이터가 동기화 단계-상태 동안에 노드들 간에 전달될 수 있다. 동기화 단계-상태가 완료될 때, 노드는 자신에게로 온 메시지를 단순히 후속 노드 또는 선행 노드로 전달하는 대신에 이들을 처리할 수 있다. 그렇지만, 노드의 라우팅 테이블이 채워져 있지 않기 때문에, 노드는 여전히 약한 라우팅 참가자로 볼 수 있다.
- <193> **라우팅 단계:** 동기화 단계-상태가 완료된 후에, 노드는 라우팅 단계-상태로 천이한다. 라우팅 단계-상태에서, 새로 동기화된 노드(예를 들어, 노드 Y)는 그의 라우팅 노드를 계산한다. 라우팅 단계-상태가 이하의 알고리즘에 따라 구현될 수 있다(모든 계산이 모듈로  $b^n$ 으로 수행됨).
- <194> RP1. 라우팅 단계-상태가 균형 절차(balancing procedure)(나중에 설명함)의 일부로서 실행 중인 경우, 후속 노드(Y.s) 및 선행 노드(Y.p)가 노드 Y가 참가하는 모든 근접 링에 존재하도록 한다. 어느 하나가 존재하지 않는 경우, 고려 중인 링 내의 이웃 노드들 중에서 그 다음으로 가장 양호한 후속 또는 선행 노드를 선택함으로써 고장난 노드(들)에 대한 대체 노드를 결정한다.
- <195> RP2.  $1 < i < n-1$ 인 경우
- <196> RP2a.  $z = Y.id \pm b^i$ 를 계산한다.



- <197> RP2b. 링 d가 가장 관련있는 근접 링이 아닌 경우, 노드 Y가 참가하고 조건  $Y.s_d.id < Y.id + b^i < Y.s_{d+i}.id$  또는  $Y.p_d.id < Y.id - b^i < Y.p_{d+1}.id$ 를 만족시키는 근접 링 d를 찾아낸다. 그렇지 않은 경우, 링 d를 가장 관련있는 근접 링으로 삼는다. 링 d는 노드 Y가 z에 가장 가까운 라우팅 파트너를 탐색해야만 하는 근접 링이다. Q가  $Y.s_d.r_{\pm i}$ 과  $Y.p_d.r_{\pm i}$  사이에서 z에 수치적으로 가장 가까운 노드라고 하자.  $|Q.id - z|$ 가  $b^i$ 의 구성가능한 퍼센트(통상적으로 20%) 내에 있는 경우, 단순히  $Y.r_{\pm i} = Q$ 가 된다. Q.id가  $(Y.s_d.id \pm b^i)$  또는  $(Y.p_d.id \pm b^i)$ 보다 z에 더 가까운 경우, 이는 노드 Y가  $Y.s_d$  또는  $Y.p_d$ 보다 근접 링 d 내의 노드 Q에 대한 더 나은 파트너 라우팅 노드라는 것을 의미한다. 따라서, updateMsg를 노드 Q로 전송하고, 아직 전송되지 않은 경우, 노드 Q가 노드 Y를  $r_{-i}$ 에서 그의 파트너 라우팅 노드로 확립할 수 있도록 i 및 노드 Y를 파라미터로서 제공한다.
- <198> RP2c. 이 단계-상태가 균형 절차의 일부로서 실행 중이고  $Y.s_d.r_{\pm i}.id == Y.p_d.r_{\pm i}.id$ 인 경우,  $(Y.s_d.id \pm b^i)$ 와  $(Y.p_d.id \pm b^i)$  사이의 수치 범위에 단지 하나의 노드가 있다. 그 노드는 후속(또는 선행) 노드의 라우팅 노드  $r_{\pm i}$ 가 가리키는 노드이다. 따라서, 단순히  $Y.r_{\pm i} = Y.s_d.r_{\pm i}$ 로 된다.
- <199> RP2d. 그렇지 않은 경우, 근접성 기준을 링 d의 근접성 기준으로 설정한 상태에서 노드 Q에 대해 RouteProximally를 호출함으로써 라우팅 파트너  $Y.r_{\pm i}$ 를 계산한다. 이것은  $Y.r_{\pm i} = Q.RouteProximally(z, updateMsg, d)$ 를 의미한다.
- <200> RP3. 이 시점에서, 노드 Y는 자신에게로 오는 메시지를 처리할 뿐만 아니라 메시지를 라우팅할 수도 있다.
- <201> RP4. 파트너 라우팅 노드의 종단점 ID에 대한 애플리케이션 계층으로부터 전송된 생존 통지 이벤트에 아직 가입하지 않은 경우, 가입을 한다. 또한, 더 이상 파트너 라우팅 노드가 아닌 노드에 대한 애플리케이션 계층과 이전에 확립된 생존 이벤트 가입을 취소한다. 예를 들어, 가입 및/또는 취소 요청이 대응하는 애플리케이션(예를 들어, 이름공간 애플리케이션)에 대한 pub-sub 논리를 구현하는 애플리케이션 계층(예를 들어, 애플리케이션 계층(121))으로 위로 전달될 수 있다. 차후의 애플리케이션 관련 생존 메시지(예를 들어, 이름공간 가입의 결과 생기는 메시지)가 애플리케이션 계층에서 수신될 때, 통지(이벤트)가 처리를 위해 다른 하위 계층(예를 들어, 다른 하위 계층(131))으로 푸시다운될 수 있다.
- <202> 도 17은 함수 계층(1751)과 애플리케이션 계층(1752) 사이에서 일어날 수 있는 다수의 생존 상호작용의 일례를 나타낸 것이다. 도 17에 도시된 바와 같이, 종단점은, 예를 들어, 다양한 노드를 표현하는(예를 들어, URL 또는 URI로 표현됨) 게시/가입 주제(publish/subscribe topic)이고, 예를 들어, 연합 기반구조 노드일 수 있다. 생존 이벤트에의 가입(Subscribe To Liveness Event)(1701)은 생존 이벤트(예를 들어, 게시/가입 주제) 가입하기 위해 함수 계층(1751)로부터 애플리케이션 계층(1752)으로 호출될 수 있다. 생존 가입 취소(Revoke Liveness Subscription)(1702)은 생존 이벤트에의 가입을 취소하기 위해 함수 계층(1751)으로부터 애플리케이션 계층(1752)으로 호출될 수 있다. 종단점 고장(End Point Down)(1703)은 종단점이 고장났을지도 모른다는 것을 알려주고 함수 계층(1751)에 선택적인 대체 종단점을 제공하기 위해 애플리케이션 계층(1752)으로부터 함수 계층(1751)으로 전송될 수 있다. 종단점 고장 이벤트(End Point Down event)(1703)은 이전의 가입(예를 들어, 생존 이벤트에의 가입(1701))에 기초하여 비동기적으로 전송될 수 있다.
- <203> 노드 고장(Node Down)(1704)은 함수 계층(1751)(또는 어떤 다른 하위 계층)이 고장난 노드를 검출하였음을 알려주고 선택적으로 애플리케이션 계층(1752)에 대체 노드를 제공하기 위해 함수 계층(1751)으로부터 애플리케이션 계층(1752)으로 호출될 수 있다. 애플리케이션 계층(1752)은 그 후에 어쩌면 고장난 노드가 검출되었음을 다른 관심있는 당사자들에게 전달할 수 있다. 노드 고장 이벤트(1704)는 함수 계층(1751) 또는 어떤 다른 하위 계층이 어쩌면 고장난 노드를 검출할 때마다 비동기적으로 전송될 수 있다. 생존 전송(Send Liveness)(1706)은 애플리케이션 계층(1752)이 (예를 들어, 노드 고장 이벤트(1704)로부터 또는 어떤 다른 대역외 메카니즘으로부터) 노드가 고장났음을 검출할 때 애플리케이션 계층(1752)으로부터 함수 계층(1751)으로 호출될 수 있다. 생존 전송 이벤트(1706)는 함수 계층(1751)으로 하여금 생존 메시지를 전송하게 할 수 있다. 생존 전송 이벤트(1706)는 또한 애플리케이션 계층(1752)이 노드가 고장났음을 검출할 때마다 비동기적으로 호출될 수 있고, (생존에의 가입을 통해) 이전에 확립된 가입에 의존하지 않는다.

- <204> 따라서, 몇몇 실시예들에서, 함수 계층(1751)이 재귀적으로 사용된다. 예를 들어, 함수 계층(1751)은 지정된 노드(예를 들어, 위 또는 아래의 특정 노드)에 대한 관심을 애플리케이션 계층(1752)에 나타낼 수 있다. 애플리케이션 계층(1752)은 지정된 노드에 관련된 통지들에 대한 애플리케이션 관련 가입을 작성하고 이어서 작성된 가입을 다른 연합 노드 내의 적절한 대응하는 애플리케이션 계층(1752) 인스턴스로 전달하기 위해 함수 계층(1751)을 재사용할 수 있다. 예를 들어, 연합 노드 내의 애플리케이션 계층(1752)이 이름 공간 pub/sub 거동을 구현하는 경우, 함수 계층(1751)은 지정된 노드에 대한 통지를 관리하는 게시/가입 관리자(publish/subscribe manager)로 이 가입을 라우팅할 수 있으며, 이 pub/sub 관리자는 관련된 연합 노드 내의 애플리케이션(1752)의 적어도 일부로서 구현된다. 그에 따라, 함수 계층(1751)은 함수 계층(1751)에 의해 발생된 가입을 라우팅하기 위해 사용된다. 유사한 재귀적 메카니즘이 탈퇴하거나 지정된 노드에 더 이상 관심이 없음을 나타내는 데 사용될 수 있다.
- <205> **동작 단계:** 라우팅 단계-상태가 완료된 후에, 노드는 동작 단계-상태로 천이한다. 노드는 고장(예를 들어, 재부팅)이 있을 때까지 동작 단계-상태에 있을 수 있다. 동작 단계-상태에서, 노드는 때때로 라우팅 파트너에게 갱신 메시지를 전송할 수 있다. 갱신 메시지(갱신 요청 및 갱신 응답 둘다)는 전송측 노드에 대한(예를 들어, 관심의 모든 근접 이웃에 대한) 이웃 노드 생존 정보를 포함할 수 있다. 이러한 전송된 생존 정보는 또한 송신자의 생존 정보도 포함할 수 있다. 갱신 메시지는 노드가 자신의 라우팅 파트너 노드를 주기적으로 갱신하기 위해 발신하는 라우팅된 메시지일 수 있다. 애플리케이션 데이터는 라우팅 파트너 갱신 동안에 전달될 수 있도록 갱신 메시지에 피기백될 수 있다. 메시지 목적지는 원하는 라우팅 인덱스에 있는 완전 라우팅 파트너(perfect routing partner)의 ID로 설정된다. 이 메시지의 Message ID 속성은 이 메시지를 처리하는 노드(들)가 최신 메시지를 판정할 수 있게 해주기 위해 애플리케이션 순서 번호를 할당받으며, 이 메시지는 근접 라우팅된다.
- <206> 갱신 메시지를 수신하는 노드는 갱신 응답으로 응답할 수 있다. 갱신 응답은 데이터가 응답측 노드의 관점에서 본 것임을 제외하고는 갱신 메시지와 동일한 데이터를 전달한다. 갱신 메시지 및 갱신 응답의 교환을 통해, 노드는 라우팅 정보를 교환할 수 있다. 때때로, 동작 노드는 라우팅 파트너를 갱신할 수 있다.
- <207> 때때로, 동작 노드는 또한 ping 메시지(예를 들어, ping 메시지(1609, 1611))를 전송할 수 있다. ping 메시지는 노드가 그의 존재를 주기적으로 공표하고 그의 이웃 내에서 그의 이웃/라우팅 노드에 관한 정보를 배포하며(예를 들어, 피기백된) 애플리케이션 데이터를 복제하기 위해 전송하는 단방향 메시지이다.
- <208> 발신지 노드는 그의 중간 선행 및 후속 이웃 노드 중 하나 이상으로 ping 메시지를 전송할 수 있다. 따라서, ping 배포 패턴(즉, 어느 노드가 ping 메시지를 전송받았는지)에 따라, 발신지 노드에 관련된 정보가 발신지 노드의 이웃 내의 링 상의 다른 노드들로 전달된다. 예를 들어, 발신지 노드는 그의 중간 선행 및 후속 노드에게만 ping 메시지를 전송할 수 있고, ping 메시지는 발신지 노드의 위치(노드 ED)로부터 외부로 링을 따라 양방향으로 발신지의 이웃의 경계쪽으로 전달된다. 다른 대안으로서, 발신지 노드는 그의 이웃 내에서 그의 선행 및 후속 노드 방향 둘다에서 매 n번째 노드로 ping 메시지를 전송할 수 있다.
- <209> ping 메시지를 수신하는 각각의 노드는 이웃 범위 관점에서 볼 때의 발신지 노드에 대한 그의 관심을 검사한다. 관심이 없는 경우, 노드는 ping 메시지를 버린다. 관심이 있는 경우, 노드는 ping 메시지를 처리하고, ping 메시지의 전달이 발신측 노드의 이웃으로 제약되어 있는 경우, 이 ping 메시지를 그의 지정된 ping 패턴에 따라 전달한다. 예를 들어, ping 메시지를 처리한 후에, 수신측 노드는 송신측 및 발신지 노드가 그의 선행 노드 집합에 있는 경우 ping 메시지를 적어도 그의 후속 노드로 전달하거나 송신측 및 발신지 노드가 그의 후속 노드 집합에 있는 경우 ping 메시지를 적어도 그의 선행 노드로 전달할 수 있다.
- <210> 따라서, ping 메시지가 발신지 노드 주변의 이웃 노드 집합의 경계에 도달할 때 이 메시지의 외부 전달이 중단된다. ping 메시지의 Message ID 속성은 이 메시지를 처리하는 노드가 발신지 노드로부터의 최신 메시지를 판정하고 중복 처리 또는 불필요한 전달을 피하기 위해 애플리케이션 순서 번호를 할당받는다.
- <211> 다시 도 16을 참조하면, ping 메시지(1609)가 이웃 노드로부터 함수 계층(1651)에 수신될 수 있다. 애플리케이션 데이터(1612)(예를 들어, 이름공간 가입)는 ping 메시지(1609)에 피기백될 수 있다. 함수 계층(1651)은 ping 메시지에 포함된 애플리케이션 데이터를 애플리케이션 계층(1652)에 알려줄 수 있다. 이와 유사하게, 함수 계층(1651)은 동기 요청 메시지에 포함된 애플리케이션 데이터를 애플리케이션 계층(1652)에 알려줄 수 있다. 이들 전달 사례 둘다는 애플리케이션 데이터(1612)를 포함하는 이웃 상태 동기 이벤트(1603)를 애플리케이션 계층(1652)에 전송함으로써 달성될 수 있다.

- <212> 어떤 함수 계층 이벤트(예를 들어, 수신된 ping 메시지(1609))에 응답하여, 함수 계층(1651)은 애플리케이션 계층(1652)에 이웃 상태 요청(1604)을 전송할 수 있다. 이웃 상태 요청(1604)은 이웃에서 선택적으로 전달될 필요가 있는 상태를 획득하기 위해 애플리케이션 계층(1652)에 대해 호출된다. 이웃 상태 요청(1604)에 응답하여, 애플리케이션 계층(1652)은 선택적인 애플리케이션 데이터(1607)를 포함하는 이웃 상태(1606)를 함수 계층(1651)에 반환할 수 있다. 함수 계층(1651)은, 이웃 및 라우팅 파트너 노드 생존 정보는 물론 선택적인 애플리케이션 계층 이웃 상태를 전달하기 위해, 선택적인 애플리케이션 데이터(1607)를 포함하는 ping 메시지(1611)를 전송할 수 있다. 함수 계층(1651)은 또한 애플리케이션 상태를 전달하기 위해 선택적인 애플리케이션 데이터(1607)를 포함하는 동기 응답(1608)도 전송할 수 있다.
- <213> **이탈 프로토콜**
- <214> 노드가 연합으로부터 이탈하는 것이 적절한 경우, 노드는 연합으로부터 정상적으로 제거되기 위해 Depart 함수를 실시할 수 있다. 노드는 그의 직전 근접 선행 및 직후 근접 후속 노드와 경우에 따라서는 동일한 근접 이웃 내의 다른 노드들 중 하나 이상으로 이탈 메시지를 전송함으로써 기존의 연합으로부터 이탈한다. 따라서, 이탈 배포 패턴(즉, 어느 노드가 이탈 메시지를 전송받는지)에 따라, 이탈하는 노드에 관련된 정보가 이탈하는 노드의 이웃 내에서 링 상의 다른 노드들로 전달된다. 이탈 메시지는 정상 이탈하는 노드가 그의 임박한 이탈에 관하여 그의 근접 이웃들 중 적어도 하나 내의 하나 이상의 다른 노드들에 알려주기 위해 발신하는 단방향 메시지이다. 이탈하는 노드는 ping 메시지의 전달과 유사한 방식으로 (예를 들어, 그의 이웃 내에서) 이탈 메시지를 전달한다. 예를 들어, ID 30을 갖는 노드는 ID 17 및 40을 갖는 노드로 이탈 메시지(1219)를 전송할 수 있다. ID 30을 갖는 노드는 이어서 주어진 근접 링의 관점에서 볼 때 연합 기반구조로부터 그 자신을 제거할 수 있다. 유의할 점은 노드가 그가 속할지도 모르는 다른 이웃들이 아니라 하나의 근접 이웃으로부터 그 자신을 제거할 수 있다는 것이다.
- <215> ID 30을 갖는 노드가 제거된 후에는 ID 17 및 40을 갖는 노드(즉, 선행 노드 및 후속 노드)가 ID 30에 가장 가까운 노드일 가능성이 있기 때문에, ID 17 및 40을 갖는 노드는 ID 30을 갖는 노드의 이탈을 알게 된다. 따라서, ID 30으로 배달되는 장래의 메시지들은 ID 17 및 40을 갖는 노드에서 적절히 처리될 수 있다. ID 17 및 40을 갖는 노드는 ID 30을 갖는 노드의 이탈을 링(1205) 상의 다른 노드들에 전달할 수 있다. ID 30을 갖는 노드가 없는 경우, ID 17 및 40을 갖는 노드는 어쩌면 서로를 가리키고 있는 선행 노드 및 후속 노드 포인터도 재계산할 수 있다.
- <216> 이탈 메시지의 Message ID 속성은 이탈 메시지를 처리하는 노드가 발신지 노드에 의해 전송된 일련의 ping 및 이탈 메시지 중에서 최신 메시지를 결정하기 위해 ping 메시지와 동일한 애플리케이션 순서 ID를 할당받는다. 연합 근접 링으로부터의 정상 이탈은 선택적이지만 권고된다. 그렇지만, 연합은 노드가 갑작스럽게 이탈하는 경우 자가-치유하도록 설계되어 있다.
- <217> **생존**
- <218> 연합의 수명 동안에, 노드들은 연합을 유지하기 위해 생존 정보를 교환할 수 있다. 생존 정보는 생존 메시지 헤더의 형태로 연합 내에서 교환되는 거의 모든 메시지 내에 포함될 수 있다. 예를 들어, 가입 메시지, 가입 응답, 동기 메시지, 동기 응답, 갱신 메시지, 갱신 응답, 애플리케이션 관련 메시지, 생존 메시지, 및 ping 메시지 모두는 생존 정보 헤더를 포함할 수 있다. 연합 노드가 메시지 또는 응답을 전송할 때, 그 노드는 다른 노드들이 처리하기 위한 생존 정보를 포함시킬 수 있다. 생존 정보는 생존 메시지의 생존 정보 헤더에 포함될 수 있다.
- <219> 노드의 생존 상태를 나타내는 생존 정보는 이하의 속성들을 사용하여 표현될 수 있다.
- <220> [Node]: 어느 노드의 생존 상태가 표현되고 있는지를 식별해준다. [Instance ID]를 더 포함하는 [Reference Properties]에 기초하여 노드가 식별될 수 있다.
- <221> [Reference Properties]: WS-어드레싱 규격에 명시된 요소 정보 항목. WS-어드레싱은 참조 속성 집합에 포함시키기 위한 [Instance ID] 참조 속성을 정의한다.
- <222> [Instance ID]: 노드의 특정 인스턴스를 식별해주는 번호. 증분하는 부트 카운트가 노드의 인스턴스 ID로서 사용될 수 있다.
- <223> [Phase]: 식별된 노드의 단계를 전달한다.
- <224> [Phase-State Value] 표시된 노드 인스턴스가 달성한 것으로 알려진 최상위 단계-상태(inserting,

synchronizing, routing, operating)를 전달한다.

- <225> [Phase.Unknown Indication] 현재의 단계를 알고 있는지 모르고 있는지를 전달하는 표시자
- <226> [Freshness]: 정보의 신선도(freshness)를 전달하며, 그의 값은 0 내지 MaxFreshness의 범위에 있다. 값이 높을수록, 정보가 더 신선하며, 0은 정보가 없음을 암시하고 MaxFreshness는 프로토콜 정의 상수이다.
- <227> [Color]: 노드가 속해 있는 근접성 등가 클래스를 식별해준다. 동일한 컬러값을 갖는 2개의 노드는 항상 가장 가깝게 근접해 있는 것으로 간주되는데, 그 이유는 이들 노드 둘다가 컬러값에 의해 식별되는 동일한 등가 클래스에 속해 있기 때문이다. 근접성 등가 클래스의 수는 시간이 경과하면서 더 많은 노드가 연합에 가입함에 따라 증가할 수 있다.
- <228> [Weight]: 노드 성능 메트릭을 제공하며 그의 값은 0 내지 Max Weight의 범위에 있다. 이는 대단한 계산 능력, 높은 네트워크 대역폭, 및 긴 가동시간 등의 연합 노드의 바람직한 특성을 측정한다. 값이 높을수록, 노드가 능력이 더 많으며 파트너쉽 측면에서 볼 때 노드를 더 바람직하게 만들어준다.
- <229> 어떤 환경에서, 노드의 [Node] 및 [Freshness] 속성은 [Origin] 및 [Sender] 메시지 헤더 등의 더 큰 범위에서 암시적으로 또는 명시적으로 전달되며, 그 자체로서 상기의 속성들을 생존 헤더에 또다시 포함시키는 것은 중복적인 것이 된다. 예를 들어, 메시지의 송신자는 그의 현재 단계, 컬러, 및 가중치 정보만을 전달하면 되는데, 그 이유는 그의 ID, Instance Id가 메시지 어드레싱 헤더에 제공되어 있고 그의 Freshness가 암시되어 있기 때문이다.
- <230> 생존 상태는 다음과 같이 정의되는 "<" 이항 관계에 기초하여 적어도 부분적으로 순서화되어 있을 수 있다.
- <231> 다음과 같은 경우에 "L1 < L2"이 참이다.
- <232> 1. "L1.[Node].[Name] == L2.[Node].[Name]"이 참이고 이하의 것들 중 하나가 참이며(단, 열거된 순서로 테스트가 수행되고 단락됨)
  - <233> • "L1.[Node].[Reference Properties].[Instance ID] < L2.[Node].
  - <234> [Reference Properties].[Instance ID]
  - <235> • L1.[Phase.Unknown Indication] != true AND L2.[Phase.Unknown
  - <236> Indication] != true AND L1.[Phase-State] < L2.[Phase-State]
  - <237> • L1.[Freshness] < L2.[Freshness]
- <238> 2. 또는 "L1.[Color] == L2.[Color]"가 참이고 이하의 것들 중 하나가 참임(단, 열거된 순서로 테스트가 수행되고 단락됨)
  - <239> • L1.[Phase-State] < L2.[Phase-State]
  - <240> • L1.[Weight] < L2.[Weight]
- <241> 게다가, 지정된 노드가 이용가능하지 않게 되었음(예를 들어, 고장났음)이 검출되거나 의심되는 경우, 생존 "고장" 메시지가 지정된 노드로 전송될 수 있다. 예로서, 애플리케이션 계층(예를 들어, 애플리케이션 계층(121))이 다른 애플리케이션 계층(예를 들어, 애플리케이션 계층(123)) 또는 그 다른 애플리케이션 계층을 호스팅하는 노드가 고장났음을 검출할 때, 검출하는 애플리케이션 계층은 그 노드가 고장났을지도 모른다는 것을, 예를 들어, 메시지 모델 및 관련 처리 모델(1600 및/또는 1700)에 따라 다른 하위 계층들(예를 들어, 다른 하위 계층들(131))에 통지할 수 있다. 이러한 통지는, 예를 들어, 함수 계층(1651) 등의 다른 하위 계층들로 하여금 생존 고장 메시지(liveness down message)를 전송하게 할 수 있다. 이것은 생존 고장 메시지의 발생에 대한 자극의 일례에 불과하다.
- <242> 생존 고장 메시지가 라우팅되고 따라서 고장난 것으로 의심되는 노드들에 가장 가까운 노드로 전달되기 때문에, 지정된 노드에 대한 생존 고장 메시지가 다시 지정된 노드로 전달되는 경우, 지정된 노드가 결코 고장나지 않았거나 지정된 노드가 다른 인스턴스이다(예를 들어, 다른 인스턴스 ID를 가짐). 반면에, 생존 고장 메시지가 다른 노드로 전달되는 경우, 이는 지정된 노드가 고장난 것처럼 보인다는 것을 나타낸다. 그에 따라, 생존 고장 메시지를 수신하는 노드는, 그 자신을 지정된 노드의 근접 이웃에 있는 것으로 보는 경우, 상기한 바와 같이 지



정된 노드에 대한 이탈 메시지를 그 근접 이웃에 제공하는 것은 물론 지정된 노드가 고장났을지도 모른다는 것과 수신하는 노드가 그에 대한 대체물이라는 것을 (예를 들어, Node Down(1704)을 사용하여) 그의 애플리케이션 계층에 알려줄 수 있다. 지정된 노드에 대한 생존 고장 메시지는 근접지로 라우팅될 수 있고, 이 메시지의 타겟 ID는 고장났을지도 모르는 노드의 ID로 설정된다.

#### <243> 균형 절차

<244> 본 발명의 실시예들은 짧은 기간 내에 많은 수의 노드가 연합에 가입하고 이탈하는 것에 대응하도록 설계되어 있다. 여러가지 노드에 유지되는 대수적 탐색 트리(logarithmic search tree)가 균형되어 있지 않는 경우, 즉, 링의 한쪽에 다른쪽보다 더 많은 노드가 있는 경우, 네트워크 내에서의 이러한 변화는 라우팅 지연을 야기할 수 있다. 최적의 라우팅 효율을 용이하게 해주기 위해, 어떤 기준이 만족될 때 연합에 참가하는 노드들이 균형 절차를 실행한다.

<245> 예를 들어, 이하의 조건들 중 임의의 조건이 참인 경우, 최적의 라우팅 효율을 위한 균형 라우팅 테이블을 보장하기 위해 어느 노드라도 균형 절차를 실행할 수 있다.

<246> • 구성된 수의 상기한 생존 메시지가 수신되었다.

<247> • 상기한 마지막 생존 메시지의 수신 이후에 구성된 시간이 경과하였다.

<248> • 어떤 새로운 노드가 참가하였거나 어떤 기존의 노드가 이탈하였다는 의미에서 이웃이 변화하였다.

<249> 라우팅 테이블을 균형을 이루게 하는 것은 간단한 프로세스이다. 예를 들어, 불균형 라우팅 테이블을 갖는 노드는 가입 프로토콜의 동기화 및 라우팅 단계-상태를 재실행할 수 있다.

<250> 동작 RP2b, RP2d 및 RP4이 1) 어떤 번호에 가장 가까운 라우팅 노드를 찾아내는 것, 2) 연합을 정상 이탈하는 노드가 이탈 프로토콜을 준수하는 것, 및 3) 생존 메시지를 수신하는 노드가 균형 절차를 준수하는 것과 결합되는 것이 연합 노드들이 네트워크를 상당히 빠르고 여러번 가입 및 이탈할 때 가장 빠른 치유 시스템으로 된다.

#### <251> 상태 메시지

<252> 상태 메시지는 송신자 노드가 이전에 수신자 노드로 전달한 관련 메시지의 라우팅 성공/실패를 알려주기 위해 수신자 노드에 의해 송신자 노드로 전송되는 비라우팅 메시지이다. 도 18은 요청-응답 메시지 교환 패턴의 일부를 형성하는 메시지들이 링 상의 노드들에 걸쳐 어떻게 라우팅되는지의 일례를 나타낸 것이다. 상태 메시지는 라우팅 상태가 보고 되고 있는 원래의 관련 메시지를 식별해주는 헤더를 포함할 수 있다. 그 자체로서, 상태 메시지는 노드들 사이에서 메시지가 한 노드에서 그 다음 노드로 성공적으로 라우팅되었는지를 나타내기 위해 사용될 수 있다. 예를 들어, 요청 메시지(1811)를 노드(1801)로부터 노드(1806)로 라우팅하는 것은 노드(1802, 1803, 1804, 1805)를 통해 요청(1811)을 전송하는 것을 포함한다. 대응하는 연속되는 성공 상태 메시지(상태 1817, 1818, 1819, 1820, 1821)가 노드(1806)로부터 노드(1805)로, 노드(1805)로부터 노드(1804)로, 노드(1804)로부터 노드(1803)로, 노드(1803)로부터 노드(1802)로, 노드(1802)로부터 노드(1801)로 각각 전송될 수 있다. 요청(1811)에 응답하여, 응답(1816)이 노드(1807)로부터 노드(1801)로 종단간 전송될 수 있다. 응답(1816)은 선택적이며, 단방향 메시지 교환 패턴에서는 존재하지 않을 수도 있다.

<253> 도 13은 노드가 연합 기반구조에 가입하는 방법(1300)의 플로우차트의 일례를 나타낸 것이다. 방법(1300)은 도 12a 및 도 12b의 링(1206)과 관련하여 설명될 것이다. 방법(1300)은 연합 기반구조에 가입 메시지를 발행하는 동작(동작 1301)을 포함한다. 예를 들어, ID 144를 갖는 노드가 링(1206)을 포함하는 연합 기반구조에 가입 메시지(1201)를 발행할 수 있다. 방법(1300)은 가입하는 노드로부터 가입 메시지를 수신하는 동작(동작 1308)을 포함한다. 예를 들어, 링(1206)을 포함하는 연합 기반구조 내의 기존의 노드는 가입 메시지(1201)를 수신할 수 있다.

<254> 방법(1300)은 가입 메시지를 처리 노드로 라우팅하는 동작(동작 1309)을 포함한다. 처리 노드는 가입 메시지가 라우팅되고 있을 때 연합 기반구조 내의 다른 활성 노드들보다 가입하는 노드의 ID에 수치적으로 더 가까운 ID를 갖는 노드일 수 있다. 예를 들어, 가입 메시지(1201)는 처음에 ID 64를 갖는 노드에 수신되고, ID 135를 갖는 노드로 라우팅되며, ID 151을 갖는 노드로 라우팅될 수 있다.

<255> 방법(1300)은 가입하는 노드에 대한 하나 이상의 선행 노드 및 하나 이상의 후속 노드를 계산하는 동작(동작 1310)을 포함한다. 예를 들어, ID 151을 갖는 노드는 ID 144를 갖는 노드의 직전 선행 노드 및 직후 후속 노드

를 계산할 수 있다. 링(1206) 내에서, ID 151을 갖는 노드는 ID 135를 갖는 노드가 직전 선행 노드이고 ID 151을 갖는 노드가 직후 후속 노드인 것으로 계산할 수 있다. 다른 근접 링들에 대해 유사한 계산이 행해질 수 있다.

- <256> 방법(1300)은 가입하는 노드에 대한 하나 이상의 라우팅 노드를 계산하는 동작(동작 1311)을 포함한다. 예를 들어, ID 151을 갖는 노드는 ID 144를 갖는 노드에 대한 (ID 151을 갖는 노드의 관점에서 본) 라우팅 노드를 계산할 수 있다. 링(1206) 내에서, ID 151을 갖는 노드는, 예를 들어, ID 218 및 40을 갖는 노드가 ID 144를 갖는 노드에 대한 라우팅 노드인 것으로 계산할 수 있다. 다른 근접 링들에 대해 유사한 계산이 행해질 수 있다.
- <257> 방법(1300)은 가입하는 노드로 가입 응답을 전송하는 동작(동작 1312)을 포함한다. 가입 응답은, 연합 기반구조의 현재 뷰가 주어진 경우, 처리 노드에 의해 계산되는 가입하는 노드에 대한 모든 선행 및 후속 이웃 및 라우팅 파트너 노드를 식별해줄 수 있다. 예를 들어, 가입 응답(1202)은 적어도 ID 135를 갖는 노드를 ID 144를 갖는 노드에 대한 직전 선행 노드로서 식별할 수 있고, ID 151을 갖는 노드를 ID 144를 갖는 노드에 대한 직후 후속 노드로서 식별할 수 있으며, 노드 ID 144(새로 가입하는 노드)에 대해 ID 151을 갖는 노드에서 계산된 (ID 144를 갖는 노드에 대한) 모든 라우팅 노드를 식별할 수 있다.
- <258> 방법(1300)은 가입 메시지를 처리한 연합 노드로부터 가입 응답을 수신하는 동작(동작 1302)을 포함한다. 예를 들어, ID 144를 갖는 노드는 ID 151을 갖는 노드로부터 가입 응답(1202)을 수신할 수 있다.
- <259> 방법(1300)은 직전 근접 선행 노드 및 직후 근접 후속 노드의 적어도 각각으로 동기 요청을 전송하는 동작(동작 1303)을 포함한다. 예를 들어, 이제 도 12b를 참조하면, ID 144를 갖는 노드는 ID 135 및 151을 갖는 노드로 동기 요청(1203)을 전송할 수 있다. 동기 요청(1203)은 ID 144를 갖는 노드의 임의의 이웃 노드의 식별 번호 및/또는 ID 144를 갖는 노드의 임의의 라우팅 파트너의 식별 번호를 포함할 수 있다.
- <260> ID 135 및 151을 갖는 노드는 동기 요청(1203)을 수신할 수 있다. 동기 요청(1203)을 수신한 것에 응답하여, ID 135 및 151을 갖는 노드는 대응하는 라우팅 테이블로부터 그의 이웃 및 라우팅 파트너 노드를 식별할 수 있다. ID 135 및 151을 갖는 노드는 동기 응답(1204)에 그의 식별된 이웃 및 라우팅 파트너 노드의 생존 정보를 포함시키고 이 동기 응답(1204)을 ID 144를 갖는 노드로 전송할 수 있다.
- <261> 방법(1300)은 근접 선행 및 후속 노드 각각으로부터 동기 응답을 수신하는 동작(단계 1304)을 포함한다. 예를 들어, ID 144를 갖는 노드는 ID 135 및 151을 갖는 노드로부터 동기 응답(1204)을 수신할 수 있다. 동기 응답(1204)은 링(1206) 또는 연합 기반구조 내의 다른 링들 상의 하나 이상의 노드에 대한 생존 정보를 포함할 수 있다. 동기 응답(1204)은 또한 ID 144를 갖는 노드에 대한 예상된 라우팅 파트너 노드를 식별할 수 있다.
- <262> 방법(1300)은 이웃 노드를 계산하는 동작(동작 1305)을 포함한다. 예를 들어, ID 144를 갖는 노드는 ID 135 및 151을 갖는 노드에 대한 이웃 노드들의 합집합에 기초하여 대응하는 이웃 노드를 계산할 수 있다. 이웃 노드는 가입 응답 메시지와 동기 응답 메시지의 요약된 뷰에 기초하여 계산될 수 있다.
- <263> 방법(1300)은 라우팅 노드를 계산하는 동작(동작 1306)을 포함한다. 예를 들어, ID 144를 갖는 노드는 링(1206)의 노드들 중에서 라우팅 노드를 계산할 수 있다. 가입 응답 메시지 및 동기 응답 메시지의 요약된 뷰에 기초하여 라우팅 파트너가 계산될 수 있다.
- <264> 방법(1300)은 계산된 라우팅 파트너들과 적어도 이웃 노드 정보를 교환하는 동작(동작 1307)을 포함한다. 예를 들어, ID 144를 갖는 노드 및 ID 218을 갖는 노드(계산된 라우팅 파트너)는 그 각자의 이웃 노드에 대응하는 상태 정보(예를 들어, 인스턴스 ID, 단계-상태, 기타)를 교환할 수 있다. 이들 교환은 새로 가입하는 노드가 상기 라우팅 단계-상태 텍스트에 기술된 적어도 각각의 고유한 계산된 라우팅 파트너로 갱신 메시지를 제공(라우팅)하는 것에 의해 달성된다. 갱신 메시지를 처리하는 노드는 새로 가입하는 노드로부터 이들 갱신 메시지를 수신한 것에 응답하여 대응하는 갱신 응답 메시지를 전송한다. 갱신 응답은 그 자신 및 그의 이웃 노드들에 대한 생존 정보를 적어도 포함한다.
- <265> 방법(1300)은 또한 적어도 하나의 이웃 노드로의 라우팅 테이블의 초기 전달을 개시하는 동작을 포함할 수 있다. 예를 들어, ID 144를 갖는 노드는 계산된 이웃 및 라우팅 파트너 노드를 ping 메시지에 포함시키고 이 ping 메시지를 ID 174를 갖는 노드(예를 들어, 계산된 이웃 노드들 중 하나)로 전송할 수 있다. ID 174를 갖는 노드는 ping 메시지를 수신하고 ID 144를 갖는 노드에서 발신된 생존 정보로 대응하는 라우팅 테이블을 갱신할 수 있다. ID 174를 갖는 노드는 또한 그의 대응하는 라우팅 테이블을 제2 ping 메시지에 포함시키고 이 제2 ping 메시지를 어떤 장래의 시점에서 ID 144를 갖는 노드로 전송할 수 있다. ID 144를 갖는 노드는 제2 ping 메시지를 수신하고 제2 ping 메시지에 포함된 생존 정보 내의 노드들(즉, ID 174를 갖는 노드의 라우팅 테이블

내의 노드들)로 그의 대응하는 라우팅 테이블을 갱신할 수 있다. ID 144를 갖는 노드는 링(1206) 내의 다른 이웃 노드들에 대해 ping 메시지의 전송을 반복할 수 있다.

<266> 새로 가입하는 노드가 연합에 가입할 때, 새로 가입하는 노드가 기존의 연합 멤버를 찾지 못하고 따라서 유일한 멤버로 될 수도 있다는 것을 잘 알 것이다. 따라서, 새로 가입하는 노드에 대한 할당된 선행, 후속 또는 이웃 노드가 없을 수 있다. 그에 따라, 새로 가입하는 노드는 모든 경우에서 최상의 라우팅 파트너로서 매핑된다.

<267> 게다가, 방법(1300)이 단 하나의 링(링(1206))과 관련하여 설명되었지만, 몇몇 실시예들에서, 하나의 링에 가입하는 노드가 본질적으로 하나 이상의 다른 노드에도 가입한다는 것을 잘 알 것이다. 예를 들어, 다시 간략히도 5를 참조하면, 링(551)에 가입하는 노드는 본질적으로 링(543, 531, 522, 511, 501)에도 가입한다. 따라서, 방법(1300)은 복수의 링에 가입하도록 실시될 수 있다. 다른 실시예들에서, 다수의 링에 가입할 때 방법(1300)에서의 동작들 중 일부 또는 그 전부가 반복될 수 있다. 예를 들어, 다시 도 5를 참조하면, 노드가 링(551) 및 링(514) 둘다에 가입할 때(예를 들어, 별칭 지정) 방법(1300)의 동작들 중 하나 이상이 반복될 수 있다. 어쨌든, 가입하는 노드 ID가 액세스되어 정렬 연결 리스트는 물론 가입하는 노드가 참가하게 될 대응하는 계층적으로 분할된 서브-리스트에서 가입하는 노드를 식별하는 데 사용될 수 있다. 수신하는 노드는 정렬 연결 리스트 및 각각의 분할된 서브-리스트로부터 식별된다. 가입 메시지는 정렬 연결 리스트 및 각각의 분할된 서브-리스트 내의 처리 노드로 (예를 들어, ID에 기초하여) 라우팅된다. 정렬 연결 리스트 및 각각의 분할된 서브-리스트 내의 처리 노드로부터 가입 응답이 수신된다.

<268> 도 14는 노드가 연합 기반구조에 멤버십을 유지하는 방법(1400)의 플로우차트의 일례를 나타낸 것이다. 방법(1400)이 링(1206)과 관련하여 설명될 것이다. 방법(1400)은 제1 ping 메시지를 이웃 노드로 전송하는 동작(동작 1401)을 포함한다. 제1 ping 메시지는 제1 ping 메시지를 전송하는 현재 노드가 이웃 노드의 이웃임을 나타낸다. 제1 ping 메시지는 또한 현재 노드의 라우팅 파트너 및 이웃 노드의 상태도 포함할 수 있다. 예를 들어, ID 144를 갖는 노드는 ID 151을 갖는 노드로 ping 메시지를 전송할 수 있다. 제1 ping 메시지를 수신할 시에, ID 151을 갖는 노드는 ID 144를 갖는 노드가 ID 151을 갖는 노드의 이웃이라는 것을 알게 된다. 노드(151)는 또한 이 동작의 부수 효과로서 노드(144)로부터 (링(1206) 상의 다른 노드들에 대한) 더 새로운 생존 정보를 발견할 수 있다.

<269> ping 메시지는, 예를 들어, ping 메시지가 전송될 근접 링과 연관된 구성 상태에 기초하여 지정된 빈도수로 주기적으로 반복될 수 있다. 이 빈도수는 구성 상태에 따라 변화될 수 있다. 예를 들어, WAN에 대한 지정된 ping 빈도수는 LAN에 대한 지정된 빈도수와 다를 수 있다. ping 메시지는 또한 ping 배포 패턴에 따라 전송될 수 있다. 발신측 노드에 대한 ping 배포 패턴은 ping 메시지가 링에서 양방향으로 이웃 노드로 전송되어야 함을 나타낼 수 있다. 예를 들어, ID 144를 갖는 노드는 ID 135를 갖는 노드의 방향으로도 ID 151을 갖는 노드의 방향으로도 ping을 전송할 수 있다. ping 배포 패턴 및 빈도수는, 예를 들어, 근접 링마다 변화될 수 있다.

<270> 방법(1400)은 이웃 노드로부터 제2 ping 메시지를 수신하는 동작(동작 1402)을 포함한다. 제2 ping 메시지는 적어도 제2 ping 메시지를 발신하는 이웃 노드가 현재 노드의 이웃이라는 것을 현재 노드에 알려준다. 제2 ping 메시지는 또한 발신하는 이웃 노드의 라우팅 파트너 및 이웃 노드의 상태도 포함할 수 있다. 예를 들어, ID 151을 갖는 노드는 ID 144를 갖는 노드로 제2 ping 메시지를 전송할 수 있다. 제2 ping 메시지를 수신할 시에, ID 144를 갖는 노드는 ID 151을 갖는 노드가 ID 144를 갖는 노드의 이웃이라는 것을 알게 된다. 제2 ping 메시지는 또한 링(1206) 상의 다른 노드들에 대한 생존 정보도 포함할 수 있다. 따라서, 일반적으로, ping 메시지는 이웃 내에서 교환될 수 있고 (각각의 근접 멤버십에 대한) 이웃 멤버십 및 연합 내의 노드 존재의 근사화된 공통 이웃 뷰를 유지하는 데 사용될 수 있다.

<271> 수신된 ping 메시지는 주기적으로 반복될 수 있고/ping이 발신된(발신측 노드에 의해 전송된) 근접 이웃 내의 다른 노드로 전달될 수 있다. 전달된 ping 메시지는 또한 ping 배포 패턴에 따라 전송될 수 있다. 전달하는 노드에 대한 ping 배포 패턴은 ping 메시지가 발신측 노드로부터 멀어지는 방향으로 이웃 노드로 전송되어야 함을 나타낼 수 있다. 예를 들어, ID 151을 갖는 노드는 ID 144를 갖는 노드에서 발신하는 ping을 ID 174를 갖는 노드의 방향으로 전달할 수 있다. ping 전달 배포 패턴이, 예를 들어, 근접 링마다 변화될 수 있다.

<272> 노드가 대응하는 간격으로 ping 메시지를 수신하도록 구성될 수 있다. 예상된 ping 메시지가 수신되지 않는 경우, 노드는 통신 고장을 해석하여, 예상되었지만 적어도 늦은 ping 메시지를 발신했어야만 하는 노드에 대해 다른 노드에 대한 phase.unknown indication을 참으로 설정할 수 있다.

<273> 방법(1400)은 갱신 요청 메시지를 완전 라우팅 노드로 근접 라우팅하는 동작(동작 1403)을 포함한다. 갱신 요

청 메시지는 현재 노드가 수신측 라우팅 노드의 라우팅 파트너로서 참가하고 있음을 이러한 라우팅된 갱신 요청을 수신하는 라우팅 노드에 알려준다. 갱신 요청 메시지는 또한 적어도 현재 노드의 이웃 노드의 식별자도 (예를 들어, 생존 정보의 형태로) 포함할 수 있다. 예를 들어, ID 144를 갖는 노드는 갱신 메시지(1216)를 ID 208을 갖는 노드(144로부터 64만큼 오프셋된 완전 라우팅 파트너)로 라우팅할 수 있다. 노드(210)(이전에 계산된 라우팅 노드)가 208에 가장 가깝기 때문에, 노드(210)가 라우팅된 갱신 요청을 수신하여 처리한다. 갱신 메시지(1216)를 수신할 시에, ID 210을 갖는 노드는 ID 144를 갖는 노드가 ID 210을 갖는 노드의 라우팅 파트너라는 것을 알게 된다(또는 보장된다).

<274> 방법(1400)은 처리하는 (수신하는) 라우팅 노드로부터 갱신 응답 메시지를 수신하는 동작(동작 1404)을 포함한다. 갱신 응답은 처리하는 라우팅 노드가 현재 노드의 라우팅 파트너로서 참가하고 있음을 현재 노드에 알려준다. 갱신 응답 메시지는 또한 적어도 처리하는 라우팅 파트너의 이웃 노드의 식별자를 포함할 수 있다. 예를 들어, ID 210을 갖는 노드는 갱신 응답(1207)을 ID 144를 갖는 노드로 전송할 수 있다. 갱신 응답(1207)을 수신할 시에, ID 144를 갖는 노드는 ID 210을 갖는 노드가 ID 144를 갖는 노드의 라우팅 파트너이라는 것을 알게 된다.

<275> 방법(1400)은 또한 현재 노드 및 이웃 노드가 이웃으로서 참가하고 있고 현재 노드 및 이웃 노드가 라우팅 파트너로서 참가하고 있음을 알려주기 위해 노드 정보를 적절히 갱신하는 동작을 포함할 수 있다. 예를 들어, ID 144 및 141을 갖는 노드가 (근접) 이웃에 참가하고 있음을 알려주기 위해, ID 144를 갖는 노드가 ID 151을 갖는 노드에 대응하는 노드 정보를 갱신할 수 있다. 이와 유사하게, ID 144 및 210을 갖는 노드가 라우팅 파트너로서 참가하고 있음을 알려주기 위해, ID 144를 갖는 노드가 ID 210을 갖는 노드에 대응하는 노드 정보를 갱신할 수 있다.

<276> 몇몇 실시예에서, 지정된 노드 X에 저장된 애플리케이션 상태가 신뢰성있는 플러딩 프로토콜을 사용하여 그의 Neighborhood(X) 노드들 간에 복제된다. 애플리케이션 상태 내의 각각의 항목은 할당된 소유자를 가지며, 이 소유자는 그 항목을 생성한 종단점일 수 있다. 애플리케이션 상태 내의 각각의 항목은 또한 그의 소유자에 의해 부여되는 연관된 타임스탬프(즉, 순서 번호)를 갖는다. 이 타임스탬프는 적어도 3개의 성분을 갖는다.

<277> o 소유하는 개체의 인스턴스 ID(예를 들어, 부호없는 정수). 적어도 단조(>1) 증가해야 한다.

<278> o 소유자에 의해 발생된 특정의 시퀀스를 식별해주는 시퀀스 ID(예를 들어, URI). 이 성분에 의해 동일 소유자가 다수의 독립적인 시퀀스를 발생할 수 있다.

<279> o 식별된 애플리케이션 시퀀스 ID 내에서의 오프셋을 식별해주는 서수(예를 들어, 부호없는 정수).

<280> 항목 타임스탬프는 복제 동안 대응하는 항목과 연관된 최신의 정보를 검출하는 데 사용되는데, 그 이유는 항목 타임스탬프가 <Instance ID, Sequence ID, Offset> 트리플을 갖는 적어도 부분 순서를 발생하기 때문이다. 복제되는 항목과 연관된 타임스탬프는 최신의 항목을 검출하기 위해, 로컬 항목이 있는 경우, 그 로컬 항목과 비교된다. 항목 타임스탬프는 또한 생성/갱신/삭제 동작의 idempotent 의미를 지원하는 데 사용된다. 예를 들어, 노드가 애플리케이션 상태에서의 기존의 항목을 갱신하라는 요청을 수신할 때, 갱신 요청과 연관된 타임스탬프가 로컬 항목과 연관된 타임스탬프보다 상위에 있는 경우에만, 이 갱신이 수락된다. 항목들이 단 한명의 소유자에게 할당될 수 없는 경우, 벡터 타임스탬프에 기초한 충돌 해결 기법들이 이용될 수 있다. 애플리케이션 상태 복제는 고장-허용을 제공하고 이웃 노드들에 걸쳐 부하 분산 요청을 용이하게 해준다.

<281> 선택적인 동작으로서, (어떤 기간 후에) 다른 (발신) 파트너 (라우팅 및/또는 파트너) 노드로부터의 예상된 갱신 또는 ping을 검출하지 못한 노드는 단계-상태를 모르는 것으로 간주하고, phase.unknown indication를 참으로 설정하며 이를 그 자체로서 다른 제3자 노드에 보고할 수 있다.

<282> 환언하면, 갱신 및 ping의 주기적인 발생이 필요할 수 있다. 이 요건 및 실제의 타임아웃 값이 여러가지 근접성의 속성일 수 있다. 예를 들어, 링은 (예를 들어, LAN 세그먼트 내의) 어떤 서브-링에 대해 더 제한적인 타이밍 요건을 가질 수 있고, 노드 고장 검출/보고가 비교적 신속하다. 반면에, 링은 (예를 들어, 인터넷 상의) 다른 서브-링에 대해 덜 제한적인 타이밍 요건을 가질 수 있고(또는 타이밍 요건이 전혀 없음) 사전 예방적인 노드 고장 검출/보고가 비교적 오래 걸린다(또는 존재하지 않는다).

<283> 도 15는 다른 노드에 대한 생존 정보를 발견하는 방법(1500)의 플로우차트의 일례를 나타낸 것이다. 방법(1500)은 도 12a 및 도 12b의 링(1206)과 관련하여 설명될 것이다. 일반적으로, 예를 들어, 동기(1203), 동기 응답(1204), 갱신(1216), 갱신 응답(1207), 기타 등등의 임의의 메시지가 적어도 하나의 생존 헤더를 포함할 수 있다. 몇몇 실시예들에서, 생존 헤더는 노드에 대한 <노드 ID, 인스턴스 ID, 단계 [phase-state



value].[phase.unknown indication], 신선도 값, 컬러 (근접성) 값, 가중치 값>를 포함한다. 다른 실시예들에서, 생존 헤더는 <단계 [phase-state value].[phase.unknown indication], 신선도 값, 컬러 (근접성) 값, 가중치 값>을 포함한다. 이들 다른 실시예들에서, 생존 헤더는 송신자 및 수신지 노드에 대한 노드 ID 및 인스턴스 ID를 이미 포함하는 어드레싱 헤더를 보강하기 위해 사용될 수 있다. 어드레싱 헤더가 노드 ID 및 인스턴스 ID를 이미 포함하고 있기 때문에, 이 정보가 생존 헤더로부터 생략될 수 있다.

<284> 방법(1500)은 연합 기반구조에 참가하는 노드에 대한 상태 정보를 표현하는 생존 헤더를 수신하는 동작(동작 1501)을 포함한다. 생존 헤더는 적어도 수신된 참가 노드 ID, 수신된 노드의 인스턴스 ID, 수신된 단계 값, 및 수신된 신선도 값을 포함한다. 예를 들어, ID 144를 갖는 노드는 ID 151을 갖는 노드로부터 동기 응답(1204)으로 제1 생존 헤더를 수신할 수 있다. 제1 생존 헤더는 ID 174를 갖는 노드에 대한 <참가 노드 ID, 인스턴스 ID, 단계 값 [phase-state value].[phase.unknown indication], 신선도 값, 컬러(근접성) 값, 및 가중치값>을 포함할 수 있다. 단계-상태 값(예를 들어, Inserting, Syncing, Routing, Operating)은 제1 신선도 값일 때에 ID 174를 갖는 노드의 표현된 단계를 식별해준다. 단계 값(예를 들어, 단계-상태: [Inserting, Syncing, Routing, Operating], 및 phase.unknown)은 제1 신선도 값으로 표시된 때에 ID 174를 갖는 노드의 표현된 및/또는 검출된 단계 정보를 식별해준다.

<285> 그렇지만, 신선도 값이 통신 지연으로 인해 떨어질 수 있다. 신선도 값은 또한 시간의 경과에 따라 감소될 수도 있다. 신선도 값의 감소 곡선은 서로 다른 단계 상태(unknown을 포함함)에 대해 다를 수 있다(선형이나 대칭이 아닐 수 있다). 따라서, 서로 다른 노드 단계에 걸쳐, 신선도 값의 감소는 비선형 및/또는 비대칭일 수 있다.

<286> 방법(1500)은 현재 노드에 유지되는 참가 노드에 대한 적어도 현재 인스턴스 ID, 현재 단계 값, 및 현재 신선도 값에 액세스하는 동작(동작 1502)을 포함한다. 예를 들어, ID 144를 갖는 노드는 ID 174를 갖는 노드에 대한 이전에 수신되어 저장된 인스턴스 ID, 단계 값 [phase-state value].[phase.unknown indication], 및 신선도 값에 액세스할 수 있다.

<287> 방법(1500)은 현재 노드에서 적어도 수신된 인스턴스 ID, 수신된 단계 값, 및 수신된 신선도 값을 현재 인스턴스 ID, 현재 단계 값, 및 현재 신선도 값과 각각 비교하는 동작(동작 1503)을 포함한다. 예를 들어, ID 144를 갖는 노드는 ID 174를 갖는 노드에 대한 이전에 수신되어 저장된 인스턴스 ID, 단계 값 [phase-state value].[phase.unknown indication], 및 신선도 값을 생존 헤더로 수신된 인스턴스 ID, 단계 값 [phase-state value].[phase.unknown indication], 및 신선도 값과 비교할 수 있다.

<288> 제1 인스턴스 ID가 (순서상) ID 174를 갖는 노드에 대한 현재 저장된 인스턴스 ID보다 큰 것에 기초하여, 제1 단계-상태 값이 ID 174를 갖는 노드에 대한 현재 저장된 단계-상태 값보다 더 앞서 있는 것에 기초하여, 또는 제1 신선도 값이 ID 174를 갖는 노드에 대한 현재 저장된 신선도 값보다 큰 값인 것에 기초하여, ID 144를 갖는 노드는 ID 174를 갖는 노드에 대한 (예를 들어, ID 151을 갖는 노드로부터 수신된) 현재 상태 정보가 오래되어 쓸모 없는 것으로 판정할 수 있다. ID 144를 갖는 노드는 또한 적어도 하나의 phase.unknown indication(현재 저장되어 있거나 생존 헤더로 수신됨)이 단계-상태가 검출/전송되었을 때 그 단계-상태를 알고 있었다는 것을 나타내는 것으로 판정할 수 있다.

<289> 방법(1500)은 이 비교에 기초하여 참가 노드에 대한 상태 정보가 현재 노드에서 갱신되어야 하는지를 판정하는 동작(동작 1504)을 포함한다. 예를 들어, ID 174를 갖는 노드에 대한 값들의 비교에 기초하여, ID 144를 갖는 노드는 ID 174를 갖는 노드에 대한 상태 정보가 갱신되어야 하는지를 판정할 수 있다. ID 174를 갖는 노드에 대한 오래된 상태 정보를 갱신하는 것은 현재 저장된 값(예를 들어, 인스턴스 ID, 단계-상태 값, phase.unknown indication, 신선도 값)을 생존 헤더에 포함된 값으로 대체하는 것을 포함할 수 있다. 예를 들어, ID 144를 갖는 노드는 ID 174를 갖는 노드가 더 앞선 단계-상태로 천이했다는 것을 나타내기 위해 ID 174를 갖는 노드에 대한 상태 정보를 갱신할 수 있다.

<290> 몇몇 실시예들에서, 참가 노드와의 통신이 단절되었을지도 모른다고 검출될 수 있다. 예를 들어, ID 144를 갖는 노드는 ID 151을 갖는 노드와의 통신이 단절된 것으로 검출할 수 있다. 간략히 도 17을 참조하면, (ID 151을 갖는 노드의 종단점에 대한) 이전의 생존에 대한 가입 이벤트(1701)에 응답하여, 애플리케이션 계층(1752)은 (ID 151을 갖는 노드의 종단점에서의) 종단점 고장 이벤트(1703)를 함수 계층(1751)으로 전송할 수 있다. 이들 실시예에서, 이러한 검출된 생존 조건이 생존 정보에 나타내어져 있을 수 있으며, Phase.Unknown 표시자는 마지막 기지의 단계 상태값과 함께 참으로 설정된다.

- <291> 방법(1500)은 또한 연합 기반구조 내의 다른 제2 노드로부터 제2 생존 헤더를 포함하는 메시지를 수신하는 동작을 포함할 수 있다. 예를 들어, ID 144를 갖는 노드는 제2 생존 헤더를 포함하는 상태 메시지를 (ID 103을 갖는 노드 또는 링(1206)의 어떤 다른 노드로부터) 수신할 수 있다. 제2 생존 헤더는 ID 174를 갖는 노드에 대한 <참가 노드 ID, 제2 인스턴스 ID, 제2 단계 값 [phase-state value].[phase.unknown indication], 제2 신선도 값, 제2 컬러(근접성) 값, 제2 가중치 값>을 포함할 수 있다. 제2 단계 값(예를 들어, 단계-상태 : [Inserting, Syncing, Routing, Operating], 및 phase.unknown indication)은 제2 신선도 값의 때에 ID 174를 갖는 노드의 표현된/검출된 단계를 식별해준다.
- <292> 다른 대안으로서, 제1 생존 헤더를 수신한 후에, ID 144를 갖는 노드는 ID 174를 갖는 노드와 직접 통신하려고 할 수 있다. 통신이 성공적인 경우, ID 174를 갖는 노드는 어드레싱 헤더에 노드 ID 및 제2 인스턴스 ID를 가짐과 동시에 <제2 단계 값, 제2 신선도 값, 제2 컬러(근접성) 값, 제2 가중치 값>을 포함하는 생존 헤더를 갖는 메시지(예를 들어, 동기 응답)를 반환할 수 있다. 고장이 검출되는 경우, ID 144를 갖는 노드는 내부 생존 상태 변화(예를 들어, freshness = max, 및 phase.unknown indication = true)를 야기하고 상태 변화가 다른 노드로부터 수신된 것처럼 상태 변화를 처리한다. 이러한 상태 변화는 최고 신선도 값을 갖는다.
- <293> 방법(1500)은 또한 제2 인스턴스 ID, 제2 단계 값, 및 제2 신선도 값을 현재 인스턴스 ID, 현재 단계 값, 및 현재 신선도 값과 각각 비교하는 동작(동작 1506)을 포함할 수 있다. 예를 들어, ID 103을 갖는 노드로부터 상태 메시지를 수신한 후에, ID 144를 갖는 노드는 (순서상) 제2 인스턴스 ID가 제1 인스턴스 ID보다 큰 것에 기초하여, 제2 단계 값이 제1 단계 값보다 더 앞서 있는 것에 기초하여, 또는 제2 신선도 값이 제1 신선도 값보다 더 큰 것에 기초하여, ID 151을 갖는 노드에 대한 현재 상태 정보가 오래되어 쓸모 없는 것으로 판정할 수 있다.
- <294> 방법(1500)은 또한 이 비교에 기초하여 참가 노드의 상태 정보가 갱신되어야 하는지를 판정하는 동작을 포함할 수 있다. 예를 들어, ID 174를 갖는 노드에 대한 값들의 비교에 기초하여, ID 144를 갖는 노드는 ID 174를 갖는 노드에 대한 상태 정보가 갱신되어야 하는지를 판정할 수 있다. ID 174를 갖는 노드에 대한 오래된 상태 정보를 갱신하는 것은 현재 저장된 값들(예를 들어, 인스턴스 ID, 단계-상태 값, phase.unknown indication 또는 신선도 값)을 제2 생존 헤더에 포함된 값들로 대체하는 것을 포함할 수 있다. 예를 들어, ID 144를 갖는 노드는 ID 174를 갖는 노드가 보다 앞선 단계-상태로 천이하였음을 나타내기 위해 ID 174를 갖는 노드에 대한 상태 정보를 갱신할 수 있다.
- <295> 몇몇 실시예들에서, 같은 컬러 값의 컨텍스트 내에서 단계 값이 비교된다. 상기한 바와 같이, 노드는 다수의 근접 링에 참가할 수 있다. 다수의 근접 링에의 참가가 보다 특징적인 링에의 참가가 (공통의 스파인(spine)을 따라) 보다 일반적인 링에의 참가를 암시하는 결과로서 일어날 수 있다. 예를 들어, 다시 도 5를 참조하면, 링(532)에의 노드의 참가가 또한 이 노드가 링(522, 511, 501)에 참가하게 되는 것을 암시한다. 따라서, 보다 특징적인 링에 대한 컬러는 또한 모든 부모 근접 링도 나타낸다. 또한, 앞서 설명한 바와 같이, 다수의 근접 링에의 참가는 한 링 내의 노드가 (어쩌면 서로 다른 스파인을 따라) 하나 이상의 다른 링으로 별칭 지정될 때 일어날 수 있다. 예를 들어, 예를 들어, 여전히 도 5를 참조하면, 링(532)에 참가하는 노드는 링(531)(또는 심지어 링(531, 522, 511, 501)에의 참가를 암시하는 링(541))으로 별칭 지정될 수 있다. 따라서, 한 링(예를 들어, 링(531))에 대한 컬러는 다른 링(예를 들어, 링(532))의 피어 컬러(또는 근접성)로서 볼 수 있다.
- <296> 노드가 별칭 지정의 방식으로 복수의 근접 링에 참가할 때, 노드에 대한 단계 값(예를 들어, 단계-상태 값 및/또는 phase.unknown indication)이 서로 다른 근접 링 간에 다르게 될 가능성이 얼마간 있다. 따라서, 다른 노드에 대한 상태 정보를 수신하는 노드는 현재 상태 정보가 그 노드 및 컬러에 대해 갱신되어야 하는지를 판정하기 전에 상태 정보(컬러)에 대한 대응하는 근접 링을 식별한다. 예를 들어, ID 144를 갖는 노드는 수신된 상태 정보를 현재 상태 정보와 비교하기 전에 ID 174를 갖는 노드에 대응하는 수신된 상태 정보에 대한 대응하는 근접 링을 식별할 수 있다.
- <297> 적절한 근접 링을 식별하는 것은 수신된 컬러 값을 하나 이상의 현재 컬러 값과 비교하는 것을 포함할 수 있다. 수신된 컬러 값 및 현재 컬러 값이 같은 경우, 예를 들어, 현재 인스턴스 ID, 현재 단계 값, 및 현재 신선도 값 등의 다른 상태 정보가, 예를 들어, 수신된 인스턴스 ID, 수신된 단계 값, 및 수신된 신선도 값 등의 대응하는 수신된 상태 정보와 비교될 수 있다. 반면에, 수신된 컬러 값 및 현재 컬러 값이 다른 경우, 추가의 비교가 행해지지 않는다.
- <298> 컬러 값들 간의 동등성은 다양한 방식으로 얻어질 수 있다. 예를 들어, 컬러 값들 간의 동등성은 현재 컬러 값 및 수신된 컬러 값이 동일한 근접 링(예를 들어, 링(532))을 나타낼 때 얻어질 수 있다. 게다가, 컬러 값들 간의 동등성은 보다 특징적인 컬러 값이 대응하는 부모 컬러 값(예를 들어, 동일한 스파인을 따라 있는 다른 링)

과 비교될 때 얻어질 수 있다. 예를 들어, 링(532)에 대한 컬러 값을 링(511)(또는 링(522 또는 501))에 대한 컬러 값과 비교하는 것에 의해 동등성이 얻어질 수 있다. 따라서, 자식 근접성이 부모 근접성이지만, 더 특정적이다.

<299> 따라서, 일반적으로, 연합 기반구조 내의 현재 동작 노드는 다른 노드에 대한 표현된 및 검출된 생존 상태 정보를, 그 다른 노드와의 통신이 단절된 것으로 보일 때조차도, 교환할 수 있다.

### <300> 부트스트랩 메카니즘

<301> 일반적으로, 노드가 연합의 활성 멤버가 되기(예를 들어, 가입하기) 위해서는, 노드는 자신이 가입하고자 하는 리프 링(leaf ring)의 활성 멤버가 이미 되어 있는 적어도 하나의 다른 노드와 통신해야만 한다. 이러한 초기 형태의 통신이 이용가능하도록 해주는 데 도움을 주기 위해, 연합은 부트스트랩 메카니즘(bootstrapping mechanism)을 이용할 수 있다. 다른 유형의 통신이 리프 링의 활성 멤버를 식별하는 데 실패한 경우, 또는 보안 제약조건이 새로 가입하는 노드가 처음에 씨드 노드 등의 일련의 특수 노드 중 적어도 하나와 통신할 것을 요구하는 경우, 부트스트랩 메카니즘이 최후의 수단으로서 사용될 수 있다. 즉, 다른 유형의 통신이 실패하는 경우 또는 보안 요건으로 인해, 리프 링의 활성 멤버 노드를 식별하기 위해 부트스트랩 메카니즘이 사용될 수 있다.

<302> 몇몇 실시예들에서, 씨드 노드는 연합과의 통신을 부트스트랩하는 데 사용된다. 씨드 노드는 어떤 유형의 근접 시간 통신(cross (inter) proximity communication)에 대한 공지의 진입점을 제공한다. 씨드 노드는 기반구조 고장/복구 및 일반적 역동성으로 인한 링 분할을 치유하는 데 도움이 된다. 각각의 링은 연합에 기본적인 부트스트랩 속성을 제공하기 위해 적어도 하나의 동작 씨드 노드를 가질 수 있다.

<303> 피어 씨드 노드는 근접성을 위한 적어도 모든 활성 씨드 노드로 이루어져 있는 그 근접성을 위한 링 구조(예를 들어, 이중 연결 리스트)를 유지하기 위해 그 자신들끼리 통신을 할 수 있다. 모든 다른 씨드 노드의 존재(활성) 상태에 대한 적어도 전체적인 지식을 각각의 씨드 노드에 제공하기 위해 전용 씨드 노드 동기화 프로토콜(dedicated seed node synchronization protocol)이 사용될 수 있다. 활성 씨드 노드는 그 자신이 들어 있는 근접 리프 링은 물론 이 리프 링의 모든 다른 선조 링의 멤버 노드이다. 따라서, 씨드 노드는, 예를 들어, 씨드 노드의 리프 링에서 루트 링까지의 근접 링들의 전체적인 스펙트럼을 나타낼 수 있다. 그에 따라, 씨드 노드는 이들 근접 링 각각 내에서 이용가능성이 높고 잘 알려진 진입 노드로서 기능할 수 있다. 그 결과, 씨드 노드에 관한 존재 상태는 연합 내에서의 여러가지 형태의 통신(예를 들어, 근접시간 통신)에 유용할 수 있다. 그에 따라, 씨드 노드는, 예를 들어, 가입하는 노드를 위한 잘 알려진 "가입 지점(join point)"으로서 기능하는 것, 링 보안 기관으로서 기능하는 것, 기반구조 분할을 치유하는 것을 돕는 것, 및 각각의 근접 노드에 대한 안정된 "진입 노드"로서 기능하는 것과 같은 다수의 특별한 속성을 제공할 수 있다.

<304> 프레즌스 데이터(presence data)를 제공하기 위해, 씨드 노드의 참가 및 순차적인 이탈이 그의 근접 노드 각각 내의 랑데뷰 지점에서 안정된 진입 노드로서 등록될 수 있다. 예를 들어, 등록 메시지가 문자열 "Proximity:/"의 SHI-1 해쉬를 목적지 ID로 갖는 일정한 URI로 라우팅될 수 있다. 일 실시예에서, 안정된 진입 노드로서 기능하는 씨드 노드가 그 자신을 이러한 방식으로 등록하지만, 선택된 비씨드 노드도 그 자신을 씨드 노드에 대해 여기에 기술된 동일한 또는 유사한 프로토콜을 사용하여 동일한 방식으로 등록할 수 있는 다른 실시예들이 있다. 안정된 진입 노드(씨드 노드 등)가 등록될 때, 이 안정된 진입 노드는 그 자신이 멤버로 되어 있는 각각의 링에 알려줄 수 있다. 따라서, 이 일정한 URI에 의해 식별되는 랑데뷰 지점에 유지되는 정보는 본질적으로 안정된 진입 노드 및 그의 대응하는 링 멤버십의 리스트이다. 그에 따라, 어떤 노드라도 이용가능한 안정된 진입 노드 및 그의 링 멤버십의 리스트를 획득하기 위해 이 일정한 URI에 의해 식별되는 랑데뷰 지점을 참조할 수 있다.

<305> 일 실시예에서, 안정된 진입 노드는 이들 참가 및 이탈 이벤트를 직접 등록한다. 다른 실시예에서, 안정된 진입 노드는 이들 이벤트를 그의 직속 근접 링 내의 랑데뷰 지점에 직접 등록하고, 그 랑데뷰 지점은 등록하는/등록 해제하는 안정된 진입 노드가 속해 있는 나머지 근접 링 각각 내의 모든 다른 적절한 랑데뷰 지점을 갱신하는 것을 투명하게 (직접적으로 또는 간접적으로) 용이하게 해준다. 이러한 안정된 진입 노드의 등록 정보를 유지 및 전달하기 위해 연합의 애플리케이션 상태 시퀀싱 및 전달 속성이 사용될 수 있다. 예를 들어, 저장된 애플리케이션 상태를 노드의 이웃 노드들에 복제하기 위해 신뢰성있는 플러딩 프로토콜이 사용될 수 있다.

<306> 안정된 진입 노드의 프레즌스 데이터를 루트 링 쪽으로 승격시키는 것에 의해, 연합 내의 다른 노드들이 모든 근접 링 내의 적어도 하나의 진입 노드를 탐색할 수 있게 된다. 탐색을 수행하는 노드의 리프 링 및 원하는 근

접 링의 LCAR(Lowest Common Ancestor Ring) 내의 상기 결정된 랑데뷰 지점 쪽으로 노드 탐색 메시지를 라우팅함으로써 진입 노드 탐색이 용이하게 될 수 있다. 예를 들어, 도 5를 참조하면, 링(541) 내의 노드는 링(533) 내의 노드와 통신하고자 할지도 모른다. 그렇지만, 링(541) 내의 노드는 링(533) 내의 노드에 대해 직접적으로 알 수는 없다. 따라서, 링(541) 내의 노드는 링(522)(링(541) 및 링(533)의 LCAR)으로 노드 탐색 메시지를 전송할 수 있다. 진입 노드 프레즌스 정보(그 진입 노드에 의해 발신된 등록 메시지로 인해 시스템 내에 존재하게 됨)를 처리하는 링(522) 내의 랑데뷰 지점 노드는 링(533) 내의 적어도 등록된 안정된 진입 노드에 대한 연락 정보를 사용하여 탐색 응답 메시지를 반환할 수 있다.

<307> 몇몇 실시예들에서, 안정된 진입 노드는 다양한 근접지에 대한 프레즌스 데이터를 유지하는 안정된 진입 노드로서 특별히 구성된 씨드 노드이다. 다른 실시예들에서, 다른 유형의 노드들도 다양한 근접지에 대한 프레즌스 데이터를 유지하는 안정된 진입 노드로서 기능할 수 있으며 또한 다른 동작을 수행하도록 구성될 수도 있다. 예를 들어, 어떤 다른 유형의 노드가 (예를 들어, 관리자에 의해) 이용가능성이 높도록, 따라서 안정된 진입 노드로서 적당하도록 (즉, 상기한 바와 같이 등록되도록) 구성될 수 있다. 그렇지만, 이 다른 유형의 노드들은 부가의 씨드 노드 기능을 포함하지 않을 수 있다(예를 들어, 링 보안 기관으로서 신뢰되지 않을 수 있다). 몇몇 실시예들에서, 그의 지속 근접지에 대한 진입 노드 존재 상태를 유지하는 랑데뷰 지점은 그 자신을 선조 링 또는 링들 내의 안정된 진입 노드로서 등록할 수 있다.

### <308> 근접지간 통신

<309> 본 발명의 실시예들은 또한, 예를 들어, 링들로 이루어진 트리의 서로 다른 근접 브랜치 내의 노드들 간의 통신과 같은, 근접지간 통신을 용이하게 해줄 수 있다. 근접 분할된 링 기반구조 내의 하나 이상의 근접 링, 어쩌면 모든 근접 링과 통신하기 위해 및/또는 이들 근접 링 간에 통신하기 위해 근접지간 통신이 사용될 수 있다. 이제 도 5a를 참조하면, 도 5a는 분할 트리(500)의 일부분들에 부가 레벨의 상세를 갖는 근접 유도 분할 트리(500)의 일례를 나타낸 것이다. 도 5a의 다양한 노드들 간에 근접지간 통신이 행해질 수 있다.

<310> 도 5a에 나타낸 바와 같이, 링(500)의 분할 트리는 그에 부가하여 링(513) 아래에 여러가지 서브-링을 포함하고 있다. 이 부가적인 서브-링 각각은 정렬 연결 리스트의 분할을 나타낸다. 도 5에 대해 앞서 설명한 바와 같이, 분할 트리(500) 내에서, 루트 링(501)은 기준(571)(제1 관리 도메인 경계 기준)에 기초하여 서브-링(511, 512, 513, 514)을 포함하는 복수의 서브-링으로 분할된다. 또한, 도 5에 대해 앞서 설명한 바와 같이, 서브-링(511)은 기준(581)(제2 관리 도메인 경계 기준)에 기초하여 서브-링(521, 522, 523)을 포함하는 복수의 서브-링으로 추가로 분할될 수 있다. 서브-링(522) 아래의 다른 서브-링들은 다른 기준에 기초하여 더 분할된다.

<311> 앞서 설명한 바와 같이, 분할 트리(500) 내에서, 각각의 노드는 단 하나의 ID를 가지며 루트에서 시작하여 리프에 이르는 대응하는 분할 경로(스파인)를 따라 링들에 참가한다. 예를 들어, 서브-링(552)에 참가하는 각각의 노드는 또한 서브-링(543, 531, 522, 511) 및 루트(501)에도 참가한다.

<312> 도 5a에서, 서브-링(513)은 또한, 예를 들어, 국가 관할권(state jurisdiction) 등의 다른 기준에 기초하여 서브-링(561, 562)을 포함하는 복수의 서브-링으로 더 분할될 수 있다. 서브-링(562)은, 예를 들어, 도시 관할권 등의 다른 기준에 기초하여 서브-링(571, 572)을 포함하는 복수의 서브-링으로 더 분할될 수 있다. 그에 따라, 도 5a 내에서, 링(541)으로부터 루트 링(501)으로의 스파인을 따라 위로 갔다가 다시 아래로 루트 링(501)으로부터 링(572)으로의 통신을 필요로 하지 않고 링(541)의 노드로부터 링(572)의 노드로 메시지를 전송하기 위해 근접지간 통신이 사용될 수 있다.

<313> 근접지간 통신은 링들로 이루어진 트리로 구현되는, 예를 들어, 브로드캐스트, 멀티캐스트, 또는 애니캐스트 등의 통신 패턴의 일부로서 포함될 수 있다. 브로드캐스트는 링들로 이루어진 트리 내의 모든 활성 노드에 메시지를 전송하는 것을 포함할 수 있다. 멀티캐스트는 링들로 이루어진 트리 내의 일군의 노드들에게 메시지를 전송하는 것을 포함할 수 있다. 애니캐스트는 링들로 이루어진 트리 내의 적어도 하나의 노드에게 메시지를 전송하는 것을 포함할 수 있다.

<314> 도 19a는 근접지간 통신을 용이하게 해주는, 링들로 이루어진 근접 유도 분할 트리(1900)의 일례를 나타낸 것이다. 링들로 이루어진 분할 트리(1900) 내에서, 루트 링(1901)은 선택된 기준(예를 들어, 제1 관리 도메인 경계 기준)에 기초하여 서브-링(1, 2, 3, 4)을 포함하는 복수의 서브-링으로 분할된다. 서브-링(1)은 서브-링(11, 21, 31, 41)으로 더 분할된다. 서브-링(11)은 서브-링(111, 211, 311)으로 더 분할된다. 서브-링(21)은 서브-링(121, 221, 321, 421, 521)으로 더 분할된다. 명시적으로 도시되어 있지는 않지만, 예를 들어, 서브-링(2, 3, 4, 31, 41) 등의 다른 서브-링들도 더 분할될 수 있다.



- <315> 링들로 이루어진 근접 유도 분할 트리(1900)에서의 링들의 번호 부여 방식은 첫번째 숫자 이후의 숫자들이 링의 부모 링을 나타내도록 구성되어 있다. 예를 들어, 링 "311"에서 "11"은 링(11)이 링(311)의 부모 링이라는 것을 나타낸다. 이와 유사하게, 링 "41"에서 "1"은 링(1)이 링(41)의 부모 링이라는 것을 나타낸다. 전역 링(1901)은, 예를 들어, 링(1, 2, 3, 4) 등의 단 하나의 숫자로 번호가 매겨진 링들의 부모 링이다. 도 5a의 링들로 이루어진 근접 유도 분할 트리(500)와 유사하게, 링들로 이루어진 분할 트리(1900)는 여러가지 근접 기준에 기초하여 분할될 수 있다.
- <316> 이 설명 및 첨부된 특허 청구의 범위에서, 표기법 R["<번호>"]는 링 번호를 말하는 데 사용된다. 예를 들어, R["11"]은 링(11)을 말한다. 이 설명 및 첨부된 특허 청구의 범위에서, 표기법 N["<번호>"]는 노드 번호를 말하는 데 사용된다. 예를 들어, N[1311]은 노드(1311)를 말한다.
- <317> 링들로 이루어진 트리에서의 통신을 용이하게 해주기 위해, 노드는 링을 대응하는 진입 노드와 대응시키는 엔트리 테이블(entry table)을 유지할 수 있다. 앞서 설명한 바와 같이, 링들로 이루어진 분할 트리(1900)의 구성에 기초하여, 부모 링은 그의 자식 링 각각에 속하는 노드들 전부를 포함한다. 예를 들어, 링(11)은 R["111"], R["211"] 및 R["311"]에 속하는 모든 노드를 포함한다. 따라서, 링 R["111"], R["211"] 및 R["311"] 내의 임의의 노드로 메시지를 가져오기 위해서 링(11)으로 메시지를 전송하는 것으로 충분하다. 그에 따라, 몇몇 실시예들에서, 노드의 엔트리 테이블이 노드가 존재하는 링의 관점에서 볼 때 비교적 다른 그 링들에 대한 엔트리로 감소될 수 있다. 예를 들어, N[1111]은 단순히 R["21"] - N[1121] - 에 대한 엔트리를 유지할 수 있는데, 그 이유는 다수의 자식 링, 즉 R["21"]에 대한 엔트리를 유지하는 것이 중복적이기 때문이다.
- <318> **방계 링 세트의 생성 및 유지**
- <319> 이 설명 및 첨부된 특허 청구의 범위에서, 지정된 링의 임의의 피어 링(peer ring)은 지정된 링의 "방계 링(collateral ring)"으로 정의된다. 이 설명 및 첨부된 특허 청구의 범위에서, 지정된 링의 선조 링의 임의의 피어 링도 지정된 링의 "방계 링"으로 정의된다. 지정된 링의 방계 링은 또한 지정된 링에 포함된 모든 노드의 방계 링이다.
- <320> 따라서, 예를 들어, 여전히 도 19a를 참조하면, R["211"]은 R["111"]의 방계 링인데, 그 이유는 R["211"]이 R["111"]의 피어이기 때문이다. R["211"]은 또한 R["111"]에 포함된 임의의 노드(예를 들어, N[1311] 등)의 방계 링이다. 게다가, R["21"]은 R["111"]의 방계 링인데, 그 이유는 R["21"]이 R["11"] (즉, R["111"]의 선조)의 피어이기 때문이다. R["21"]은 또한 R["111"]에 포함된 임의의 노드(예를 들어, N[1111] 등)의 방계 링이다.
- <321> 이 설명 및 첨부된 특허 청구의 범위에서, "방계 링 세트(collateral ring set, CRS)"는 지정된 링 또는 지정된 링 내의 노드들의 관점에서 볼 때 하나 이상의 방계 링의 세트로서 정의된다. 예를 들어, 도 19a에서, R["221"]은 물론 R["221"] 내의 임의의 노드(예를 들어, N[8221] 등)에 대한 방계 링 세트는 R["11"], R["121"], R["31"], R["41"], R["2"], R["3"], 및 R["4"]를 포함한다.
- <322> 따라서, 근접지간 통신을 용이하게 해주기 위해, 노드는 하나 이상의 방계 링 및 하나 이상의 방계 링으로의 하나 이상의 대응하는 진입 노드를 포함하는 CRS 엔트리 테이블을 유지할 수 있다. CRS 엔트리 테이블은 하나 이상의 <방계 링, 1 내지 N개의 진입 노드> 항목(단, N은 어떤 정수임)을 포함하는 데이터 구조일 수 있다. 예를 들어, 이 데이터 구조는 형식 <방계 링\_01, 진입 노드\_01, 진입 노드\_02, ...>을 가질 수 있으며, 여기서 생략 부호는 방계 링\_01로의 하나 이상의 부가의 진입 노드를 나타낸다.
- <323> CRS 엔트리 테이블을 생성하기 위해, 노드는 링들로 이루어진 트리에서 상태를 전파하는 데 사용되는 랑데뷰 프로토콜 메시지(예를 들어, ping 메시지, 갱신 메시지, 및 갱신 응답), 애플리케이션 메시지, 및 링들로 이루어진 트리에서의 지정된 통신 패턴(예를 들어, 브로드캐스트, 멀티캐스트, 및 애니캐스트)을 용이하게 해주기 위해 사용되는 메시지에 대한 로컬 지식을 사용할 수 있다.
- <324> 노드는 노드가 참가하는 모든 레벨의 링들로부터, 예를 들어, 라우팅 테이블 정보 등의 로컬 지식을 사용할 수 있다. 예를 들어, 여전히 도 19를 참조하면, N[1121]은 R["1"] 내의 N[1311]의 이웃일 수 있다. N[1121]의 관점에서 볼 때 N[1311]의 하나의 방계 링이 R["21"]이다. 그에 따라, N[1311]은 쌍 (R["21"], N[1121]) (이 경우에, 단 하나의 진입 노드를 갖는 항목)을 N[1311]의 CRS 엔트리 테이블에 삽입할 수 있다. 이 유형의 로컬 지식을 사용함으로써, 노드는 엔트리들을 CRS 엔트리 테이블에 삽입할 수 있다.
- <325> 노드들은, 특별한 목적의 메시지에 부가하여, 그렇지 않았으면 다른 목적으로(예를 들어, 이웃 및 라우팅 파트



너 상태 정보를 전달하기 위해) 사용되는 메시지를 통해 그의 CRS 엔트리 테이블 정보를 다른 노드로 전달할 수 있다. 예를 들어, 노드는 이웃 노드로 전송되는 ping 메시지에 또한 라우팅 파트너 노드들 간에 교환되는 갱신 메시지 및 갱신 응답에 CRS 엔트리 테이블 상태를 포함시킬 수 있다. 다른 노드로부터 CRS 엔트리 테이블 상태를 수신하는 노드는 수신된 CRS 엔트리 테이블 상태를 사용하여 그 자신의 CRS 엔트리 테이블을 보강 및/또는 유지할 수 있다.

<326> 예를 들어, 노드가 (예를 들어, 랑데뷰 프로토콜 계층에서) 그의 이웃/파트너 노드와 랑데뷰 ping/갱신 메시지를 교환할 때, 노드는 또한 그의 CRS 엔트리 테이블(의 적어도 일부, 경우에 따라서는 그 전부)을 교환할 수 있다(그의 이웃/파트너로부터 수신된 CRS 엔트리 테이블 상태를 사용하여 그 자신의 테이블을 갱신할 수 있다). 예를 들어, N[1311]이 (링(1901)의 컨텍스트 내에서) R["3"] 내의 임의의 노드에 대해 사전에 알고 있지 않은 것으로 가정한다. 그렇지만, N[1311]은 (R["1"] 내의) 이웃 N[8221]을 가지며, N[8221]의 CRS 엔트리 테이블은 엔트리 (R["3"], N[8223])을 갖는다. 적어도 N[1311] 및 N[8221]이 이웃이기 때문에, N[1311] 및 N[8221]은 때때로 서로에게 ping 메시지를 전송할 수 있다. N[8221]은 N[1311]로 전송되는 ping 메시지에 그의 CRS 엔트리 테이블(의 적어도 일부, 경우에 따라서는 그 전부)을 포함시킬 수 있다. 따라서, N[1311]이 N[8221]로부터 ping 메시지를 수신할 때, ping 메시지는 N[8221]의 CRS 엔트리 테이블을 포함할 수 있다. N[8221]의 CRS 엔트리 테이블로부터, N[1311]은 항목 <R["3"], N[8223], ...>을 식별하고 항목 <R["3"], N[8223], ...>을 그 자신의 CRS 엔트리 테이블에 포함시킨다. N[1311]은 또한 그의 CRS 내의 다른 링들에 대한 다른 항목들을 식별할 수 있으며 그 다른 항목들을 그 자신의 CRS 엔트리 테이블에 포함시킬 수 있다.

<327> CRS 엔트리 테이블 상태도 이와 마찬가지로 링 내의 라우팅 파트너들 간에 교환되는 갱신 메시지 및 갱신 응답을 통해 교환될 수 있다. 또한, 임의의 라우팅 프로토콜 메시지는 CRS 엔트리 노드에 관하여 (예를 들어, 생존 정보) 알아내는 데 사용될 수 있다. 또한, CRS 엔트리 테이블을 유지하기 위한 특정의 메시지가 사용될 수 있다.

<328> CRS 엔트리 테이블 상태는 또한 링들로 이루어진 트리에서 지정된 통신 패턴(예를 들어, 브로드캐스트, 멀티캐스트, 애니캐스트)을 용이하게 해주는 메시지들에서 발견될 수 있다. 예를 들어, 링들로 이루어진 트리 내의 모든 노드로 메시지를 브로드캐스트하기 위해, 브로드캐스트 알고리즘은 브로드캐스트에 특유한 여러가지 유형의 메시지를 사용할 수 있다. 이들 브로드캐스트 관련 메시지를 전송하는 노드는 브로드캐스트 관련 메시지 내에 CRS 엔트리 테이블 상태를 포함시킬 수 있다. 이와 유사하게, 일군의 노드 내의 모든 노드로 메시지를 멀티캐스트할 때, 멀티캐스트 알고리즘은 멀티캐스트에 특유한 여러가지 유형의 메시지를 사용할 수 있다. 이들 멀티캐스트 관련 메시지를 전송하는 노드는 멀티캐스트 관련 메시지 내에 CRS 엔트리 테이블 상태를 사용할 수 있다. 이들 멀티캐스트 관련 메시지를 전송하는 노드는 멀티캐스트 관련 메시지 내에 CRS 엔트리 테이블 상태를 포함시킬 수 있다. 마찬가지로, 메시지를 적어도 하나의 노드로 애니캐스트할 때, 애니캐스트 알고리즘은 애니캐스트에 특유한 여러가지 유형의 메시지를 사용할 수 있다. 이들 애니캐스트 관련 메시지를 전송하는 노드는 CRS 엔트리 테이블 상태를 애니캐스트 관련 메시지 내에 포함시킬 수 있다. 통신 패턴 관련 메시지를 수신하는 노드는 적절한 엔트리(예를 들어, <방계 링, 엔트리 노드> 항목)를 포함시킬 수 있고 이들 엔트리로부터의 상태의 일부 또는 그 전부를 그 자신의 CRS 엔트리 테이블에 포함시킬 수 있다.

<329> CRS 상태는 또한 애플리케이션들 간에 교환되는 애플리케이션 컴포넌트 메시지에서 발견될 수 있다. 간략히 도 1을 참조하면, 애플리케이션 계층(121, 122 및/또는 123)은 CRS 상태를 포함하는 애플리케이션 컴포넌트 메시지를 교환할 수 있다. CRS 상태를 포함하는 애플리케이션 컴포넌트 메시지를 수신할 시에, 애플리케이션 계층은 기존의 CRS 상태를 보강하기 위해 CRS 상태를 아래로 대응하는 다른 하위 계층까지(예를 들어, 랑데뷰 프로토콜 계층까지) 전송할 수 있다.

<330> CRS 상태는 애플리케이션에 의해 제공되고 및/또는 배선 연결될 수 있으며, 링들로 이루어진 트리 내의 노드들에 의해 구성가능하다.

<331> 다시 도 19a를 참조하면, 노드가 동일한 링에 대한 복수의 진입 노드를 알고 있음을 발견한 경우, 그 노드는 링에 대한 진입 노드들 중 2개 이상을 유지하기로 결정할 수 있다. 어떤 실시예들에서, 노드는 유지된 진입 노드들 중 하나의 진입 노드를 랜덤하게 선택한다. 다른 실시예들에서, 유지된 진입 노드들 중 지정된 진입 노드를 선택하는 것을 돕기 위해 정책이 적용된다. 정책은 선택된 진입 노드가 매번마다 동일한 진입 노드이어야 함을 나타낼 수 있다. 다른 대안으로서, 정책은 선택된 진입 노드가 유지된 진입 노드들 간에 변동되어야 함을 나타낼 수 있다. 예를 들어, 어떤 실시예들에서, 유지된 진입 노드들 간에 부하가 균등하게 분산될 수 있도록 진입 노드가 라운드-로빈 방식으로 선택된다.

- <332> 게다가, 다수의 진입 노드가 유지될 때, 예를 들어, 제1 선택된 진입 노드가 고장나거나, 사용 중이거나, 또는 어떤 다른 이유로 이용할 수 없는 경우, 한 노드가 효율적으로 다른 진입 노드로 천이할 수 있다.
- <333> 다른 대안으로서, 노드는 링에 대한 단 하나의 진입 노드를 유지할 수 있다.
- <334> 때때로, 노드는 링에 대한 부가의 진입 노드를 검출할 수 있다. 어떤 실시예들에서, 노드가 단 하나의 진입 노드를 유지하거나 부가의 진입 노드를 저장할 메모리가 없는 경우, 노드는 새로 수신된 진입 노드가 기존의 진입 노드를 대체할지를 결정할 수 있다. 어떤 실시예들에서, 노드는 이하의 파라미터, 즉 진입 노드의 거리, 진입 노드 정보의 신선도, 및 진입 노드의 가중치(예를 들어, 구성 선호)를 사용하여 각각의 후보 진입 노드에 대한 순위를 계산하기 위해 함수를 사용할 수 있다. 더 높은 순위를 갖는 진입 노드가 유지된다.
- <335> CRS 엔트리 테이블이 완성되거나 완성되지 않을 수 있다. 완성된 CRS 엔트리 테이블은 노드의 CRS 내의 각각의 링에 대해 적어도 하나의 진입 노드를 포함한다. 예를 들어, 노드(1311)에 대한 완성된 CRS 엔트리 테이블은 링(111, 211, 21, 31, 41, 2, 3, 4) 각각으로의 적어도 하나의 진입 노드를 포함한다.
- <336> 상기 메커니즘을 이용하여, 근접 링 계층구조 내의 하나 이상의 링 및/또는 노드의 관점에서 볼 때 완성된 CRS 엔트리 테이블을 작성하는 것이 가능할 수 있다. 어떤 실시예들에서, 라우팅 테이블 상태 정보만이 링들로 이루어진 트리에서의 각각의 노드에 대한 완성된 CRS 엔트리 테이블을 형성하는 경우가 있을 수 있다. 예를 들어, 링들로 이루어진 트리(1900)에서 전달되는 라우팅 테이블 관련 정보(예를 들어, ping 메시지, 갱신 요청 메시지, 및 갱신 응답 메시지)가 도식된 링들 각각에 있는 모든 노드에 대한 완성된 CRS 엔트리 테이블을 형성할 수 있다.
- <337> 그렇지만, 다른 분산 네트워킹 환경에서, 링의 동적 성질 및/또는 엔트리 테이블 관련 상태의 교환에서의 지연이 완성된 엔트리 테이블의 작성을 방해할 수 있다. 달리 말하면, 이들 다른 환경에서, 노드 또는 링이 임의의 주어진 때에 알아채지 못하는 링 또는 노드의 방계 링 세트에 속하는 하나 이상의 링이 있을 수 있다. 예를 들어, 도 19a를 다시 참조하면, N[8004]가 이제 막 가입했고 그 이전에 R["4"]에 속하는 노드가 없었던 것으로 가정하면, 메시지 트래픽(예를 들어, ping/갱신 메시지) 또는 다른 활동으로 인해 전체 링 기반구조에 걸쳐 그 정보의 배포가 일어날 때까지 대부분의 노드가 R["4"]에 대한 진입 노드를 가지고 있지 않게 된다. 따라서, 노드에 완성된 CRS 엔트리 테이블을 유지하는 것이 네트워크 역동성(예를 들어, 노드 고장, 통신 장애, 통신 지연, 노드 추가, 기타 등등)으로 인해 항상 가능한 것은 아니다. 그에 따라, 많은 환경들에서, 노드는 노드의 CRS 내의 노드들 전부보다 적은 노드들에 대한 진입 노드를 포함하는 부분 CRS 엔트리 테이블을 유지한다.
- <338> **CRS 엔트리 테이블을 사용한 근접지간 통신**
- <339> 노드는 (송신측 링 및 목적지 링의 LCAR로 메시지를 라우팅할 필요 없이) CRS 엔트리 테이블 내의 정보를 사용하여 근접지간 통신을 전송할 수 있다. 여전히 도 19a를 참조하면, CRS 엔트리 테이블 내의 적절한 엔트리에 기초하여, N[1311](예를 들어, 게시자 노드)은 근접지간 통신을 R["111"], R["211"], R["21"], R["31"], R["41"], R["2"], R["3"], 및 R["4"] 중 하나 이상으로 직접 전송할 수 있다. 예를 들어, 어떤 시점에서, N[1311]에 대한 CRS 엔트리 테이블은 이하의 엔트리들을 포함할 수 있다.
- <340> R["2"] : N[1112]
- <341> R["3"] : N[8223]
- <342> R["21"] : N[1121]
- <343> R["31"] : N[1131]
- <344> R["111"] : N[1111]
- <345> R["211"] : N[1211].
- <346> 이들 엔트리는 R["111"], R["211"], R["21"], R["31"], R["2"], 및 R["3"]으로의 직접 통신을 위해 사용될 수 있다. 예를 들어, N[1311]은 R["111"]로 통신(52)을 전송할 수 있고, N[1311]은 R["211"]로 통신(51)을 전송할 수 있으며, N[1311]은 R["21"]로 통신(53)을 전송할 수 있다(N[1121]은 R["21"] 및 R["221"] 둘다의 멤버임). N[1311]은 R["31"]로 통신(56)을 전송할 수 있고, N[1311]은 R["2"]로 통신(57)을 전송할 수 있으며, N[1311]은 R["3"]으로 통신(58)을 전송할 수 있다. 시간이 흐름에 따라, N[1311]은 R["41"] 및 R["4"]에 대한 엔트리(예를 들어, <방계 링, 엔트리 노드> 항목)를 식별할 수 있다. 이들 엔트리는 갱신된 로컬 지식으로부터, 라데뷰 프로토콜 메시지(예를 들어, ping 메시지, 갱신 메시지, 및 갱신 응답)에 포함된 CRS

엔트리 테이블로부터, 통신 패턴 관련 메시지 내에 포함된 CRS 엔트리 테이블 상태를 통해, 그리고, 예를 들어, 애플리케이션 컴포넌트 등의 기타 메카니즘을 통해 식별될 수 있다.

#### <347> 하향 라우팅 알고리즘

<348> 라우팅 연합(CRS 엔트리 테이블을 갖거나 갖지 않음)에서, 메시지가 자신이 발신된 리프 링에 대한(또는 리프 링 내의) 선조가 아닌 지정된 근접 링 내의 주어진 노드 ID에 가장 가까운 노드로 라우팅되는 경우가 있을 수 있다(이후부터 "하향 라우팅(downward routing)"이라고 함). 이것의 일례는 발신측 링 내의 노드가 방계 링에 대한 진입 노드를 모르고 있을 때 발신측 링 내의 노드로부터 방계 링으로의 근접지간 통신을 용이하게 해주는 것이다. 도 19b는 링들로 이루어진 근접 유도 분할 트리(1900)의 일례의 다른 뷰를 나타낸 것이다. 예를 들어, 이제 도 19b를 참조하면, N[1311]이 R["4"] 또는 R["41"] 내의 노드로 보내지는 통신을 전송하는 경우가 있을 수 있다.

<349> 라우팅 연합이 주어진 경우, 이하의 함수가 정의될 수 있다.

<350> RouteDown(M, P, ID): 연합이 메시지 M을 근접지 P 내의 ID에 가장 가까운 노드로 전달한다. 근접지 P는 발신측 노드의 리프 링에 대한(또는 발신측 노드의 리프 링 내의) 선조가 아닌 연합(리프 또는 중간) 내의 임의의 근접 링일 수 있다.

<351> 어떤 실시예들에서, 메시지 M은 발신측 노드의 리프 링 및 타겟 근접 링 P의 LCAR 내의 노드 위로는 결코 라우팅되지 않는다. 예를 들어, N[1311]로부터 R["41"]로 하향 라우팅을 실시하기 위해, 메시지를 R["1"] (R["311"] 및 R["41"]의 LCAR) 위로 라우팅할 필요가 없다. 그렇지만, 다른 실시예들에서, 적절한 경우, 메시지가 LCAR 위로 라우팅될 수 있다.

<352> RouteDown(M, P, ID) 함수는 타겟 근접 링 P의 멤버인 것으로 알려져 있는 진입 노드를 식별하는 것을 포함할 수 있다. 송신측 노드는 여러가지 서로 다른 메카니즘을 사용하여 타겟 근접 링 내의 진입 노드를 식별할 수 있다. 송신측 노드는 발신측 노드(제1 송신측 노드)일 수 있다. 송신측 노드는 또한 발신측 노드 또는 다른 중간 노드로부터 메시지를 수신하고 나서 이 메시지를 전달하는 중간 노드일 수 있다.

<353> 송신측 노드는, 예를 들어, 구성 또는 로컬 캐싱된 정보 등의 로컬 지식을 (예를 들어, CRS 엔트리 테이블에 부가하여) 사용하여 타겟 근접 링에 대한 진입 노드를 식별할 수 있다. 예를 들어, N[1311]은 링들로 이루어진 트리(1900) 내의 링에 관한 구성 또는 로컬 캐싱된 정보를 포함하는 캐쉬 및 구성(1902)에 액세스할 수 있다. 어떤 환경에서, 연합에 관한 로컬 지식은 연합 외부의(즉, 대역외) 통신 메카니즘을 통해 획득된다. 로컬 지식은 정확한 타겟 근접 링 P 내의 임의의 노드를 식별하는 데 사용될 수 있다. 정확한 타겟 근접 링 내의 임의의 노드가 발견된 경우, 메시지 M이 그 노드들 중 하나로 전달될 수 있다. 예를 들어, N[1311]은 캐쉬 및 구성(1902)을 사용하여 N[8004]를 식별하고 메시지(1903)를 R["4"]로 전달할 수 있다.

<354> 송신측 노드는 CRS 엔트리 테이블을 사용하여 타겟 근접 링 P 내의 임의의 진입 노드를 찾아낼 수 있다. 타겟 근접 링 내의 임의의 진입 노드가 발견된 경우, 메시지 M이 타겟 근접 링 P 내의 진입 노드로 전달될 수 있다. 예를 들어, N[7521]이 메시지(1903)의 목적지일 때, N[1311]은 N[6521]을 R["521"]에 대한 진입 노드로 식별할 수 있다. 그에 따라, 메시지(1903)는 N[6521]으로(또는 R["521"] 내로) 라우팅될 수 있다. R["521"] 내에서, N[6521]은 이어서 링내 통신(intra-ring communication)을 사용하여 N[7521]로 메시지(1903)를 라우팅하려고 할 수 있다. 따라서, 송신측 노드가 타겟 근접 링에 대한 진입 노드를 식별할 수 있는 경우, 송신측 노드는 정확한 타겟 근접 링으로 메시지를 전달한다. 예를 들어, N[6521]이 R["21"]에 대한 진입 노드를 식별할 수 있는 경우, N[6521]은 메시지(1903)를 R["21"]로 전달한다(점선).

<355> 반면에, 송신측 노드가 (예를 들어, 로컬 지식 또는 CRS 엔트리 테이블로부터) 목적지 타겟 근접지의 임의의 진입 노드를 식별할 수 없는 경우, 송신측 노드는 타겟 근접 링에 대한 임의의 선조 링의 진입 노드를 찾아내기 위해 그의 CRS 엔트리 테이블을 검사할 수 있다. 예를 들어, N[1311]이 R["121"] 내의 노드로 메시지를 전송하고자 하는 경우가 있을 수 있다. 그렇지만, N[1311]은 R["121"]에 대한 진입 노드를 식별할 수 없을지도 모른다. 그에 따라, N[1311]은 N[6521](R["21"]에의 진입 노드)을 찾아내기 위해 CRS 엔트리 테이블(1904)을 참조할 수 있다. N[1311]은 메시지를 N[6521]로 전달할 수 있고, N[6521]은 이제 메시지의 송신측 노드가 된다.

<356> 논리적으로, 메시지(1903)는 이제 R["21"]에 "있는" 것으로 볼 수 있는데, 그 이유는 N[6521]도 R["21"]의 노드이기 때문이다. N[6521]은 이어서 R["221"] (타겟 근접지)에의 진입 노드를 식별하려고 할 수 있다. 예를 들어, N[6521]은 로컬 지식, CRS 엔트리 테이블을 참조할 수 있고, 및/또는 RouteDown 알고리즘을 (재귀적으로) 적용할 수 있다. N[6521]이 N[3221]을 R["221"]에의 진입 노드로 식별하는 경우가 있을 수 있다. 그에 따라,

N[6521]은 메시지(1903)를 N[3221]로 전송할 수 있다. N[3221]은 이어서 링내 통신을 사용하여 메시지(1903)를 R["221"] 내의 적절한 노드로 라우팅할 수 있다.

- <357> 적절한 경우(예를 들어, 도 19b에 도시된 것보다 깊이가 더 깊은 트리에서), 타겟 근접 링에의 진입 노드가 식별될 때까지 또는 이용가능한 선조 링이 더 이상 없을 때까지 메시지가 타겟 근접 링에 점점 더 가까운 선조로 전달될 수 있다.
- <358> 송신측 노드는 또한 적어도 요청 타겟 근접지를 포함하는 진입 노드 탐색 요청(Entry Node Lookup Request)을 진입 노드 디렉토리 메카니즘(예를 들어, 연합을 부트스트랩하는 데 사용됨)으로 라우팅할 수 있다. 진입 노드 탐색 요청의 라우팅은 송신측 노드 및 타겟 링의 LCAR로 제약될 수 있다. 예를 들어, N[1311]은 R["31"]에 대한 진입 노드를 요청하기 위해 랑데뷰 지점(7651)으로 탐색 요청(1906)을 라우팅할 수 있다. 진입 노드 디렉토리 메카니즘은 잠재적인 진입 노드의 리스트를 반환할 수 있다. 예를 들어, 랑데뷰 지점(7651)은 잠재적인 진입 노드의 리스트(N[8431]을 포함함) 및 타겟 근접지에 대한 랑데뷰 지점에 등록된 적어도 씨드 노드를 탐색 응답(1907)으로 반환할 수 있다.
- <359> 송신측 노드는 (랑데뷰 지점으로부터 전송된) 탐색 응답 메시지에서 식별되는 새로 발견된 노드를 고려하고 메시지 M을 그 진입 노드들 중 하나로 전달한다. 새로 발견된 노드를 고려함으로써 CRS 엔트리 테이블은 물론 다른 로컬 캐싱된 노드 프레즌스 정보가 보장되고 다른 방식으로 유지될 수 있게 된다. 탐색 응답 메시지는 또한 다른 노드 프레즌스 정보를 포함할 수 있다. 예를 들어, 송신측 노드에 중요한 것으로 알려진 특정의 진입 노드가 송신측 노드의 CRS 엔트리 테이블에 포함될 수 있다.
- <360> 따라서, 하나 이상의 이용가능한 메카니즘이 타겟 근접 링에 적어도 하나의 진입 노드를 생성하는 경우, 메시지 M이 (최초의 송신측 노드, 타겟 근접지의 선조 링 내의 진입 노드, 또는 탐색된 진입 노드에 의해) 타겟 링 내의 진입 노드들 중 적어도 하나로 전송된다. 메시지 M은 타겟 근접 링 내의 진입 노드이 메시지 M을 ID로 라우팅하라는 명령어를 포함할 수 있다. 반면에, 이용가능한 메카니즘 중 어느 것도 타겟 근접 링 내에 진입 노드를 생성하지 않는 경우, RouteDown 요청이 실패하여 다시 최초의 송신측 노드 쪽으로 갈 수 있다.
- <361> 어떤 실시예들에서, 발신측 노드는 엔드-투-엔드 RouteDown 메시지 헤더를 메시지에 첨부하며, 여기서 RouteDown 메시지 헤더는 타겟 근접 URI를 명시하고 있다. 상기 메카니즘들 중 어느 것이 다수의 "그 다음 홉" 노드를 식별하는 경우, 단 하나의 노드가 선택될 수 있다. 단 하나의 노드를 선택할 때(예를 들어, 동점을 깨뜨릴 때), 타겟 근접지 P에 더 가까운 노드가 선택될 수 있고, 그 다음에 더 높은 가중치를 갖는 노드가 선택될 수 있으며, 그 다음에 M의 목적지 ID에 더 가까운 노드가 선택될 수 있다. 어떤 메카니즘에 의해서도 선택이 행해지지 않은 경우, 노드가 랜덤하게 선택될 수 있다. 메시지 M을 전달하려는 시도가 실패하는 경우, 다수의 후보가 있을 때, 장애없는 후보 노드가 있는 한, 전달이 재시도될 수 있다.
- <362> 애플리케이션 메시지가 발신자로부터 중간자를 통해 최종 목적지 노드로 전달될 때 역방향 상태/고장 메시지 경로가 설정될 수 있다. 메시지가 목적지로 전달되거나 고장이 검출된 경우, 대응하는 고장/상태 메시지가 이 경로를 따라 발신자 쪽으로 반송될 수 있다.
- <363> 도 19c는 링들로 이루어진 근접 유도 분할 트리(1900)의 일례의 일부분을 분할하여 나타낸 것이다. 도 19d는 링들로 이루어진 근접 유도 분할 트리(1900)의 일례로부터의 링(11)을 확대하여 나타낸 도면이다. 도 20은 링들로 이루어진 트리 내의 노드에 대한 방계 링 세트를 유지하는 방법(200)의 플로우차트의 일례를 나타낸 것이다. 방법(2000)은 도 19c 및 도 19d의 링, 노드, 메시지 및 데이터와 관련하여 설명될 것이다.
- <364> 방법(2000)은 노드가 노드에 대한 방계 링 세트 엔트리를 저장하도록 구성된 방계 링 세트 엔트리 테이블에 액세스하는 동작(동작 2001)을 포함한다. 각각의 방계 링 세트 엔트리는 노드의 방계 링 및 노드의 방계 링으로의 적어도 하나의 대응하는 진입 노드를 나타내도록 구성되어 있다. 예를 들어, N[1311]은 N[1311]에 대한, <방계 링, 1 내지 N개의 진입 노드> 형식(단, N은 어떤 정수임)으로 방계 링 세트 엔트리를 저장하도록 구성된 CRS 엔트리 테이블(1904)에 액세스한다. 따라서, CRS 엔트리 테이블(1904)은 0개 이상의 <방계 링, 진입 노드> 항목을 포함할 수 있고, 각각의 포함된 항목은 N[1311]의 방계 링 및 그 방계 링으로의 하나 이상의 대응하는 진입 노드를 나타낸다. 예를 들어, 도 19c에 도시된 바와 같이, CRS 엔트리 테이블(1904)은 N[8651]이 R["51"] (N[1311] 및 R["311"]의 방계 링)로의 1 내지 N개의 진입 노드 중 하나임을 나타내는 CRS 엔트리 <R["51"], N[8651], ...>를 포함한다.
- <365> 방법(2000)은 링들로 이루어진 트리의 구성에 관한 정보를 유지하는 이용가능한 자원으로부터 방계 링 세트 엔트리 테이블 정보를 발견하는 동작(동작 2002)을 포함한다. 예를 들어, N[1311]은 링들로 이루어진 트리(190



0)의 구성에 관한 정보를 유지하는 자원으로부터 방계 링 세트 엔트리 테이블 정보를 발견할 수 있다. 앞서 설명한 바와 같이, 노드가 여러가지 서로 다른 CRS 엔트리 테이블 정보의 자원들을 이용할 수 있다. 예를 들어, 노드는, 예를 들어, 로컬 구성 및 캐시 정보 등의 로컬 지식에 액세스할 수 있고, 링들로 이루어진 트리에서 상태를 전달하는 데 사용되는, 예를 들어, ping 메시지, 갱신 메시지, 및 갱신 응답 등의 라데뷰 프로토콜 메시지에 포함된 CRS 관련 상태에 액세스할 수 있으며, 애플리케이션 메시지에 포함된 CRS 관련 상태에 액세스할 수 있고, 링들로 이루어진 트리에서, 예를 들어, 브로드캐스트, 멀티캐스트 및 애니캐스트 등의 지정된 통신 패턴을 용이하게 해주는 데 사용되는 메시지로부터의 CRS 관련 상태에 액세스할 수 있다.

<366> 따라서, 도 19c에서, N[1311]이 로컬 구성(1921)에 액세스하는 경우가 있을 수 있다. 로컬 구성(1921)으로부터, N[1311]은 N[1111]이 R["111"] (N[1311] 및 R["311"]의 방계 링)로의 진입 노드들 중 적어도 하나임을 나타내는 CRS 엔트리 <R["111"], N[1111], ...>를 발견할 수 있다. N[1311]은 또한 애플리케이션 메시지(1971)를 수신할 수 있다. 애플리케이션 메시지(1971)로부터, N[1311]은 CRS 정보(1972)를 발견할 수 있다. CRS 정보(1972)는 N[1311]의 CRS에 링들에 대한 CRS 상태를 포함하거나 포함하지 않을 수 있다. N[1311]은 또한 통신 패턴 관련 메시지(1973)도 수신할 수 있다. 통신 패턴 관련 메시지(1973)로부터, N[1311]은 CRS 상태(1974)를 발견할 수 있다. CRS 상태(1974)는 N[1311]의 CRS 내의 링들에 대한 CRS 상태를 포함하거나 포함하지 않을 수 있다.

<367> 이제 도 19d를 참조하면, N[1311]은 라데뷰 프로토콜 메시지에서 CRS 상태를 발견하고 이를 교환할 수 있다. N[1111]이 R["111"]의 멤버이고, N[1211]이 R["211"]의 멤버이며, N[1311]이 R["311"]의 멤버이기 때문에, 노드 N[1111], N[1211], N[1311] 각각도 역시 R["11"]의 멤버이다. 앞서 설명한 바와 같이, 공통 링의 멤버인 노드들은 라우팅 테이블 정보를 유지하기 위해 ping 메시지, 갱신 메시지, 및 갱신 응답을 교환할 수 있다. 따라서, R["11"]의 노드들은 R["11"]에 대한 라우팅 테이블 정보를 유지하기 위해 ping 메시지, 갱신 메시지, 및 갱신 응답을 교환할 수 있다. CRS 상태가 교환되는 ping 메시지, 갱신 메시지, 및 갱신 응답에는 물론 노드들 간의 다른 라데뷰 프로토콜 및 애플리케이션 메시지 트래픽에 포함될 수 있다.

<368> 예를 들어, N[A11](R["11"] 내의 N[B11]의 이웃)은 CRS 상태(1932)를 포함하는 ping 메시지(1931)를 N[1311]로 전송할 수 있다. CRS 상태(1932)로부터, N[1311]은 N[1131]이 R["31"] (N[1311] 및 R["311"]의 방계 링)로의 진입 노드임을 나타내는 CRS 엔트리 <R["31"], N[1131]>를 발견할 수 있다. CRS 엔트리(1932)는 N[A11]의 CRS 엔트리 테이블 내의 CRS 엔트리의 전체 또는 부분 리스트일 수 있다. N[1311]은 또한 CRS 상태를 포함하는 ping 메시지를 그의 이웃들에게 전송할 수 있다. 예를 들어, N[1311]은 CRS 상태(1934)를 포함하는 ping 메시지(1945)를 N[E11](R["11"] 내의 N[1311]의 이웃)에게 전송할 수 있다. CRS 엔트리(1934)는 CRS 엔트리 테이블(1904)로부터의 CRS 엔트리의 전체 또는 부분 리스트를 포함할 수 있다.

<369> N[1311]은 또한 CRS 관련 정보를 포함하는 갱신 메시지 및 갱신 응답을 전송 및 수신할 수 있다. 예를 들어, N[1311]은 CRS 엔트리(1934)를 포함하는 갱신 메시지(1933)를 N[D11](R["11"] 내의 N[1311]의 라우팅 파트너)에게 전송할 수 있다. N[D11]은 CRS 엔트리(1938)를 포함하는 갱신 응답(1937)을 N[1311]에게 전송하는 것으로 응답할 수 있다. CRS 엔트리(1938)는 N[D11]의 CRS 엔트리 테이블 내의 CRS 엔트리의 전체 또는 부분 리스트일 수 있다. 이와 유사하게, N[1311]은 CRS 엔트리(1942)를 포함하는 갱신 메시지(1941)를 N[C11](R["11"] 내의 N[1311]의 라우팅 파트너)로부터 수신할 수 있다. CRS 엔트리(1942)는 N[C11]의 CRS 엔트리 테이블 내의 CRS 엔트리의 전체 또는 부분 리스트일 수 있다. N[1311]은 CRS 엔트리(1934)를 포함하는 갱신 응답(1943)을 N[C11]에게 전송하는 것으로 응답할 수 있다.

<370> 노드는 또한 이용가능한 자원으로부터 CRS 엔트리가 더 이상 유효하지 않을지도 모른다는 것을 (직접적으로 또는 간접적으로) 알려주는, 예를 들어, 진입 노드에 접촉할 수 없음을 알려주는 방계 링 세트 엔트리 테이블 관련 정보도 수신할 수 있다. CRS 관련 상태를 전송하는 데 사용되는 자원은 또한 CRS 엔트리가 더 이상 유효하지 않을지도 모른다는 것을 의미하는 것으로 해석될 수 있는 표시를 전송하는 데도 사용될 수 있다. 따라서, 때때로 노드가 하나 이상의 CRS 엔트리가 그의 CRS 엔트리 테이블에 추가되게 하는 CRS 관련 상태는 물론 더 이상 적절하지 않을지도 모르는 하나 이상의 CRS 엔트리를 제거하게 할 수 있는 표시를 수신하는 경우가 있을 수 있다.

<371> 방법(2000)은 발견된 방계 링 세트 엔트리 테이블 정보에 기초하여 적절한 방계 링 세트 엔트리 상태를 사용하여 방계 링 세트 엔트리 테이블을 갱신하는 동작(동작 2003)을 포함한다. 각각의 적절한 방계 링 세트 엔트리 상태는 노드의 방계 링 및 노드의 방계 링으로의 적어도 하나의 대응하는 진입 노드를 포함한다. 예를 들어, N[1311]은 도 19c 및 도 19d에서 수신된 CRS 엔트리를 CRS 엔트리 테이블(1904)에 포함시킬 수 있다. N[131



1]은 또한 어쩌면 더 이상 적절하지 않은 것으로 (예를 들어, 구성, 랑데뷰 프로토콜 메시지, 애플리케이션 메시지, 또는 통신 패턴 관련 메시지에) 표시된 CRS 엔트리를 CRS 엔트리 테이블(1904)로부터 제거할 수 있다. 그에 따라, 랑데뷰 연합의 구조 변화를 적절히 반영하기 위해 노드의 CRS 엔트리 테이블이 갱신될 수 있다.

<372> 도 19e는 링들로 이루어진 근접 유도 분할 트리(1900)의 일례의 다른 뷰를 나타낸 것이다. 도 19e에는 도 19c 및 도 19d에서 교환된 CRS 상태에 기초하여 채워져 있을 수 있는 CRS 엔트리 테이블(1904)이 도시되어 있다. 도 21은 링들로 이루어진 트리에서 근접시간 통신을 전송하는 방법(2100)의 플로우차트의 일례를 나타낸 것이다. 방법(2100)은 도 19f의 노드, 링, 메시지 및 데이터와 관련하여 설명될 것이다.

<373> 방법(2100)은 노드가 노드의 방계 링으로 메시지를 전송하는지를 판정하는 동작(동작 2101)을 포함한다. 예를 들어, N[1311]은 메시지(1976)를 R["2"]로 전송해야 한다는 표시를 수신할 수 있다. 메시지가 방계 링으로 전송되어야 한다는 표시는 다른 노드로부터 수신될 수 있고, 이 다른 노드는 라우팅 논리의 함수, N[1311]에서의 애플리케이션, 멀티캐스트 설비, 브로드캐스트 설비, 애니캐스트 설비, 기타 등등을 의미할 수 있다.

<374> 방법(2100)은 노드가 노드에 대한 방계 링 세트 엔트리를 저장하도록 구성되어 있는 방계 링 세트 엔트리 테이블에 액세스하는 동작(동작 2102)을 포함한다. 각각의 방계 링 세트 엔트리는 노드의 방계 링 및 노드의 방계 링에 적어도 하나의 대응하는 진입 노드를 나타내도록 구성되어 있다. 예를 들어, N[1311]은 CRS 엔트리 테이블(1904)에 액세스할 수 있다. CRS 엔트리 테이블(1904) 내의 각각의 CRS 엔트리는 N[1311]의 방계 링 및 N[1311]의 방계 링에 적어도 하나의 진입 노드를 나타낼 수 있다. 예를 들어, 엔트리 <R["111"], N[1111], ...>은 R["111"]이 N[1311]의 방계 링이고 N[1111]이 R["111"]에 적어도 하나의 진입 노드 중 하나임을 나타낸다.

<375> 방법(2100)은 노드의 방계 링 세트 엔트리 테이블로부터 방계 링에 대한 적어도 하나의 방계 링 세트 엔트리를 식별하는 동작(동작 2103)을 포함한다. 적어도 하나의 방계 링 세트 엔트리 각각은 방계 링의 적어도 하나의 진입 노드를 나타낸다. 예를 들어, N[1311]은 CRS 엔트리 테이블(1904)로부터 R["2"]에 대한 엔트리 <R["2"], N[1112], ...>를 식별할 수 있다. 엔트리 <R["2"], N[1112], ...>는 N[1112](어쩌면 다른 노드들)가 R["2"]에 대한 진입 노드임을 나타낸다.

<376> 식별된 방계 링 세트 엔트리에 포함된 대응하는 진입 노드의 수에 기초하여(예를 들어, 2개 이상의 진입 노드가 있는 경우), 방법(2100)은 또한 방계 링에 대한 복수의 진입 노드를 적절한 진입 노드들의 서브셋 또한 어쩌면 단 하나의 적절한 진입 노드로 해석하는 동작을 포함할 수 있다. 예를 들어, 적절한 진입 노드들의 서브셋이 발신측 근접지 P와 타겟 근접지 P 간의 근접성에 기초하여, 선택된 노드 가중치에 기초하여, 목적지 ID에의 근접성에 기초하여 결정될 수 있거나, 또는 랜덤하게 선택될 수 있다.

<377> 방법(2100)은 적어도 하나의 표시된 진입 노드로 메시지를 전송하는 동작(동작 2104)을 포함한다. 예를 들어, N[1311]은 메시지(1976)를 N[1112]로 전송할 수 있다. 메시지를 적어도 하나의 노드로 전송하는 것은 메시지를 복수의 노드 중의 모든 진입 노드로, 해석된 적절한 진입 노드의 서브셋 내의 각각의 진입 노드로, 또는 단 하나의 적절한 진입 노드로 전송하는 것을 포함할 수 있다. 어떤 실시예들에서, 하나의 진입 노드에 대해 메시지 장애가 일어나는 경우, 하나 이상의 다른 진입 노드가 시도될 수 있다. 송신측 노드가 고장의 결과로서 새로운 노드를 식별하는 것도 가능하다.

<378> 도 22는 링들로 이루어진 트리에서 근접시간 통신을 전송하는 방법(2200)의 플로우차트의 일례를 나타낸 것이다. 방법(2200)은 도 19f 및 도 19g의 노드, 링, 메시지 및 데이터와 관련하여 설명될 것이다.

<379> 방법(2200)은 발신측 노드가 링들로 이루어진 트리 내에서 타겟 근접 링 내의 주어진 노드 ID에 가장 가까운 목적지 노드로 메시지를 라우팅하려고 하는지를 판정하는 동작(동작 2201)을 포함한다. 타겟 근접 링은 발신측 노드의 방계 링 또는 발신측 노드의 방계 링의 서브-링일 수 있다. 예를 들어, N[1311]은 메시지(1998)를 R["1221"] (타겟 근접 링) 내의 ID 30 쪽으로 라우팅해야 한다는 표시를 수신할 수 있다. 메시지가 방계 링 또는 방계 링의 서브-링으로 전송되어야 한다는 표시는 다른 노드, N[1311]에 관련된 애플리케이션, 멀티캐스트 설비, 브로드캐스트 설비, 애니캐스트 설비, 기타 등등으로부터 수신될 수 있다.

<380> 방법(2200)은 타겟 근접 링 및 타겟 근접 링의 선조 링 중 적어도 하나의 멤버 노드인 것으로 알려진 하나 이상의 진입 노드를 식별하는 동작(동작 2202)을 포함한다. 예를 들어, N[1311]은 노드 ID 56을 갖는 N[41221]를 R["1221"]에의 진입 노드로서 식별할 수 있다. N[41221]을 식별하는 데 여러가지 서로 다른 메커니즘이 사용될 수 있다. N[1311]은 타겟 근접 링에의 진입 노드를 식별하려고 하기 위해 로컬 지식을 참조할 수 있다. 예를 들어, N[1311]은 R["1221"]에의 진입 노드를 식별하려고 하기 위해 캐쉬 및 구성(1902)을 참조할 수 있다.

- <381> N[1311]은 또한 (예를 들어, 타겟 근접 링으로의 진입 노드가 식별되지 않은 경우) 타겟 근접 링의 선조 링으로의 진입 노드를 식별하기 위해 CRS 엔트리 테이블을 참조할 수 있다. 서브-링 R["321"]이 노드 N[4321]을 R["21"]로의 진입 노드로서 제공하는 경우가 있을 수 있다. 이와 마찬가지로, R["2221"]이 R["221"]로의 진입 노드로서 노드 N[12221]을 제공하는 경우가 있을 수 있다. R["1221"]로의 진입 노드가 식별되지 않은 경우, N[1311]은 R["221"]로의 진입 노드(예를 들어, N[12221] 등)를 식별하려고 할 수 있다. N[1331]은 또한 R["21"]로의 진입 노드(예를 들어, N[4321] 등)를 식별하려고 할 수 있다.
- <382> 타겟 근접지로의 진입 노드가 식별되지 않은 경우, 노드가 지정된 타겟 근접지의 관점에서 볼 때 더 멀리 있는 선조 내의 진입 노드를 식별하려고 하기 전에 더 가까운 선조 링 내의 진입 노드를 식별하려고 하는 경우가 있을 수 있다. 예를 들어, R["1221"]으로의 진입 노드가 식별되지 않은 경우, N[1311]은 먼저 R["221"] 내의 진입 노드를 식별하려고 할 수 있다. R["221"]로의 진입 노드가 식별되지 않은 경우, N[1311]은 R["21"] 내의 진입 노드를 식별하려고 할 수 있다.
- <383> N[1311]은 또한, 예를 들어, 씨드 노드 등의 부트스트랩 메카니즘을 이용할 수 있다. 예를 들어, N[1311]은 진입 노드 탐색 요청을, 예를 들어, 랑데뷰 지점 N[7651] 등의 기지의 랑데뷰 지점으로 라우팅하여, 기지의 (등록된) 진입 노드(씨드 노드가 일레임)를 요청할 수 있다. 탐색 요청에 응답하여, 랑데뷰 지점은 기지의 진입 노드를 포함하는 탐색 응답 메시지를 반환(전송)할 수 있다. 예를 들어, 탐색 응답(1997)이 랑데뷰 지점 N[7651]로부터 N[1311]로 반환될 수 있다. 탐색 응답(1997)은 랑데뷰 지점 N[7651]에 등록된 진입 노드의 위치를 포함할 수 있다.
- <384> 이들 메카니즘 중 하나 이상이 N[4221]을 R["221"]에의 진입 노드로서 식별할 수 있다.
- <385> 어떤 실시예들에서, 어떤 진입 노드 식별 메카니즘이 다른 진입 노드 식별 메카니즘보다 이전에 이용된다. 예를 들어, 송신측 노드는 타겟 링의 선조 링으로의 진입 노드를 식별하려고 하거나 진입 노드 탐색 요청을 라우팅하기 전에 로컬 지식을 참조할 수 있다. 이들 동일 실시예에서, 송신측 노드는 또한 진입 노드 탐색 요청을 라우팅하기 전에 타겟 링의 선조 링으로의 진입 노드를 식별하려고 할 수 있다. 그렇지만, 다른 실시예들에서, 진입 노드 식별 메카니즘이 다른 순서로 이용될 수 있거나 생략될 수 있다.
- <386> 방법(2200)은 메시지를 식별된 진입 노드로 전송하는 동작(동작 2203)을 포함한다. 이 메시지는 진입 노드가 메시지를 해석하여 타겟 근접 링 내의 표시된 목적지 노드에 가장 가까운 노드 ID를 갖는 노드로 보내야 함을 나타내고 있다. 예를 들어, 실선으로 나타낸 바와 같이, N[1311]은 메시지가 해석되어 노드 ID 30로 보내져야 한다는 표시에 따라 메시지(1998)를 N[41221](R["1221"]로의 진입 노드로서 노드 ID 56을 가짐)로 전송할 수 있다. N[41221]은 자신이 알고 있는 노드 ID 30에 가장 가까운 노드 ID가 노드 ID 25를 갖는 N[61221]인지를 판정하기 위해 그의 라우팅 테이블 및/또는 이웃에 액세스할 수 있다. 이와 마찬가지로, N[61221]은 자신이 알고 있는 노드 ID 30에 가장 가까운 노드 ID가 노드 ID 28을 갖는 N[71221]인지를 판정하기 위해 그의 라우팅 테이블 및/또는 이웃에 액세스할 수 있다. N[71221]은 그의 노드 ID, 즉 노드 ID 28이 노드 ID 30에 가장 가까운 기지의 노드 ID인지를 판정하고 메시지를 전달하기 위해 그의 테이블 및/또는 이웃을 참조할 수 있다.
- <387> 앞서 설명한 라우팅 알고리즘들은 R["221"] 내에서 메시지(1998)를 라우팅하는 데도 사용될 수 있다.
- <388> 메시지가 선조 링 또는 타겟 근접 링의 멤버인 식별된 진입 노드로 전송될 때, 식별된 진입 노드에서 방법(2200)이 재귀적으로 적용될 수 있다. 즉, 선조 링으로의 식별된 진입 노드는 이번에는 타겟 근접 링으로의 진입 노드를 식별하려고 할 수 있다. 예를 들어, 점선으로 나타낸 바와 같이, 방법(2200)의 적용에 의해 메시지(1998)가 진입 노드 N[12221](서브-링 R["2221"]에 의해 제공되는 R["221"]에 대한 진입 노드)로 전송되는 경우가 있을 수 있다. (R["2221"]의 관점에서 볼 때) N[12221]은 R["1221"]로의 진입 노드를 식별할 수 있다. 따라서, N[12221]에서의 방법(2200)의 재귀적 적용에 의해 메시지(1998)가 N[41221]로 전송될 수 있다.
- <389> 선조 링 내의 식별된 진입 노드가 타겟 근접 링으로의 진입 노드를 모르고 있는 경우, 식별된 진입 노드는 타겟 근접지에 더 가까운 다른 선조 링으로의 진입 노드를 식별하려고 할 수 있다. 선조 링으로의 진입 노드는 메시지를 타겟 근접 링의 보다 가까운 선조 링의 진입 노드로 전달할 수 있다.
- <390> 예를 들어, 점선으로 나타낸 바와 같이, 방법(2200)의 적용에 의해 메시지(1998)가 진입 노드 N[4321](서브-링 R["321"]에 의해 제공되는 R["21"]의 진입 노드)로 전송되는 경우가 있을 수 있다. 그렇지만, (R["321"]의 관점에서 볼 때) N[4321]은 R["1221"] 내의 진입 노드를 식별하지 못할지도 모른다. 따라서, (R["321"]의 관점에서 볼 때) N[4321]은 R["221"] 내의 진입 노드를 식별할지도 모른다. 따라서, N[4321]에서의 방법(2200)의 첫 번째 재귀적 적용에 의해 메시지(1998)가 N[12221]로 전송될 수 있다. N[12221]에서의 방법(2200)의 두 번째 재

귀적 적용에 의해 메시지(1998)가 N[41221]로 전송될 수 있다.

- <391> 그렇지만, N[4321]이 R["1221"]로의 진입 노드를 식별한 경우, N[4321]은 메시지(1998)를 진입 노드로 직접 전송할 수 있다. 따라서, 선조 링으로의 진입 노드는 타겟 근접 링으로 또는 적절한 경우 타겟 근접 링의 보다 가까운 선조 링의 진입 노드(통상적으로 전달하는 진입 노드의 CRS 내에 있음)로 메시지를 전달할 수 있다. 상기한 바와 같이, 보다 가까운 선조 링으로의 진입 노드는 메시지를 타겟 근접 링으로 또는 적절한 경우 타겟 근접 링의 훨씬 더 가까운 선조 링(전달하는 "훨씬 더 가까운" 진입 노드의 CRS에 있음)으로 전달할 수 있다. 이 프로세스는 (예를 들어, 방법(2200)을 재귀적으로 적용함으로써) 타겟 근접 링 진입 노드에 도달할 때까지 반복될 수 있다.
- <392> 타겟 근접 링에 도달될 때, 타겟 근접 링 내에서 메시지를 라우팅하기 위해 앞서 설명한 라우팅 알고리즘이 사용될 수 있다.
- <393> 도 6 및 이하의 설명은 본 발명이 실시될 수 있는 적절한 컴퓨팅 환경에 대한 간략하고 개괄적인 설명을 제공하는 것을 의도한 것이다. 꼭 그럴 필요는 없지만, 본 발명은 일반적으로 컴퓨터 시스템에 의해 실행되는 프로그램 모듈 등의 컴퓨터 실행가능 명령어와 관련하여 설명된다. 일반적으로, 프로그램 모듈은 특정의 태스크를 실행하거나 특정의 추상 데이터 유형을 구현하는, 루틴, 프로그램, 객체, 컴포넌트, 데이터 구조 등을 포함한다. 컴퓨터 실행가능 명령어, 관련 데이터 구조, 및 프로그램 모듈은 본 명세서에서 개시되는 방법의 동작을 실행하는 프로그램 코드 수단의 예를 나타낸다.
- <394> 도 6을 참조하면, 본 발명을 실시하기 위한 예시적인 시스템은 처리 장치(621), 시스템 메모리(622), 및 시스템 메모리(622)를 비롯한 여러가지 시스템 컴포넌트를 처리 장치(621)에 결합시키는 시스템 버스(623)를 포함하는 컴퓨터 시스템(620) 형태의 범용 컴퓨팅 장치를 포함한다. 처리 장치(621)는 본 발명의 기능을 비롯하여 컴퓨터 시스템(620)의 기능을 구현하도록 설계된 컴퓨터 실행가능 명령어를 실행할 수 있다. 시스템 버스(623)는 메모리 버스 또는 메모리 컨트롤러, 주변 장치 버스, 및 여러가지 버스 아키텍처 중 임의의 것을 이용하는 로컬 버스를 포함한 몇가지 유형의 버스 구조 중 임의의 것일 수 있다. 시스템 메모리는 판독 전용 메모리(ROM)(624), 및 랜덤 액세스 메모리(RAM)(625)를 포함한다. 기본 입/출력 시스템(BIOS)(626)은, 시동 중과 같은 때에, 컴퓨터 시스템(620) 내의 구성요소들 간의 정보 전송을 돕는 기본 루틴을 포함하며, ROM(624)에 저장될 수 있다.
- <395> 컴퓨터 시스템(620)은 또한 자기 하드 디스크(639)에 기록을 하고 그로부터 판독을 하는 자기 하드 디스크 드라이브(627), 이동식 자기 디스크(629)에 기록을 하거나 그로부터 판독을 하는 자기 디스크 드라이브(628), 및, 예를 들어, CD-ROM 또는 기타 광 매체 등 이동식 광 디스크(631)에 기록을 하거나 그로부터 판독을 하는 광 디스크 드라이브(630)도 포함할 수 있다. 자기 하드 디스크 드라이브(627), 자기 디스크 드라이브(628), 및 광 디스크 드라이브(630)는 각각 하드 디스크 드라이브 인터페이스(632), 자기 디스크 드라이브 인터페이스(633), 및 광 드라이브 인터페이스(634)에 의해 시스템 버스(623)에 접속된다. 드라이브 및 관련 컴퓨터 판독가능 매체는 컴퓨터 시스템(620)에 대한 컴퓨터 실행가능 명령어, 데이터 구조, 프로그램 모듈, 및 기타 데이터의 비휘발성 저장을 제공한다. 본 명세서에서 설명하는 예시적인 환경이 자기 하드 디스크(639), 이동식 자기 디스크(629), 및 이동식 광 디스크(631)를 이용하지만, 자기 카세트, 플래쉬 메모리 카드, DVD, 베르누이 카트리지, RAM, 및 ROM 등을 비롯한, 데이터를 저장하기 위한 기타 유형의 컴퓨터 판독가능 매체가 사용될 수 있다.
- <396> 운영 체제(635), 하나 이상의 애플리케이션 프로그램(636), 기타 프로그램 모듈(637), 및 프로그램 데이터(638)를 비롯한 하나 이상의 프로그램 모듈을 포함하는 프로그램 코드 수단은 하드 디스크(639), 자기 디스크(629), 광 디스크(631), ROM(624), 또는 RAM(625)에 저장될 수 있다. 사용자는 키보드(640), 포인팅 장치(642), 또는, 예를 들어, 마이크, 조이스틱, 게임 패드, 또는 스캐너 등의 기타 입력 장치(도시하지 않음)를 통해 컴퓨터 시스템(620)에 명령 및 정보를 입력할 수 있다. 이들 및 기타 입력 장치는 시스템 버스(623)에 결합된 입/출력 인터페이스(646)를 통해 처리 장치(621)에 접속될 수 있다. 입/출력 인터페이스(646)는 논리적으로, 예를 들어, 직렬 포트 인터페이스, PS/2 인터페이스, 병렬 포트 인터페이스, USB(Universal Serial Bus) 인터페이스, 또는 IEEE1394 인터페이스(즉, FireWire 인터페이스) 등 다양한 서로 다른 인터페이스 중 임의의 것을 나타내거나, 논리적으로 심지어 서로 다른 인터페이스의 조합도 나타낼 수 있다.
- <397> 모니터(647) 또는 기타 디스플레이 장치도 비디오 인터페이스(648)를 통해 시스템 버스(623)에 접속된다. 스피커(669) 또는 기타 오디오 출력 장치도 오디오 인터페이스(649)를 통해 시스템 버스(623)에 접속된다. 예를 들어, 프린터 등의 기타 주변 출력 장치(도시하지 않음)도 컴퓨터 시스템(620)에 접속할 수 있다.

- <398> 컴퓨터 시스템(620)은, 예를 들어, 사무실 규모 또는 전사적 컴퓨터 네트워크, 홈 네트워크, 인트라넷, 및/또는 인터넷 등의 네트워크에 접속가능하다. 컴퓨터 시스템(620)은 이러한 네트워크를 통해, 예를 들어, 원격 컴퓨터 시스템, 원격 애플리케이션, 및/또는 원격 데이터베이스 등의 외부 소스와 데이터를 교환할 수 있다.
- <399> 컴퓨터 시스템(620)은 컴퓨터 시스템(620)이 외부 소스로부터 데이터를 수신하고 및/또는 외부 소스에 데이터를 전송하는 데 이용하는 네트워크 인터페이스(653)를 포함한다. 도 6에 나타난 바와 같이, 네트워크 인터페이스(653)는 링크(651)를 통해 원격 컴퓨터 시스템(683)과의 데이터 교환을 용이하게 해준다. 네트워크 인터페이스(653)는 논리적으로, 예를 들어, 네트워크 인터페이스 카드 및 대응하는 NDIS(Network Driver Interface Specification, 네트워크 드라이버 인터페이스 규격) 스택 등, 하나 이상의 소프트웨어 및/또는 하드웨어 모듈을 나타낼 수 있다. 링크(651)는 네트워크의 일부(예를 들어, 이더넷 세그먼트)를 나타내고, 원격 컴퓨터 시스템(683)은 네트워크의 노드를 나타낸다.
- <400> 마찬가지로, 컴퓨터 시스템(620)은 컴퓨터 시스템(620)이 외부 소스로부터 데이터를 수신하고 및/또는 외부 소스에 데이터를 전송하는 데 이용하는 입/출력 인터페이스(646)를 포함한다. 입/출력 인터페이스(646)는 컴퓨터 시스템(620)이 외부 소스로부터 데이터를 수신하고 및/또는 외부 소스에 데이터를 전송하는 데 이용하는 링크(659)를 통해 모뎀(654)(예를 들어, 표준 모뎀, 케이블 모뎀, 또는 디지털 가입자 회선(DSL) 모뎀)에 결합된다. 도 6에 나타난 바와 같이, 입/출력 인터페이스(646)는 링크(652)를 통해 원격 컴퓨터 시스템(693)과의 데이터 교환을 용이하게 해준다. 링크(652)는 네트워크의 일부를 나타내고, 원격 컴퓨터 시스템(693)은 네트워크의 노드를 나타낸다.
- <401> 도 6은 본 발명에 대한 적절한 동작 환경을 나타내지만, 본 발명의 원리는, 필요한 경우 적절한 변경을 하여, 본 발명의 원리를 실시할 수 있는 임의의 시스템에서 이용될 수 있다. 도 6에 나타난 환경은 예시적인 것에 지나지 않으며, 본 발명의 원리가 실시될 수 있는 다양한 환경의 작은 일부분조차도 대표하지 않는다.
- <402> 본 발명에 따르면, 노드, 애플리케이션 계층, 및 기타 하위 계층은 물론 라우팅 테이블 및 노드 ID를 비롯한 관련 데이터가 저장되어, 컴퓨터 시스템(620)과 연관된 임의의 컴퓨터 관독가능 매체로부터 액세스될 수 있다. 예를 들어, 이러한 모듈의 일부 및 관련 프로그램 데이터의 일부는 운영 체제(635), 애플리케이션 프로그램(636), 프로그램 모듈(637), 및/또는 프로그램 데이터(638)에 포함되어 시스템 메모리(622)에 저장될 수 있다.
- <403> 예를 들어, 자기 하드 디스크(639) 등의 대용량 저장 장치가 컴퓨터 시스템(620)에 결합되어 있을 경우, 이러한 모듈 및 관련 프로그램 데이터도 대용량 저장 장치에 저장될 수 있다. 네트워크화된 환경에서는, 컴퓨터 시스템(620) 또는 그 일부에 대해 나타난 프로그램 모듈은 원격 컴퓨터 시스템(683) 및/또는 원격 컴퓨터 시스템(693)과 연관된 시스템 메모리 및/또는 대용량 저장 장치 등의 원격 메모리 저장 장치에 저장될 수 있다. 이러한 모듈의 실행은 앞서 설명한 것처럼 분산 환경에서 수행될 수 있다.

### 산업상 이용 가능성

- <404> 본 발명은 본 발명의 정신 또는 필수적인 특성을 벗어나지 않고 기타 특성의 형태로 실시될 수 있다. 설명한 실시예는 모든 점에서 제한하는 것이 아닌 예시적인 것으로 간주되어야 한다. 따라서, 본 발명의 범위는 상기의 설명에 의해서보다는 첨부된 특허 청구의 범위에 의해 나타내진다. 특허 청구의 범위의 의미 및 균등성의 범위에 들어가는 모든 변경은 이러한 범위에 포함되는 것으로 한다.

### 도면의 간단한 설명

- <18> 도 1은 연합 기반구조(federation infrastructure)의 일례를 나타낸 도면.
- <19> 도 2는 요청을 파트너들에게 간접적으로 라우팅하는 것을 용이하게 해주는 컴퓨터 아키텍처의 일례를 나타낸 도면.
- <20> 도 3은 정렬 리스트(sorted list) 및 대응하는 링의 형태로 된 연합 기반구조 내의 노드들 간의 이항 관계(binary relationship)의 일례를 나타낸 도면.
- <21> 도 4는 근접 라우팅(proximal routing)을 용이하게 해주는 링들로 이루어진 링의 일례를 나타낸 도면.
- <22> 도 5는 근접 라우팅을 용이하게 해주는 링들로 이루어진 근접 유도 분할 트리(proximity induced partition tree)의 일례를 나타낸 도면.
- <23> 도 5a는 도 5의 링들로 이루어진 분할 트리의 일부분에 부가적인 상세를 갖는, 도 5의 링들로 이루어진 근접 유



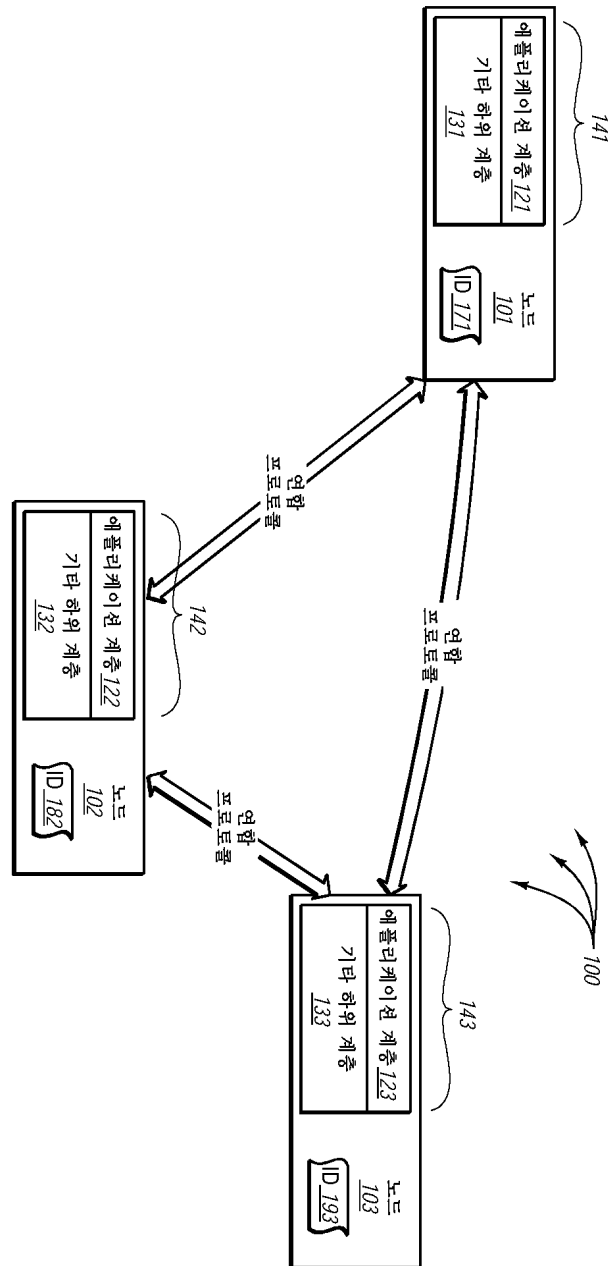
도 분할 트리의 일례를 나타낸 도면.

- <24> 도 6은 본 발명의 원리들에 적당한 동작 환경을 나타낸 도면.
- <25> 도 7은 근접성 기준을 고려하는 노드 라우팅 테이블(node routing table)을 채우는 방법의 플로우차트의 일례를 나타낸 도면.
- <26> 도 8은 연합 기반구조의 노드들을 분할하는 방법의 플로우차트의 일례를 나타낸 도면.
- <27> 도 9는 노드 라우팅 테이블을 채우는 방법의 플로우차트의 일례를 나타낸 도면.
- <28> 도 10은 목적지 노드 쪽으로 메시지를 수치적으로 라우팅하는 방법의 플로우차트의 일례를 나타낸 도면.
- <29> 도 11은 목적지 노드 쪽으로 메시지를 근접적으로 라우팅하는 방법의 플로우차트의 일례를 나타낸 도면.
- <30> 도 12a는 기존의 연합 내에서 멤버쉽을 설정하는 노드의 일례를 나타낸 도면.
- <31> 도 12b는 메시지를 교환하는 연합 기반구조 내의 노드의 일례를 나타낸 도면.
- <32> 도 13은 연합 기반구조 내에서 멤버쉽을 확립하는 방법의 플로우차트의 일례를 나타낸 도면.
- <33> 도 14는 연합 기반구조 내에서 멤버쉽을 유지하는 방법의 플로우차트의 일례를 나타낸 도면.
- <34> 도 15는 다른 노드에 대한 생존 정보(liveness information)를 탐색하는 방법의 플로우차트의 일례를 나타낸 도면.
- <35> 도 16은 메시지 모델 및 관련 처리 모델의 일례를 나타낸 도면.
- <36> 도 17은 함수 계층(function layer)과 애플리케이션 계층 사이에서 일어날 수 있는 다수의 생존 상호작용(liveness interaction)의 일례를 나타낸 도면.
- <37> 도 18은 링 상의 노드들에 걸쳐 라우팅되는 요청-응답 메시지 교환 패턴의 일부를 형성하는 메시지들의 일례를 나타낸 도면.
- <38> 도 19a는 근접시간 통신을 용이하게 해주는, 링들로 이루어진 근접 유도 분할 트리의 일례를 나타낸 도면.
- <39> 도 19b는 도 19a의 링들로 이루어진 근접 유도 분할 트리의 다른 일례를 나타낸 도면.
- <40> 도 19c는 도 19a의 링들로 이루어진 근접 유도 분할 트리의 일례의 일부분을 분할하여 나타낸 도면.
- <41> 도 19d는 도 19a의 링들로 이루어진 근접 유도 분할 트리의 일례에서의 중간 링을 확대하여 나타낸 도면.
- <42> 도 19e는 도 19a의 링들로 이루어진 근접 유도 분할 트리의 또다른 일례를 나타낸 도면.
- <43> 도 19f는 도 19a의 링들로 이루어진 근접 유도 분할 트리의 또다른 일례를 나타낸 도면.
- <44> 도 19g는 도 19f의 일부분을 확대하여 나타낸 도면.
- <45> 도 20은 링들로 된 트리에서의 노드에 대한 방계 링 세트를 유지하는 방법의 플로우차트의 일례를 나타낸 도면.
- <46> 도 21은 링들의 트리에서의 근접시간 통신을 전송하는 방법의 플로우차트의 일례를 나타낸 도면.
- <47> 도 22는 링들의 트리에서의 근접시간 통신을 전송하는 다른 방법의 플로우차트의 일례를 나타낸 도면.

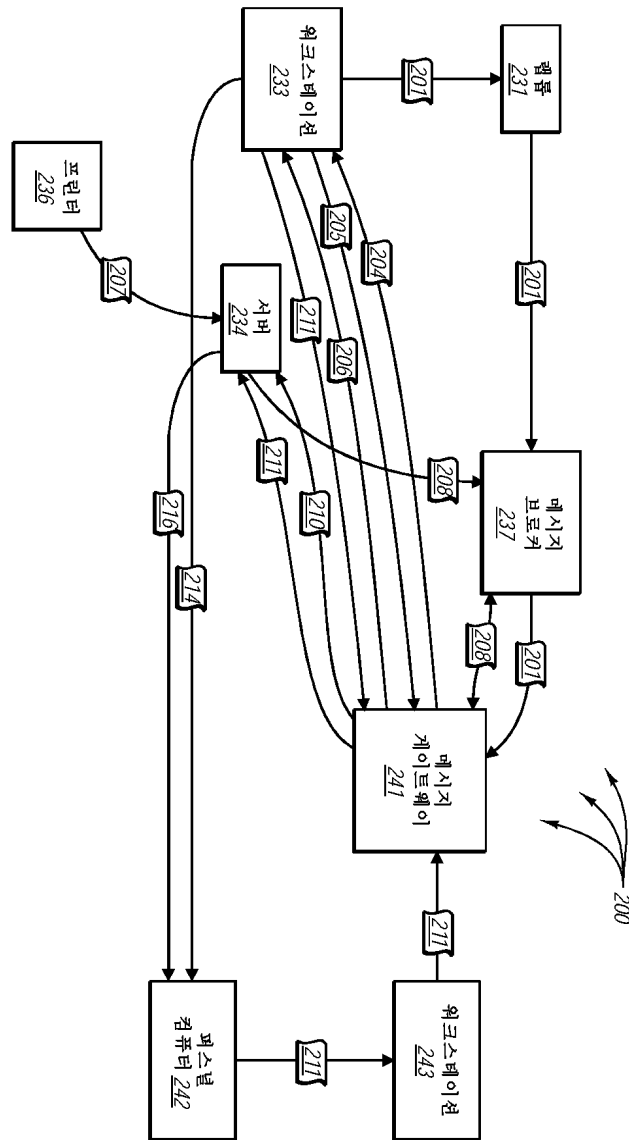


도면

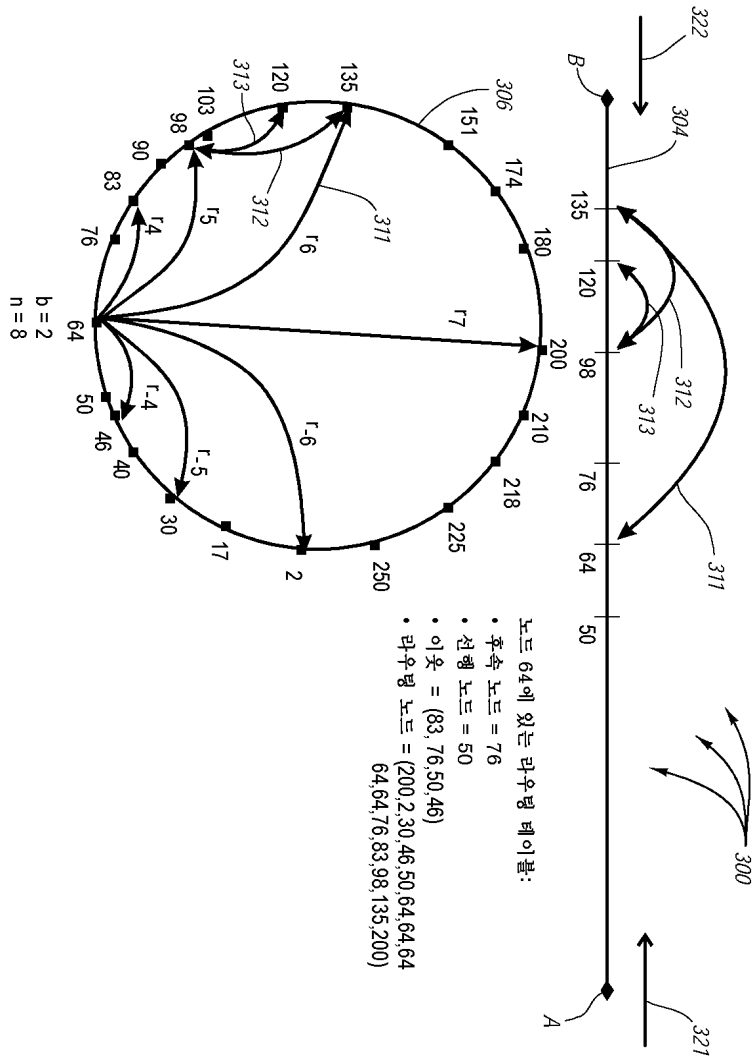
도면1



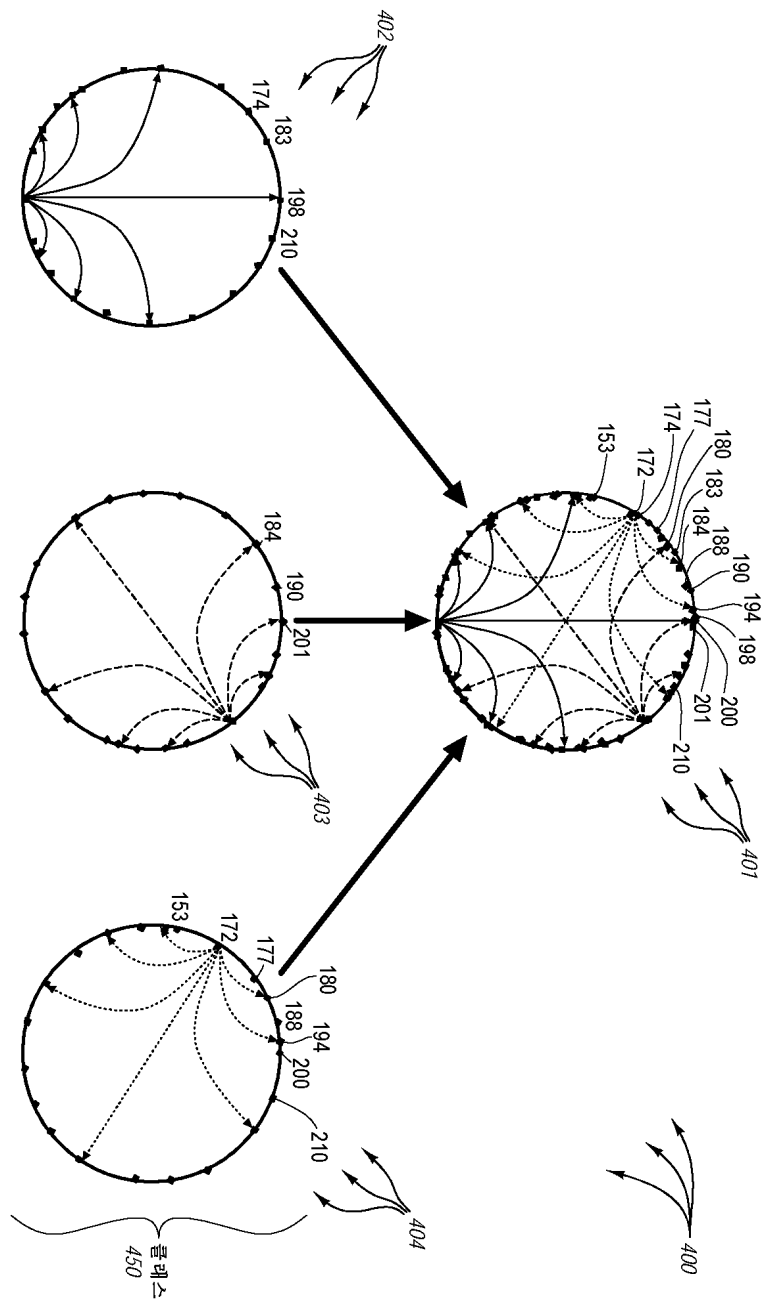
도면2



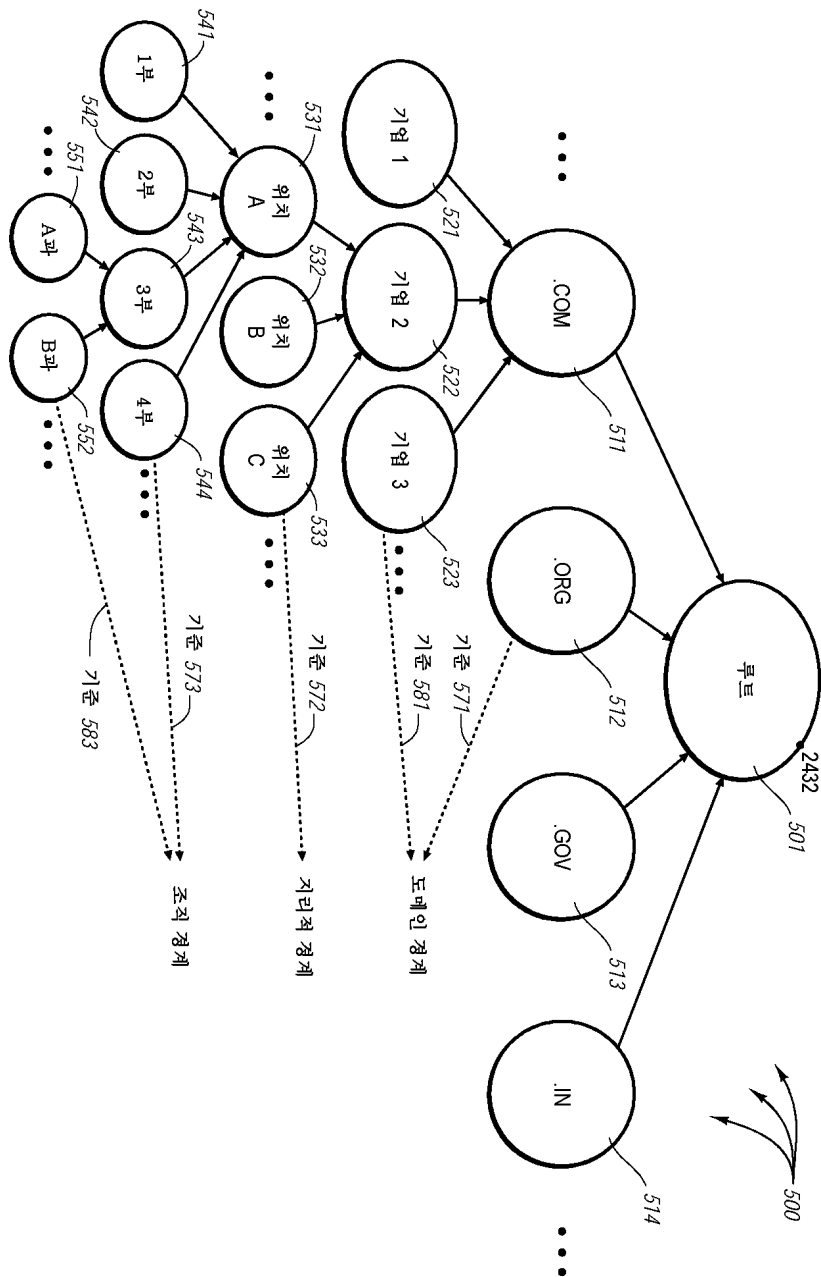
도면3



도면4

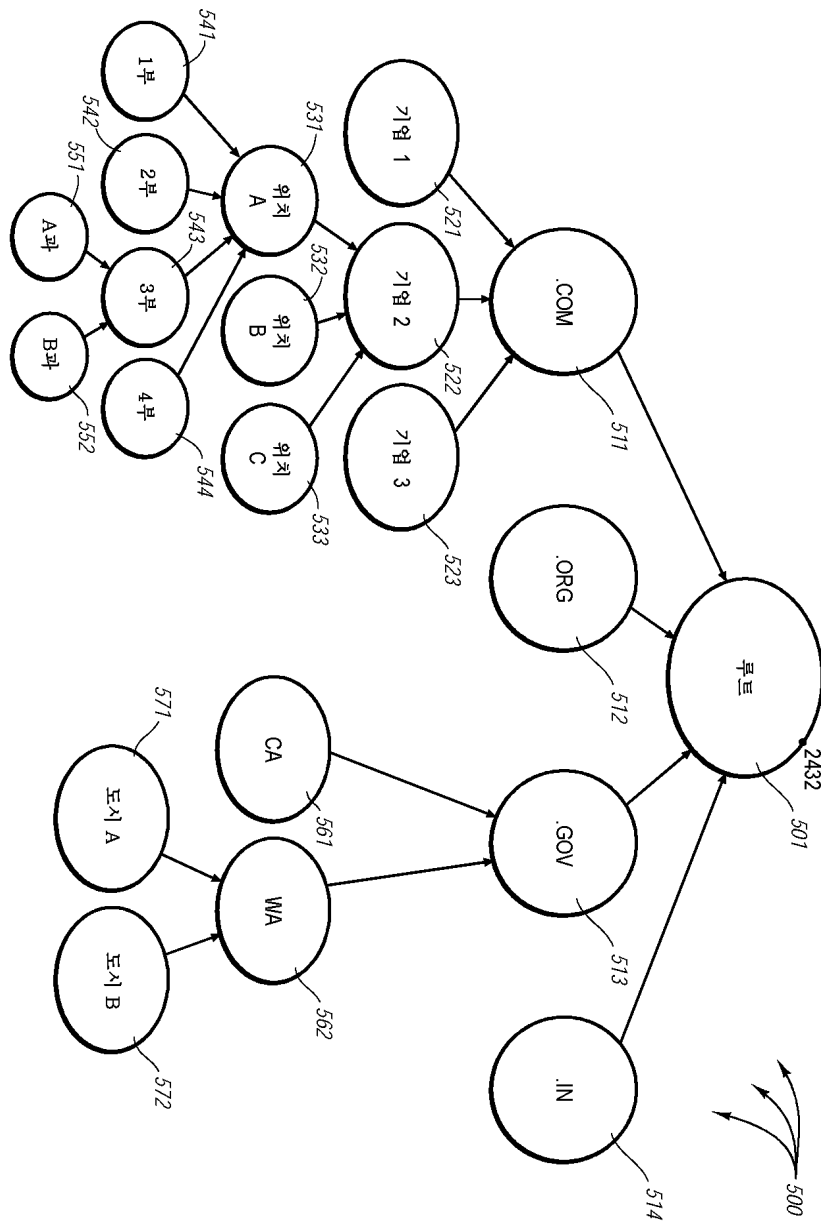


도면5

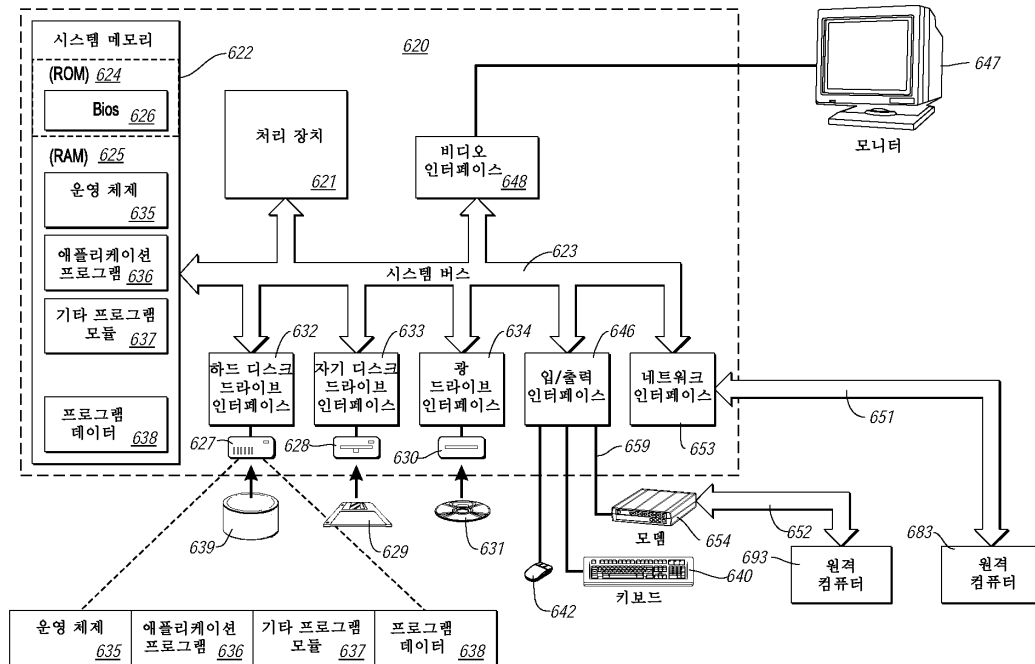




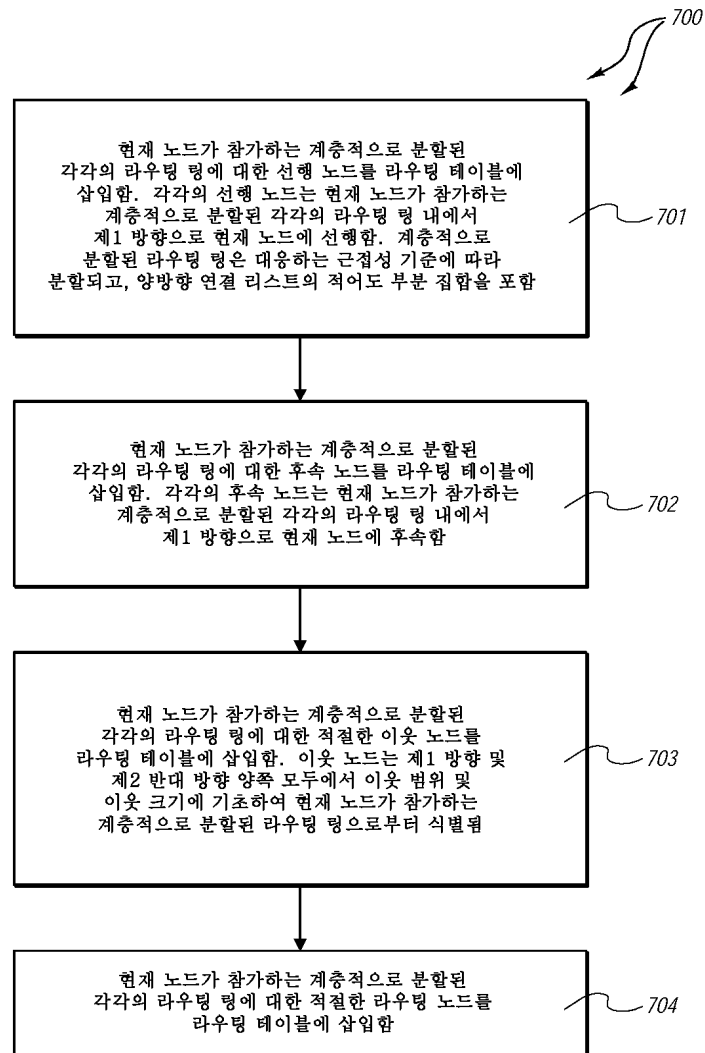
도면5a



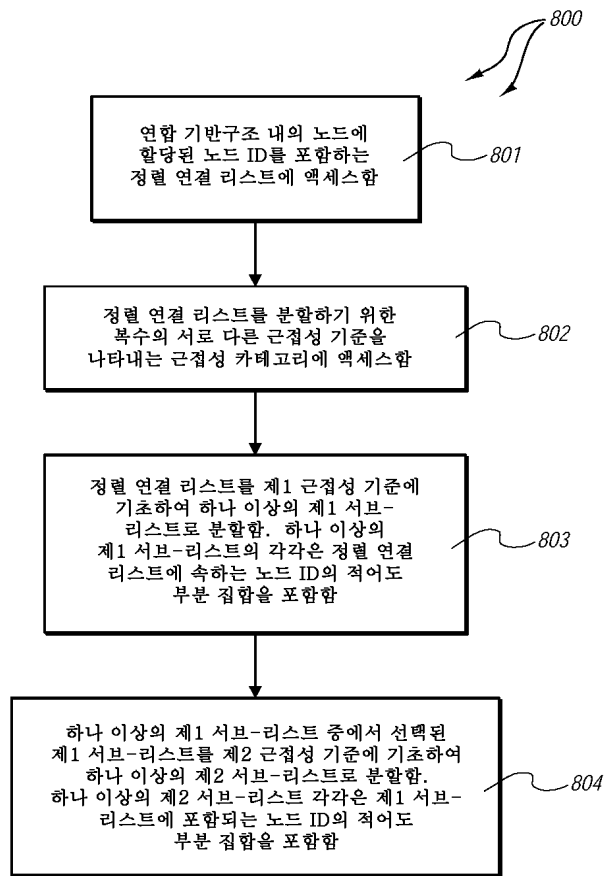
도면6



도면7

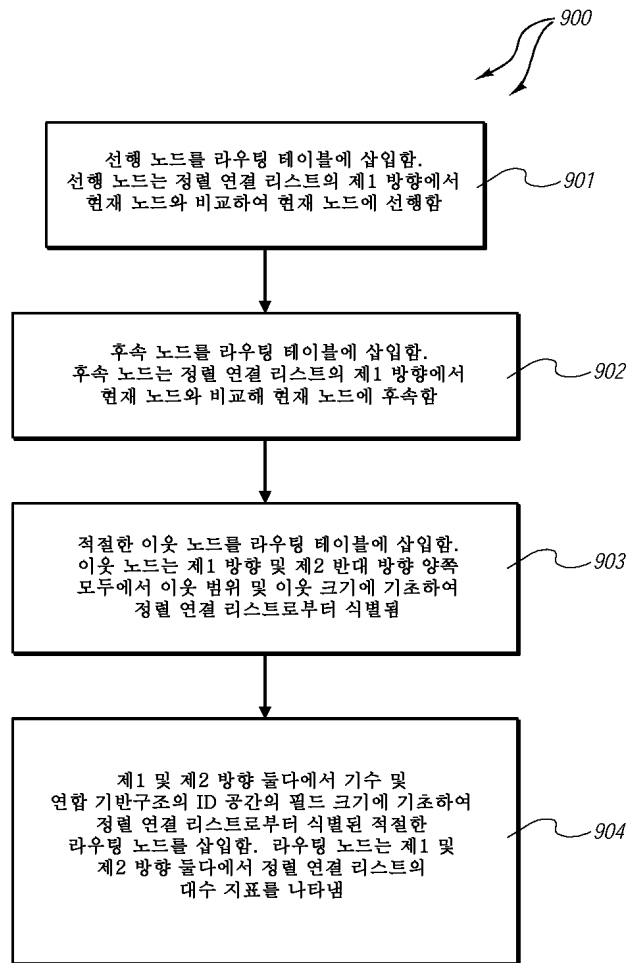


도면8

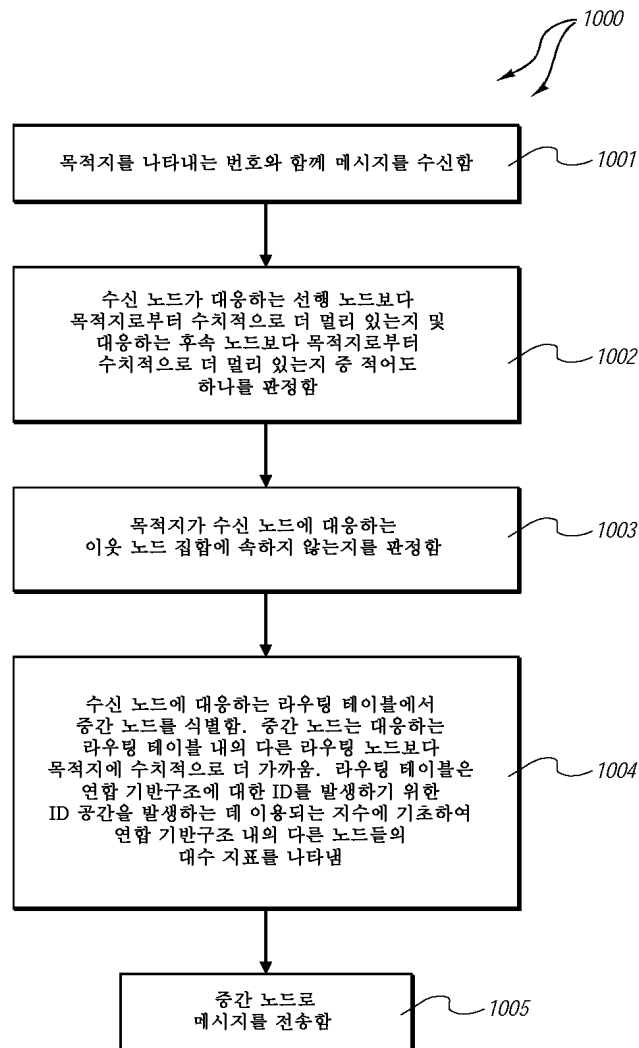




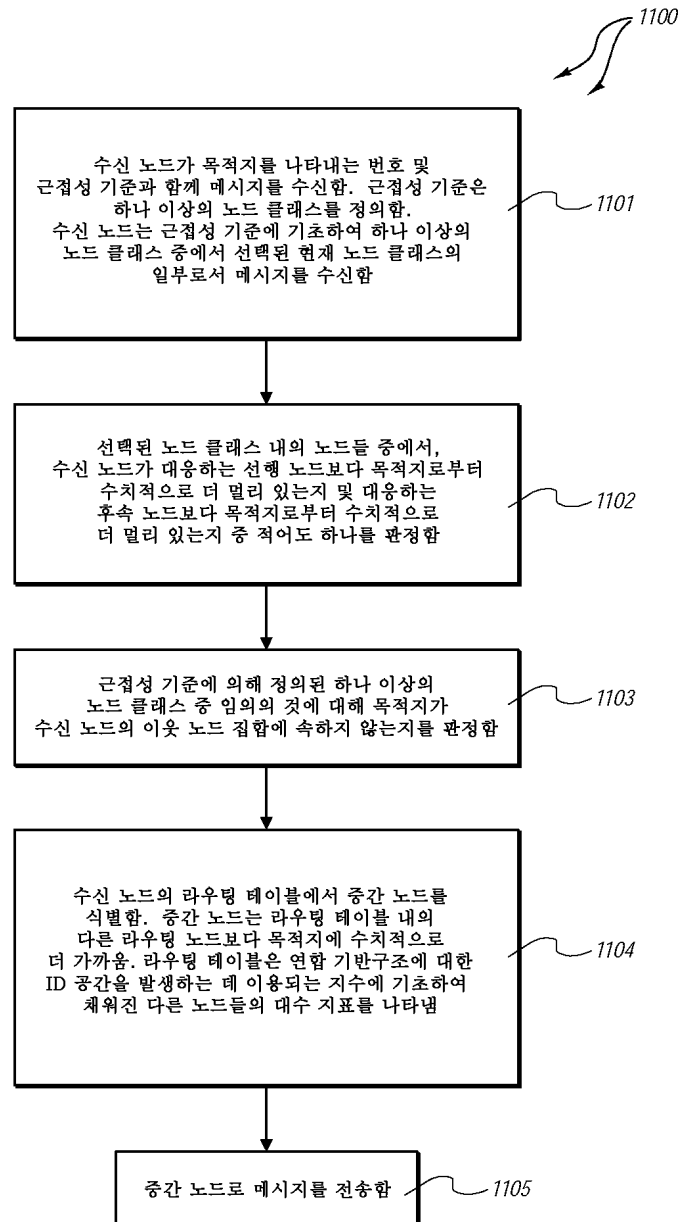
도면9



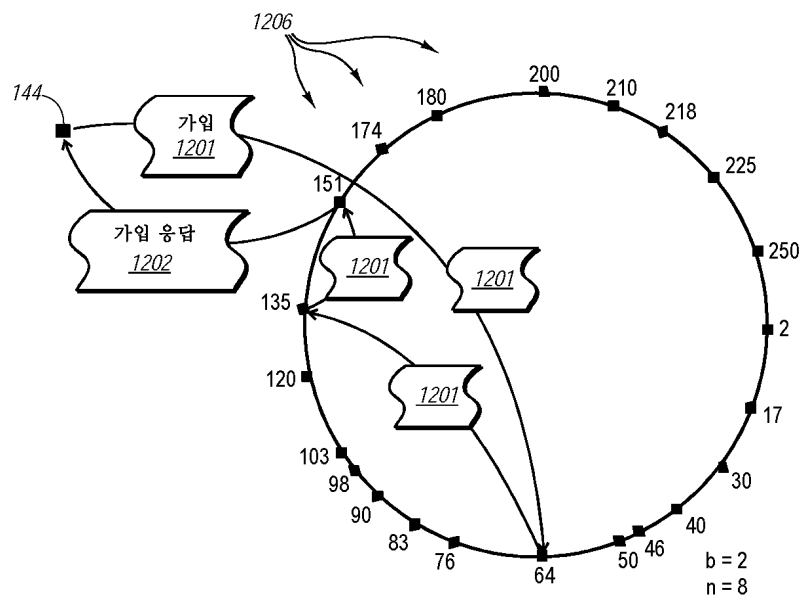
도면10



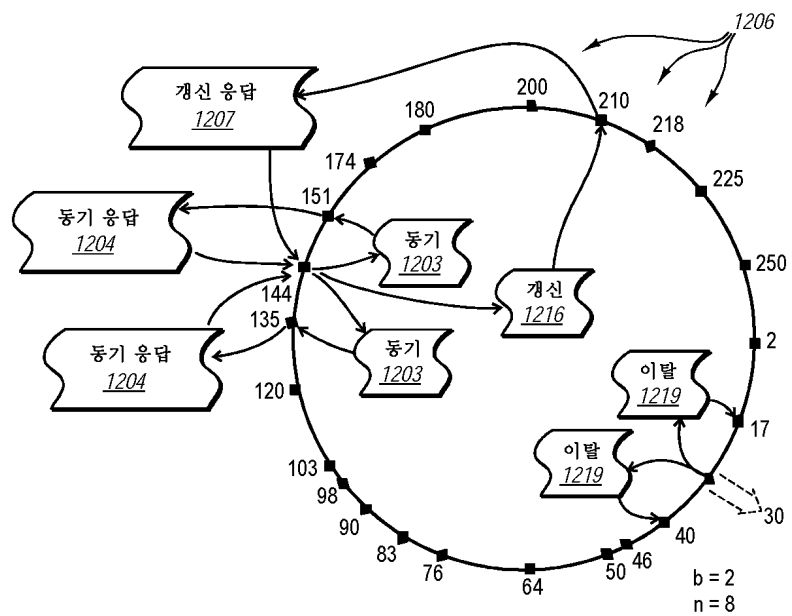
도면11



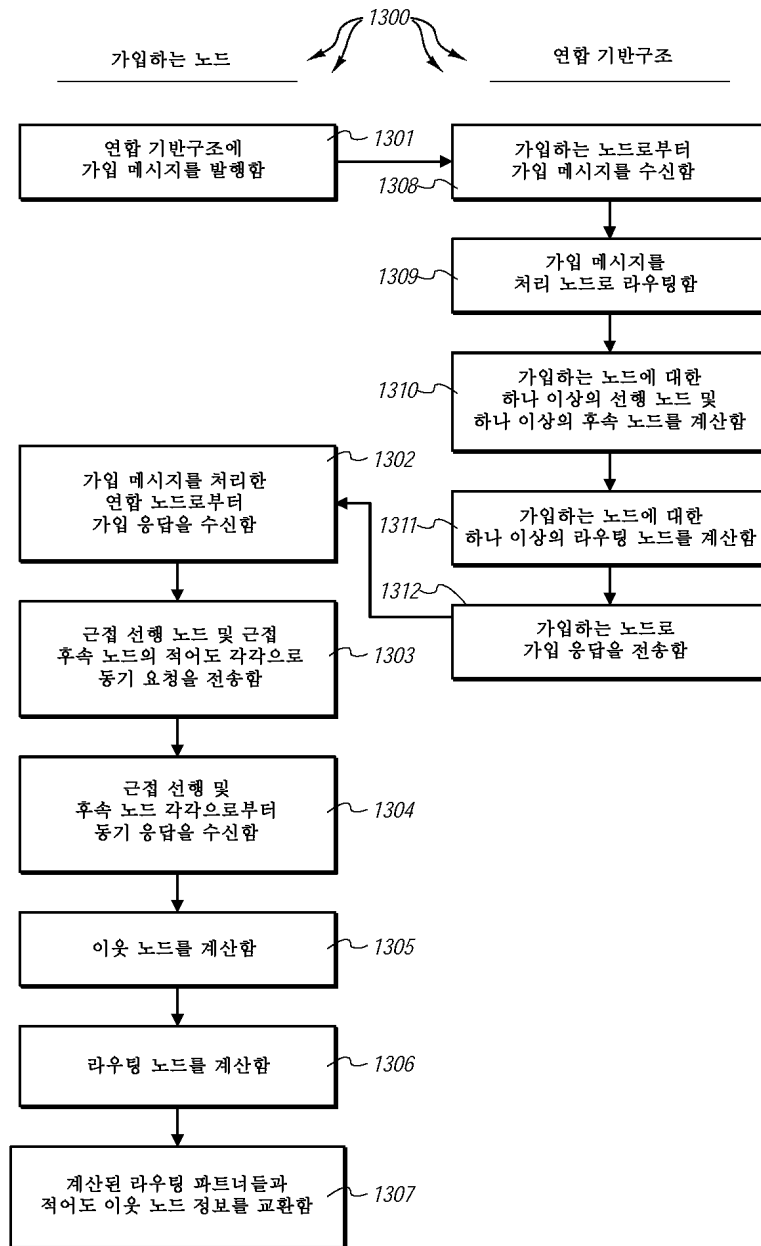
도면12a



도면12b

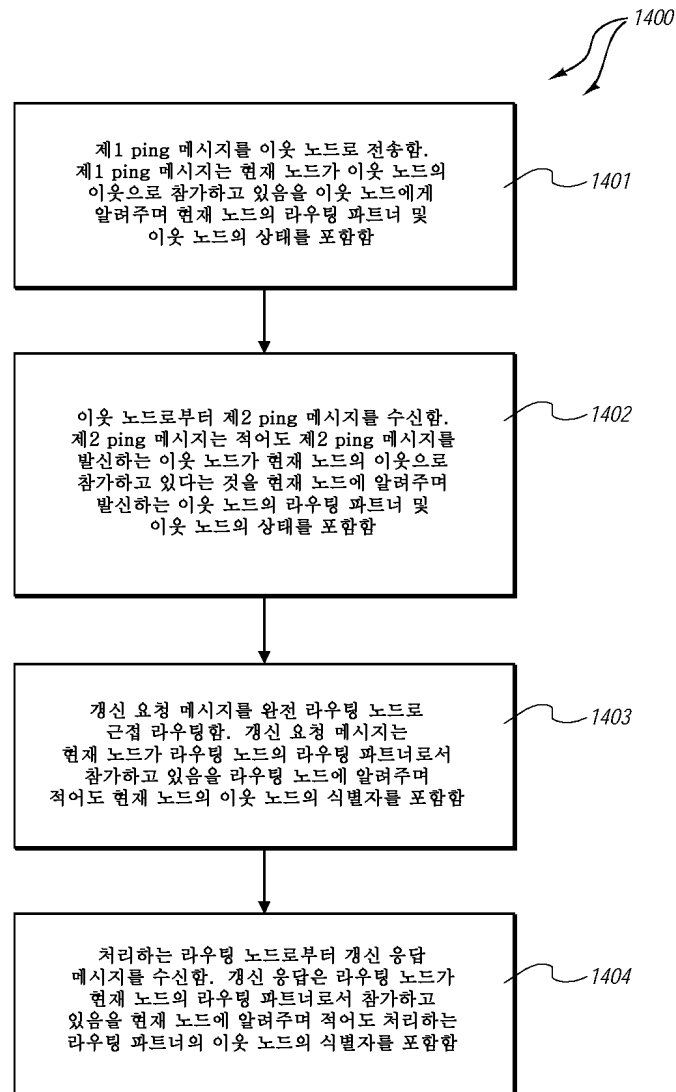


도면13

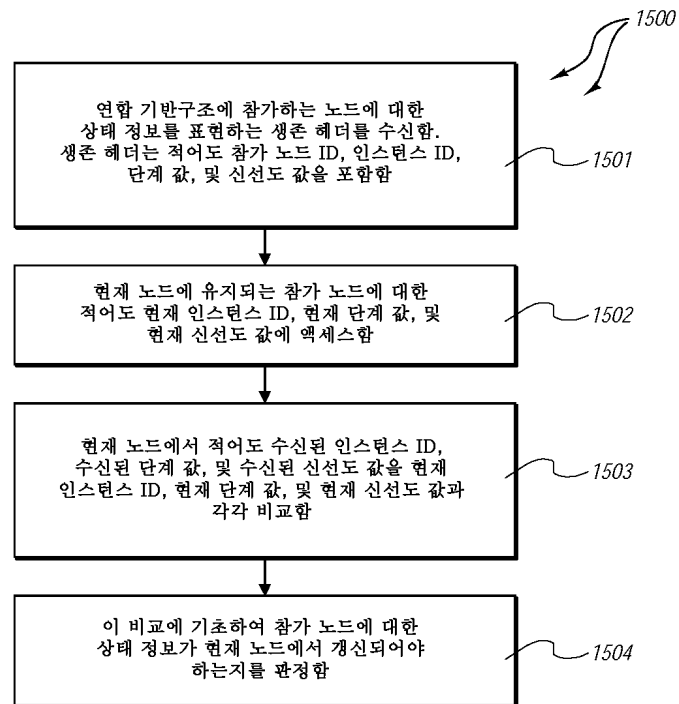




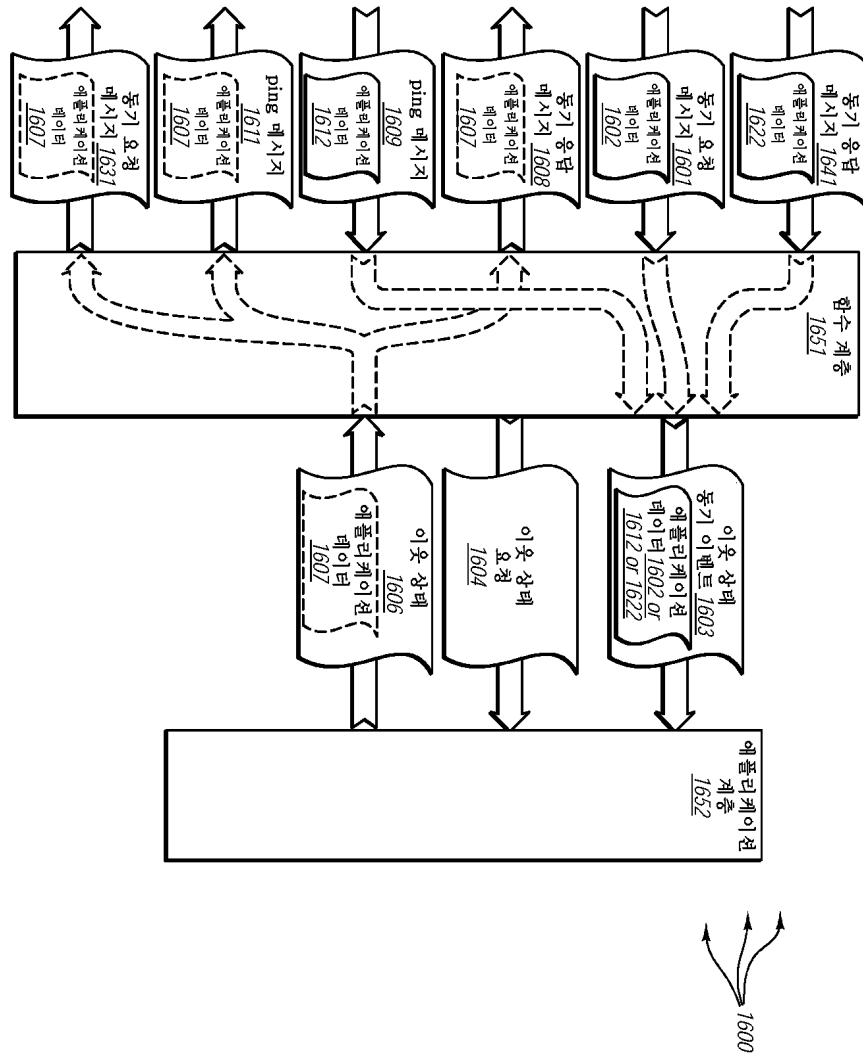
도면14



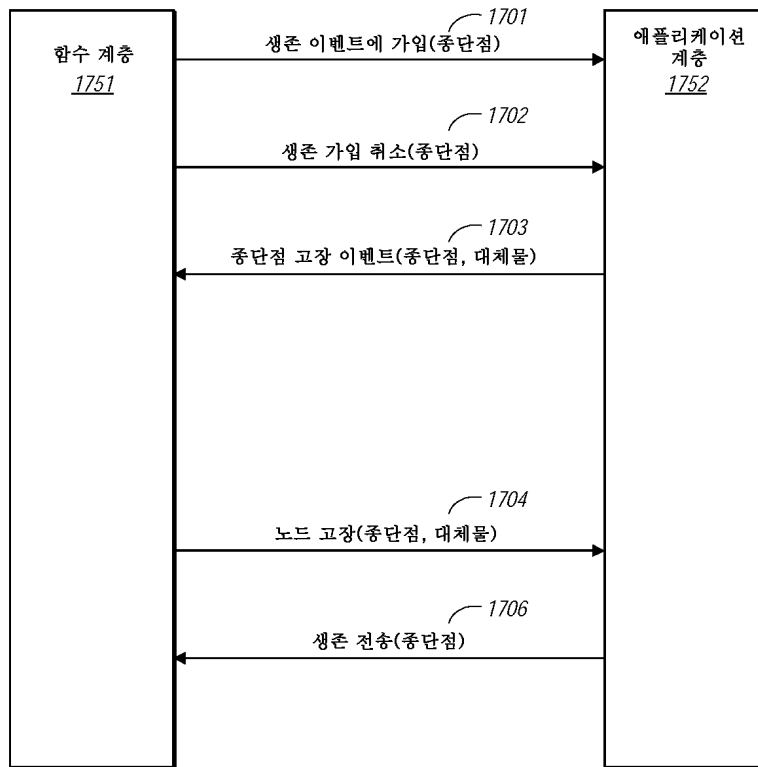
도면15



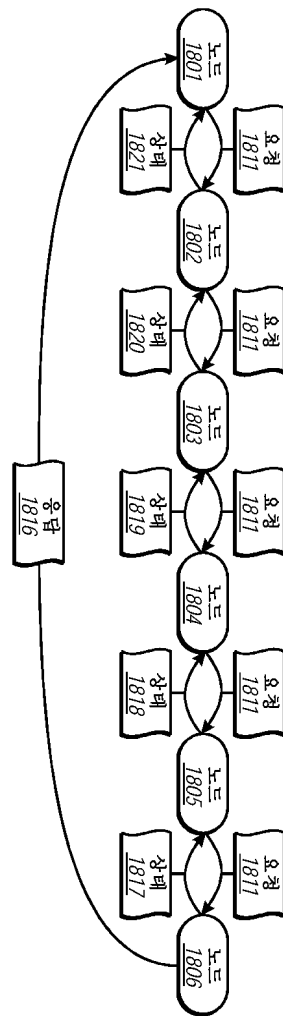
도면16



도면17

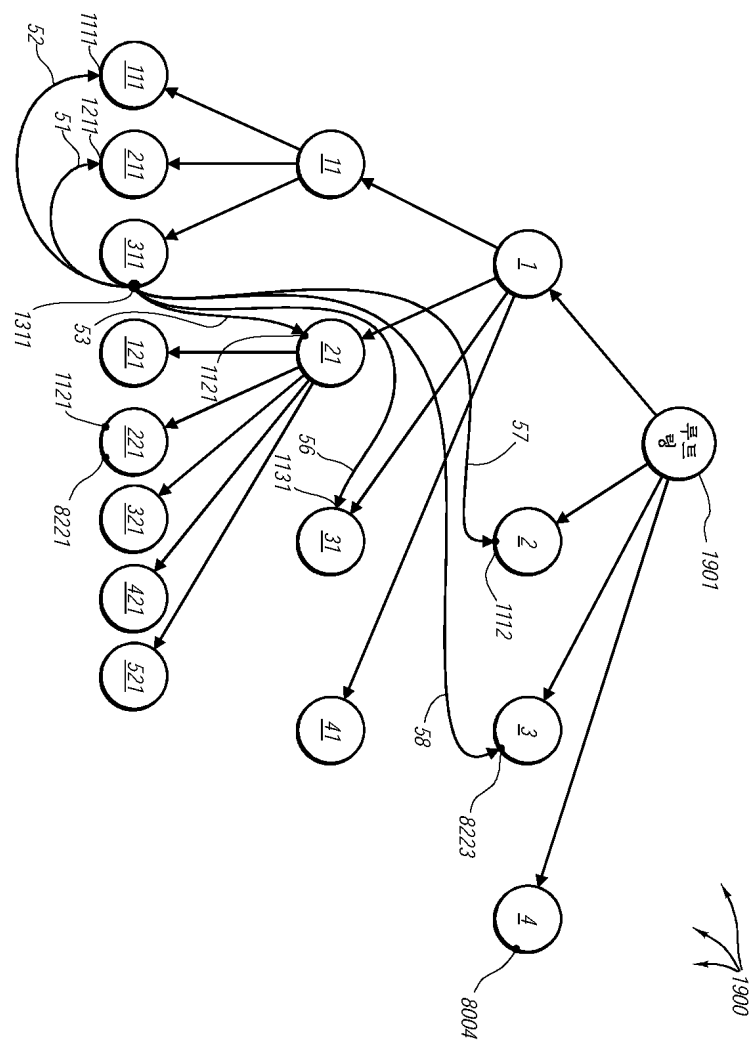


도면18

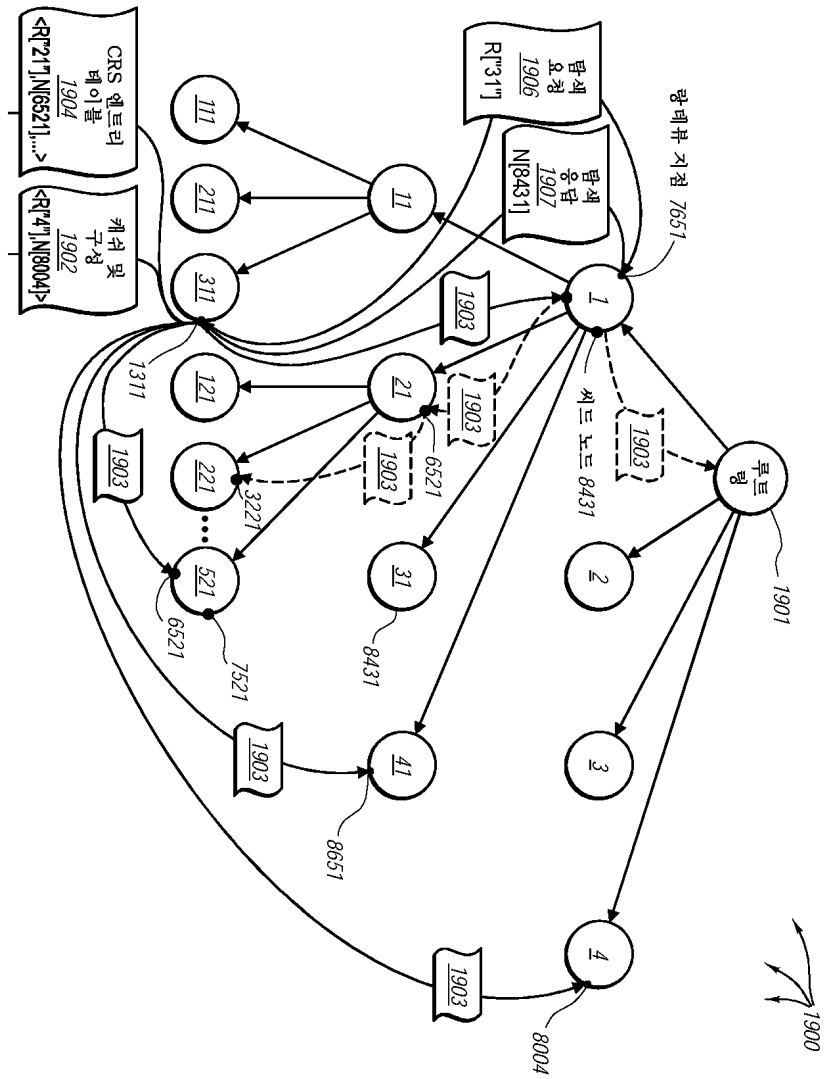




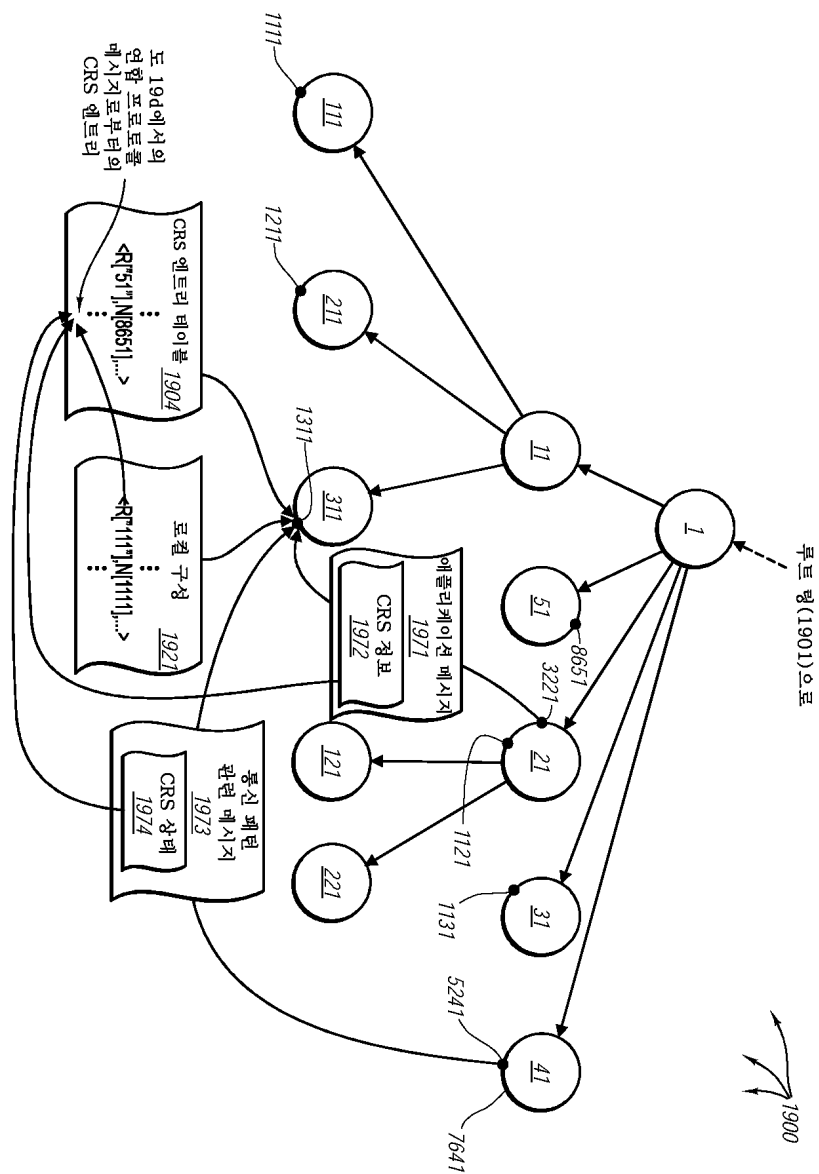
도면19a



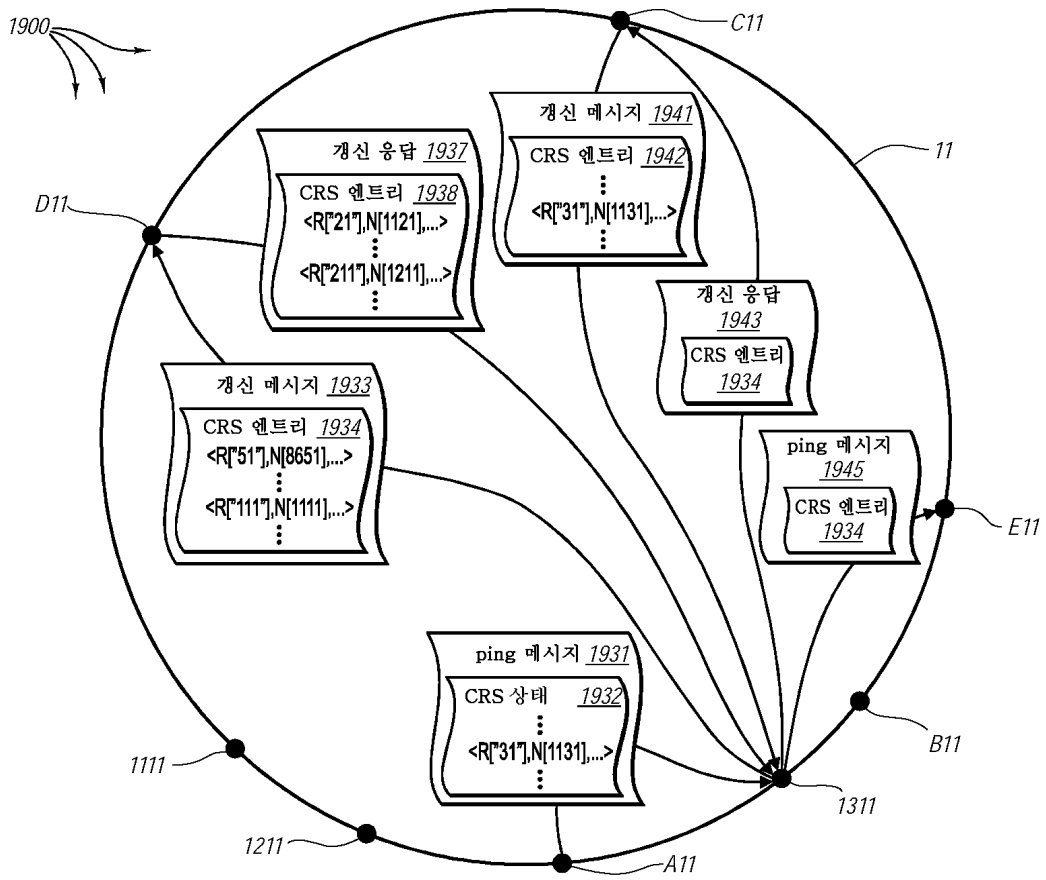
도면19b



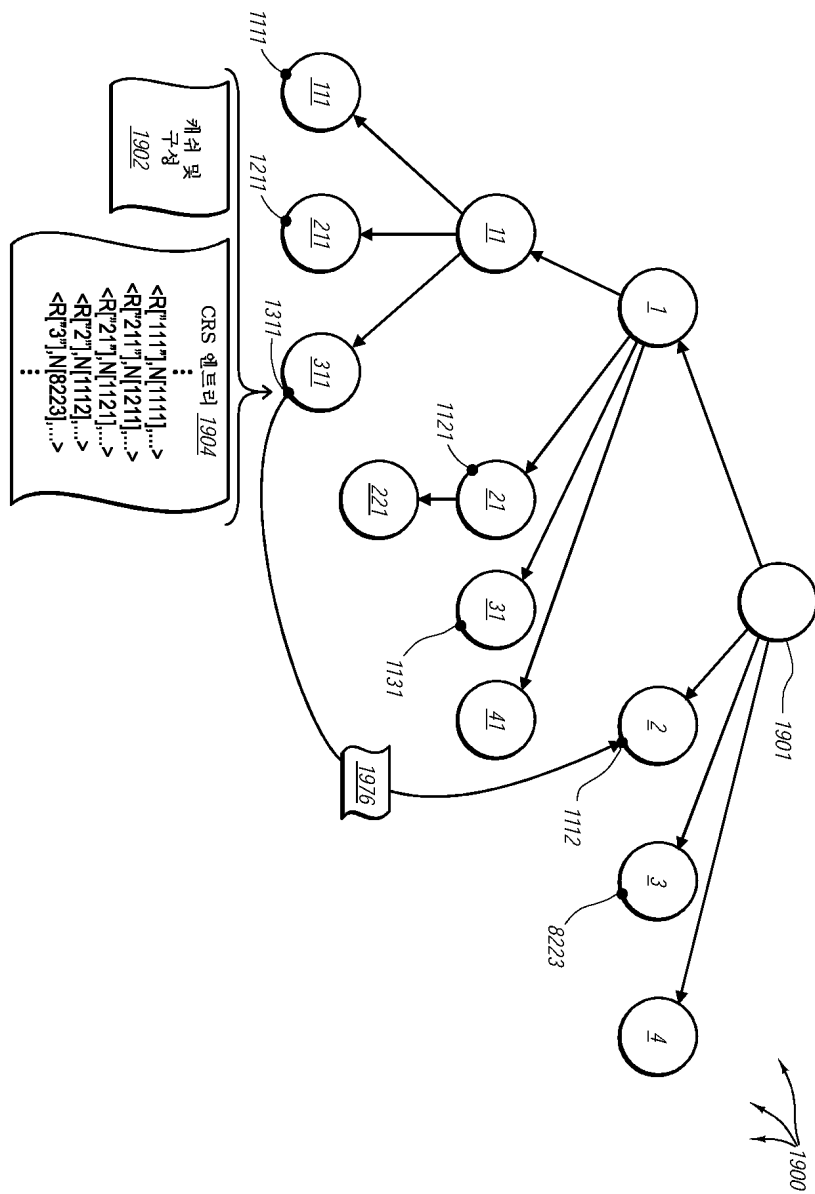
도면19c



도면19d

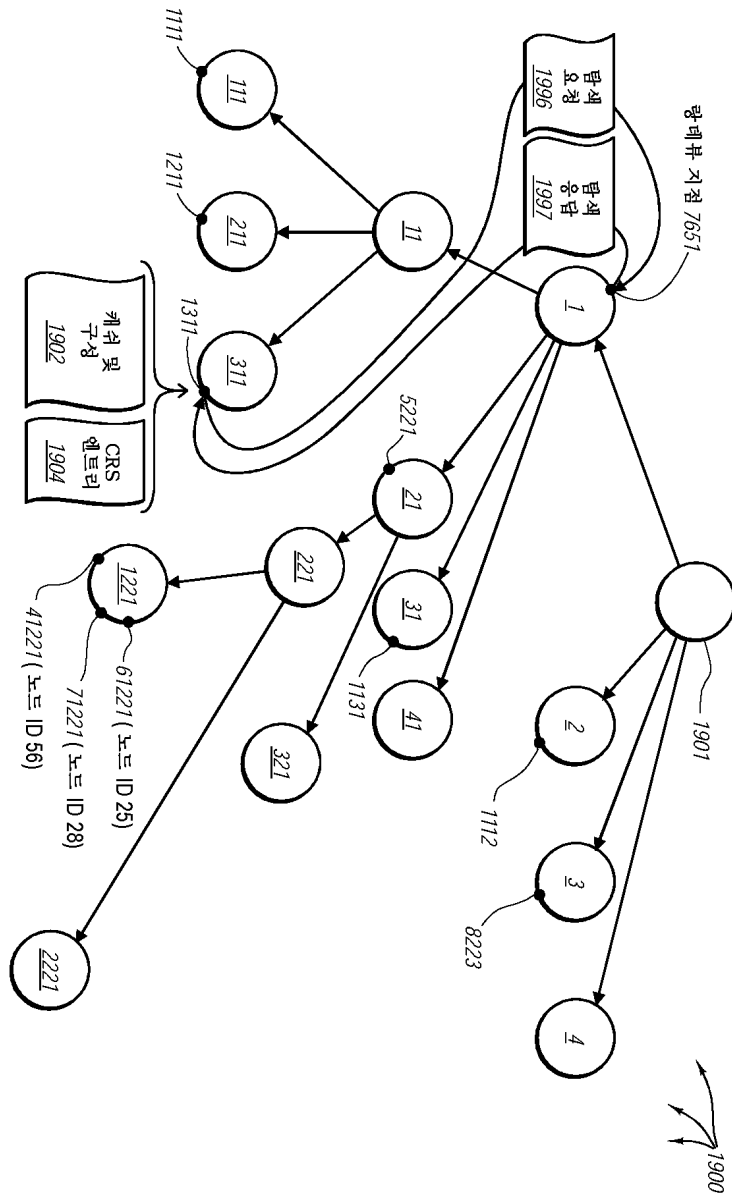


도면19e

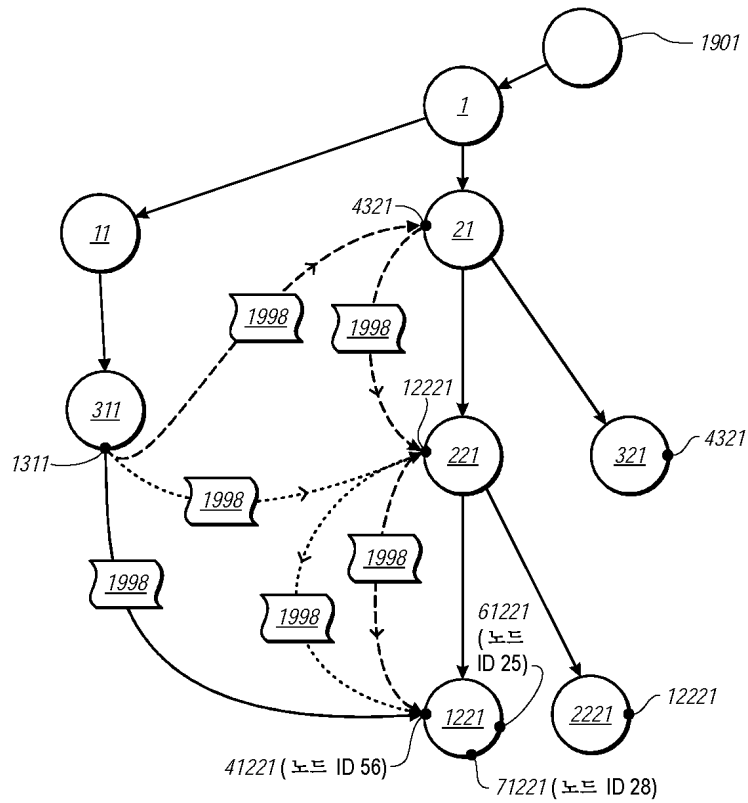




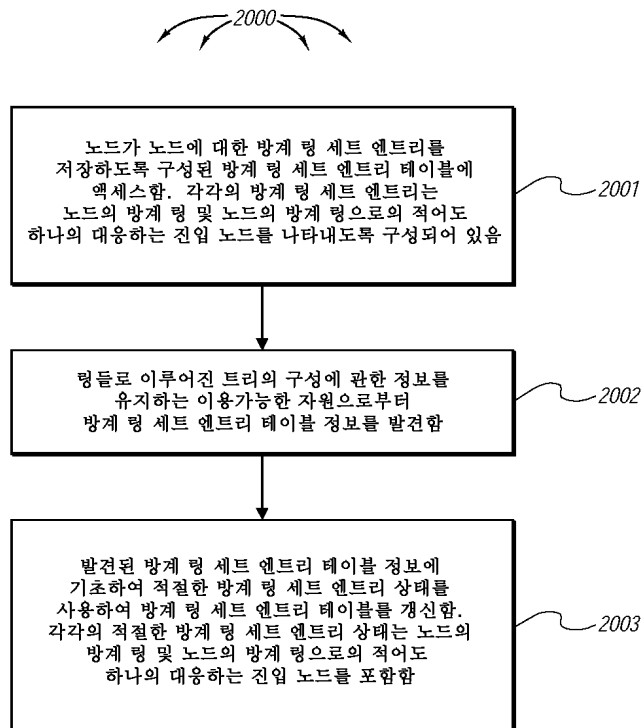
도면19f



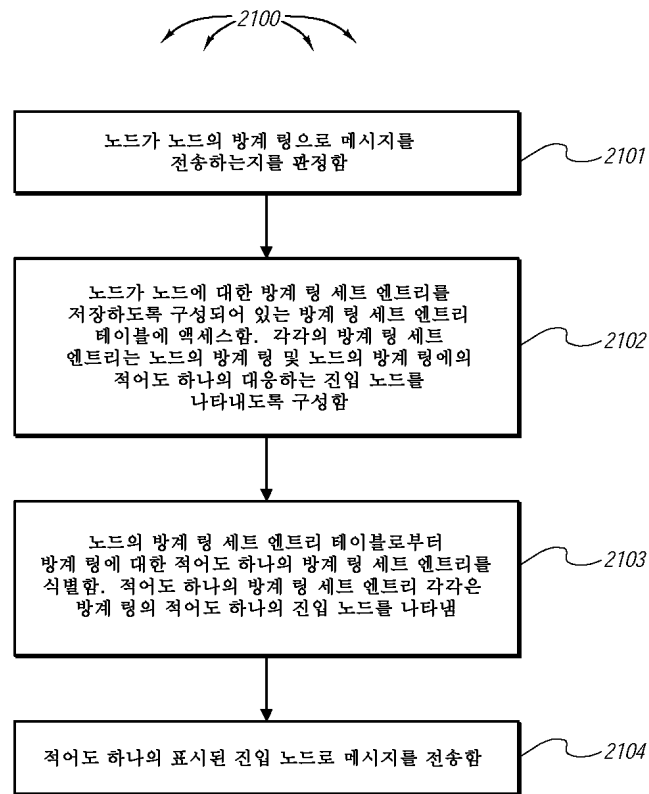
도면19g



도면20



도면21



도면22

